



برنامه‌نویسی مبتنی بر وب

رشته کامپیوتر - نرم افزار

دانشگاه آزاد اسلامی

سازمان سما

دانشکده فنی و حرفه‌ای سما واحد اندیشه

مهندس رضوانی

فهرست مطالب

مقدمه

1

1

نسخه های مختلف نرم افزار Visual Studio

2

تنظیمات اولیه محیط Visual Studio

17

انواع پروژه ها در Visual Studio 2008 & 2010

18

چگونه یک سایت خوب طراحی کنیم؟

18

طراحی استاندارد از نظر شکلی و تکنیکی

19

طراحی استاندارد از نظر فلسفی و مفهومی

19

تولید سایت با حداقل configuration

20

بالا بردن Ranking سایت

20

برای تولید برنامه تحت وب از چه زبانی استفاده کنیم؟

23

طراحی Layout

23

فایل style sheet

29

فایل Master Page

33

چند style مفید برای قسمت Main صفحه

33

Style چپ به راست و راست به چپ

34

Style مربوط به Hyperlink

35

تعریف Template

38

کدنویسی

38

کنترل های Server Side

41

دکمه reset

41

Layout ورود اطلاعات

43

صفت AutoEventWireup

44

موجوات برزخی (تگهای HTML با ID)

44

ListBox و DropDownList

45	String.format یعنی چه؟
45	صفت IsPostBack
45	مفهوم refactoring
49	آدرس دهی
49	آدرس دهی مطلق
49	آدرس دهی نسبی، نسبت به جایی که هستیم
50	آدرس دهی نسبی، نسبت به root
51	طراحی CMS
51	فولدر App_Data
54	مکانیزم Error Handling
62	امکان اضافه نمودن صفحات به صورت داینامیک در CMS
67	امکان مدیریت نمودن صفحات به صورت داینامیک در CMS
70	امکان ویرایش فایل CSS به صورت داینامیک در CMS
74	مفاهیم Object Oriented و کلاس
74	Object Oriented چیست؟
74	کلاس چیست؟
75	Name Space چیست؟
75	Class , Field , Property , Method چیست؟
76	Member استاتیک و غیر استاتیک چیست؟
76	Method Overloading چیست؟
77	Error Handling
78	دسته بندی Property های کنترلها
78	Property های نوع اول
78	Property های نوع سوم
79	Property های نوع دوم
79	Property های نوع چهارم
79	Property های نوع پنجم
79	کنترل Validator
79	کنترل Required Field Validator
82	سه نکته مهم در .NET
82	چیزهایی که باعث Post Back می شوند
83	صفحه Login و Registration با کنترل Field Required Validator
88	کنترل Regular Expression Validator
88	وظیفه این کنترل
88	Property های این کنترل
89	راهکارهای پیشنهادی جهت نوشتن Regular Expression
90	کنترل Compare Validator
90	وظیفه این کنترل

90	Property های این کنترل
91	کنترل Range Validator
91	وظیفه این کنترل
91	Property های این کنترل
91	کنترل Custom Validator
91	وظیفه این کنترل
91	Property های این کنترل
93	صفت validation group
96	کنترل Validation Summary
96	وظیفه این کنترل
97	Property های این کنترل
98	مفهوم User Control
102	Ajax
108	روش کار با ADO.Net
108	قسمت بانک اطلاعاتی
108	تنظیمات اولیه بانک اطلاعاتی - SQL Configuration & SQL Services
110	تعریف user
110	تعریف بانک اطلاعاتی (Database)
111	تعریف جدول (Table)
112	تعریف Primary key ، Foreign key و Alternate key
117	قسمت برنامه نویسی
123	کار با بانک اطلاعاتی با کمک wizard
128	روش وصل شدن به بانک اطلاعاتی و کار کردن با بانک اطلاعاتی به صورت دستی
138	بحث Web Application
138	خصوصیات Global.asax
138	Session Start & Applicatio Start
138	Session End & Applicatio End
140	متغیرهایی در سطح Application , session, ViewState , cookie
140	متغیرهای Application
142	ماجرای session id چیست؟
142	تعریف Session Time out
142	متغیر session
145	تعریف Session Clear , Session Abandon
146	تعریف cookie
148	Centralized Error Handling
148	انواع خطاها در برنامه‌های مبتنی بر web
148	اهداف Centralized Error Handling
149	Try throw catch در کنار Centralized Error Handling یا در کنار آن
149	انواع خطاها
151	Exceptions

153	کاربرد صفحه Offline.htm
161	انواع mode های مربوط به Custom Error
163	مبحث Authentication & Authorization
164	:Authentication
165	:Autorization
165	Allow
165	Deny
167	دستور یا کد برای Login کردن
167	در Login.aspx.cs
168	در برنامه UserLoginStatus.aspx.cs
169	در برنامه Logout
171	تعریف menu
174	دیگر پراپرتی های منوها:
174	ADD RUN TIME MENU
176	ایجاد منو با استفاده از Web.sitemap
176	حسن این menu
176	ایرادات این menu
179	مبحث چند زبانی
180	استفاده از فایل های Resource
190	خواندن از فایل Resource
191	ارسال Email
192	تکنیک ارسال email

جلسه اول

مقدمه

در این کلاس از محیط visual studio 2010 استفاده می کنیم. برای نصب محیط Visual Studio بهتر است که اول IIS را نصب کنید و بعد visual studio را نصب نمایید وگرنه دچار مشکل می شوید و بعد از آن SQL 2008 R2 را نصب می کنید. اگر فراموش کردید که IIS را نصب کنید و اول Visual studio را نصب کردید نگران نباشید، قابل اصلاح می باشد. در 2002, 2003 Visual Studio حتماً باید IIS را نصب می کردید تا بتوانید Web Application ایجاد کنید ولی در 2005 به بعد می توانید به صورت file system کار کنید و یک IIS Simulation را run می کند و از طریق آن عمل می کند. در Visual Studio 2008 مفهوم وب سایت ایجاد شد. در Visual Studio 2008 امکان ایجاد وب سایت به سه صورت وجود دارد، یا به صورت IIS ای است یا file System ای یا FTP است. Visual Studio 2010 هم در راستای مفاهیم 2008 حرکت کرده است. نکته در خصوص ویندوز و آنتی ویروس: حتماً ویندوز و آنتی ویروس را original استفاده کنید.

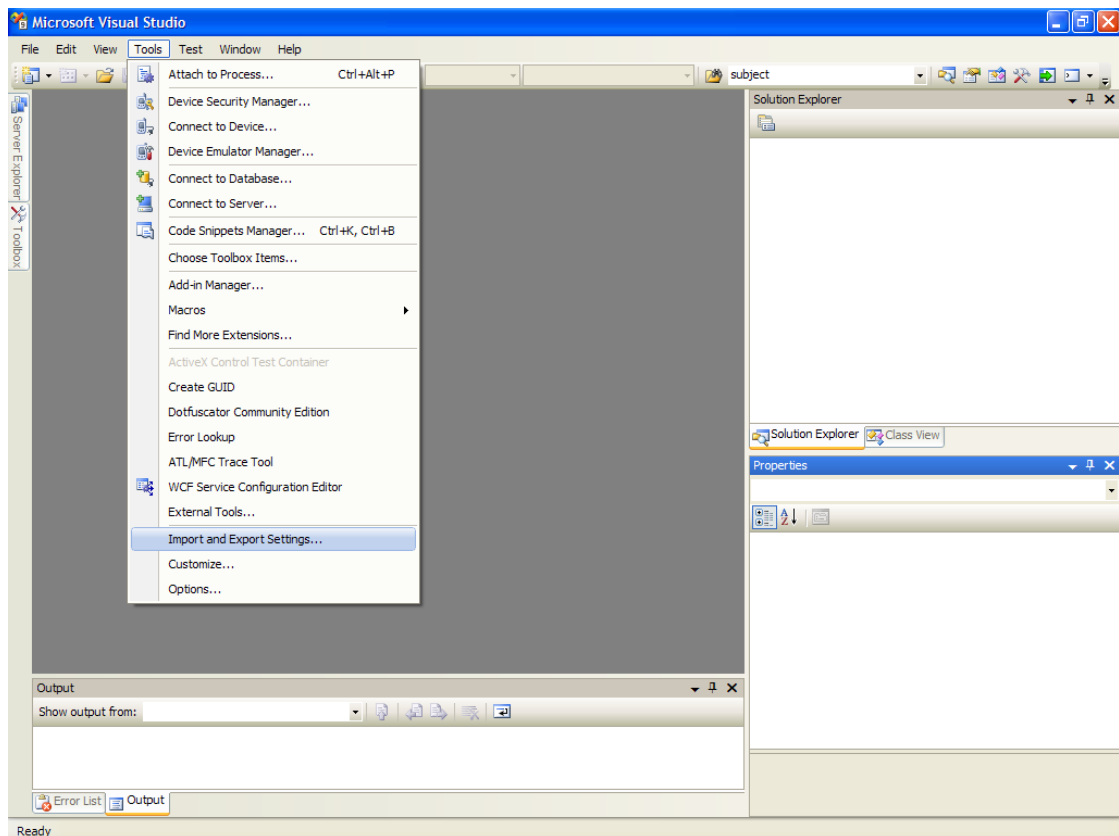
نسخه های مختلف نرم افزار Visual Studio

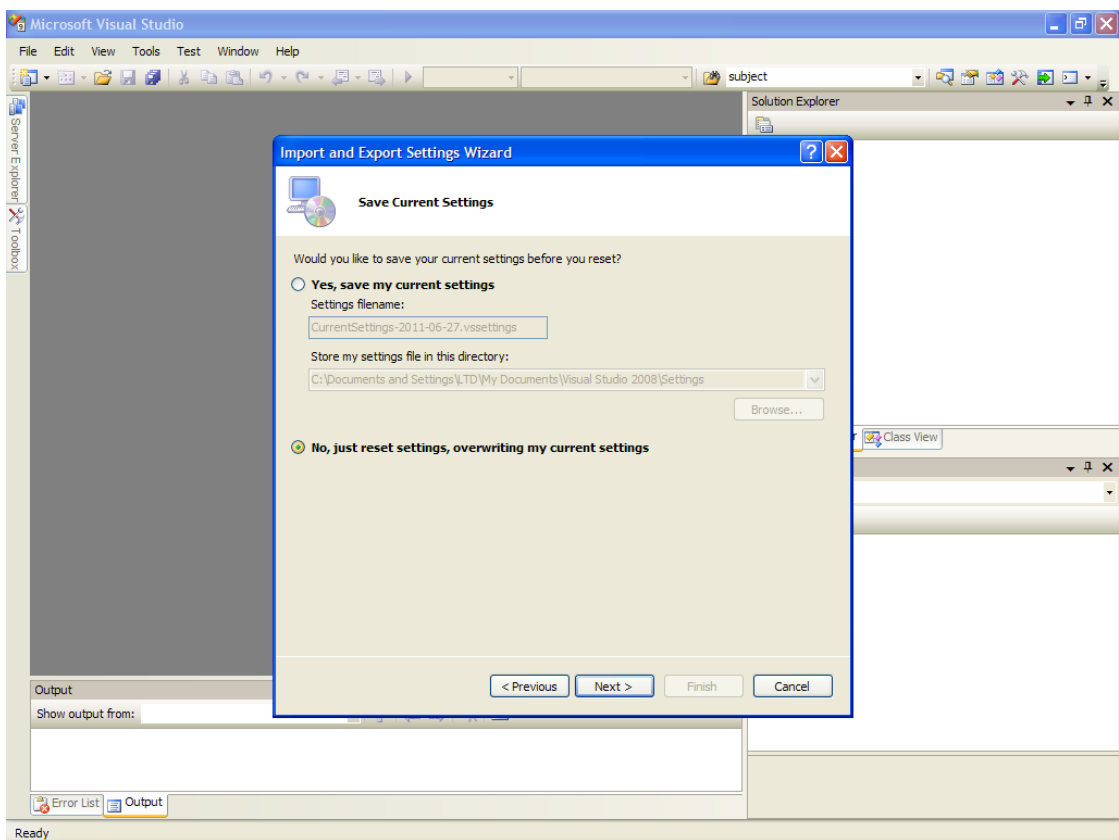
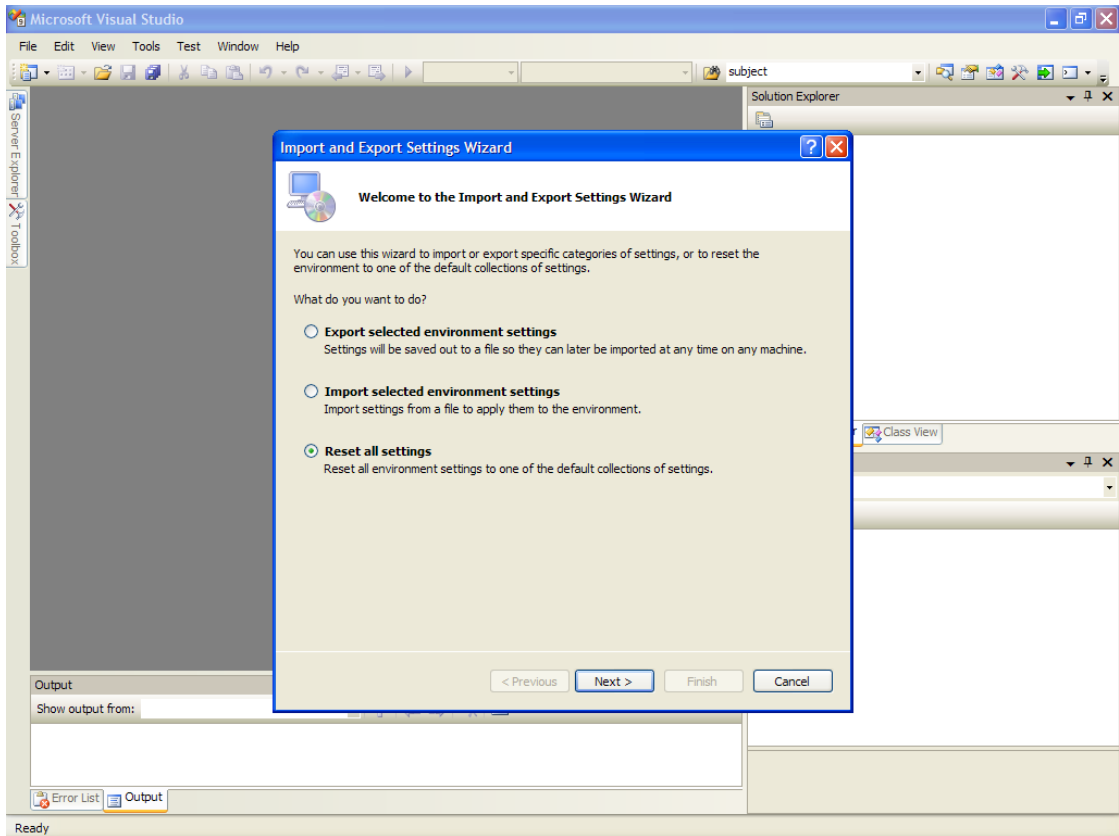
Visual studio 2010 چون Crack شده است، پس از وصل شدن به اینترنت دچار مشکلاتی می شود و یک سری منوی آن از کار می افتد و خطا می دهد. در واقع پس از وصل شدن به اینترنت این نرم افزار کاملاً مانیتور می شود. Microsoft در نسخه 2005 اقدام به ارائه Expression edition نمود ولی واقعاً محدود بود. ولی در 2008 و بعد از آن در 2010 نسخه expression آن قوی شد و محدودیت آن فقط در این است که نمی توان در Team Work از آن استفاده نمود و در واقع از Source safe و ... پشتیبانی نمی کند، وگرنه قابلیت های آن کامل است. اگر developer هایی هستید که تنهایی کار می کنید یا تعامل شما در کارها زیاد نیست، محیط expression این نرم افزار، بهترین انتخاب است. چون کاملاً free است و به مرور دچار مشکل نمی شود. پس Visual Studio 2010 Package expression کاملاً original را نصب نمایید. اولین بار که وارد Visual studio می شوید، یک پنجره باز می شود که می پرسد محیط Visual studio شما به چه شکل باشد، حتماً گزینه C# development را انتخاب کنید حتی اگر از زبان VB.net استفاده می کنید، چون قویترین محیط را در اختیار شما می گذارد.

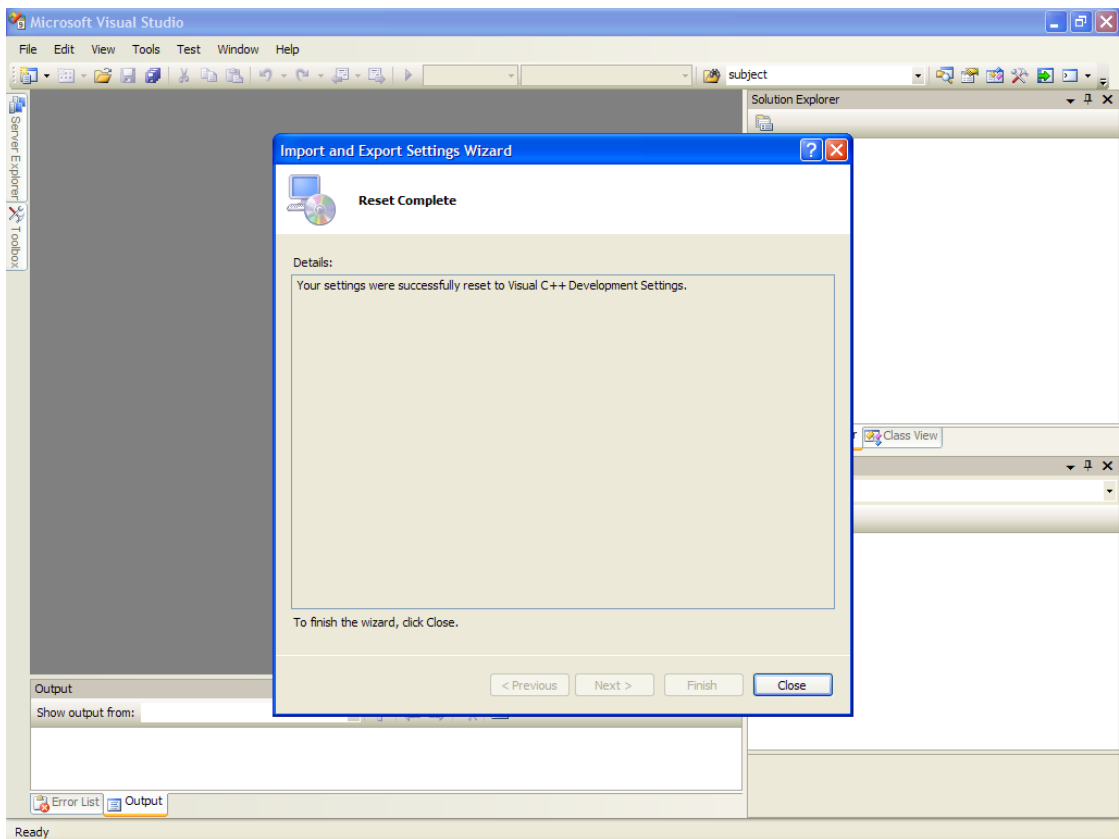
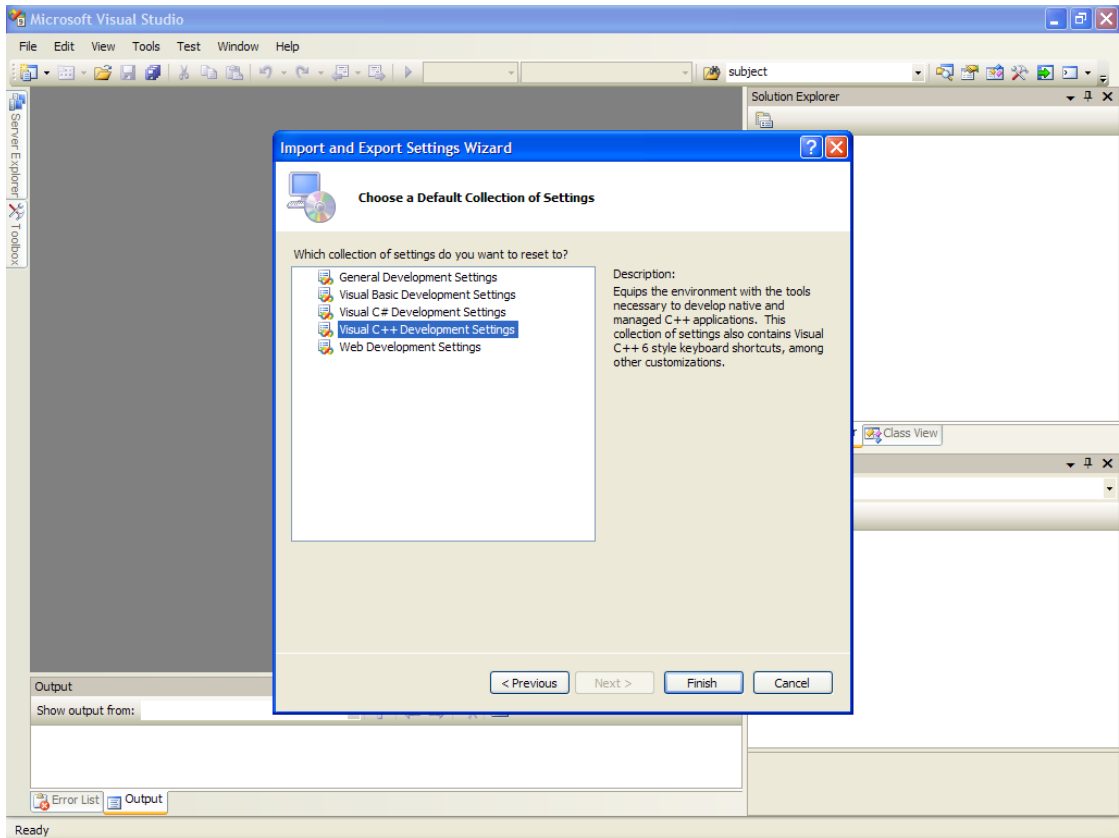
تنظیمات اولیه محیط Visual Studio

تنظیمات VS یک سری تنظیمات Start up دارد که باید در ابتدای کار صورت گیرد. چون نمی دانیم که اگر از قبل نرم افزار بر روی کامپیوتر نصب شده است، چگونه و با چه تنظیماتی بوده، پس اول کار آن را resear می کنیم. Snapshot ها گرفته شده و در فایل [DT Microsoft Visual Studio 2010 Settings 1390 01 03 Version 02](#) قرار داده شده است.

1 - مرحله اول resear کردن تنظیمات قبلی است. برای این کار مطابق شکل ذیل عمل کنید.
 نکته: اگر در محیط Visual studio 2010 expression هستی باید وارد قسمت extending visual studio شوید و Import & export setting را از گزینه Tools انتخاب نمایید. Reseat of setting را می زنیم. حتماً expert settings را انتخاب می کنیم ولی اگر نسخه enterprise است C# setting را انتخاب می کنیم.

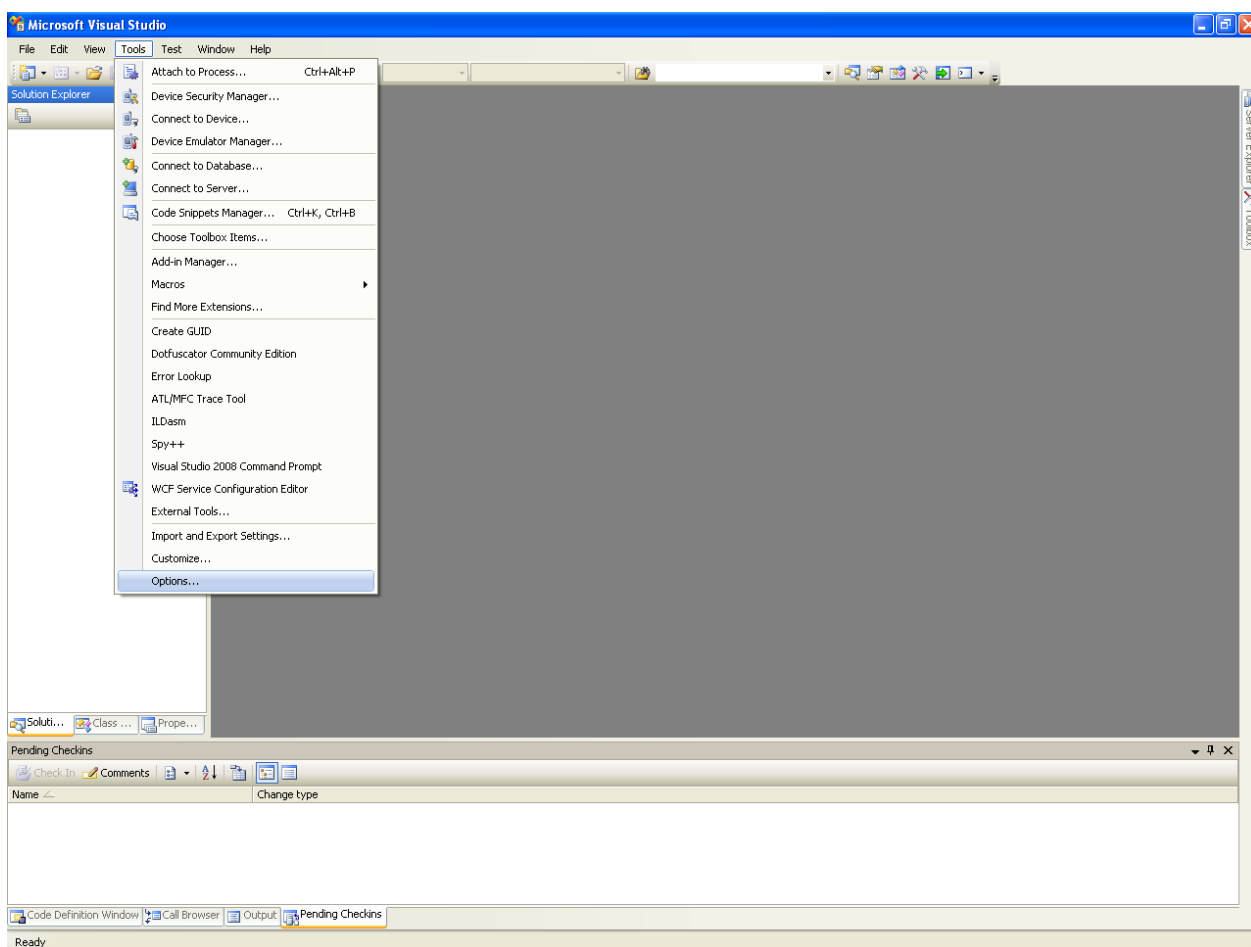


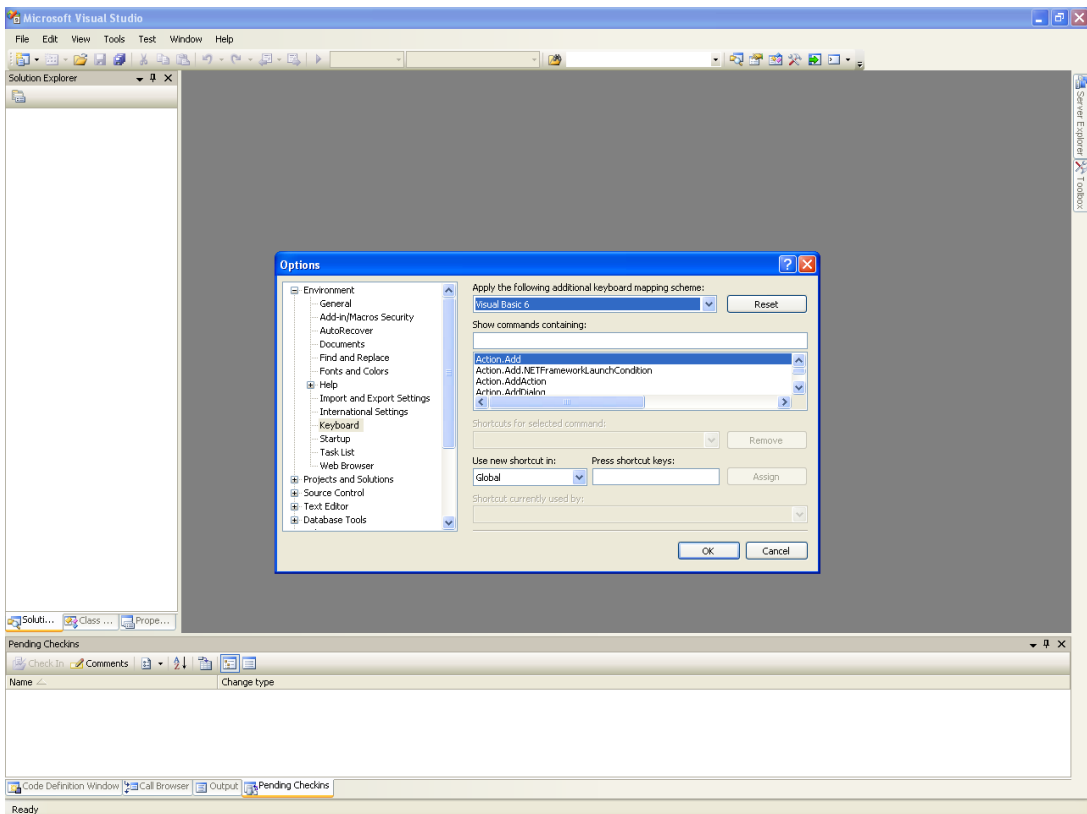
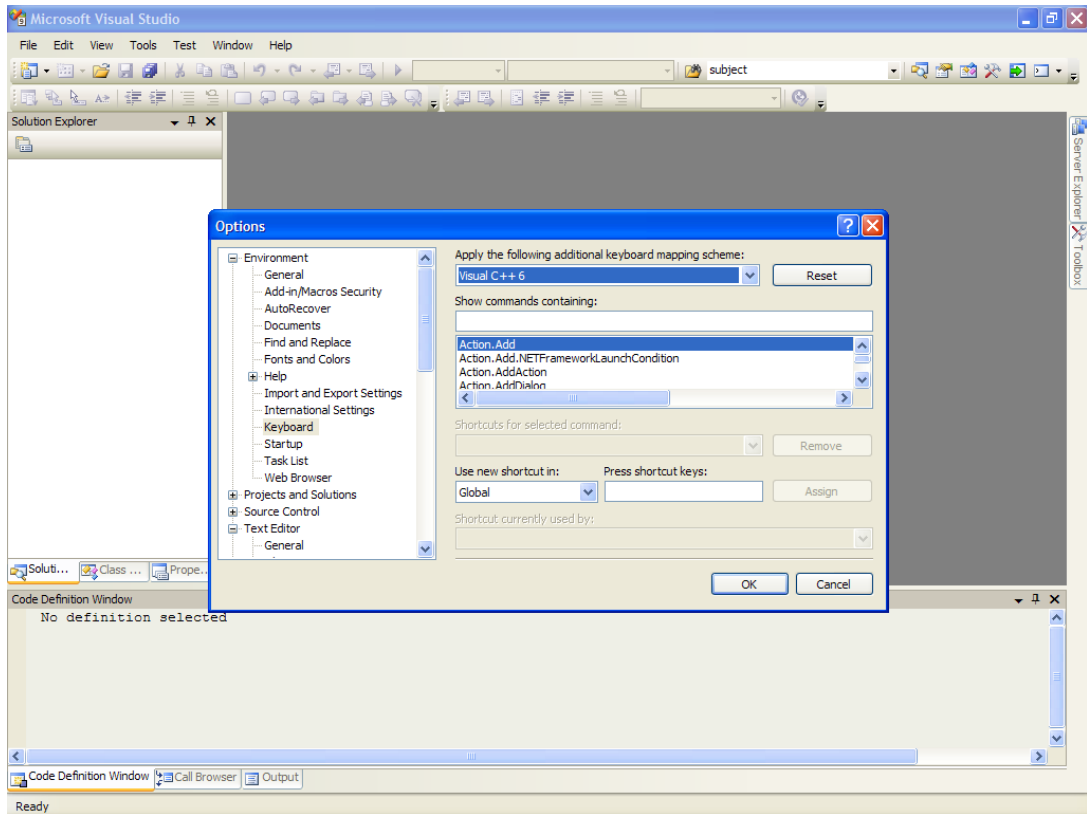


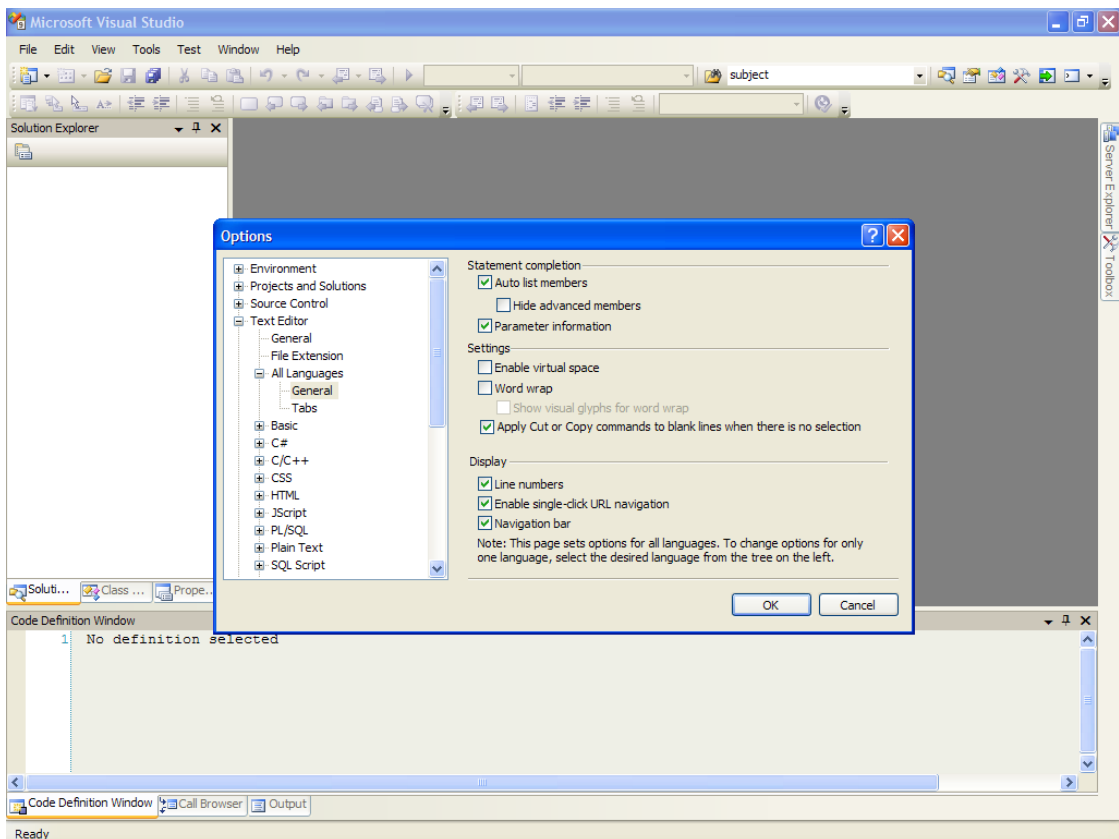
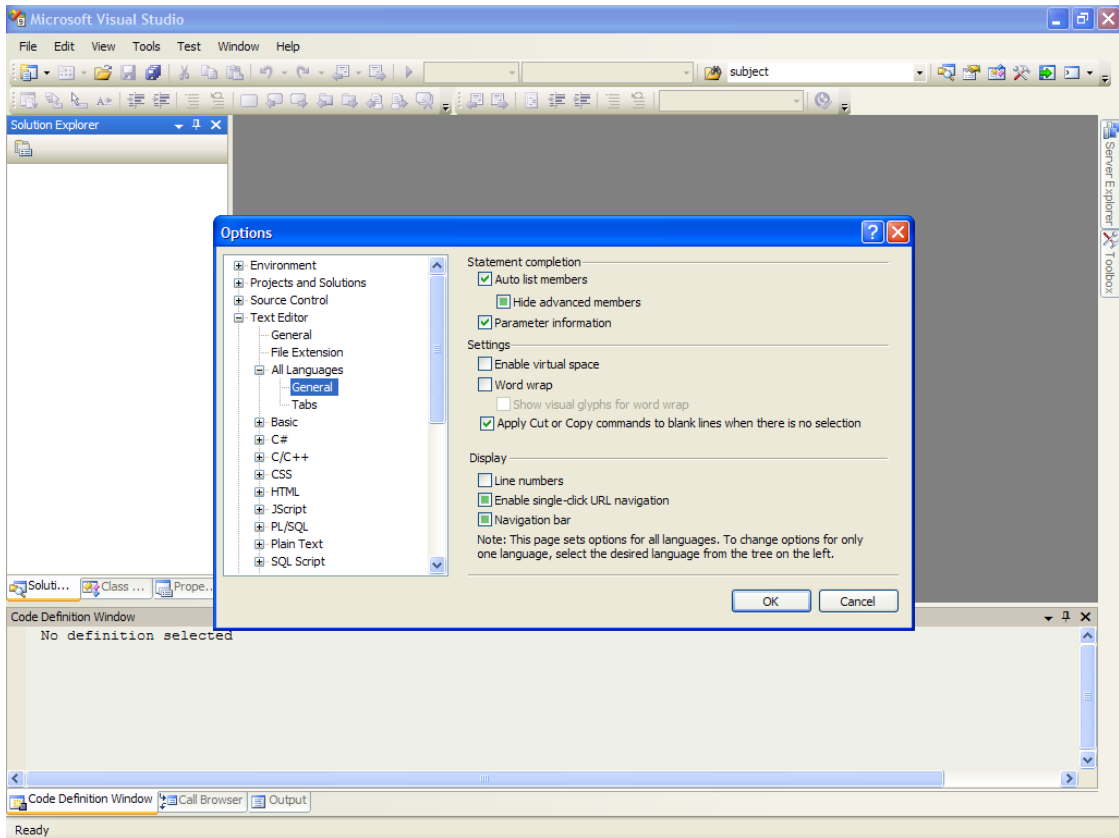


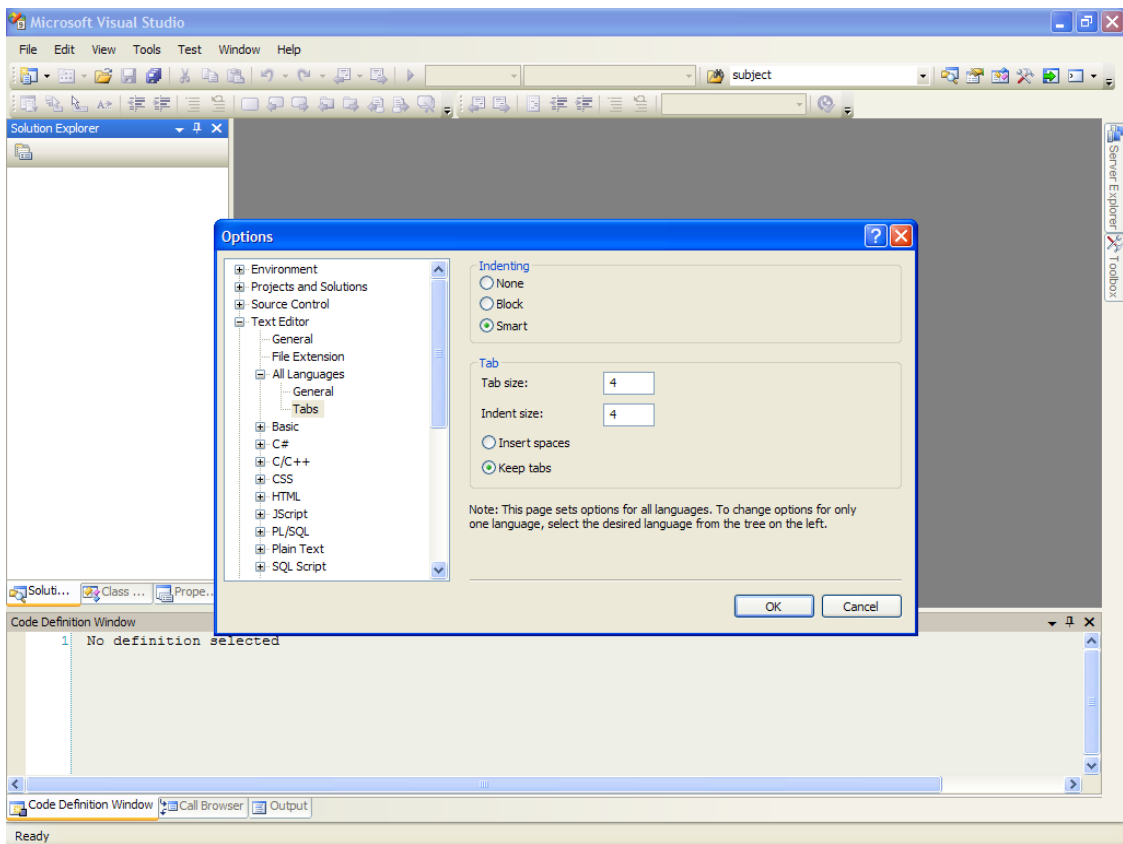
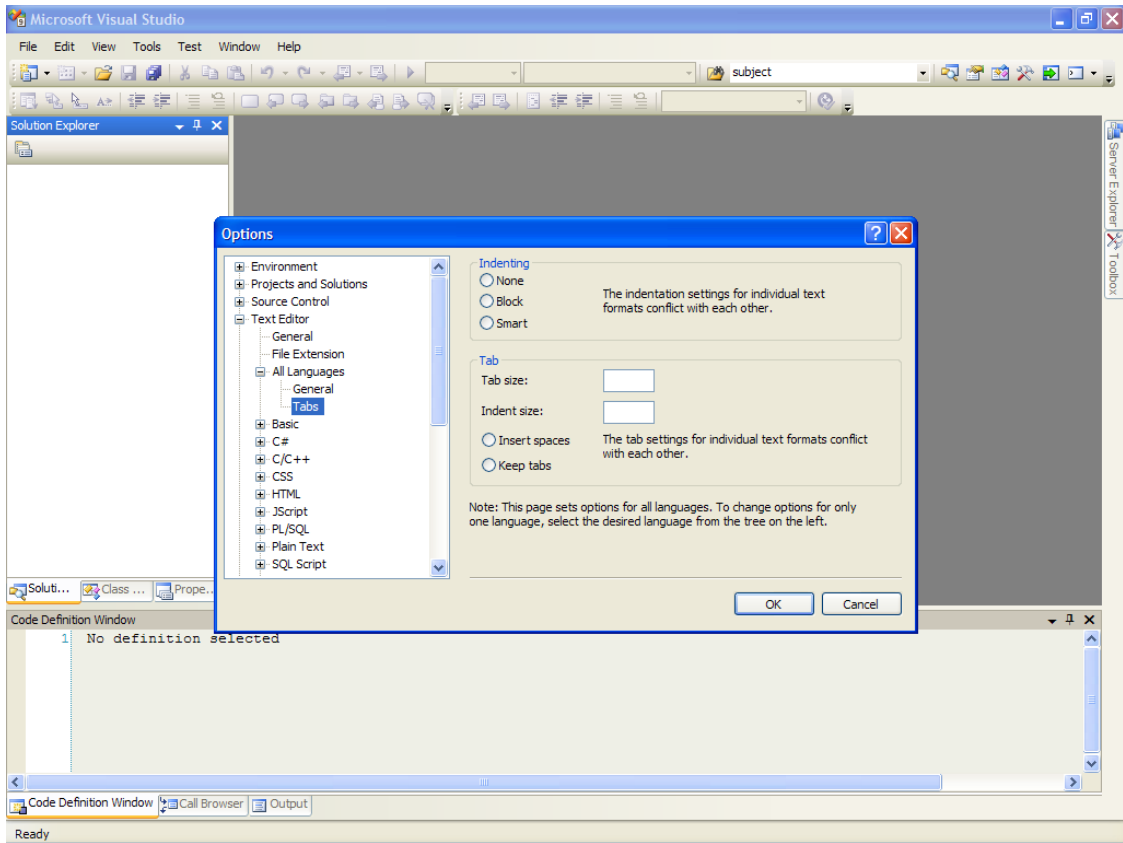
2 - در مرحله بعد تنظیمات مورد نظر خود را از طریق option → tool اعمال می نماییم. Snapshot مربوط به این تنظیمات در ذیل آمده است.

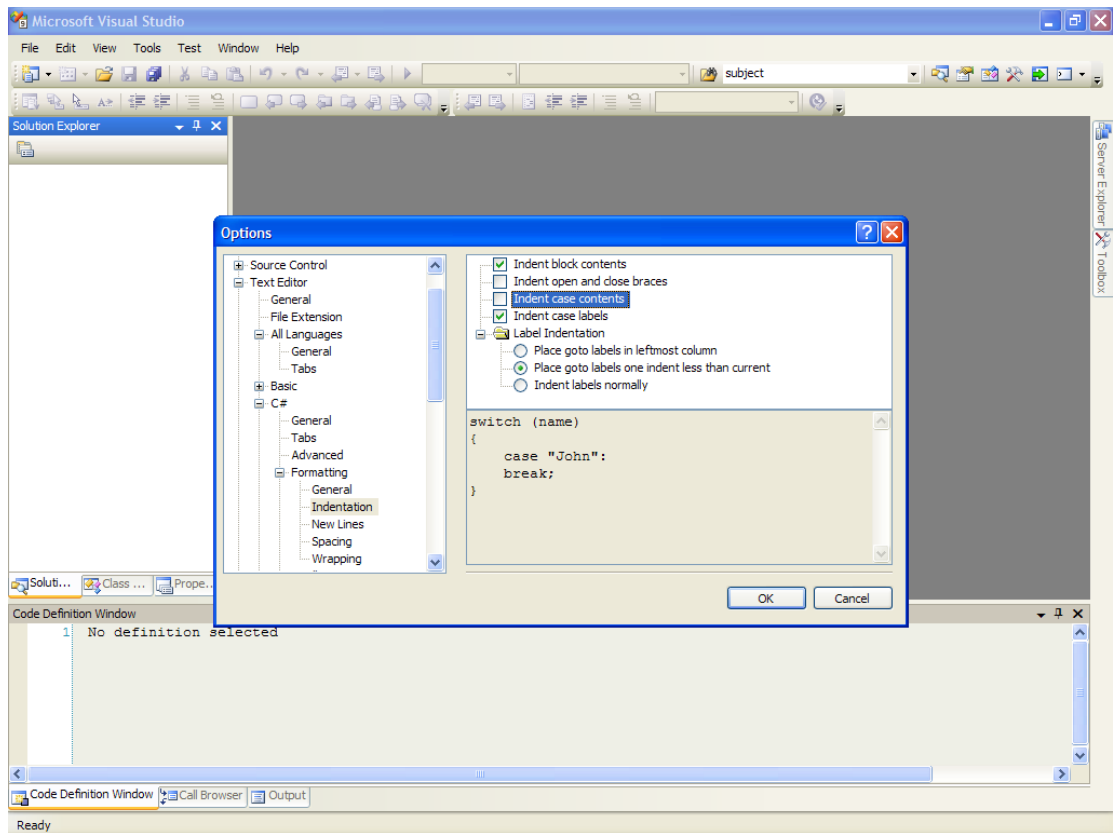
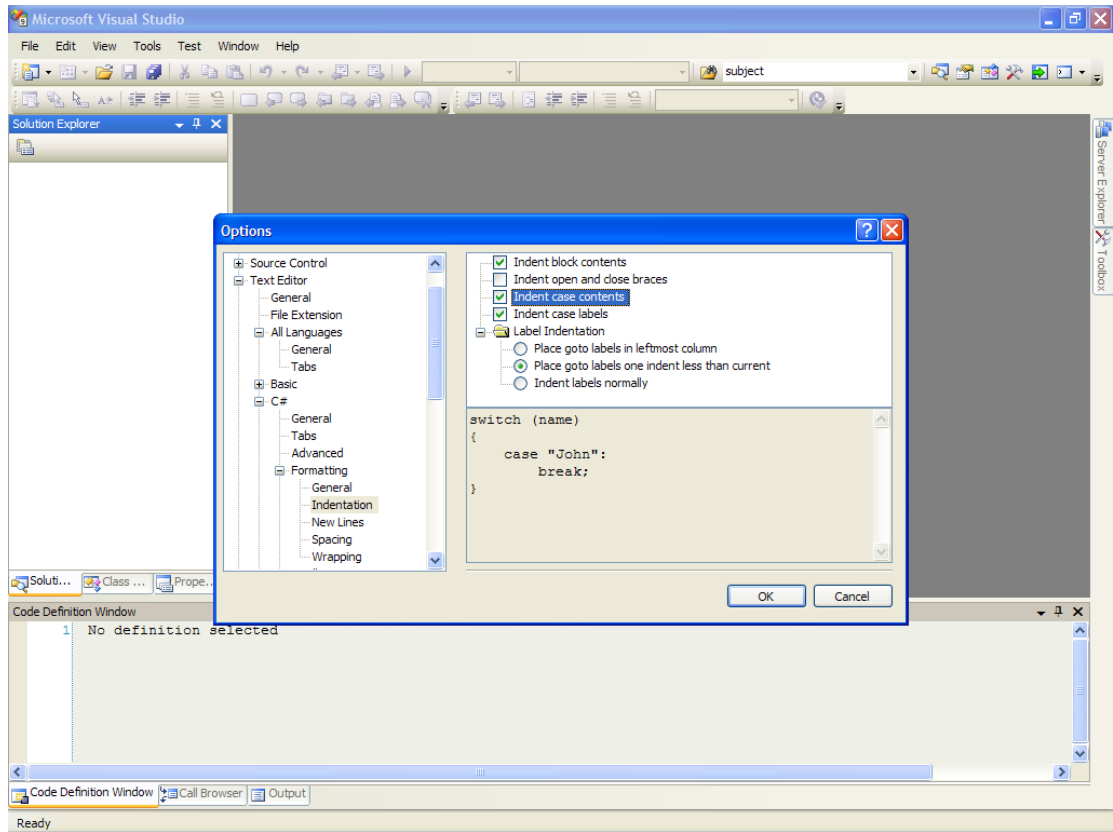
نکته: هدف از این تنظیمات، بهینه سازی محیط کدنویسی می باشد. یک سری قوانین و یک سری تاکتیک داریم. مثلاً آخرین سبک کدنویسی می گوید، همیشه برای بلاکهای خود آکولاد باز و بسته بگذارید. در زبانهای C based می توانید برای Case ها آکولاد باز و بسته نگذارید. خود محیط Visual studio نیز بر این اساس برای Case ها عملیات indent را انجام نمی دهد. در حالی که می خواهیم indent ها وجود داشته باشد. این تنظیمات را یکبار انجام می دهید و export می کنید به عنوان یک فایل نگهداری می کنید و هر جا خواستید آن را import می کنید. یا مثلاً در قسمت HTML Tag دو تا باگ داریم، یکی برای تگ form که indent نمی کند. یک گزینه داریم به نام format the whole document یا همان (ctrl+k+d) که کد ما را اصلاح می کند.

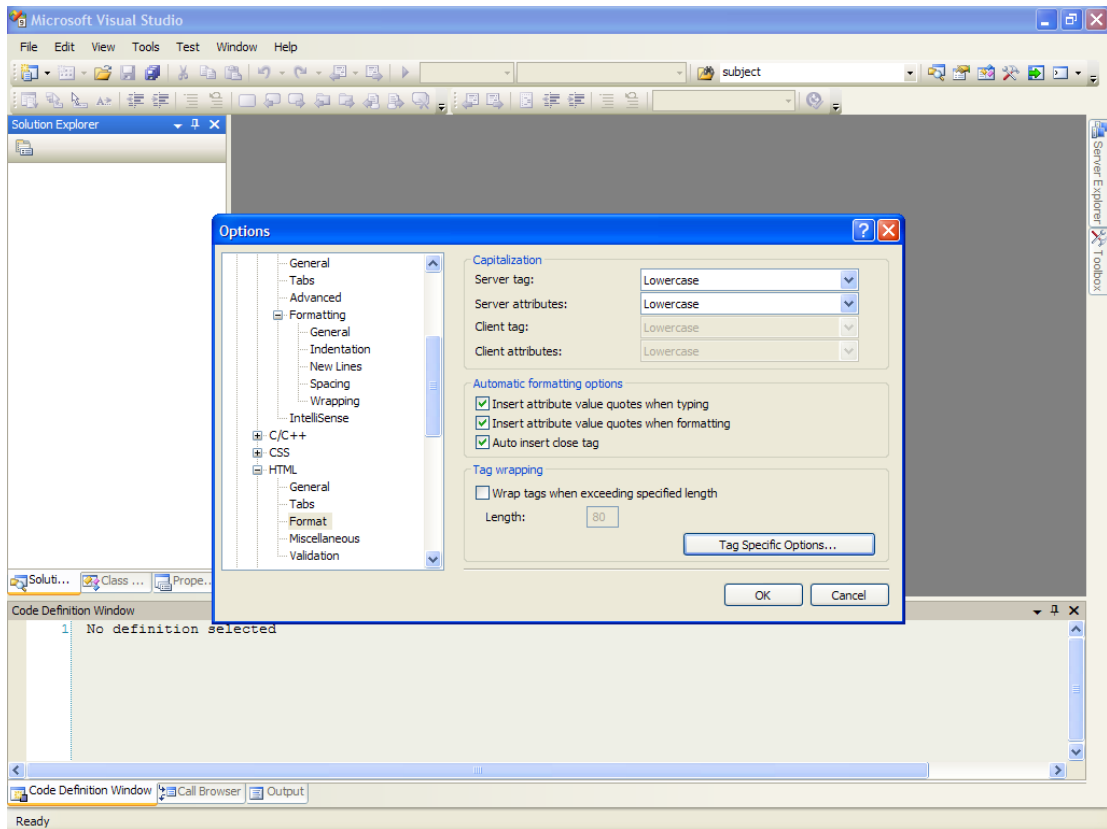
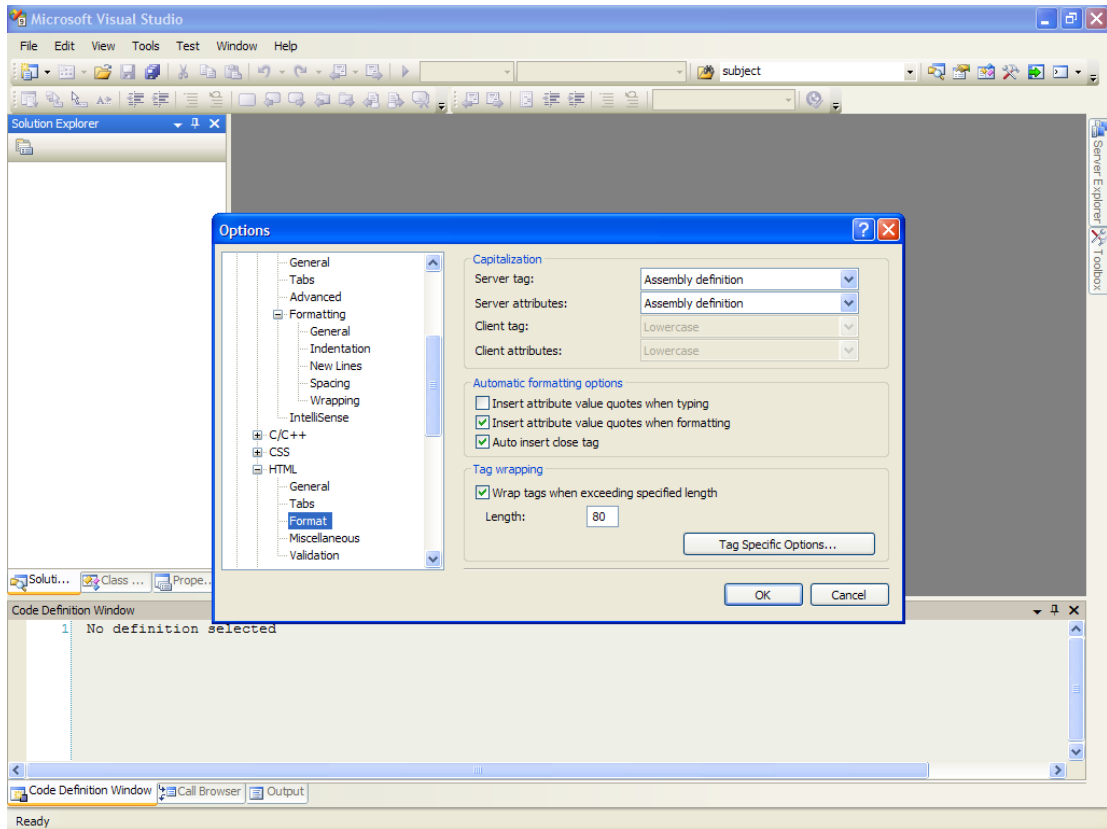




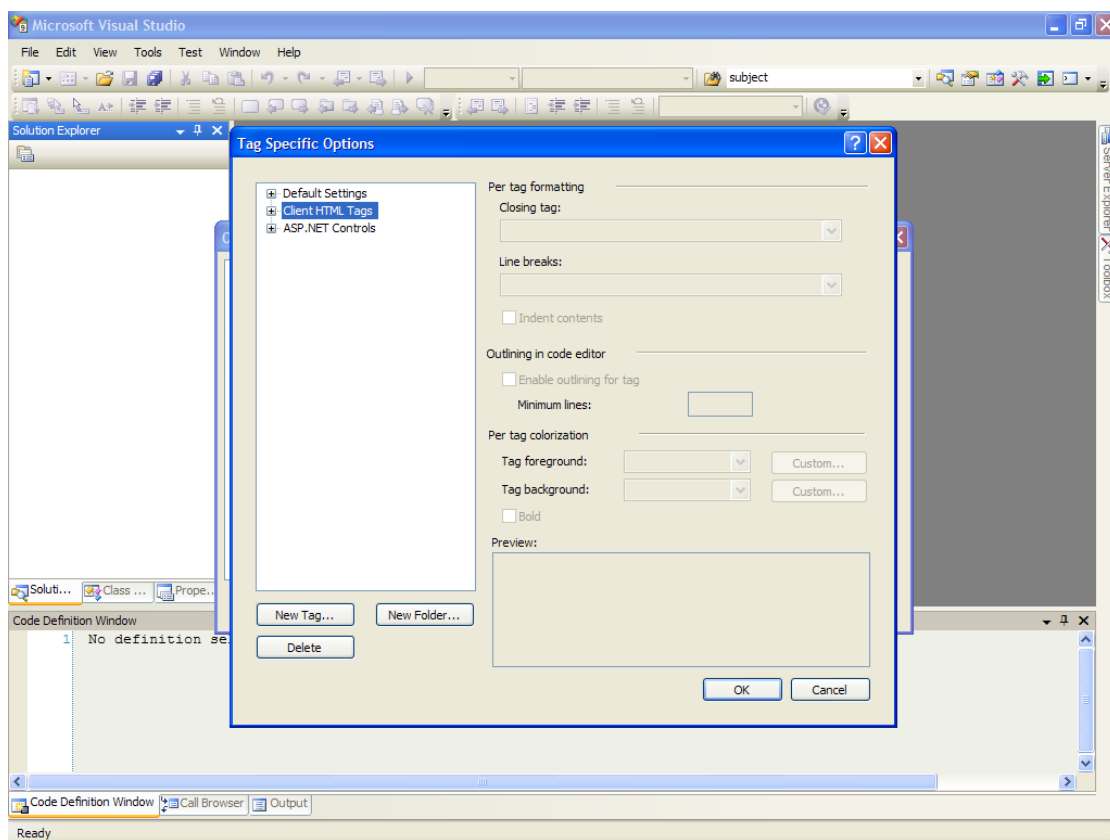


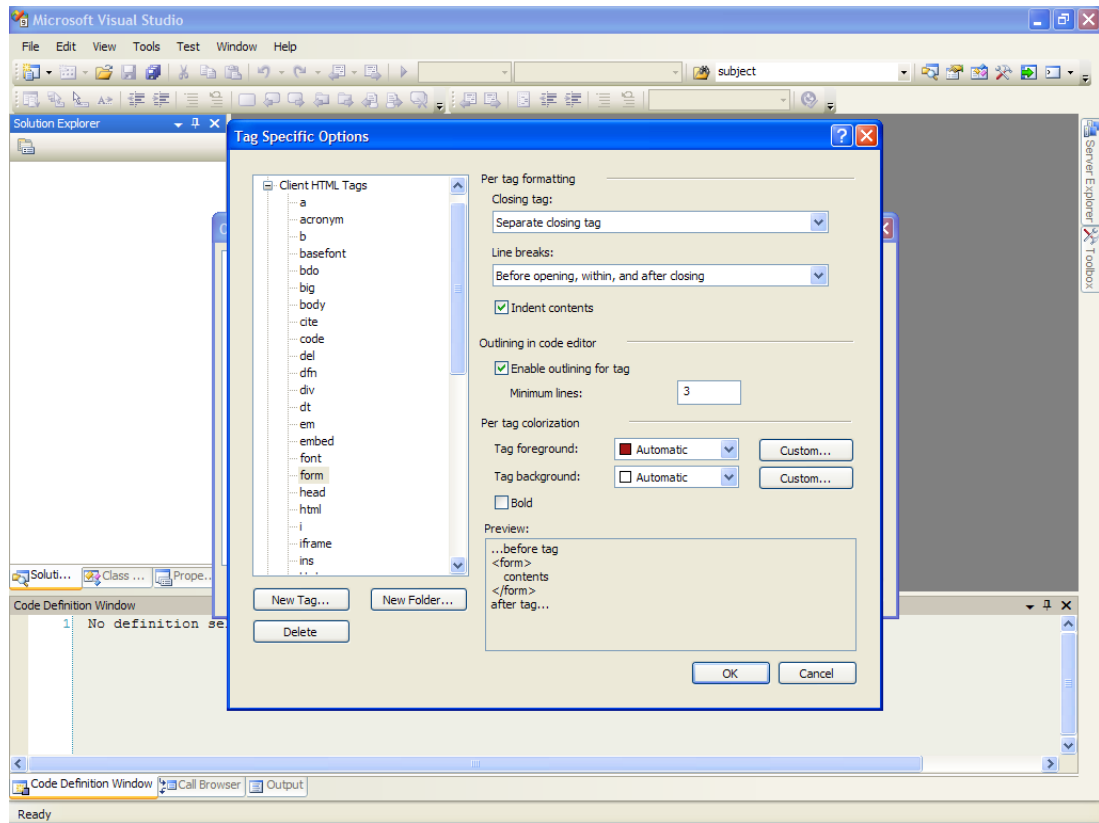
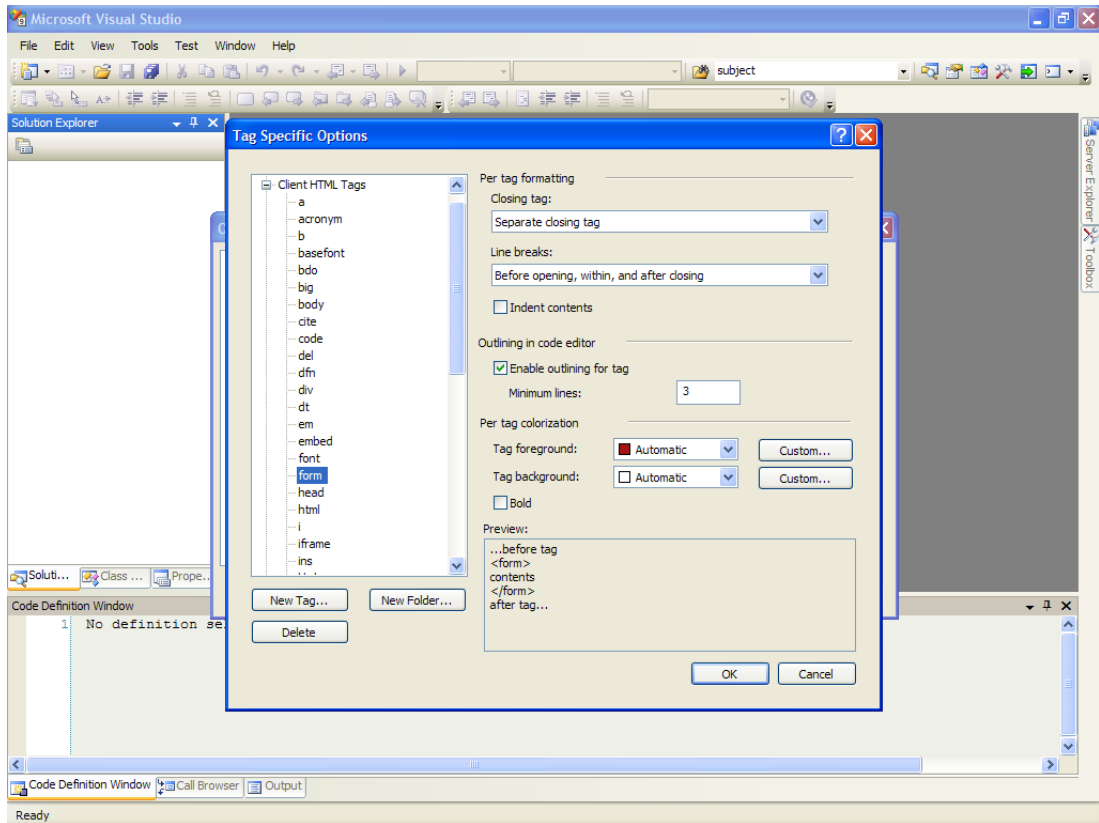


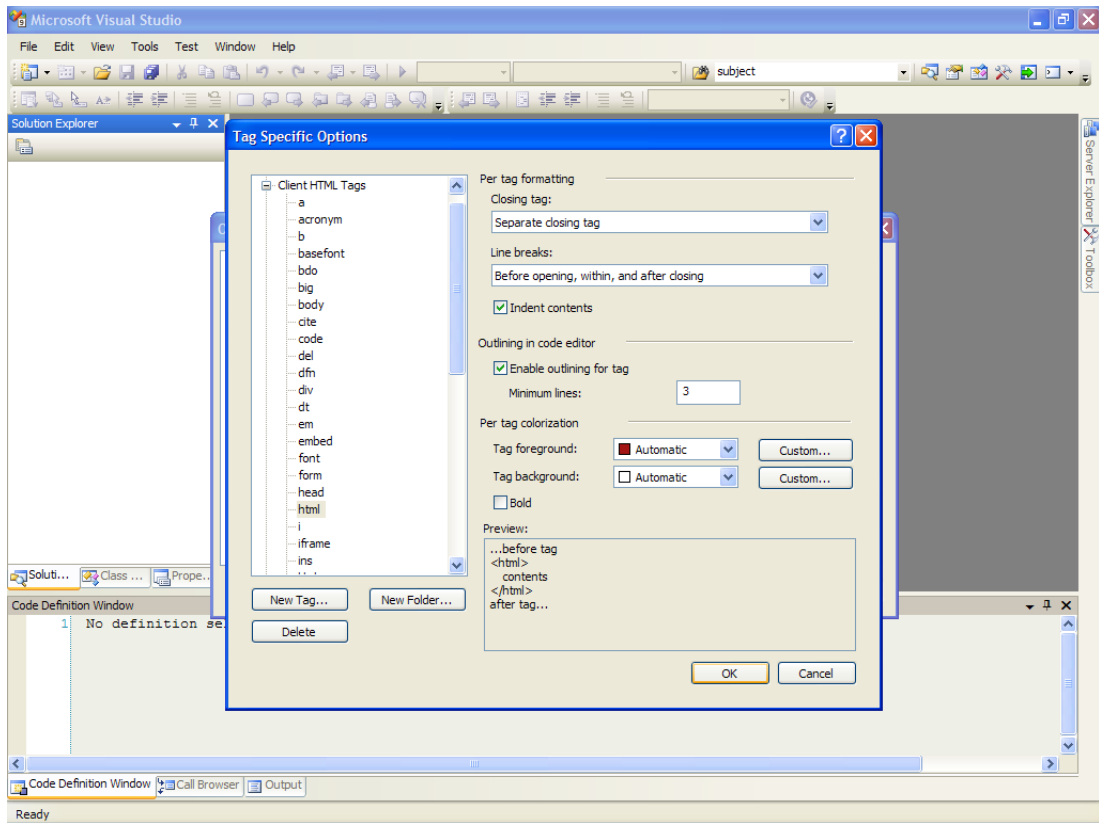
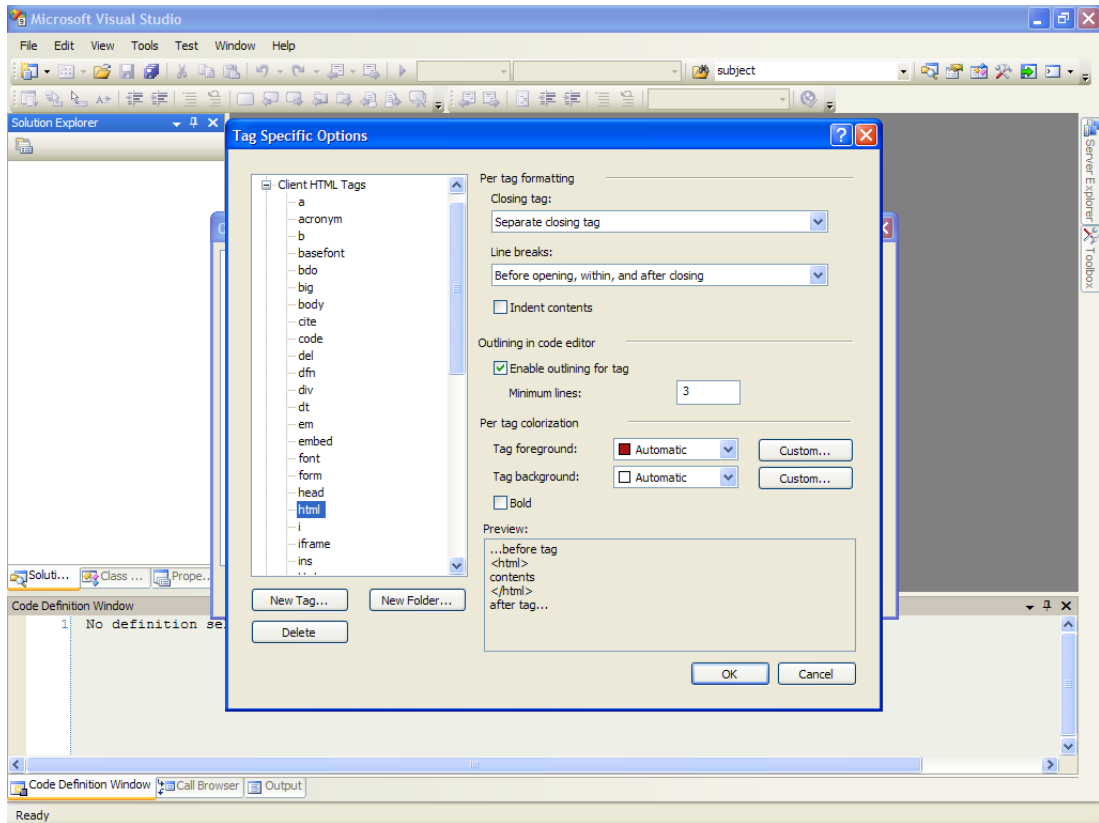


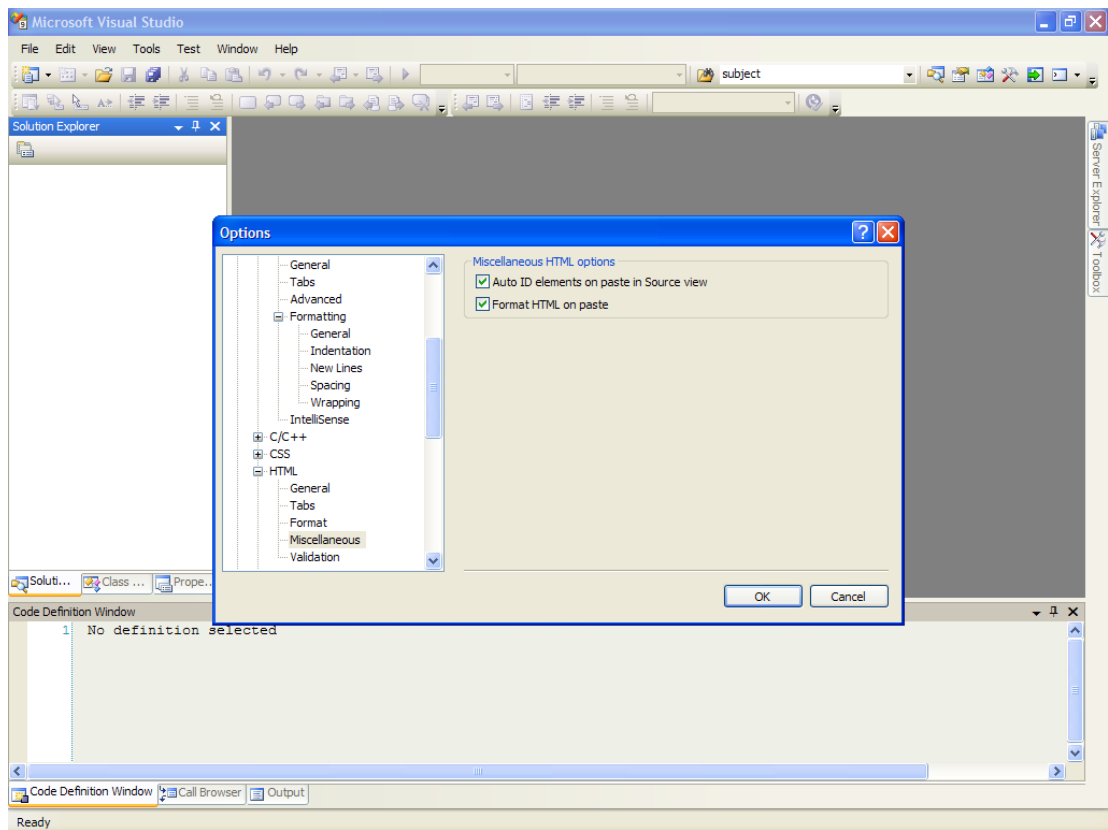
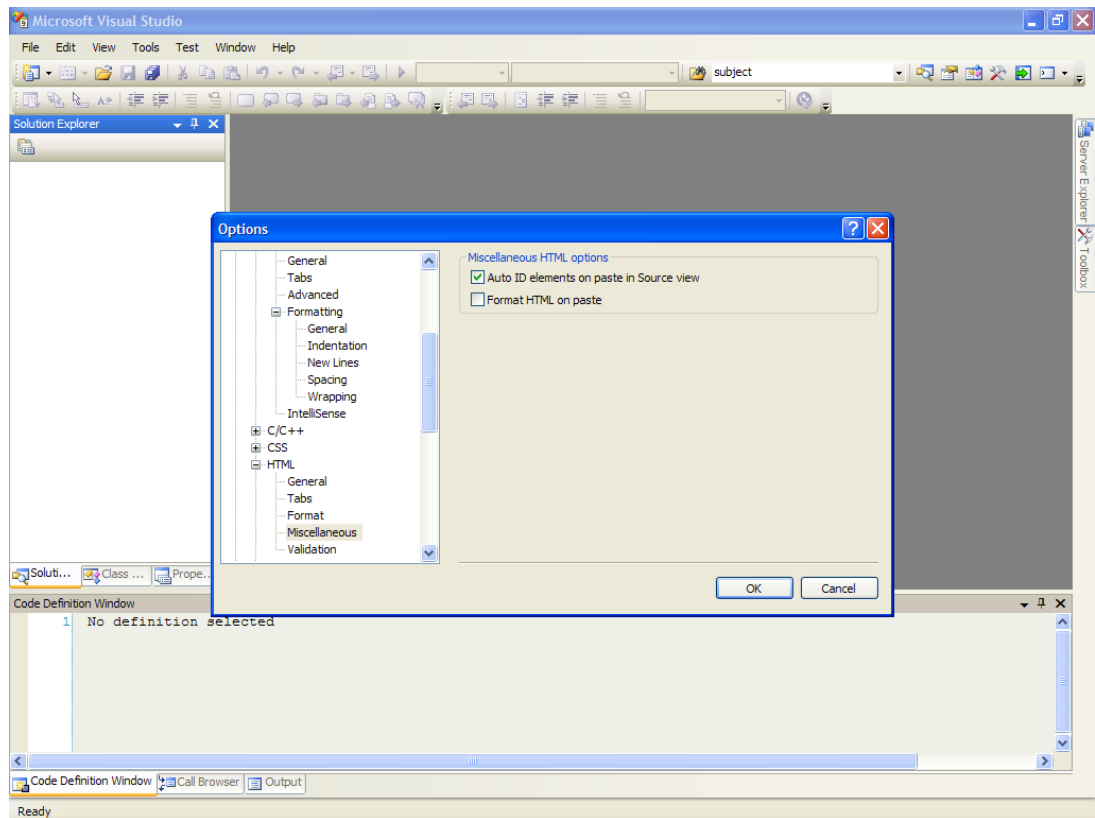


با کلیک بر روی دکمه Tag Specific option... صفحه ذیل بر روی صفحه قبلی باز می‌شود.

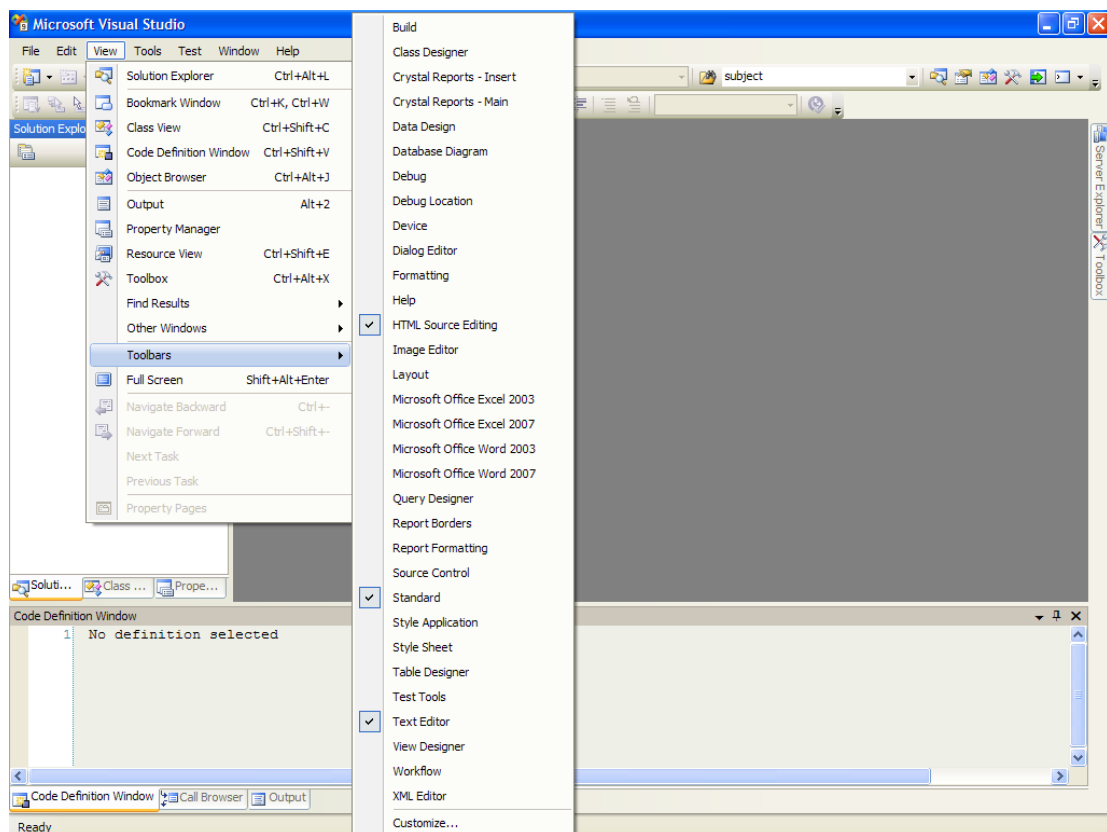




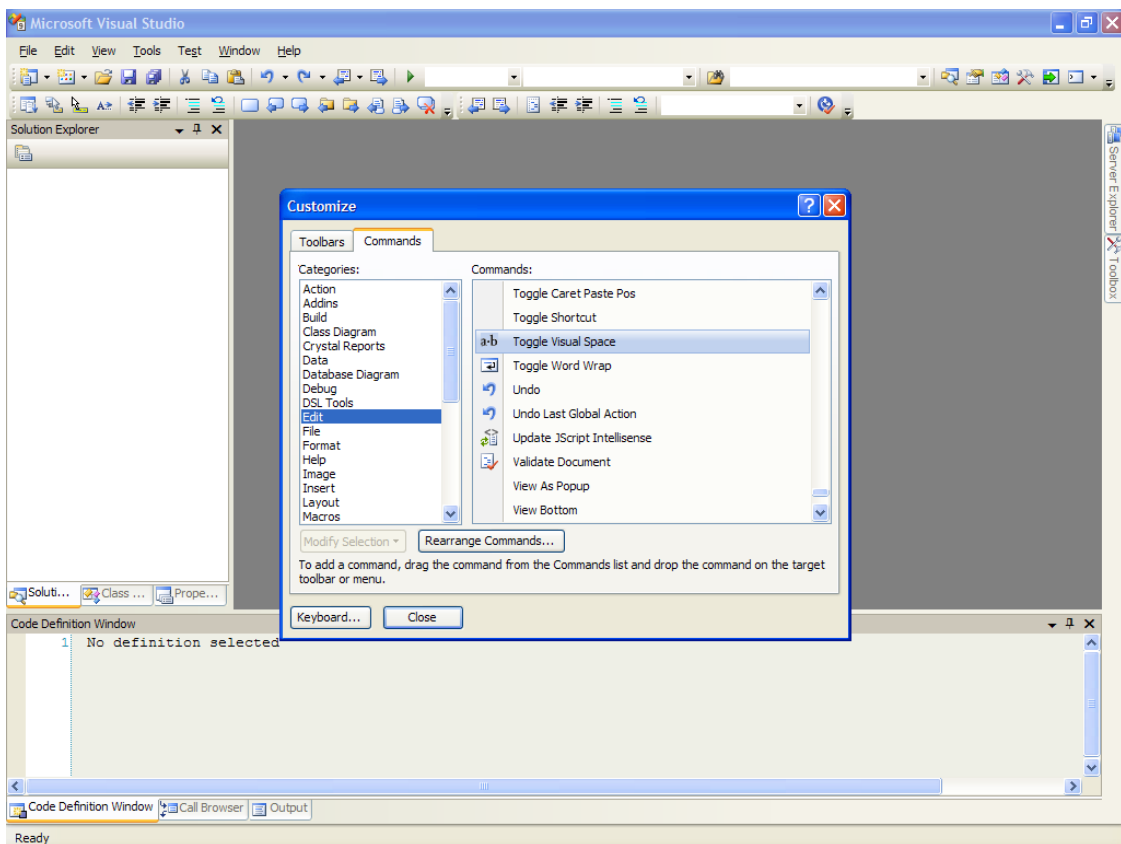
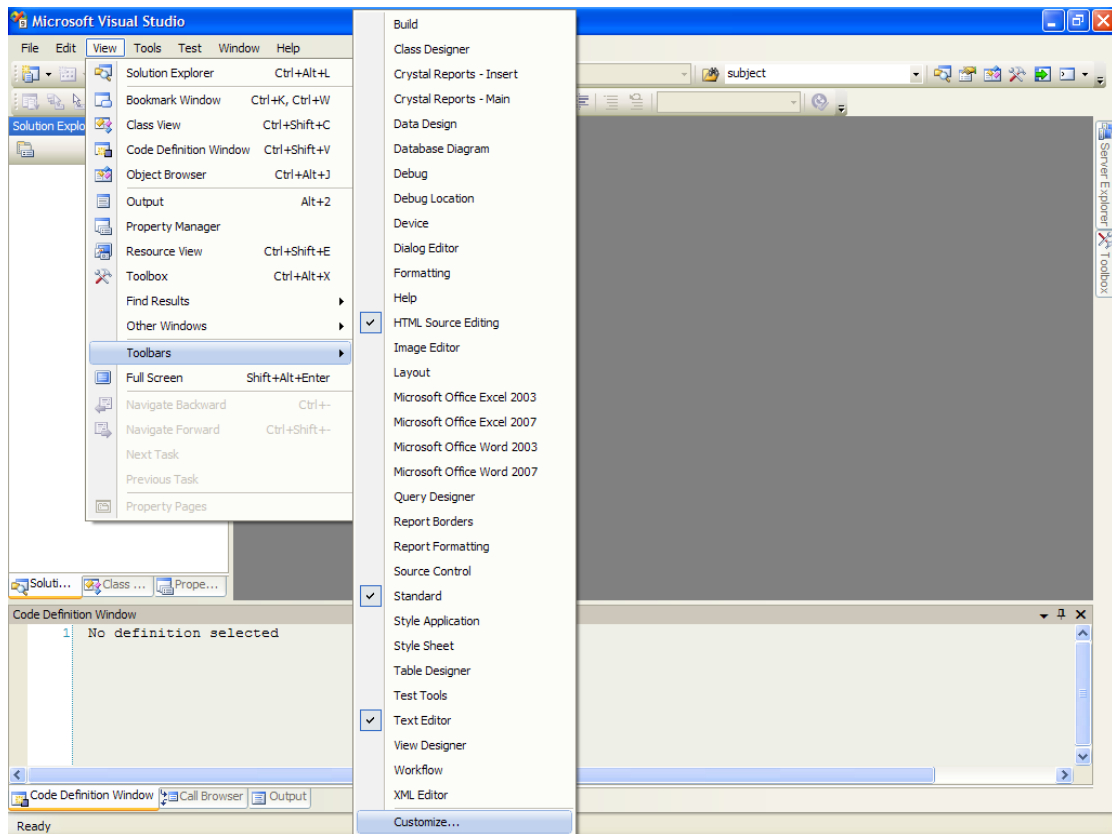




3 - 3 toolbar با نامهای HTML Source Editing ، Standard و Text Editor را اضافه می‌نماییم. از طریق مسیر
View → toolbars



4 - اضافه نمودن ابزار جدید به toolbar ها، تنظیم دیگری است که باید انجام گردد. در visual Studio کدهایی که می نویسیم به صورت پیش فرض مجهز به A.b. نیستند. یعنی اینکه به جای space و tab فلش و نقطه رنگی می گذارد. پس باید آن را اضافه کنیم. در 2010 اگر بخواهیم به یکی از toolbar ها موردی اضافه کنیم. دکمه add commands را خواهیم زد. در این قسمت می رویم و a.b را انتخاب می کنی. در 2008 باید drag & drop کنیم در 2010 انتخاب و ok و سپس سر جای خود قرار می گیرد.



نکته: برنامه foxit reader که جهت نمایش و ویرایش فایل‌های PDF است و کل آن 8 تا 9 مگابایت بیشتر نیست و free است و می‌توانید به آن متن اضافه کنید و آن را highlight کنید.

نکته: یک فایل iso یک image file است، بعضی ها اول image را write بعد نصب می کنند. پس فقط فایل ایزو را mount کنید به آن فایل و مثل یک CD عمل می کند. برنامه Virtual Clone Drive برای تبدیل فایل image به یک درایو مجازی می باشد.

نکته: در هنگام نصب visual Studio 2010 ورژن expression می توانید فقط قسمت Web development آن را نصب نمایید.

نکته: ماکروسافت فقط donetframework 2 و 3.5 با Service pack 1 و 4 را پشتیبانی می کند.

انواع پروژه‌ها در Vistual Studio 2008 & 2010

در VS 2010 دو نوع پروژه می توان اضافه نمود.

ASP.Net Web Form MVC هم وجود دارد که بعد از SP1 مربوط به VS2008 به بازار آمد و آقای اسکات کسی است که کلیه تفکرات ماکروسافت در این زمینه مربوط به او می باشد. آقای اسکات در یک ساعت تفکر MVC را پایه گذاری نموده است. MVC به هیچ عنوان قویتر و عجیبتتر از Web Form نیست و بسته به سلیقه یکی را انتخاب نمایید. در مورد خود Web form دو روش وجود دارد. یکی تولید Web application و دیگری تولید Web site است. روش Web application ای روش file system ای است و یک simulator iis اجرا می کند. ولی وقتی که Web site را انتخاب می کنید می پرسد که HTTP یا FTP یا file system ای کار می کنید.

ما در اینجا وب سایت را یاد می گیریم چون گرفتاری وب سایت بیشتر است و اگر آن را یاد بگیرد بعداً می تواند web application درست کنید. مثلاً در web application هر جایی خواستید کلاس ایجاد می کنید ولی در وب سایت باید حتماً یک دایرکتوری App-code داشته باشید و درون آن ایجاد کنید.

ما روش IIS را نیز به روش File system ترجیح می دهیم، دلیل آن این است که گرفتاری آن بیشتر است و اگر به مشکل خوردیم می توانیم در سر کلاس متوجه آن شده و آن را حل نماییم.

اگر در خصوص نسخه net framework سوال پرسید و سوال کرد که از چه نسخه ای استفاده کنم، بگوئید 3.5. چون گیر و گرفتاری آن از نظر security کمتر است.

وقتی که IIS را نصب می کنید در درایو ویندوز، یک سری دایرکتوری ایجاد می شود. برای تشخیص درایو وینوز یک Cmd بزیند و در ابتدای آن درایو ویندوز را نشان می دهد. در C که الان درایو ویندوز ما است، یک دایرکتوری با نام Inetpub و درون آن نیز WWWroot تشکیل می دهند. زمانی که با file system کار می کنید هیچ توصیه خاصی وجود ندارد و هر جا که دوست دارید پروژه خود را ایجاد کنید. وقتی که HTTP کار می کنید اولاً باید IIS نصب شده باشد و حتماً باید آدرسی که می نویسیم باید با <http://localhost> شروع شده باشد. هیچوقت داخل خود localhost وب سایت خود را ایجاد نکنید و یک لایه جلوتر بروید در خود localhost درست نکنید، بیشتر از یک لایه هم به هیچ عنوان جلوتر نروید، چون IIS به مشکل بر می خورد. از نقطه و space هم استفاده نکنید. پس در اینجا از آدرس ذیل استفاده می کنیم.

<http://localhost/abfa>

تبدیل وب سایت به وب application یا MVC کاملاً پرهزینه است ولی تبدیل وب سایت IIS ای به فایل سیستمی و یا برعکس پرهزینه نیست و زحمت زیادی ندارد، ولی مسلماً یک پروژه فایل سیستمی باید فایل سیستمی باز شود و برعکس. وقتی که وب سایت درست می کنید، تعدادی فایل می سازد که ابتدا آن را پاک می کنیم و به مرور اگر لازم شد، اضافه می کنیم. طراحی Layout سایت را در ابتدا انجام می دهیم و یک صفحه Template درست می کنیم و بعد شروع می کنیم به Copy/paste و ... مثلاً از Master/page در ابتدا استفاده نمی کنیم. پس روشها را به صورت سلسله مراتبی جلو می رویم.

ما یک Solution داریم و یک Project. روی Solution از Add new item استفاده نکنید بلکه بر روی پروژه اینکار را کنید. پس Project → right click → add new item → web form را انتخاب کنید. یک سری از نامها را تغییر ندهید، یکی صفحه Login و دیگری صفحه default. پس صفحه home page را حتماً Default و صفحه ورود را Login بگیرید و در دایرکتوری هم نگذارید در داخل شاخه اصلی بگذارید. نه اینکه اشکال داشته باشد ولی به یک سری مشکلاتی بر می‌خورید که حل آنها ارزشی ندارد، پس استاندارد عمل کنید. فعلاً در این مرحله دو تیک Place code و Master page را نزنید. خیلی مفید هستند ولی ما مرحله به مرحله حرکت می‌کنیم و الان نوبت آن نیست. هدف ما در این دوره، صرفاً تولید یک سایت نیست. تولید یک سایت استاندارد با قابلیت اینترنتی شدن با حداقل Configuration است.

چگونه یک سایت خوب طراحی کنیم؟

چگونه یک سایت استاندارد درست کنیم. اینکه یک سایت فقط در IE درست دیده شود، هدف ما نیست. اولین قدم ما تولید سایت‌های استاندارد است. استاندارد دو بخش دارد. یکی بحث شکلی و تکنیکی است و دیگری فلسفی، مفهومی و ماهیتی است. منظور از شکلی و تکنیکی یعنی این که اگر یک سری تگ اشتباه به کار ببرید و یک سری attribute را نگذارید، درست نمایش داده نمی‌شود. مثلاً شما حق استفاده از تگ Marque را ندارید یا Blink همینطور، اولی مال IE و دومی مال Netscape است و در مرورگرهای دیگر درست دیده نمی‌شود. پس چه تگ‌هایی را استفاده کنیم، مسئله این است.

طراحی استاندارد از نظر شکلی و تکنیکی

در سایت JQuery یکی از ویژگی‌هایی که دارد این است که Cross browser است، یعنی در همه Browser ها کار می‌کند. سایتی Cross browser است که در IE6 به بعد و در Chrome و در Safari3 به بعد، Opera 9 به بعد دیده شود. ولی اینکه در netscape دیده نمی‌شود اصلاً مهم نیست. یک سری Browser در بعضی از مراکز تحقیقاتی درست می‌شود. بعضی از Browser ها اصلاً جدید نیست. مثلاً MSExplorer که همان IE است، یعنی موتور آن همان موتور IE Browser است. یک پنجره سفید در IE دیده می‌شود که مهمترین Component یک IE است که Microsoft به عنوان یک component تولید نموده است. اگر از این موتور استفاده کنید و اطراف آن را طراحی کنید، در واقع مرورگر جدیدی طراحی نکرده‌اید. بعضی ها برای خودشان مرورگر طراحی می‌کنند و کار سختی هم نیست. چون مرورگرهای استاندارد مرتباً toolbar های جدید به آن اضافه می‌شود و به مرور نصف صفحه را toolbar ها اشغال می‌کنند ولی مرورگری که خودتات بنویسید اینگونه نیست. به عنوان یک آدم حرفه ای همه نرم‌افزارهای (browser) ذیل را باید داشته باشید، تا بتوانید تست کنید. ولی مسئله این است که دو Browser از یک نوع با ورژن مختلف که نمی‌توان نصب کرد.

خیلی از سایتها ما را فیلتر کرده‌اند و نمی‌توان از روی سایت آن، دانلود نمود. پس چه کار کنیم. یک سایت به نام FileHippo است که یک سایت رسمی در دنیا است. این سایت 720 امین سایت دنیا است و کلیه تهیه‌کنندگان گردن کلفت دنیا با این سایت تعامل دارند و با سایت‌های کرکی تفاوت دارد. این سایت تقریباً ورژن همه Software ها را دارد و اجازه دانلود می‌دهد. مثلاً Firefox را Search می‌کنید و آخرین ورژن و آرشیو ورژن های آن را می‌بینید. مثلاً در Firefox از دو به بعد است، پس Firefox 2 و Firefox 5 را دانلود می‌کنید.

در خصوص ورژن پایین، باید حتماً Release آن را بردارید. در خصوص ورژن بالا، هر چه باشد نسخه Beta یا Candidate یا هر چه باشد باید بر دارید. و در آن تست کنید. و البته هر کدام را در دو نسخه 32 بیتی و 64 بیتی باید دانلود نمایند. چه کنیم که بر روی سیستم همه اینها باشد و همه با هم نصب شود. مفهوم Virtualization بسیار مطرح است. سه محیط virtual PC مطرح داریم. یکی VMWare است که پولی است و مسلماً open source هم نیست و ---- که مجانی است ولی Open source نیست. Virtual Box است که مال Oracle یا Sun است و هم free است و هم open source است. این نرم

افزار را هم از روی همین سایت دانلود نمایید و استفاده کنید. آخرین ورژن را بر روی سیستم خود نصب کنید و بر روی Virtual می‌توانید minimum ها را نصب نمایید.

نکته: اگر بخواهید با سیستم عامل اندروید یا ویندوز 8 کار کنید، از این Virtual ها استفاده کنید.

پس اولین هدف تولید صفحات استاندارد است. استاندارد یک بحث تکنیکال دارد که باید در Browser درست دیده شود.

طراحی استاندارد از نظر فلسفی و مفهومی

ولی بحث بعدی فلسفی و مفهومی است. حتماً باید تگ HTML داشته باشیم و حتماً باید یک دانه باشد. در مورد , Head body و ... در همان استاندارد می‌گوید که باید در Head باید یک تگ Title بگذارید. این بحث فلسفی است و نه تاکتیکال ، در یک سایت تب های مختلف باز می‌کنید و می‌بینید که حتماً Untitled است و این بسیار بد است. برای بعضی از تگها یک سری attribute ها Required است. ولی alt یک صفت لازم و واجب است. Alternative مخفف Alt text است و به معنی tooltrip نیست یعنی به هر دلیلی اگر عکس دانلود نشد، alt در درون آن نوشته می‌شود. باید همه عکسها alt داشته باشد، که اگر عکس دانلود نشد، خواننده سایت بداند، چه چیزی در آن بوده است. IE لطف کرده و alt را به عنوان Tooltip هم نشان می‌دهد. ولی در حالت عادی اگر Tooltip می‌خواهید باید از تگ Title استفاده کنید.

سایت باید بسیار سبک باشد، چون می‌خواهیم سایت سریع download و باز شود. بعضی از افراد در وب کارهای مربوط به ویندوز را انجام می‌دهند. مثلاً Component ای به نام Teleric استفاده می‌کنند تا وب را به ویندوز نزدیک کنند، این سنگین و پر از باگ است. اگر نمی‌خواهید UI مربوط به وب استفاده کنید، پس از ویندوز Based استفاده کنید. برای پروژه‌های اینترنت خوب است، ولی برای برنامه های تحت وب جوابگو نیست و این موجودات مثل Teleric سایت را بسیار لخت و کند می‌کند.

در ویندوز از Component های کرک شده استفاده می‌کنید ولی در مورد وب، اگر host آنطرف آب باشد، host را بلاک می‌کنند و با پلیس Interpol سرو کار دارید. البته host ایران باشد مشکلی نیست. البته Host ایران اصلاً پیشنهاد نمی‌شود. کل پهنای باند ایران 2 گیگابایت است. پس در ایران اصلاً جوابگو نیست. در ضمن host داخلی را می‌توان به راحتی dos attack کرد و به مشکل بر می‌خورد. پس تولید سایت استاندارد و سبک هدف ما است.

تولید سایت با حداقل configuration

هدف بعدی تولید سایت با حداقل configuration است.

در مورد فضای host یا فضا Shared یا Dedicate است. برای سازمان ما باید فضای share استفاده کرد. چون dedicate خیلی گران است. در Shared 300 یا 200 تا سایت مختلف روی یک host است و هزینه آن ماکزیمم سالی 100 هزار تومان است با کلی امکانات ولی در dedicate حداقل آن ماهی 250 هزار تومان است با حداقل امکانات. اما در مورد فضای Shared نمی‌توان بگویید این Patch را روی SQL اجرا کن. پس Configuration خود را حداقل نمایید. برای اینکه این استانداردها را مطالعه کنید. سایتی داریم با عنوان www.w3.org. تمام استانداردهای وب که استاندارد شدند یا در حد بتا هستند و می‌خواهند استاندارد شوند و خط مشی های وب را در این سایت رقم می‌زنند. در این سایت سه چیز خیلی مهم است. اولاً فایل ebook مربوط به HTML 4.01. یک نسخه از این را پرینت و صحافی کنید.

هدف ما تولید صفحات xhtml 4.01 است یعنی هم از استاندارد Html 4.01 تبعیت کند و هم از XML 1.0. کتاب بعدی XML 1.0 است که باید از آن نیز تبعیت کنید و صحافی کنید و یک نسخه آن را به عنوان مرجع نگهداری کنید.

در این سایت یک سری Validator وجود دارد. چک کنید که آیا این سایت شما، یک سایت Valid است یا خیر. پس در Search همین سایت بنویسید Html validator خودش قسمت search را انجام می‌دهد. به سه روش می‌توانید صفحات خود را چک کنید.

- 1 - آدرس URL یا URI خود را بنویسید.
- 2 - یا سایت مورد نظر خود را upload کنید.
- 3 - از روش Direct input یا همان copy/paste استفاده نمایید و Source صفحه وب خود را در قسمت مربوطه کپی کنید.

خوبی آن این است که خطا و Error را می‌گوید و دلیل آن هم می‌گوید که می‌تواند جنبه آموزشی داشته باشد. مثلاً یک خطایی که در مورد این سایت می‌گوید، این است که برای Img ، صفت ALT نگذاشتید.

بالا بردن Ranking سایت

با بالا بردن Ranking سایت از جنبه‌های مختلف. مثلاً سرویس ای که می‌گوید، id مربوط به yahoo messenger الان واقعاً در چه وضعیتی است؟ اولین سایت که فقط این سرویس را ارائه نموده است الان 800 میلیون دلار ارزش دارد. تولید سایت‌های چند زبانه را یاد می‌گیریم. یک روش آن این است که می‌خواهید هم عربی و هم انگلیسی داشته باشد. پس سایت را فارسی طراحی می‌کنید و بعد که قسمت فارسی سایت نهایی شد، بعد Copy/past و در یک ورژن عربی و در دیگری انگلیسی می‌کنید. این کار درست نیست. چون عملاً سه سایت وب دارید و موتور شما دچار مشکل می‌شود. پس به گونه ای طراحی کنید که سایت یکی باشد و layout تغییر کند.

اگر سایت شما مشکلی ندارد W3C تایید می‌دهد و شما می‌توانید بر روی سایت خود قرار دهید و این کلاس سایت شما را بالا می‌برد. به هیچ عنوان بدون گرفتن تایید W3C علامت تایید آن را بر روی سایت خود نگذارید. به این دلیل که یک درسی به نام HTTP referrer داریم که تشخیص می‌دهد که User از کجا وارد سایت شما شده است. اگر کسی واقعی باشد، با کلیک بر روی این Icon این مسئله را تایید یا رد می‌کند.

در این دوره با دو سری موجودات سر و کار داریم ، یکی HTML و دیگری Server side هستند و با ASP شروع می‌شوند. همه این موجودات بر روی مرورگر کاربر تبدیل به HTML می‌شوند، چون Browser که ASP بلد نیست ، پس این کدها Render می‌شوند به HTML . Microsoft ادعا کرده است که من کد ASP را به HTML استاندارد Render می‌کنم و در 90 درصد اوقات اینگونه است ولی در چند درصد اوقات خود Microsoft خودش به صورت استاندارد render نکرده است. به طور کل ترکیب دو چیز برای ما مهم است یکی XHTML 1.0 و CSS 2.0 .

البته XHTML 1.1 هم آمده است، که آن هم استاندارد است ولی خود IE8 هنوز HTML 5 را پشتیبانی نمی‌کند، بنابراین دلیل اینکه شاید از آخرین ورژن استفاده نمی‌کنیم و از استاندارد XHTML 1.1 استفاده می‌کنیم، این است که می‌خواهیم از ورژنی استفاده کنیم که هم استاندارد باشد هم browser compatibility داشته باشد. یعنی در تمام مرورگرهای استاندارد پشتیبانی شود.

برای تولید برنامه تحت وب از چه زبانی استفاده کنیم؟

این قسمت بیشتر مربوط به مدیران و تصمیم‌گیرانی است که می‌خواهند دلایل منطقی برای انتخاب خود داشته باشند. از چه زبانی باید استفاده شود. net. یا Php یا Jee یا ...

در دنیا تنها زبانی که با net. قابل مقایسه است (در حد egoverment) ، همان JEE است. PHP و ... این کار را نمی‌کند و قابل مقایسه نیست. البته صلاحیت این زبانها رد نمی‌شود ولی قابل مقایسه نیستند، ولی سایتی مانند Phacebook خودش با php طراحی شده است.

برای پروژه Egoverment بیشتر net. و JEE قابل قبول است. ولی در مقام مقایسه این دو زبان می‌توان گفت:

- پروژه‌های .net ای بسیار سریعتر هستند (حدود 4 برابر) . Bench mark مربوط به Pet shop این را تست کرده است.

- پروژه‌های .net 7 تا 10 برابر Source کمتری می‌خواهد.

البته مسلماً .net نیز معایبی دارد.

علی‌رغم گفته‌های ناپخته عده‌ای، پروژه‌های .net از نظر Security مشکلی ندارند. بزرگترین پروژه ایران، سیستم مناقصه‌گران ایرانی حدود 4300 کاربر جاری در ساعت 4 بعداز ظهر دارد. سیستم معاملات و مناقصه‌گران جمهوری اسلامی از .net استفاده شده است و کلی hacker از کشورهای مختلف استخدام و این سایت توسط آنها چک شده است و کاملاً امن و Secure می‌باشد.

امنیت در لایه‌های مختلف باید پیاده‌سازی شود:

- در حوزه سخت‌افزار router را secure کنید در بحث سخت افزار هم امنیت را رعایت کنید که این دو تا به شما مربوط نمی‌شود.

- در حوزه مهندسی نرم‌افزار هم تکنیکهای امنیتی وجود دارد که باید اینها را رعایت کنید. و این تکیه به شما مربوط است.

.net هم مثل همه تکنولوژیها می‌تواند ریسک داشته باشد و یا اینکه فوق‌العاده باشد.

دو پارامتر برای رتبه‌بندی سایتها داریم. Page google rank که مربوط به هر صفحه است و اهمیت صفحه شما در اینترنت را نشان می‌دهد و هر چه این عدد بزرگتر باشد یعنی صفحه شما بین صفحات دیگر طرفدار و مراجعه کننده بیشتری داشته است. یکی دیگر از فاکتورها alexa rank است که نشان دهنده مراجعه کننده سایت نمی‌باشد، مثلاً یک سایت می‌تواند دارای الکسا رنک 10 باشد، ولی مراجعه کننده آن صفر باشد.

اینکه لینک یک سایت را در سایت‌های مختلف بگذارند الکسا رنک را افزایش می‌دهد ولی گوگل رنک در خصوص مراجعه کنندگان است و این دو مقوله متفاوت است.

وقتی وارد alexa می‌شود، 10 سایت پربیننده‌ترین را نشان می‌دهد.

- 1 - google
- 2 - facebook
- 3 - youtube
- 4 - ...

این می‌تواند بر اساس کشور و شهر فیلتر شود.

در قسمت top site By country ایران را انتخاب نمایید. Blogfa سومین سایت پرترفدار است. در حالی که Blogfa با .net نوشته شده است و این همه طرفدار دارد. البته این سایت در سطح دنیا هم 200 امین سایت است.

Mihanblog با PHP نوشته شده است. در سطح دنیا شاید حدود 11000 باشد.

پس طرفدار یک سایت .net ای با سایت مشابه خود به زبان PHP بیشتر است. این مسائل را بدانید که اگر یک روز این تکنولوژی را انتخاب کردید بتوانید از آن دفاع کنید و بی‌دلیل آن را زیر سوال نبرند.

نکته: چند کار ذیل را انجام دهید:

- کتاب مدیریت زمان برایان تریسی را مطالعه نمایید.
- تکنیکهای Search در search engine ها را یاد بگیرید. مثلاً در Google بگوئید: site فقط در آن سایت می‌گردد. در خود Google جستجو کنید، که چگونه درست Search کنیم و مقاله 4 ، 5 صفحه ای که پیدا شد را مطالعه کنید.
- کتابها و مقالات علمی را در زمینه برنامه‌نویسی و #C و غیره را یاد بگیرید. یک کتاب ترجمه از انتشارات wrox پیدا کنید و آن را مطالعه نمایید. مثلاً سایت Asp.net مربوط به Microsoft را باز کنید و هر روز مطالعه کنید.

یک سری سایت است که باید با آن آشنا شوید:

- 1 - IranianExperts که مربوط به خود استاد است. که ضمن اینکه می توانید از این سایت استفاده کنید، این سایت جنبه آموزشی دارد. در این سایت رفتارهای مختلف را چک کنید و رفتارهای مختلف آن را چک کنید.
- 2 - گروه IranianExpert در یاهو است که کلیه source code ها در اینجا قرار دارد. در این گروه عضو شوید. در این گروه ابتدا join شوید و از دفعات بعد Login نمایید.
آدرس فایلها - بعد از اینکه join شدید، وارد C# شوید و در آن یک سری فایل و فولدر هست. در داخل این فولدر یک فولدر به نام Visual Studio 2008 است (در این سایت Site path map وجود دارد، که مسیر کامل دایرکتوریها را نشان می دهد). ابتدا فایل Learning-asp-net-general-2011-01-07-version-33.zip هست که در دوره های جدید تدریس می شود، آن را download کنید. ولی در ترمهای گذشته تر، پروژه های تدریس شده به صورت کامل با تمام Validation ها Register ها و ... هستند کاملاً قابل استفاده هستند. Learning-asp-net-globalization-2010-10-03-version-6.zip و غیره ...

جلسه دوم

طراحی Layout

امروز طراحی Layout را شروع می‌کنیم. طراحی Layout یا به صورت Frame set ای یا Multi frame صورت می‌گیرد یا به صورت Single frame. Multi frame امروزه مورد استفاده قرار نمی‌گیرد و منسوخ شده است. نمونه multiframe به شکل ذیل می‌باشد. که در آن هر قسمت یک Frame را تشکیل می‌دهد.

Frame1	
Frame3	Frame2

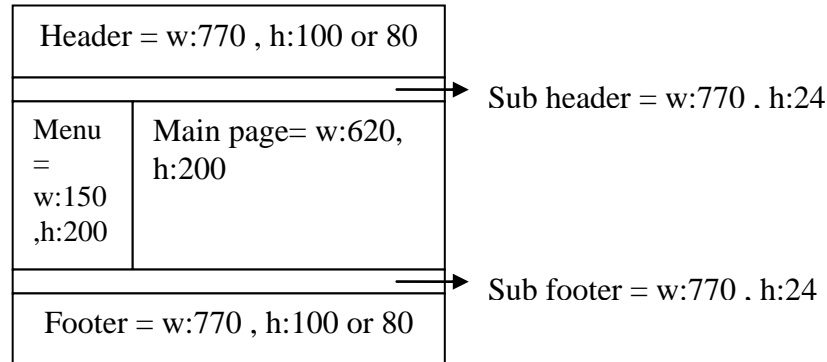
این روش مشکلاتی داشت و تشکیل هم نبود. بنابراین در این کلاس طراحی Single frame را یاد می‌گیریم. خود single frame انواع مختلفی دارد یا به صورت fix است یا relative. خود Relative دلیل ندارد که کاملاً به صورت درصدی (percentage) باشد. بلکه می‌تواند قسمتی از آن Fix و قسمتی دیگر درصدی باشد. مثلاً شاید بخواهیم قسمت منوها ثابت باشد، و فقط قسمت اصلی درصدی باشد. بنابراین وقتی که صفحه کش می‌آید و بزرگ می‌شود، منوها ثابت می‌ماند و قسمت اصلی کش می‌یابد. هر دو روش استاندارد است و بستگی به شرایط و کاربرد دارد. در این کلاس هدف ما تولید Web Site است مثلاً نوشتن یک اتوماسیون تحت وب و معمولاً طراحی Fix خوش دست تر است و در اینجا بیشتر به دنبال طراحی Fix هستیم. ولی یکی را که بلد باشید، با کمی تغییر می‌توانید طراحی خود را به نوع دیگری تبدیل کنید.

پس در تمام Sample های کلاس، از Single frame استفاده می‌کنیم و طراحی را به صورت fix انجام می‌دهیم. البته خود طراحی fix نیز به دو صورت می‌تواند انجام گیرد. به صورت Table و به صورت Div. ولی Div چیز دیگری است و کسی که بتواند با div طراحی کند، یک سر و گردن بالاتر است. ولی در حال حاضر از روش table استفاده می‌کنیم.

فایل style sheet

برای طراحی سایت به کمک جدول، ابتدا ظاهر سایت را در قالب یک جدول کلاسیک در نظر می‌گیرید. جدول کلاسیک جدولی است که تا به آن نگاه کنید، بگویید $m*n$ است. پس اگر جدولی که به شما داده شده بود، کلاسیک نبود، اول خطها را

به هم وصل کنید تا کلاسیک شود. و بعد به سراغ merge کردن بروید. بعد به تمام سطرها شماره بدهید. به ازای تمام سطرها یک row ایجاد کنید. بعد صفت rowspan , Colspan داریم که نشان دهنده این است که چند سطر و ستون merge شده اند. پس در حال حاضر layout ذیل را به صورت fix طراحی می کنید.



Resolution 800*600 در حال حاضر به صورت استاندارد استفاده می شود. البته درصد بالایی از افراد از این layout استفاده نمی کنند و از layout 768*1024 استفاده می نمایند، ولی وضوح 600*800 استاندارد است، و ما نیز می خواهیم مطابق استاندارد عمل کنیم. ابتدا صفحه را در سایز 600*800 تنظیم می کنید و بعد بقیه را طراحی می کنید. اگر Scroll bar افقی ایجاد شود، این خارج از استاندارد است و باید آن را درست کنید. البته نباید عرض صفحه را 800 در نظر بگیرید بلکه باید 770 در نظر بگیرید و باید حدود 25 تا برای Scrol bar عمودی در نظر بگیرید. پهنای مربوط به header مقدار 80 تا 100 خواهد بود. و sub header باید 24 باشد و در خصوص footer و sub footer هم به همان صورت، یعنی سایز آن 80 یا 100 و 24 است. در خصوص منو نیز عرض آن 150 و عرض صفحه اصلی هم باید 620 باشد. طول منو و قسمت اصلی هم 200 در نظر بگیرید و در صفحات وب وقتیکه طول و عرض را تعیین می کنید، حداقل آن را مشخص می کنید و اگر بزرگتر شد، خودش بزرگ می شود.

ما می خواهیم اسکلت سایت را با این دیدگاه ایجاد کنیم و بعد با Copy/paste بقیه صفحات را درست می کنیم. به گونه ای صفحات را درست کنید که اگر به هر دلیل مشتری تغییر خواست، فقط تغییر در یک جا صورت گیرد. پس باید تا جایی که می توانیم قسمت Format , style را در یک فایل دیگر ببریم تا تغییرات در دفعات بعد کمتر باشد. شاید هم بخواهید، قسمت گرافیک را به کس دیگری بسپاریم، پس بهتر است کلیه قسمت های مربوط به formatting در فایل stylesheet قرار گیرد. البته این کار درست نیست و اصل طراحی را باید خود برنامه نویس سایت انجام دهد.

یک پروژه جدید از نوع web site درست کنید با نام abfa و سپس فایل های اضافی را پاک کنید و یک web form به آن پروژه اضافه کنید. در بالای صفحه `<%@ Page Language="C#" %>` خواهید دید. کلیه عبارت قرار گرفته بین `<% %>` Property های صفحه را مشخص می کند. در ابتدا صفات صفحه بسیار کم و ساده است و فقط زبان برنامه نویسی مشخص شده است، ولی به مرور زمان این خط طولانی تر می شود و صفات جدیدی به صفحه اضافه می شود. ولی چه دلیلی دارد که در بالای هر page زبان مشخص شود؟ دلیل آن این است که در وب لازم نیست که تمام صفحات از یک زبان تبعیت کند. می تواند 20 صفحه C# باشد و 10 صفحه VB داشته باشید. این از جذابیت وب است. در وب زبان، per page است و نه Per application. و در ابتدا که C# را انتخاب کردید، به عنوان پیش فرض انتخاب می کنید و بعد می توانید هر یک از صفحات را با C# یا VB بنویسید.

خط بعد تگ Doctype است که نشان دهنده این است که با چه استانداردی صفحه خوانده و فهمیده شود. سه مدل برای این کار وجود دارد. Strict و Frameset و Transitional که بهترین آن Transitional است. Frameset از مد افتاده و strict شورش را در آورده است و زیادی سختگیرانه به مسئله نگاه می کند.

XHTML 1.0 یعنی از HTML 4 و از XML 1 تبعیت می کند. پس مثلاً Case sensitive است. پس وقتی که این استاندارد را می نویسد باید به قواعد آن احترام بگذارد.

یک قسمت به نام Script است که بعداً کدهای C# در آن نوشته می شود.

در حوزه استاندارد یک html و یک head و یک body است که خود template آن را دارد. یک تگ form , div هم قرار داده است. در .net نمی توان بیش از یک تگ form در یک صفحه گذاشت. البته این راه حل دارد، که بعداً گفته می شود.

```

<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

</div>
</form>
</body>
</html>

```

تعیین مشخصات
صفحه، زبان و ...

تعیین استانداردهای
تفسیر صفحه

نگاهی استاندارد صفحات HTML . که
حتماً باید در هر صفحه HTML وجود
داشته باشند.

سعی کنید که همیشه مطابق عرف عمل کنید، یعنی پروژه های خود را به صورتی که قرارداد است، درست نمایید. مثلاً وقتی که فایل style sheet درست می کنید، باید یک فولدر با نام CSS درست کنید و داخل آن یک فایل stylesheet درست کنید. وقتیکه تبلیغات درست می کنید، باید با سایز مشخصی درست کنید.

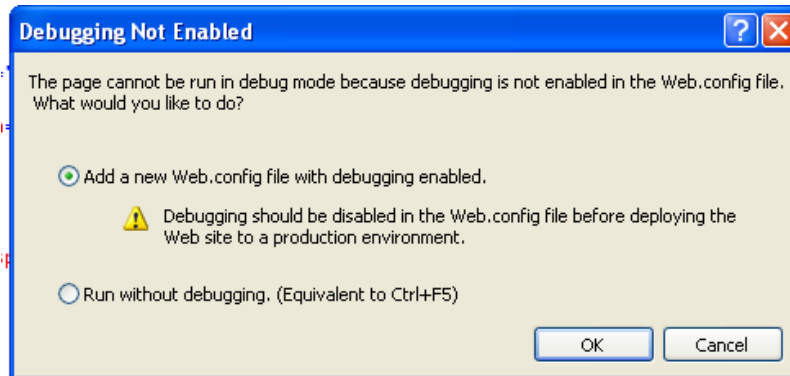
در .net نمی توان بیش از یک تگ form در یک صفحه گذاشت. البته این راه حل دارد، که بعداً بیان خواهد شد. قبلاً عرف بود که فایل Stylesheet را style یا stylesheet نامگذاری می کردند، ولی الان نام آن را Site می گذارند. برای اینکه سایت از این stylesheet استفاده کند و layout بگیرد، باید آن را معرفی کنید. که از VS2005 به بعد می توانید، فایل مورد نظر را به داخل صفحه ای که می خواهد از آن فرمت استفاده کند، drag & drop کنید و تایپ نکنید شاید با خطا تایپ کنید، بنابراین سعی کنید که از Drag & Drop استفاده کنید.

در CSS حتماً باید مقیاس و واحد سایزهای مورد نظر را تعیین کنید ولی در html پیش فرض آن px است و نیاز به بیان نیست.

برای تعیین Style کلاس تعریف می کنیم. یک کلاس table.Layout تعریف می کنیم. وقتیکه می خواهیم یک Style مربوط به یک تگ به صورت unique باشد پس از class استفاده نکنید و از id استفاده کنید و بنابراین در stylesheet هم از نام #Layout به عنوان یک id استفاده می کنیم. Cellspacing معادل CSS ای ندارد، ولی Cellpadding معادل CSS ای آن

همان padding است که می‌توان در فایل stylesheet ست نمود. حتماً برای تمام جداول خصوصیت , cellpadding را set نمایید. چون در ظاهر جداول تاثیرگذار است. سعی کنید که نامگذاری بخشهای مختلف صفحه با دقت انجام گیرد. از نام Header, Footer, subheader ,subfooter, menu, main استفاده می‌کنیم.

برای تعیین سایزها از اعداد فرد استفاده نکنید و از اعداد زوج برای تعیین سایزها استفاده کنید. کد space همان ** ** است. البته این space ای است که بین کلمات اتصال ایجاد می‌کنید و خصوصیت دیگر آن این است که هر چند تا از آن را که در صفحه بگذارید، سایز همه آنها یک است. اگر مثلاً بین ایز و ایران یکی بگذاریم، دیگر این دو کلمه از هم جدا نمی‌شود. یک باگ در IE ورژنهای پایینتر از 8 وجود دارد، که وقتی که در داخل cell یا همان td چیزی ننوشتید، border آن را نشان نمی‌دهد، حتی اگر صریحاً برای آن border تعریف کرده باشید. می‌توانید برای رفع این مشکل در تمام td ها یک علامت قرار دهید، تا تمام Border ها در ابتدای کار و وقتی که جدول خالی است، نشان داده شود. برای اولین بار که پروژه را اجرا می‌کنیم، چون web config وجود ندارد، یک سوال می‌پرسد، آن را OK کنید.



بسیاری از فایل‌های CSS و javascript و ... Cache می‌شود، پس اگر پس از ایجاد تغییر در این فایلها و اجرای برنامه، تغییر مورد نظر را مشاهده نمودید بسته به مرورگرتان، با کمک دکمه Ctrl + F5 یا F5، صفحه مرورگر خود را update کنید. هنگامی که سایتی را طراحی می‌کنید، معطل تایید شدن Layout نشوید. کار خود را شروع کنید. چون اسکلت را که درآوردید، می‌توانید پروژه را Start کنید، تغییر layout به شرط تعیین منطقی و درست اسکلت برنامه، بعداً هم امکان‌پذیر است و برای تغییر ظاهر سایت فقط folder های Images و CSS تغییر می‌کند.

اصولاً چشم انسان از رنگهای تیره به روشن و از چیزهای برجسته به فرورفته تمرکزش بیشتر می‌شود. بنابراین , header , footer , menu و رنگ تیره و بیرون آمده (برجسته) باید داشته باشند و main رنگ روشن تر و تورفته دارد. برای subheader , subfooter , menu , footer , header باید صفت Outset برجسته و thin یعنی نازک را انتخاب نمایید. رنگ استاندارد استفاده کنید، در رنگهای استاندارد دو به دو عددها یکسان است. این رنگها تضمین شده هستند، بصورتیکه در تمام browser ها به همان شکل دیده شوند.

برای main حالت inset یا تورفته و با رنگ روشن تر استفاده کنید. **Format the whole document** برای مرتب کردن source برنامه است. Icon آن در express نیست و بنابراین از **Ctrl+K+D** Shortcut استفاده کنید.

در ذیل، صفحه و فایل CSS طراحی شده برای layout تعریف شده در بالا را مشاهده می‌کنید.

```
<!-- Default Layout Page -->
<%@ page language="C#" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<script runat="server">
</script>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>

    <link href="CSS/Site1.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <form id="frmMain" runat="server">
      <center>
        <table id="layout" cellspacing="0" border="1">
          <tr>
            <td id="header" colspan="2">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="subHeader" colspan="2">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="menu">
              &nbsp;
            </td>
            <td id="main">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="subFooter" colspan="2">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="footer" colspan="2">
              &nbsp;
            </td>
          </tr>
        </table>
      </center>
    </form>
  </body>
</html>

```

```
/* Css File */
```

```
#layout
{
  width: 770px;
}
```

```
td
{
  padding: 0px;
```

```
#menu
{
```

```
width: 150px;
```

ID جدول اصلی
Layout می باشد.

صفت Cellpadding را
برای کلیه سلولها ست


```

        height: 200px;
        border: outset thin;
        background-color: #0066FF;
    }

#main
{
    width: 620px;
    height: 200px;
    border: inset thin;
    background-color: #FFFFFF;
}

#header, #footer
{
    height: 80px;
    border: outset thin;
    background-color: #0066FF;
}

#subHeader, #subFooter
{
    height: 24px;
    border: outset thin;
    background-color: #CCCCCC;
}

```

در زمانهای قدیم که از 2003 , VS 2002 استفاده می‌شد، اگر از یک Layout ، صد عدد کپی تهیه می‌کردیم، اگر یک تغییری ایجاد می‌شد، باید در تمام صفحات تغییر اعمال می‌شد. پس در هر قسمت Layout یک usercontrol قرار می‌دادیم. در این صورت، برای ایجاد تغییر در بعضی از صفحات، می‌شد usercontrol را تغییر داد. مشکل این تفکر چیست؟ اگر مشتری خواهد Layout را شخم بزند، مثلاً به هر دلیلی بخواهد، اول footer بیاید و بعد subfooter و بعد header و بعد از آن menu قرار گیرد و بعد از آن main و بعد از آن یک menu جدید که نبوده اضافه شود و در آخر هم Subheader بیاید. دیگر با استفاده از usercontrol و stylesheet نمی‌توان این کار را انجام داد. ماکروسافت از ورژن 2005 به بعد این مشکل را به صورت فوق‌العاده‌ای حل کرد که البته این روش فقط مختص .net است و زبانهای دیگر نمونه‌ای برای این مسئله ندارند.

در برنامه windows based ما می‌توانیم 1000 تا فرم داشته باشیم که تمام اینها از کلاس form ماکروسافت inherit می‌شوند. به این معنی که هر ویژگی کلاس پدر داشته باشد، به فرزند منتقل می‌شود. اگر بخواهیم در 1000 فرم bgcolor صورتی شود، fade in , fade out شود و ... در تمام فرمها باید انجام شود. بنابراین یک کلاس به نام form base بسازید که این کلاس از کلاس form به ارث ببرد و تمام فرمهای شما از این فرم به نام form base به ارث ببرند. این تفکر هم user interface را پشتیبانی می‌کند و هم functionality هم کارهای هنری و ...

در برنامه تحت وب نیز می‌توان از همین روش استفاده نمود. به صورتیکه وقتی که یک فرم aspx درست می‌کنم در کنار آن یک فایل به نام aspx.cs ایجاد می‌کند، پس من می‌توانم هزار تا page داشته باشیم که تمام آنها از یک Page ، inherit شوند. پس این مسئله در وب نیز مثل windows based ها است و می‌توان یک کلاس به نام Page Base درست کنیم و تمام Page ها را از آن inherit کنیم. مشکل آن، این است که این page فقط functionality دارد و user interface ندارد و در صورتی که تغییری در Layout بخواهند که از طریق Css نتواند انجام پذیرد، باز هم مشکل قبلی را داریم و باید در تمام صفحات تغییرات را اعمال کنیم. بنابراین از 2005 به بعد مفهوم Masterpage معرفی شد که در آن می‌گوییم که Page مورد نظر ما از Master page ، use کند ولی باز هم از Page base , inherit شده است و فقط خوراک Layout را از Master page می‌گیرد. این کمک می‌کند که بدون دست زدن به صفحات بتوان تغییرات Layout را در تمام صفحات اعمال نمود.

بنابراین صفحات ما Layout خود را از Masterpage use می کنند (اینجا نوع ساده‌ای از رابطه association برقرار است و رابطه، یک رابطه inheritance نیست به این نکته توجه داشته باشید). پس صفحات ما باز هم کلاسهای آن از base page به ارث می‌برند و فقط خوراک Layout خود را از Masterpage می‌گیرند.

فایل Master Page

داخل Master page ، قسمتی به نام ContentPlaceHolder قرار دارد که باید در قسمت Main یعنی جایی که صفحه به صفحه تغییر می‌کند، قرار گیرد.

پس یک Masterpage می‌سازیم. توجه داشته باشید که اولین Master page که می‌سازیم نام آن Master page است. اگر اسم Master page را Master page بگذاریم گرفتار می‌شویم. اولین master page را default می‌گذاریم. هیچکدام از Check box ها را تیک نمی‌زنیم.

صفحات Masterpage مثل صفحات عادی است به جز اینکه یک Contentplaceholder در آن قرار دارد. در 2005 در Masterpage ، فقط در تگ فرم می‌توانستید Contentplaceholder بنویسید و این در بعضی جاها دردسرساز بود. این امکان که در بیرون فرم و حتی در header هم Cph بگذارید از 2008 امکان‌پذیر شد. آیا در یک فرم می‌توان بیش از یک CPH داشته باشیم؟ بله ولی استاندارد نیست و در layout های غیر نرمال مجبور می‌شوید این کار را کنید و همیشه Layout خود را استاندارد طراحی کنید، تا مجبور نشوید از دو CPH در یک فرم استفاده کنید. سراغ اینجور layout های غیر نرمال نروید. دو تا CPH بیشتر نداشته باشید، یکی در head و دیگری در main . اسم آنها را cphMain , cphHead بگذارید.

تذکر: در Masterpage قسمت مربوط به لینک فایل stylesheet را drag drop کردم و قبل از Placeholder گذاشتم دلیل دارد. در masterpage تگ معرفی فایل CSS قبل از PCH بگذارید.

تگ دیگر Contentplaceholder را درون main می‌گذارم. به این نکته توجه داشته باشید که CPH را در کجا قرار می‌دهید. در CPH هیچ چیزی نگذارید. CPH صفحاتی که از masterpage استفاده نموده‌اند، می‌آید در این قسمت overwrite می‌شود، پس در CPH مربوط به Masterpage چیزی ننویسید، error نمی‌دهد ولی کاری نمی‌کند، در واقع مثل این است که هیچ کدی ننوشته‌اید. کد مورد نظر برای Master page جهت طراحی Layout بالا عبارت است از:

```
<!-- Master page 1 #-->
<%@ master language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>
    <link href="CSS/Site3.css" rel="stylesheet" type="text/css" />
    <asp:contentplaceholder id="cphHead" runat="server">
  <!--
    نوابغ دقت داشته باشند که در این قسمت چیزی ننویسند
  -->
```

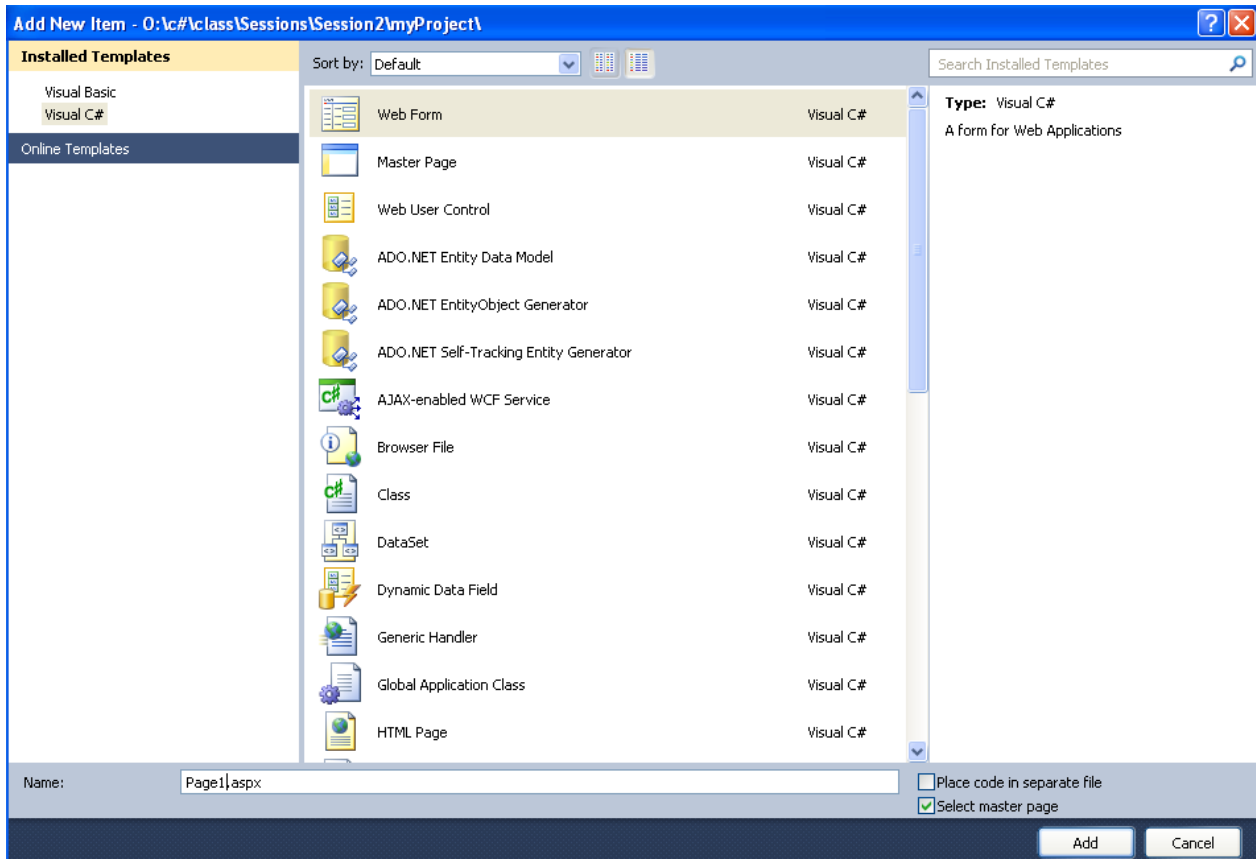
```

        </asp:contentplaceholder>
    </head>
    <body>
        <form id="frmMain" runat="server">
            <center>
                <table id="layout" cellspacing="0">
                    <tr>
                        <td id="header" colspan="2">
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td id="subHeader" colspan="2">
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td id="menu">
                            &nbsp;
                        </td>
                        <td id="main">
                            <asp:contentplaceholder id="cphMain"
runat="server">
                                <!--
                                -->
                                </asp:contentplaceholder>
                            </td>
                    </tr>
                    <tr>
                        <td id="subFooter" colspan="2">
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td id="footer" colspan="2">
                            &nbsp;
                        </td>
                    </tr>
                </table>
            </center>
        </form>
    </body>
</html>

```

نوابغ دقت داشته باشند که در این قسمت چیزی ننویسند

الان دو صفحه درست می‌کنم که در آن تیک Select master page انتخاب شده است. در یک پروژه می‌توان چند master page داشته باشیم. مثلاً برای کسانی که login کردند صفحات به یک شکل نمایش داده شود، و کسی که login نکرده است به یک شکل دیگر صفحات را ببیند.



یا مثلاً برای بعضی از ناظران خارجی بخواهیم یک سری گزارش با آیتمهای زیاد و در نمای بسیار بزرگ نمایش دهیم که باید یک masterpage جداگانه برای آنها طراحی شود. در داخل صفحه Page1 دیگر head, body, style نیست و فقط در آن CPH های masterpage و یک قسمت برای script نویسی وجود دارد. نام این دو cph را نیز عوض کنید. و نام آن را cphheader , cphmain بگذارید.

```
<!-- page 1 -->
<%@ page title="Home" language="C#" masterpagefile="~/Site.master" %>

<script runat="server">
</script>

<asp:content id="cntHead" contentplaceholderid="cphHead" runat="Server">
</asp:content>

<asp:content id="cntMain" contentplaceholderid="cphMain" runat="Server">
</asp:content>
```

هر صفحه‌ای که درست می‌کنید در ابتدا به آن title دهید. فرض کنید اگر بخواهیم الان صفحه 2 را اجرا کنیم و set as default page من صفحه 1 باشد، و برنامه را نیز اجرا نموده‌ام. لازم نیست اجرای برنامه را قطع کنید و دوباره set as default page را مشخص کنید، فقط کافی است که در همان address bar نام آن را تایپ کنید. اگر Master page یک صفحه را بخواهیم عوض کنیم. باید چه کار کنیم. در بالای صفحه یک masterpage file ایجاد شده است که کافی است که نام آن را تغییر دهید. (masterpagefile="~/Site.master") اگر مشتری بخواهد Layout را تغییر دهد، من هیچ کاری با صفحات نداریم و فقط با Master page و stylesheet کار داریم. مثلاً مشتری فرمت ذیل را می‌خواهد.

- footer - 1
- Subfooter - 2
- Header - 3
- Right menu - 4
- Main - 5
- Left menu - 6
- Sub header - 7

به راحتی این کار امکان پذیر است و فقط کافی است که در Master page جای قسمت های مختلف را با هم عوض نمایید و چند تا تغییر جزئی ایجاد کنید به شکل ذیل:

```
<!-- Master page 2 # -->

<%@ master language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">
</script>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>

    <link href="CSS/Site4.css" rel="stylesheet" type="text/css" />

    <asp:contentplaceholder id="cphHead" runat="server">
    <!--
      نوابغ دقت داشته باشند که در این قسمت چیزی ننویسند
    -->
    </asp:contentplaceholder>
  </head>
  <body>
    <form id="frmMain" runat="server">
      <center>
        <table id="layout" cellspacing="0">
          <tr>
            <td id="footer" colspan="3">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="subFooter" colspan="3">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="header" colspan="3">
              &nbsp;
            </td>
          </tr>
          <tr>
            <td id="menuLeft">
              &nbsp;
            </td>
            <td id="main">
              <asp:contentplaceholder id="cphMain"
runat="server">
                <!--
```

نواغ دقت داشته باشند که در این قسمت چیزی ننویسند

```

-->
</asp:contentplaceholder>
</td>
<td id="menuRight">
    &nbsp;
</td>
</tr>
<tr>
<td id="subHeader" colspan="3">
    &nbsp;
</td>
</tr>
</table>
</center>
</form>
</body>
</html>

```

چند style مفید برای قسمت Main صفحه

Style چپ به راست و راست به چپ

Style ای که در آن یک ناحیه را فارسی یا انگلیسی کنیم. البته بهتر است بگوییم Ltr , Rtl کنیم و نگوییم فارس و انگلیسی. چون می‌خواهیم به درد هر زبانی که از راست به چپ و یا چپ به راست است، بخورد به آن .ltr , .rtl می‌گوییم. طبق آخرین استاندارد ماکروسافت وقتی که نام namespace و folder بنویسیم تمام حروف آن را با حروف بزرگ می‌نویسیم. اگر اسم کلاس یا متغیر abbreviation است فقط حرف اول را بزرگ می‌نویسیم. اگر استاندارد Microsoft بود، باید L مربوط به ltr را بزرگ می‌نوشت.

CSS برای چپ به راست به صورت ذیل تنظیم می‌کنیم:

Font family : verdana باشد و font size 8 یا 10 باشد. text align را Left استفاده کنید. برای زبانهایی که حالت چسبیده ندارند به هیچ عنوان Justified استفاده نکنید و direction را Ltr بگیرید.

CSS برای راست به چپ به صورت ذیل تنظیم می‌کنیم:

font family : Tahoma و font-size :10 و text-align: justify; و direction:rtl;

```

.ltr
{
    direction: ltr;
    text-align: left;
    font-size: 10px;
    font-family: Verdana;
}

.rtl
{
    direction: rtl;
    text-align: justify;
    font-size: 10px;
    font-family: Tahoma;
}

```

دو فونت Tahoma , arial به صورت multi language است و وقتی آن را باز می‌کنیم تمام زبانهای دنیا حتی چینی و ژاپنی را هم support می‌کند. Tahoma قشنگتر است.

اگر یک ناحیه می‌خواهید هم right to left باشد و هم border باشد، اینها را در یک style sheet نگذارید، چند style sheet تعریف کنید و بعد به صورت ترکیبی استفاده کنید. اصلاً در هم آن قشنگ نیست. پس سعی کنید که کلاس‌بندی مرتبی داشته باشید و خصوصیات شبیه به هم را در کنار هم قرار دهید.

```
.withBorder
{
    margin: 10px;
    padding: 10px;
    border: outset thin;
}
```

پس یک کلاس به نام withboarder ایجاد نمایید. و در آن بگویید که padding , margin آن 4px باشد و boarder آن outset باشد و thin و ...

Style مربوط به Hyperlink

در خصوص hyperlink: در stylesheet چیزی به نام pseudo داریم که برای مرتبط کردن یک سری مفاهیم به یک سری event است. تلفظ آن zoodo است.

اینکه حرف اول را بکشد و بقیه حروف جملات بعدی را با آن پر کند. همه اینها pseudo است ولی مهمترین آنها

1. link
2. visited
3. hover
4. active

است که ترتیب آنها موقع نوشتن مهم است.

در Tahoma وقتی که از Underline استفاده می‌کنید، یک خط روی نقطه‌ها می‌اندازد، بنابراین در هنگام گذاشتن link، خوانایی متن کم می‌شود. برای از بین بردن آن باید چه کار کرد؟

Visited ها در سایت ما بهتر است که همان آبی باشد، در خصوص سایتهایی که خیلی لینک زیاد است، باید لینک ویزیت شده رنگ دیگری داشته باشد. آخرین لینکی هم که زدیم بهتر است قرمز باشد، تا بدانیم آخرین لینکی که کلیک کردیم کدام است. (active).

در بعضی از سایتها لینکها خط ندارد و وقتی رفتین روش خط می‌گذارد. صفتی که زیر متنها خط underline می‌کشد، اسم آن text-decoration است.

در تمام قسمتها text-decoration را none می‌زنم و بعد Border تعریف می‌کنم و می‌توان مکان border را تعیین کرد و فقط boarder-bottom را تعریف می‌کنیم و به رنگ آبی #0000FF.

With boarder , left to right , right to left , pseudo از اهمیت زیادی در style ها برخوردار است.

```
a:link
{
    color:#0000FF;
    text-decoration: none;

    padding-bottom: 1px;
    border-bottom: solid 1px Blue;
```

```

}

a:visited
{
    color:#0000FF;
    text-decoration: none;
    padding-bottom: 1px;
    border-bottom: solid 1px Blue;
}

a:hover
{
    color:#FF0000;
    text-decoration: none;

    padding-bottom: 1px;
    border-bottom: solid 0px;
}

a:active
{
    color:#FF0000;
    text-decoration: none;
    padding-bottom: 1px;
    border-bottom: solid 1px Red;
}

```

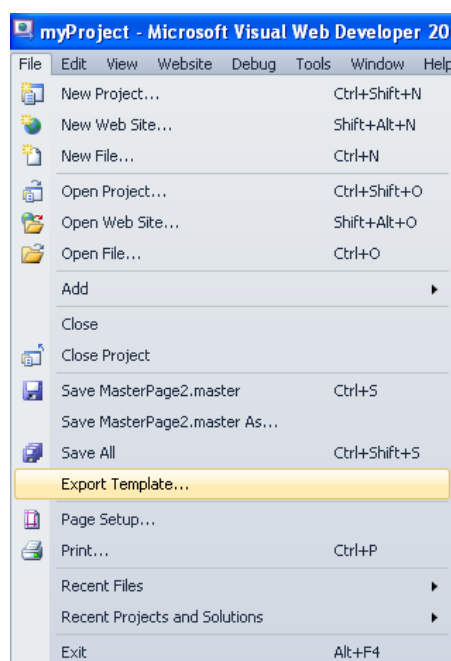
هر سایتی باید یک فولدر CSS داشته باشیم و بعد site.css داشته باشیم و باید site.master درست کنیم و یک فایل default.aspx داشته باشیم.

تعریف Template

Template می تواند per page , per site باشد.

برای تهیه Template ابتدا web site → New بسازید و به صورت file system ای ایجاد کنید و فولدر CSS و بعد site.css و site.master و web.config را ایجاد یا کپی می کنیم.

File → export template →



می پرسد که از کل پروژه template درست کنم و یا از item ای از پروژه.

Export Template Wizard

Choose Template Type

This wizard will allow you to export a project or project item from the current solution to a template which future projects can then be based upon.
Which type of template would you like to create?

Project template
A project template will allow a user to create a new project based on your exported project. A user will be able to utilize your template from the New Project dialog box for client projects and from the New Website dialog box for websites.

Item template
An item template will allow a user to add your item to one of their existing project. Your template will be available to the user from the Add New Item dialog box.

From which project would you like to create a template?
myProject

What language category should this template appear under in the New Project dialog box?
Visual C#

< Previous Next > Finish Cancel

Template از اهمیت خاصی در پروژه های web based برخوردار است، چرا که بسیاری از صفحات و تنظیمات وجود دارد که باید به صورت ثابت در تمام پروژه ها صورت گیرد.
Project template را انتخاب می کنیم و نام آن را تعیین می کنیم. توضیح دارید، بنویسید و سپس finish را بزنید.

Export Template Wizard

Select Template Options

Template name:
myProject

Template description:
<No description available>

Icon Image:
Browse...

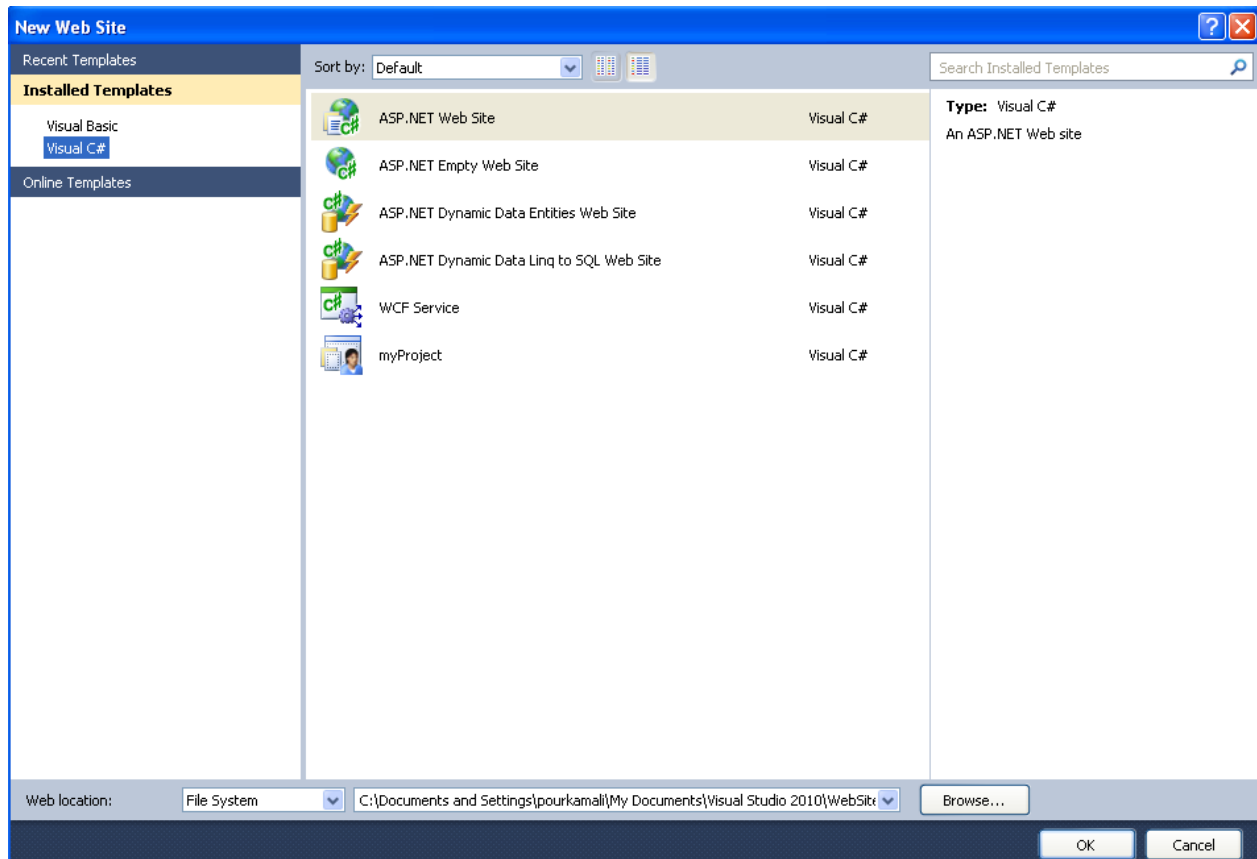
Preview Image:
Browse...

Output location:
C:\Documents and Settings\pourkamali\My Documents\Visual Studio 2010\My Exported Templates\myProject.z

Automatically import the template into Visual Studio
 Display an explorer window on the output files folder

< Previous Next > Finish Cancel

در فولدر My exported templates در مسیر My Documents\Visual Studio 2010\My Exported Templates\myProject.zip یک فایل زیپ درست می‌کند. و برای استفاده هم این فایل zip در این فولدر هر کامپیوتری که باشد، این template در لیست مشاهده می‌شود و پس دیگر این template را در لیست template ها خواهند دید. برای بارگذاری یک پروژه از نوع یک template باید از Installed Templates نام template خود را انتخاب کنید.



طرح سوم

کدنویسی

در داخل ContentPlaceHolder در قسمت main کد ذیل را بنویسید:

```
<asp:Content ID="cntMain" ContentPlaceHolderID="cphMain" runat="Server">
  <asp:Label ID="lblFirstName" runat="server" Text="First Name" />
  <asp:TextBox ID="txtFirstName" runat="server" MaxLength="20" AccessKey="1" />
  <br />
  <asp:Label ID="lblLastName" runat="server" Text="Last Name" />
  <asp:TextBox ID="txtLastName" runat="server" MaxLength="30" AccessKey="2" />
  <br />
  <br />
  <asp:Label ID="lblFullName" runat="server" Text="Full Name" />
  <asp:TextBox ID="txtFullName" runat="server" MaxLength="50" ReadOnly="true" />
  <br />
  <asp:Label ID="Label1" runat="server" Text="Full Name" />
  <asp:TextBox ID="txtFullName2" runat="server" MaxLength="50" Enabled="false" />
  <br />
  <asp:Button ID="btnSubmit" runat="server" Text="(S)ubmit" AccessKey="S"
    OnClick="btnSubmit_Click" />
  <input type="reset" value="(R)eset" accesskey="R" />
</asp:Content>
```

سه نوع دکمه یا در واقع سه type برای دکمه (button) وجود دارد. Input Submit و Input reset و Input button. دکمه از نوع reset را بعداً بررسی می‌کنیم ولی دکمه از نوع submit اطلاعات را به server ارسال می‌کند.

کنترل‌های Server Side

هر چیزی که در صفحه aspx به صورت HTML بنویسید عیناً به سمت client منتقل می‌شود. یک سری موجودات Server side داریم که آنها ابتدا تبدیل به HTML می‌شوند و بعد به سمت client می‌روند که به این عملیات تبدیل شدن، render شدن می‌گویند.

به ارسال داده از صفحه‌ای به سمت سرور Post back می‌گویند. در صفحات web based بعد از کلیک بر روی دکمه submit بلافاصله textbox ها پاک می‌شود که این از بزرگترین معضلات وب است. ولی .net آن را حل کرده است. در تکنولوژی و محیط غیر .net اگر تعدادی textbox داشتید و دکمه submit می‌زدید کل textbox ها پاک می‌شد و اگر می‌خواستید که textbox ها مقدار خود را حفظ نمایند، شما باید می‌رفتید کلی کد می‌نوشتید، تا این اطلاعات را در جایی نگهداری و مجدداً textbox ها را پر کنید. بنابراین .net موجودات server side را ایجاد نمود. این موجودات server side مخصوص .net می‌باشد.

به چه موجوداتی server side می‌گویند؟ به موجوداتی که با asp: آغاز می‌شوند و قطعاً عیناً به سمت Client منتقل نخواهد شد، چرا که مرورگرها فقط HTML را می‌شناسند و بنابراین این موجودات render می‌شوند. برای موجودات server side باید حتماً run at server را بنویسید وگرنه error می‌دهد. دادن ID برای موجودات server side به صورت optional است. دادن ID کی اجباری است؟ اگر در C# می‌خواهید برای این موجود کد بنویسید یا از این کنترل در کنترل دیگر استفاده کنید، باید ID بگذارید. البته اگر حرفه‌ای هستید باید برای تمام کنترل‌های server side حتماً ID بگذارید. Abbreviation موجودات server side با client side یکسان هستند مثلاً label را با lbl شروع کنید. وقتی که نمی‌خواهید بین تگی چیزی بنویسید بهتر است که تگ باز و بسته را یکجا قرار دهید. (یعنی مثلاً به جای نوشتن تگ textbox به صورت ذیل:

```
<asp:TextBox ID="Password" runat="server"></asp:TextBox>
```

این تگ را به این صورت می‌نویسیم.

```
<asp:TextBox ID="Password" runat="server"/>
```

پس از render شدن، تگ asp:label به span تبدیل خواهد شد. توسعه اکید می‌شود که هیچگاه برای label ، Hard code ، بنویسید و از تگ استفاده کنید. به جای استفاده از input با نوع textbox از textbox استفاده کنید. ما به ازای بعضی از html ها معادل server side ای داریم ولی بعضی از html ها معادل Server side ای نداریم. مثل دکمه با نوع reseat معادل server side ای ندارد. اگر html بنویسید عیناً به client فرستاده می‌شود ولی موجودات server side باید اول در سمت سرور render شوند، پس بار اضافی به server تحمیل می‌کنند، پس با حکمت از موجودات server side استفاده کنید. مثلاً تگ Asp.table را وقتی استفاده کنید که بخواهید جدول داینامیک داشته باشیم وگرنه حتماً از table عادی استفاده کنید. برای Layout به هیچ عنوان از asp.table استفاده نکنید.

نکته 1: به طور کل به ازای هر html معادل server side ندارید.

نکته 2: بهتر است که در جاهایی که می‌توانید از تگ HTML عادی استفاده کنید، از معادل server side آن استفاده ننمایید.

نکته 3: هیچوقت label را hardcoded ننویسید و معادل asp یا server side آن استفاده کنید.

نکته 4: قیافه برنامه وقتی که موجودات client side یا Server side هستند فرقی نمی‌کند. کد تولید شده هم ظاهراً فرقی ندارد.

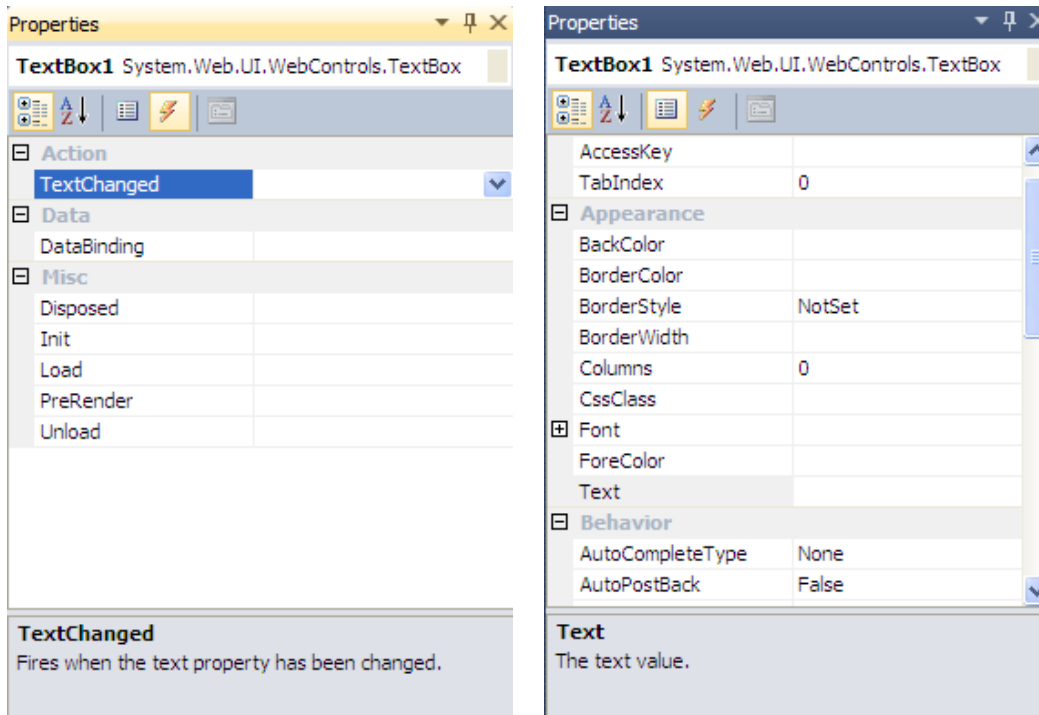
چرا به جای اینکه از کنترل input با تایپ type: text استفاده کنیم از کنترل textbox استفاده کردیم؟ ظاهراً هر دو به یک چیز تبدیل (render) می‌شود. Value های کنترل من یا همان state آن حفظ می‌شود. مهمترین ویژگی آن View State آن است. MVC تکنولوژی است که تنها چیزی که در آن نیست، View State است. و View State تفاوت بسیار اساسی و مهمی است. پس صفحات asp کلاسیک می‌تواند کنار .net باشد ولی MVC به دلیل View State نبودن آن، کاملاً دنیای متفاوتی است و form based نیست و نمی‌تواند در کنار .net باشد. بنابراین در یک پروژه نمی‌تواند دو تا فرم MVC و دو تا فرم معمولی وجود داشته باشد.

اگر به کنترل‌های server side ، ID ندهید خودش ID می‌گیرد از 1 شروع می‌کند و 2 و 3 و

پس همیشه سعی می‌کنیم که به کلیه متغیرهای Server side ، ID دهیم.

وقتی وارد کدنویسی می‌شویم ، وارد قسمت design می‌شویم و بعد Properties می‌گیریم و بعد در قسمت event ها که آیکن آن به صورت رعد و برق است ، event مورد نظر را انتخاب و روی آن دابل کلیک می‌کنید تا raise شود. چیزی به نام Default Event وجود دارد که به معنای مهمترین event یک دکمه است. مثلاً در مورد دکمه، click مهمترین event است. در خصوص Default event با دابل کلیک بر روی دکمه نیز ، event default ، raise می‌شود.

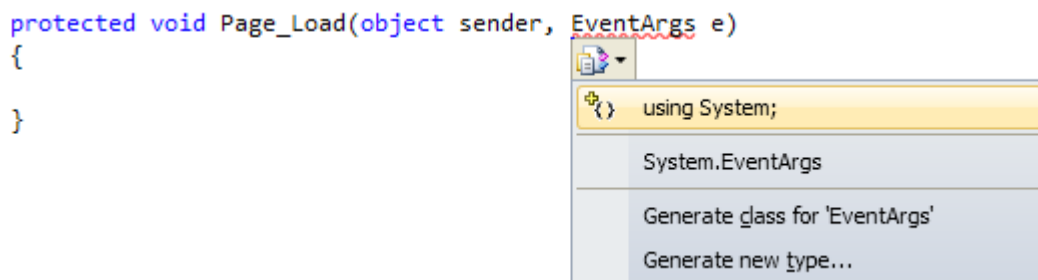
در windows based کلی event داشتیم ولی در اینجا کم شده است. در موجودات server side فقط event هایی که بعد از Post back صدا زده می شوند، دیده می شود.



در صفحه `aspx.cs` ، کلیه `using` ها را پاک کنید. بعد `error` ای به صورت ذیل می دهد.

The type or namespace name 'EventArgs' could not be found (are you missing a using directive or an assembly reference?)

بر روی پارامترهایی که `error` داده است، کلیک راست نمایید (اگر نخواهید `right click` کنید، `Ctrl+9+` ، shortcut آن است) و بعد باز کنید، دو راه حل پیشنهاد می دهد که یا باید دوباره `using` را بگذاریم و یا اینکه مسیر اصلی آن را که خودش پیشنهاد می دهد بگذاریم. ما در اینجا راه حل دوم را انتخاب می کنیم، یعنی مسیر کامل را می نویسیم.



در مد `design` بر روی دکمه مورد نظر `double click` می کنیم، یک صفت `onclick` به دکمه اضافه می شود و مقدار آن نام تابعی است که در قسمت کدنویسی (`script`) قرار گرفته است.

بعد از زدن `submit` صفحه `Postback` می شود.

در `fullname` من نباید بتوانم تایپ کنم و باید در آن حاصل `first name` , `last name` قرار گیرد. برای اینکار یا صفت `read only` آن را `true` می کنیم یا `enabled` آن را `false` می کنیم.

نکته: ID در هر صفحه باید unique باشد.

به عنوان تست هم read only و هم enabled را تست می‌کنیم.

Enabled را false کنید، خاکستری کم‌رنگ می‌شود و با tab و shift tab هم نمی‌توانید وارد آن شوید. در بعضی از browser ها امکان select , copy هم ندارید. در خصوص read only اینگونه نیست.

توصیه و تاکتیک: اگر شما یک صفحه دارید که همه اطلاعات داخل آن را به صورت غیر قابل edit به دیگران نشان دهید مثل profile محصول به دلیل خاکستری رنگ شدن آن خوانایی را کم‌رنگ و بد می‌شود و در بعضی از مرورگرها مثل chrome خیلی بدرنگ و ناخوانا می‌شود، بنابراین باید از صفت read only استفاده نمایید و آن را true کنید. در صفحاتی که بعضی از فیلدها را می‌شود تغییر داد و بعضی را نمی‌شود، مثل صفحه update profile که می‌خواهید با یک نگاه به کاربر نشان دهید که بعضی از فیلدها را می‌شود تغییر داد و بعضی را نمی‌شود، پس در اینجا از enabled استفاده کنید و آن را false نمایید.

شکستن کدها: گاهی کدهایی که می‌نویسید طولانی است ایراد دارد. اول اینکه کل source code را در یک نگاه نمی‌بینید و باید scroll دهید و دوم اینکه موقع print گرفتن شکل نمی‌باشد.

برای همه text box ها، هم در پروژه های وب و هم در ویندوز حتماً maxlength را set کنید، هم از نظر user friendly بودن بهتر است، هم جلوی error را می‌گیرد و هم جلوی هک شدن سایت را می‌گیرد.

اعداد 20 و 30 هم در دنیا استاندارد است. فیلد نام به صورت استاندارد به طول 20 و نام خانوادگی 30 می‌باشد و اگر نخواهید دو تکیه کنید، طول 50 را به عنوان طول استاندارد انتخاب کنید.

در پروژه‌های وب می‌توانید از access key استفاده کنید و در این صورت دکمه alt و آن حرف را که بگیرد، بر روی آن کنترل focus می‌شود. برای فارسی راهنما بنویسید ولی برای انگلیسی، در قسمت text در دو طرف حرف مورد نظر پرانتز بگذارید، و کاربر متوجه می‌شود که این حرف به صورت access key قابل استفاده است. Access key در فرمهای پرکاربرد اهمیت دارد.

تکنیک چیزهایی است که باید کسی که چیزی را بلد است، آن را بداند ولی در مورد تاکتیک می‌تواند منحصر به شرکت شما باشد و باید تاکتیکها در ابتدای کار بیان شود. تاکتیکها همان رویه‌هایی است که استاندارد کدنویسی ما است که باید کسی که در ابتدا کار وارد می‌شود به آن توضیح داده می‌شود.

زمانی که IIS ای کار می‌کنید، IIS نمی‌تواند بیش از یک پروژه را در حین واحد در مد debug اجرا کند. البته اگر **Ctrl+F5** را بزنید یعنی **debug نکن** و مشکلی نیست ولی در غیر این صورت error می‌دهد.

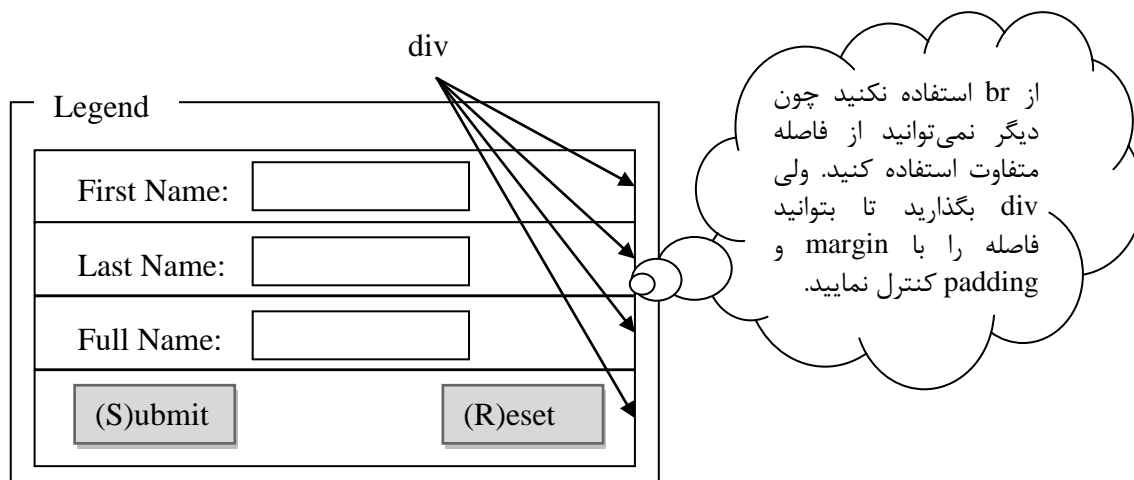
دکمه reset

دکمه reset چه کاری انجام می‌دهد. به آخرین حالتی که از طرف server آمده است، بر می‌گرداند. پس ترجمه reset هم یعنی برگرداندن به حالت اولیه که از طرف server آمده است.

Layout ورود اطلاعات

برای فرمهای ورود اطلاعات دو نوع استاندارد داریم.

یک جدول درست می‌کنیم که در یک ستون label ها را می‌گذاریم و در یک ستون هم textbox ها و در ستون آخر هم با 2 Collspan ، یک دکمه قرار داده می‌شود. الان مدل FieldSet مورد استفاده قرار می‌گیرد. که به تازگی بسیار مد شده است. یک فولدر CSS و بعد داخل آن site.css . درون آن ltr را بنویسید.



معمولاً چیزی که در Legend می‌نویسند با title یکی است. برای field set خود در css مشخصات آن را تعریف می‌کنیم. تا یکنواخت شوند.

```
.ltr
{
    font-size: 10px;
    font-family: Verdana;
    text-align: left;
    direction: ltr;
}

fieldset
{
    margin: 10px;
    padding: 0px;
}

div.field
{
    margin: 4px;
    padding: 4px;
}

div.buttons
{
    margin: 4px;
    padding: 4px;
}
```

اگر یک دکمه بیشتر ندارید، اسم آن را btnSubmit بگذارید و text آن را هر چیزی که دوست دارید، بگذارید. و اگر هم چند دکمه دارید حتماً نام یکی از آنها btnSubmit باشد. حکمت آن را بعداً می‌گوییم.

```
<asp:Content ID="cntMain" ContentPlaceHolderID="cphMain" runat="Server">
    <fieldset>
        <legend>Home </legend>

        <div class="field" runat="server">
            <asp:Label ID="lblFistName" runat="server" Text="First Name" />
            <br />
```

```

        <asp:TextBox ID="txtFirstName" runat="server" MaxLength="20" AccessKey="1"
/>
    </div>

    <div class="field" runat="server">
        <asp:Label ID="lblLastName" runat="server" Text="Last Name" />
        <br />
        <asp:TextBox ID="txtLastName" runat="server" MaxLength="30" AccessKey="2"
/>
    </div>

    <div id="divFullName" runat="server" class="field">
        <asp:Label ID="lblFullName" runat="server" Text="Full Name" />
        <br />
        <asp:TextBox ID="txtFullName" runat="server" MaxLength="50"
Enabled="false" />
    </div>

    <div class="buttons">
        <asp:Button ID="btnSubmit" runat="server" Text="(S)ubmit" AccessKey="S"
OnClick="btnSubmit_Click" />
        <input type="reset" value="(R)eset" accesskey="R" />
    </div>
</fieldset>

```

در کد به اندازه کافی از enter استفاده کنید، مثلاً field set را به خط بالای آن نچسبانید.

صفت AutoEventWireup

همه صفحات یک property دارند با نام autoeventwireup که این تا Visual Studio 2002 ، به صورت پیش فرض false بود و الان true است.

```

<%@ Page Language="VB" AutoEventWireup="true" CodeFile="Test.aspx.vb" Inherits="Test"
%>

```

این صفت فقط در خصوص رویدادهایی مانند page load , page error این event ها همیشه اتفاق می افتد. بقیه event ها باید صدا زده شوند. ولی در جایی نمی بینید که Page Load صدا زده شده باشد. ولی این اتفاق می افتد و همیشه موقع فراخوانی اولیه صفحه و در هنگام Post Back صدا زده می شود و این بخاطر autoeventwireup است. در VS2005 به بعد، این صفت پیش فرض True است.

برنامه ای به نام Code analyzer داریم که کد را آنالیز می کند و یک سری توصیه می کند. مثلاً توصیه می کند تا به جای "" که همان null string است، از کد string.empty استفاده کنیم. دستور string.isnull و ... هم همینطور است. Post back همیشه اتفاق می افتد هم بار اول هم دفعات بعد Post back اتفاق می افتد.

در وب اگر که visible را false کنید اصلاً در هنگام render کد آن generate نمی شود، پس عملاً وجود ندارد. اما مشکلی که وجود دارد این است که div آن هنوز وجود دارد. پس باید کاری کنید که div به صورت invisible شود، تا در هنگام render شدن، تولید نشود.

Page load را اگر بنویسید همیشه و در هر شرایطی حتی بعد از Post back هم Page load اتفاق می افتد و در هنگام کلیک دکمه، اول Page load اتفاق می افتد و بعد عملیات کلیک.

موجوات برزخی (تگهای HTML با ID)

یک سری موجوداتی وجود دارد که به آن موجودات برزخی می گویند، که آنها تگ html ای است که به آن ID داده شده و runat آن server شده است.

```
<div id="divFullName" runat="server" class="field">
  <asp:Label ID="lblFullName" runat="server" Text="Full Name" />
  <br />
  <asp:TextBox ID="txtFullName" runat="server" MaxLength="50" Enabled="false" />
</div>

<script runat="server">
  protected void page_Load(object sender, EventArgs e)
  {
    divFullName.Visible = false;
  }

  protected void btnSubmit_Click(object sender, EventArgs e)
  {
    txtFirstName.Text =
      txtFirstName.Text.Trim();
    txtLastName.Text =
      txtLastName.Text.Trim();
    txtFullName.Text =
      (txtFirstName.Text + " " + txtLastName.Text).Trim();
    if (txtFullName.Text != string.Empty)
    {
      divFullName.Visible = true;
    }
  }
</script>
```

Ctrl+K+U می گذارد و Comment، Ctrl+K+C می گذارد. Comment را بر می دارد.

حتماً به عنوان یک آدم حرفه ای Place code in separate file را چک بزنید. خوبی آن این است که اگر پسورد ftp شما لو رفت، دیگر در صفحه aspx شما، کد شما وجود ندارد که لو برود.

تمام کارهایی که در این فرم انجام دادید را می توانستید، با javascript یا jquery انجام دهیم. Javascript بسیار به درد نخور است، بنابراین jquery یک لایه زیبا بر روی javascript ایجاد نمود و قابلیت های آن را بالا برد. با Ajax و jquery آن می توانید صفحات سبک و زیبایی را طراحی کنید.

ListBox و DropDownList

این دو کنترل شامل یک collection از Item ها می باشد. که دارای صفاتی همچون text, value, selected, enabled می باشد. صفت Text همان آیتمی است که دیده می شود. صفت value عددی است که در پشت آن قرار گرفته است. Selected یعنی از ابتدا انتخاب شده باشد یا خیر؟. enabled نشان دهنده این است که آیا آن آیتم قابل انتخاب می باشد، یا خیر؟

به هیچ عنوان listbox خود را hardcoded نگذارید. حتی برای جنسیت هم table بگذارید. اصلاً به صورت hard code پر نکنید. مثلاً برای جنسیت یک جدول طراحی کنید و درون آن چندین نوع مختلف از بیان جنسیت را قرار دهید. حتماً (زن و

مرد (خانم و آقا) (خانم و آقای) را در جدول خود بگذارید و مانور زیبایی روی آن داشته باشید، یعنی اگر قرار است فرد جنسیت خود را انتخاب کند، لیست از زن و مرد- پر شود. اگر قرار است که یک چیزی مانند فرم تعهد نامه است که فرد از جانب خود فرم را پر می کند، لیست با خانم و آقای- پر شود و ...

بعد از render شدن، تمام item ها تبدیل به option می شوند و value همان value و text در بین تگ نوشته خواهد شد. چیزهایی که به s جمع ختم می شوند، مربوط به Collection است.

در خصوص کدنویسی آن، ابتدا لیست یا combo box را clear کنید. بعد یک item با add به انتهای لیست اضافه کنید.. اگر موقع add کردن فقط یک string بدهید، value , text یکسان می شود که مناسب نیست، بنابراین به لیست خود List item اضافه کنید که هم دارای text باشد و هم دارای value .

روش دوم: ابتدا clear می کنید و بعد مرتباً یک item list می سازید و مرتباً به listbox خود add می کنید.

String.format یعنی چه؟

String.Format("Item{0}",intIndex);

یعنی هر جایی که {0} را مشاهده نمود، متغیر intIndex را بجای آن قرار دهد.

Int A=10;

Int B=15;

Int C=20;

در زبان VB یا هر زبان دیگری می توان به صورت ذیل رشته S را مقداردهی نمود:

String S= "A = "+A+" , B = "+B+" , A= "+ A ;

در C# می توان به صورت ذیل این کار را انجام داد:

String S = string. Format("A={0},B={1},A={0}" , A, B);

در کد بالا، هر جایی که {0} را مشاهده کند، به جای آن متغیر A را قرار می دهد و هر کجا که {1} را مشاهده نماید، به جای آن متغیر B را قرار می دهد.

امروز دو درس مهم فلسفی داشتیم، یک مفهوم View State بود که با کنترل های server side حاصل می شود. دوم: Page Load همیشه اتفاق می افتد، پس نباید لیست را در Page Load همیشه پر کرد.

صفت IsPostBack

صفحه یک Property به نام IsPostBack دارد. پس if (Page.IsPostBack == false) معنی آن اینست که فقط دفعه اول، این کارها را انجام بده. دفعه اول PostBack صورت نمی گیرد و از دفعات بعد، PostBack صورت می گیرد.

IsPostBack به صورت Boolean است. اگر یک Boolean را می خواهید با True مقایسه کنید، درست نیست بنویسید ==true هیچی نگذارید همان معنی را می دهد، یعنی if (page.IsPostBack) برای مقایسه با true استفاده می شود. ولی برای نقیض آن به صراحت از == false استفاده می شود.

در زبانهای C Based برای یک دستور می توان { } را نگذاشت ولی شما همیشه بگذارید.

مفهوم refactoring

اصولاً همیشه یک تابع تعریف می کنید و سپس آن را صدا می زنید. شاید بگویید، من دوست دارم بعداً یک تابع به نام X وجود داشته باشد، پس اول می نویسید و بعد . Ctrl + را کلیک می نمایید و خودش آن را می سازد.

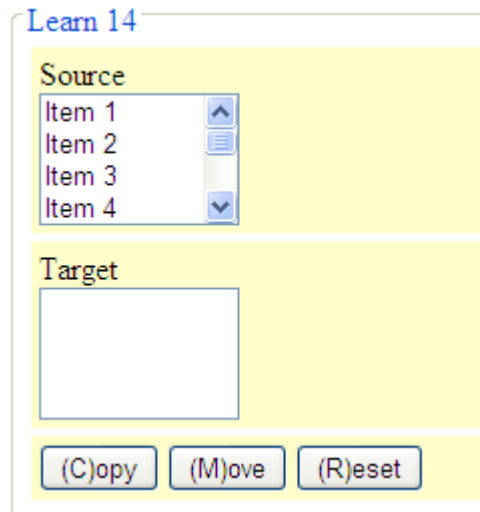
باید برای تمام کدها documentation داشته باشید، مهمترین document ها XML Documentation است. با زدن سه بار دکمه اسلش امکان توضیح وجود دارد.

```

/// <summary>
///
/// </summary>

```

به عنوان تمرین، layout ذیل را طراحی نمایید و کد آن را بنویسید.



با انتخاب چند آیتم از Source ListBox با نگه داشتن دکمه Ctrl و کلیک بر روی آیتم مورد نظر و سپس کلیک دکمه Copy، آیتمها به Target ListBox منتقل می‌شود و اگر آیتم به صورت تکراری انتخاب شود و Copy کلیک شود، آیتمهای تکراری کپی نشود. با کلیک بر روی دکمه Move آیتمهای انتخاب شده از Source حذف شود و به Target منتقل شود. کد آن به صورت ذیل می‌باشد:

/// طراحی Layout صفحه

```

<fieldset>
  <legend>
    Learn 14
  </legend>

  <div class="field">
    <asp:label id="lblSource" runat="server" text="Source" />
    <br />
    <asp:listbox id="lstSource" runat="server"
selectionmode="Multiple"
                accesskey="1" width="100" />
  </div>

  <div class="field">
    <asp:label id="lblTarget" runat="server" text="Target" />
    <br />
    <asp:listbox id="lstTarget" runat="server"
selectionmode="Multiple"
                accesskey="2" width="100" />
  </div>

  <div class="buttons">
    <asp:button id="btnCopy" runat="server" text="(C)opy"
onclick="btnCopy_Click" accesskey="C" />
    <asp:button id="btnMove" runat="server" text="(M)ove"
onclick="btnMove_Click" accesskey="M" />
    <input type="reset" value="(R)eset" accesskey="R" />
  </div>

```

</fieldset>

```

// کد برنامه
protected void Page_Load(object sender, System.EventArgs e)
{
    //اگر بار اول بود
    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
    lstSource.Items.Clear();

    System.Web.UI.WebControls.ListItem oListItem = null;

    for (int intIndex = 1; intIndex <= 9; intIndex++)
    {
        string strValue =
            intIndex.ToString();

        string strText =
            string.Format("Item {0}", intIndex);

        oListItem =
            new System.Web.UI.WebControls.ListItem(strText, strValue);

        lstSource.Items.Add(oListItem);
    }
}

/// <summary>
/// تمام آیتم های انتخاب شده در مبداء را به مقصد کپی می نمایم
/// </summary>
protected void btnCopy_Click(object sender, System.EventArgs e)
{
    //روی تمام آیتم های مبداء لوپ می زنیم
    for (int intIndex = 0; intIndex <= lstSource.Items.Count - 1; intIndex++)
    {
        //برای سهولت کار، آیتم مربوطه را در داخل یک متغیر کمکی ذخیره می نمایم
        System.Web.UI.WebControls.ListItem oCurrentListItem =
            lstSource.Items[intIndex];

        //در صورتی که آیتم مورد نظر، انتخاب شده بود
        if (oCurrentListItem.Selected)
        {
            //آیتم مربوطه را از حالت انتخاب شده خارج می نمایم
            oCurrentListItem.Selected = false;

            //در صورتی که آیتم مربوطه در مقصد وجود نداشت
            if (lstTarget.Items.Contains(oCurrentListItem) == false)
            {
                //آیتم مربوطه را به مجموعه آیتم های مقصد اضافه می کنیم
                lstTarget.Items.Add(oCurrentListItem);
            }
        }
    }
}
}

```

```

protected void btnMove_Click(object sender, System.EventArgs e)
{
    //روی تمام آیتم های مبدا لوپ می زنیم
    for (int intIndex = 0; intIndex <= lstSource.Items.Count - 1; intIndex++)
    {
        //برای سهولت کار، آیتم مربوطه را در داخل یک متغیر کمکی ذخیره می نماییم
        System.Web.UI.WebControls.ListItem oCurrentListItem =
            lstSource.Items[intIndex];

        //در صورتی که آیتم مورد نظر، انتخاب شده بود
        if (oCurrentListItem.Selected)
        {
            //در صورتی که آیتم مربوطه در مقصد وجود نداشت
            if (lstTarget.Items.Contains(oCurrentListItem) == false)
            {
                //آیتم مربوطه را به مجموعه آیتم های مقصد اضافه می کنیم
                lstTarget.Items.Add(oCurrentListItem);
            }
        }
    }

    //چون می خواهیم آیتم های انتخاب شده در مبدا را حذف نماییم
    //و این احتمال وجود دارد که بیش از یک آیتم در مبدا، برای حذف وجود داشته باشد
    //روی تمام آیتم های مبدا از انتها به ابتدا لوپ می زنیم
    for (int intIndex = lstSource.Items.Count - 1; intIndex >= 0; intIndex--)
    {
        //برای سهولت کار، آیتم مربوطه را در داخل یک متغیر کمکی ذخیره می نماییم
        System.Web.UI.WebControls.ListItem oCurrentListItem =
            lstSource.Items[intIndex];

        //در صورتی که آیتم مورد نظر، انتخاب شده بود
        if (oCurrentListItem.Selected)
        {
            //با توجه به اندیس آیتم مربوطه، آنرا از مجموعه آیتم های مبدا حذف نمی نماییم
            lstSource.Items.RemoveAt(intIndex);
        }
    }

    //مجددا بر روی تمام آیتم های مقصد لوپ می زنیم
    for (int intIndex = 0; intIndex <= lstTarget.Items.Count - 1; intIndex++)
    {
        //برای سهولت کار، آیتم مربوطه را در داخل یک متغیر کمکی ذخیره می نماییم
        System.Web.UI.WebControls.ListItem oCurrentListItem =
            lstTarget.Items[intIndex];

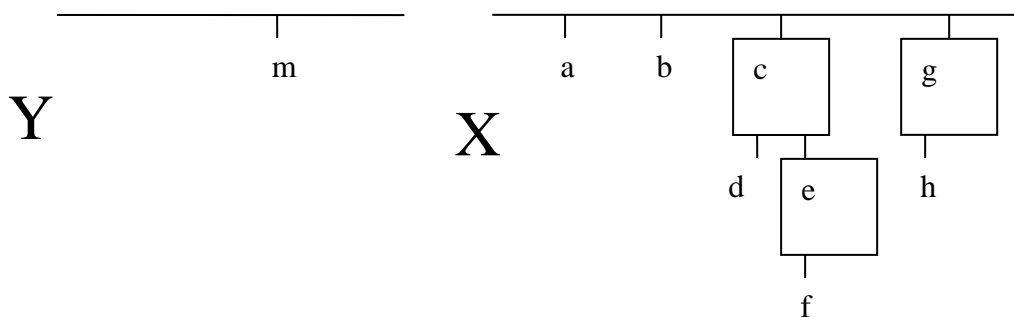
        //در صورتی که آیتم مورد نظر، انتخاب شده بود
        if (oCurrentListItem.Selected)
        {
            //آیتم مربوطه را از حالت انتخاب شده خارج می نماییم
            oCurrentListItem.Selected = false;
        }
    }
}

```

جلسه چهارم

آدرس دهی

ما یک سری فولدر و فایل در سایت داریم. وقتی که می خواهیم در یک فایل، آدرس فایل دیگر را بدهیم، چگونه باید آدرس دهی کنیم. اگر در فایل a هستیم و می خواهیم آدرس یک فایل خارج از سایت خود را بدهیم، هیچ راهی به جز استفاده از آدرس دهی مطلق ندارید.



آدرس دهی مطلق

یعنی اول پروتکل و بعد آدرس سایت و سپس نام فایل

<http://www.MyWebSite.com/Books/a.htm>

اگر فایل در سایت خودتان هم باشد، می توانید از آدرس دهی مطلق استفاده کنید، ولی اصلاً قشنگ نیست. اگر می خواهید آدرس یک شهر را از شهر دیگر بدهیم، بهتر است آدرس دهی مطلق را استفاده کنیم (وقتی که آدرس یک سایت را از سایت دیگر می دهید بهتر است که از آدرس دهی مطلق استفاده کنید) ولی وقتی که در شهر هستیم و یکی آدرس جایی در همان شهر را می پرسند، نباید آدرس مطلق دهید، باید آدرس نسبی بدهید، که البته دو نوع آدرس دهی نسبی وجود دارد، یکی نسبت به جایی که هستیم و دیگری نسبت به root.

آدرس دهی نسبی، نسبت به جایی که هستیم

آدرس دهی نسبی (نسبت به جایی که هستیم)، یعنی از جایی که الان هستیم باید کدام آدرس را طی کنیم تا به مقصد برسیم.

اگر داخل a هستیم و می خواهیم آدرس b را بدهیم. فقط بگویید b کافی است. اگر داخل f هستیم و بخواهیم آدرس b را بدهیم، باید بگویید ../b. در تمام تکنولوژیهای وب، به جز net. می توانستید به جای ../b فقط b بگذارید، این حتی در آدرس دهی داسی هم جواب می داد ولی در net. جواب نمی دهد. حتی اگر ده فولدر تو در تو داشته باشید و بخواهید آدرس

فایلی در root را به صورت نسبی (نسبت به جایی که هستیم) را بیان کنید، باید ده مرتبه علامت /.. را بگذارید و معادل دیگری ندارد.

آدرس دهی نسبی، نسبت به root

یک آدرس دهی نسبی دارید که به آن آدرس دهی نسبی (نسبت به root) می گویند و این آدرس دهی فقط به درد کنترل های server side یا در کدهای C# می خورد. و در این آدرس دهی می گویند که اگر از root به مقصد بخواهید بروید، باید چه آدرسی را طی کنید. یعنی همیشه از root آدرس را می دهید.

فایل مقصد	جایی که هستیم	آدرس مطلق	آدرس نسبی، نسبت به جایی که هستیم	آدرس نسبی نسبت به root
b d f h	a	http://www.x.com/b http://www.x.com/c/d http://www.x.com/c/e/f http://www.x.com/g/h	b c/d c/e/f g/h	~/b ~/c/d ~/c/e/f ~/g/h
b f h	d	http://www.x.com/b http://www.x.com/c/e/f http://www.x.com/g/h	../b e/f ../g/h	~/b ~/c/e/f ~/g/h
b d h	f	http://www.x.com/b http://www.x.com/c/d http://www.x.com/g/h	../../b != /b ../d ../../g/h	~/b ~/c/d ~/g/h

مثال: یک فولدر Images است و در آن یک فایل به نام Banner.jpg وجود دارد. یک فایل default.aspx وجود دارد که می خواهد Banner را نمایش دهد. از آدرس دهی نسبی (نسبت به root) استفاده می کنم.

```

```

این کاملاً غلط است، چرا که این تگ سمت client است و نمی توان از این نوع آدرس دهی (آدرس دهی نسبی، نسبت به root) استفاده نمود. ولی برای این تگ یک معادل server side داریم به نام <asp:image> که اگر از این تگ استفاده کنید، در آن بهتر است از آدرس دهی نوع سوم (آدرس دهی نسبی، نسبت به root) استفاده کنیم.

علائم مخفف (abbreviation):

b نشان دهنده blank می باشد.

/ نشان دهنده slash است.

\ نشان دهنده Back Slash است.

ما یک نوع عملیات I/O داریم که برای انجام آن، از یک سری موجودات استفاده می کنیم که فقط نیاز به آدرس فیزیکی دارند. ولی ما نباید از آدرس فیزیکی استفاده کنیم. وقتی که یک host به ما می دهند، چه داخلی و چه آن طرف آب، Host می تواند هر وقت که خواست سایت را از یک دایرکتوری به دایرکتوری دیگر یا حتی سرور دیگر منتقل کند. طبق قانون hosting نه تنها نباید از شما اجازه بگیرد، بلکه حتی نیاز به notification دادن به شما هم ندارد. تمام توابع مربوط به I/O مانند

... System.io.stream reader تمام اینها نیاز به آدرس فیزیکی دارند ولی ما نباید از آدرس فیزیکی استفاده کنیم. پس باید چه کار کنیم؟

موجودی (object) در server side داریم به نام Server. این موجود با کمک متدی به نام MapPath آدرس دهی نسبی نسبت به جایی که هستیم یا آدرس دهی نسبی، نسبت به root را می گیرد و خودش آدرس فیزیکی را بر می گرداند. بهتر است که از آدرس دهی نسبی، نسبت به root به عنوان پارامتر ورودی این تابع استفاده کنید. پس ما چهار نوع آدرس دهی داریم:

- 1 - آدرس دهی مطلق
- 2 - آدرس دهی نسبی (نسبت به جایی که هستیم)
- 3 - آدرس دهی نسبی (نسبت به root)
- 4 - آدرس دهی با کمک تابع Server.MapPath(a kind of relational address) که در واقع آدرس فیزیکی را بر می گرداند.

طراحی CMS

بر روی سایتها یک سری صفحات هستند که یک سری کنترل مخصوص با هدف خاص بر روی آن چیده شده است و به صورت ثابت هستند، اینها مهم نیستند، ولی یک سری صفحات هستند که content دارند، یعنی متن و عکس و ... دارند. وقتی چنین صفحاتی را تولید می کنید، تکنیک آن این است، که در ContentPlaceholder متن خود را می گذارید. کسی که با سایت کار می کند، اگر بخواهد متنی را تغییر دهید، یا کاری بر روی این صفحه انجام دهد، برای اینکار باید از شما کمک بگیرد، این کار خوشایند نیست و در ثانی این کار چینی برای شما به عنوان یک طراح و برنامه نویس سایت می باشد. بنابراین معمولاً تفکری را اعمال می کنند به نام CMS یا همان Content Management System که یک کتاب 400 صفحه ای است که در 1.5 جلسه یاد می گیریم. ما در ASP.NET، 8 فولدر با شخصیت داریم. یک سری فولدر داریم که این فولدرها با شخصیت هستند، یعنی برای .net. معنی دارند. یک سری فولدر بی شخصیت ولی پر کاربرد داریم، مثل image و CSS و یک سری فولدر بی شخصیت، بی کاربرد داریم، مثلاً فولدرهایی که خودتان به طور خاص برای سایت خود درست می کنید، مانند Picture و

فولدر App_Data

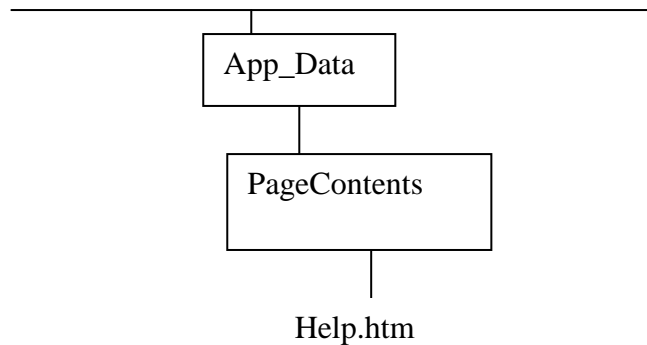
این 8 فولدر با شخصیت را تا انتهای ترم بررسی خواهیم کرد، یکی از این فولدرهای با شخصیت App_Data است، این فولدر به فولدر Secure مشهور است. اگر فایل مورد نظر شما که می خواهید download کنید در مسیر app_Data باشد، .net framework این فولدر را به عنوان یک فولدر امن در نظر می گیرد و اگر بخواهید به آن دسترسی پیدا کنید، error امنیتی می دهد، پس فایلهای امن خود را در این مسیر قرار بدهید. اگر سایت ما با بانک اطلاعاتی سر و کار داشته باشد و بانک اطلاعاتی آن سنگین نباشد، بانک خود را در این فولدر می گذاریم. در واقع SQL server Compaq edition داریم که می توانیم با این کار کنیم و فایل دیتابیس حاصل را در مسیر App_Data قرار دهیم و یا بانک اطلاعاتی access را هم در این مسیر قرار می دهند، تا امنیت آن حفظ شود.

پس گفتیم که داخل تگ <asp:content ...> باید متن و محتویات متنی و عکس قرار گیرد. پس یک فولدر PageContents در داخل App_Data درست می کنیم. و داخل آن یک صفحه Help.htm درست می کنیم و محتویات متنی را در داخل آن قرار می دهیم.

از یک تگ server side با نام <asp:litral> استفاده می کنیم که از label هم ضعیف تر است، Label حداقل به یک span تبدیل می شود، ولی این تگ خیلی سبک است و هیچ بار اضافی ندارد، یعنی هر چیزی که داخل آن بگذارید را عیناً نشان

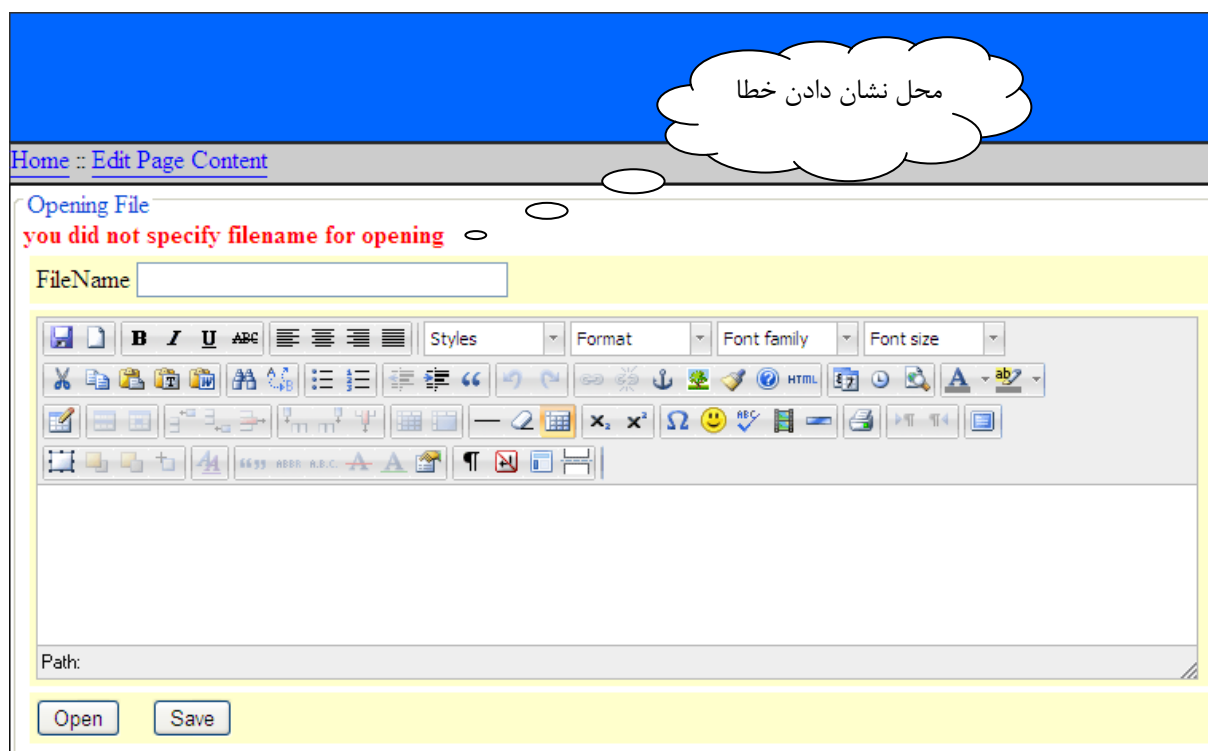
می‌دهد و بعد در Page load صفحه می‌توانید در صورتی که `if(Page.IsPostBack == false)` است عملیات initialize را انجام دهید.

```
<asp:Literal id="lit Content" runat="Server" />
```



یک صفحه به نام Manage درست می‌کنیم، که امکان load و ذخیره فایل help.htm در آن صفحه وجود داشته باشد. البته در این صورت باید فرد مسئول سیستم، خودش HTML بلد باشد، البته در مرحله بعد، کنترل Rich Text box می‌گذاریم که مثل محیط word است و فردی که با آن کار می‌کند، لازم نیست که HTML بلد باشد و می‌تواند به صورت گرافیکی تغییرات مورد نظر خود را اعمال نماید. پس اگر همه این کارها صورت گیرد، یک CMS طراحی شده است. برای شروع یک پروژه جدید، مراحل ذیل را مثل همیشه انجام می‌دهیم:

- یک فولدر به نام CSS و داخل آن یک فایل stylesheet با عنوان Site.css داشته باشید. یک سری ویژگی مثل left to right (ltr) و Right to left (rtl) و ... را باید تعریف کنید.
- بعد div.field را تعریف می‌کنیم و div.Buttons را هم تعریف می‌کنیم.
- بعد هم link , visited, hover, active و ...
- یک مورد به نام divMain داریم که برای این است که نمی‌خواهیم از ساختار جدولی برویم و می‌خواهیم به صورت div کار کنیم.
- divHeader , divFooter مثل هم در فایل Site.CSS تعریف می‌شود.
- divSubHeader , divSubFooter را هم فعلاً مثل هم تعریف می‌کنیم.
- در واقع همانهایی که در خصوص روش table ای گفته بودیم در خصوص div هم استفاده می‌شود. هر چیزی که خاصیت table دارد باید padding , margin آن را در اول کار صفر کنید.
- لینکهایی که در divmain هستند، باید خصوصیت رنگی خاصی داشته باشند که ذکر می‌کنیم، و نه همه آنها.
- بعد MasterPage ای به نام Site.master درست می‌کنیم. حتماً stylesheet را قبل از ContentPlaceHolder مربوط به head می‌گذاریم و یک ContentPlaceHolder هم که برای body داریم که در داخل آن باید divHeader , divMain و سایر div های مربوط به subfooter و footer و ... را بگذاریم.
- بعد در داخل آن using ها را پاک می‌کنیم و بعد به صورت تیپ initialize را در Page load مربوط به Master Page در جاهایی که IsPostBack نیست بگذارید.
- یک نکته مهم: سه تا صفحه هستند که نباید اسم کلاس آنها همان اسم باشد: default , profile , Login اسم کلاس آن را تغییر می‌دهیم و Page سر آن می‌گذاریم. PageDefault , PageProfile , PageLogin می‌گذاریم یعنی اسم صفحه Default است ولی اسم کلاس آن PageDefault است.



ابتدا یک فولد با شخصیت به نام App_Data به پروژه اضافه می‌کنیم. در خصوص این فولدر نباید با right click بر روی پروژه new folder را بزنیم بلکه باید **ADD Asp.Net Folder** را انتخاب کنیم که پس از انتخاب آن 8 فولدر با شخصیت را می‌بینیم و App_Data را اضافه می‌کنیم. و بعد یک فولدر به نام PageContents درست می‌کنیم و داخل آن هم default.htm استفاده می‌کنیم، که داخل آن فقط Hello word درج شده است.

برای خواندن content یک صفحه باید از یک موجودی به نام StreamReader استفاده کنید. StreamReader یک کلاس است که از icon آن می‌توانید بفهمید (در .net از آیکن‌های مختلف برای مشخص کردن کلاسها، attribute ها و method ها و ...) استفاده شده است. کلاً .net 54000 کلاس دارد که در namespace های مختلف آن را پخش کرده است.

یک سری تعاریف اولیه را یاد می‌گیریم. فایل ذیل را که با Full Path آن آمده است، مشاهده کنید.

C:\X\Y\z\t.txt

به t.txt می‌گویید، filename و به C:\X\Y\z Path. می‌گویند و به ترکیب filename و Path ، Path Name می‌گویند. البته خود ماکروسافت گاهی رعایت نمی‌کند، مثلاً در دستور StreamReader گفته پارامتر ورودی از نوع path است ولی در واقع Path name است. ولی ما در تعریف متغیرها این استانداردهای ماکروسافت را رعایت می‌کنیم.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack == false)
    {
        initialize();
    }
}

private void initialize()
{
```

```

string strFileName = "Default.htm";
string strRootRelativePath = "~/App_Data/PageContents";
string strRootRelativePathName = strRootRelativePath + "/" + strFileName;

string strPathName = Server.MapPath(strRootRelativePathName);
if (System.IO.File.Exists(strPathName))
{
    System.IO.StreamReader oStream = null;

    try
    {
        oStream = new System.IO.StreamReader(strPathName,
            System.Text.Encoding.UTF8);
        litContent.Text = oStream.ReadToEnd();
    }

    catch (System.Exception ex)
    {
        litContent.Text = ex.Message;
    }

    finally
    {
        if (oStream != null)
        {
            oStream.Close();
            oStream.Dispose();
            oStream = null;
        }
    }
}
}
}

```

مکانیزم Error Handling

هر جایی از برنامه که حدس می‌زنید که ممکن است با خطا روبه‌رو شود آن را در بلاک try catch finally قرار دهید. در واقع دستوری را که ممکن است به دلایلی با خطا مواجه شود را در داخل try بنویسید و در cache بگویید که در صورت روی دادن خطا، چه کاری انجام دهد. هر دستوری را که در دل finally بنویسید، بلافاصله اجرا می‌شود. برای بیرون آمدن از حلقه‌ها یا هر جایی می‌توانید از return و همچنین throw (پرت کردن) استفاده کنید. Finally به درد انجام عملیات finalize می‌خورد، مثلاً Close کردن فایل باز و ... یعنی یک سری کارهای لازم را که مثلاً باعث آزاد شدن حافظه یا محافظت از خراب شدن فایل یا هر چیز دیگری می‌شود، را قبل از خاتمه انجام دهید.

همه فایلها را به صورت Unicode ای ذخیره و باز کنید تا با تمام سیستمهای عامل همخوانی داشته باشد. پس از روش چهارم streamReader استفاده می‌کنیم که مسیر فیزیکی و نحوه encoding را سوال می‌کند. هم موقع open و هم موقع save ما از روش UTF8 استفاده می‌کنیم.

حال صفحه ای می‌خواهیم طراحی کنیم که بتوان از طریق آن این صفحه Default.htm را فراخوانی و edit کنیم. با توجه به اینکه این قسمت را فقط admin سیستم می‌تواند انجام دهد، پس یک فولدر به نام Admin درست می‌کنیم و در داخل آن یک صفحه با نام EditPageContent.aspx درست می‌کنیم. اصولاً چیزی را که در Legend می‌نویسیم با، Title صفحه یکی است.

یک fieldset می‌گذاریم

```

<fieldset>
    <legend>
        Edit Page Content
    </legend>

    <div id="divErrorMessage" runat="server">
        <asp:Label ID="lblErrorMessage" runat="server"/>
    </div>

    <div class="field">
        <asp:Label ID="lblFileName" runat="server" Text="FileName"/>
        <asp:TextBox ID="txtFileName" runat="server" Width="227px" />
        <asp:TextBox ID="txtPageContent" runat="server" TextMode="Multiline"
            Height="172px" Width="644px />
    </div>

    <div class="buttons">
        <asp:Button ID="btnOpen" runat="server" Text="Open"
            OnClick="btnOpen_Click"/>
        <asp:Button ID="btnSave" runat="server" Text="Save"
            onclick="btnSave_Click"/>
    </div>
</fieldset>

```

برای textbox ها
برای MaxLength
دکمه‌ها access key
هم تعریف می‌کنیم.

Textbox ها سه مد دارند، همان text و Password و multiline .

4 نوع Message وجود دارد. یکی error و دیگری information در این درس این دو نوع لازم است.

بنابراین دو تابع تعریف می‌کنیم با نامهای DisplayErrorMessage و DisplayInformationMessage ، تا بتوانیم بعداً آن را فراخوانی کنیم.

```

private void DisplayErrorMessage(string strErrorMessage)
{
    divErrorMessage.Visible = true;
    lblErrorMessage.Text = strErrorMessage;
    lblErrorMessage.CssClass = "error";
}
private void DisplayInformationMessage(string strInformationMessage)
{
    divErrorMessage.Visible = true;
    lblErrorMessage.Text = strInformationMessage;
    lblErrorMessage.CssClass = "information";
}

```

```

span.error
{
    color: Red;
    font-weight:bold;
}

```

```

span.information
{
    color: Blue;
    font-weight:bold;
}

```

<asp:label> به
render ، span
می‌شود.

یکی از روشهای hack کردن سایت روش exploite کردن سایت است، یعنی طرف در textbox تگهای html می نویسد، در asp کلاسیک خودتان باید چک کنید، ولی در .net خودش چک می کند و در صورت نوشتن tag html به شما error می دهد و شما باید به .net بفهمانید که این صفحه تحت کنترل است و گیرنده و content آن را نباید چک کنید، باید Validate Request آن را false کنید.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master"
CodeFile="EditPageContent.aspx.cs" Inherits="EditPageContent" ValidateRequest="false"
%>
```


فرق تگ HyperLink با a چیست؟ تگ hyperlink از تگهای server side ای است و می توان برای آن از آدرس دهی نسبی نسبت به root استفاده نمود. پس اگر در صفحه master page بخواهید یک لینک برقرار کنید، بهتر است از تگ hyperlink استفاده کنید، چون این صفحه می تواند خوراک صفحات مختلفی با مسیرهای مختلف باشد، پس ما جایی که الان هستیم را نمی دانیم و بهترین گزینه آدرس دهی نسبی، نسبت به root می باشد.

Snippet چیست؟ اگر بعد از زدن کلید واژه هایی مثل try ، if ، و ... دوبار دکمه tab را کلیک نمایید (tab tab)، به صورت هوشمندانه ای در تکمیل کد به ما کمک می کند.

حدود چهل و خورده ای rich text box در دنیا وجود دارد. مهمترین وب لاگ دنیا در حال حاضر World Press است.

هیچ rich text box در دنیا به صورتی که همزمان برای هر پاراگراف راست به چپ و چپ به راست را پشتیبانی کند، وجود ندارد. tinyMce را در google جستجو کنید و بعد وارد صفحه download این سایت شوید. در download کلی اطلاعات وجود دارد. تنها چیزی که باید Download کنید، TinyMce 3.4.3.2 است. نیاز به VPN ندارد.

Home
Try it
Download
Documentation
Enterprise
Develop
Forum



Version: 3.4.3.2
tinymce
Javascript WYSIWYG Editor

You are here: [Download](#)

Download

+ Share

Main packages (3.4.x)

Package	Description	Download
TinyMCE 3.4.3.2	Contains all you need for production usage.	DOWNLOAD
TinyMCE 3.4.3.2 jQuery package	Contains special jQuery build of TinyMCE and a jQuery integration plugin.	DOWNLOAD
TinyMCE 3.4.3.2 development package	This package contains development tools and full source code.	DOWNLOAD

Tinymce امکان چند زبانه شدن را دارد، به طوریکه در هر سایت، به یک زبان نشان دهد. بنابراین به قسمت Language آن بروید و زبان English و Persian را انتخاب کنید و download را بزنید.



یک فایل zip را می‌خواهد download کند، که download نمایید. این فایل zip را باز کنید و سپس یک مرحله پیش بروید تا به سه folder، Plugins, themes, langs، برسید (..\tinymce_language_pack\tinymce_language_pack). بعد داخل tiny اولیه آنقدر پیش بروید تا به همان سه فولدر برسید (tinymce_3.4.3.2\tinymce\jscripts\tiny_mce) و Paste کنید و بعد دو لایه بیابید بالا تا به فولدر jscripts برسید و بعد آن را به Root سایت خود اضافه کنید.

Languages for "tinymce"



Select language checkbox to download the language as a package, or you can [edit the languages](#) directly on our website.

<input type="checkbox"/>	Language	Code	Rev	Status	Updated
<input type="checkbox"/>	Albanian	sq	1	Incomplete	2010-12-16 19:59:16
<input type="checkbox"/>	Arabic	ar	1.9	Complete	2011-06-20 14:58:00
<input type="checkbox"/>	Armenian	hy	1	Incomplete	2010-12-16 19:39:38
<input type="checkbox"/>	Azerbaijani	az	1	Incomplete	2010-12-16 19:19:57
<input type="checkbox"/>	Basque	eu	1	Incomplete	2010-12-16 19:32:53
<input type="checkbox"/>	Belarusian	be	1	Incomplete	2010-12-16 19:21:03
<input type="checkbox"/>	Bengali	bn	1.1	Incomplete	2011-03-04 05:38:28
<input type="checkbox"/>	Bokmål, Norwegian; Norwegian Bokmål	nb	1.6	Incomplete	2011-04-13 18:47:33
<input type="checkbox"/>	Bosnian	bs	1	Incomplete	2010-12-16 19:25:22
<input type="checkbox"/>	Breton	br	1	Incomplete	2010-12-16 19:25:02
<input type="checkbox"/>	Bulgarian	bg	1.7	Complete	2011-05-04 18:40:45
<input type="checkbox"/>	Burmese	my	2.3	Incomplete	2011-01-22 13:10:47
<input type="checkbox"/>	Catalan; Valencian	ca	1.8	Complete	2011-04-28 11:03:04
<input type="checkbox"/>	Central Khmer	km	2.1	Incomplete	2011-01-25 02:57:35
<input type="checkbox"/>	Chamorro	ch	1	Incomplete	2010-12-16 19:26:58
<input type="checkbox"/>	Chinese	zh		Incomplete	2011-03-12 09:15:00
<input type="checkbox"/>	Croatian	hr	1.4	Incomplete	2011-04-15 17:25:27
<input type="checkbox"/>	Czech	cs	2.1	Complete	2011-07-13 10:56:21
<input type="checkbox"/>	Danish	da	1.3	Complete	2011-07-14 12:07:17
<input type="checkbox"/>	Divehi; Dhivehi; Maldivian	dv	1	Incomplete	2010-12-16 19:29:17

بعد بروید به صفحه‌ای که می‌خواهید text box multi line شما این کار را انجام دهد، یعنی می‌خواهید text box با مد multi line شما به صورت rich text box در آید. در آن صفحه با drag & drop بروید tiny_msce.js را بیاورید. پس خط ذیل به کد شما اضافه می‌شود:

```
<script src="tinymce_3.4.3.2/jscripts/tiny_mce/tiny_mce.js">
```

```
type="text/javascript"></script>
```

بعد در سایت به try it بروید و در آنجا انواع و اقسام rich text box را به شما نشان می‌دهد، که همراه با دموی آن می‌باشد. نوع Full featured نوع مناسبی است. به View Source بروید. باید خط /script تا انتها را بیاورید، که در واقع configuration این موجود است.

The screenshot shows the TinyMCE website interface. At the top, there is a navigation menu with links for Home, Try it (highlighted), Download, Documentation, Enterprise, Develop, and Forum. Below the menu is a banner for TinyMCE Version: 3.4.3.2, Javascript WYSIWYG Editor, featuring a puzzle graphic. The main content area has a breadcrumb trail: You are here: Try it > Full featured example. The title is 'Full featured example' with a share button and a print icon. Below the title are two tabs: 'View Example' (selected) and 'View Source'. The main content area displays a rich text editor toolbar and a text area with the following text:

This example displays all plugins and buttons that comes with the TinyMCE package.

Welcome to the TinyMCE editor demo!

Feel free to try out the different features that are provided, please note that the MCImageManager and MCFileManager specific functionality is part of our commercial offering. The demo is to show the integration.

We really recommend [Firefox](#) as the primary browser for the best editing experience, but of course, TinyMCE is [compatible](#) with all major browsers.

Got questions or need help?

If you have questions or need help, feel free to visit our [community forum](#)! We also offer Enterprise [support](#) solutions. Also do not miss out on the [documentation](#), its a great resource wiki for understanding how TinyMCE works and integrates.

```
<script type="text/javascript">
tinyMCE.init({
    // General options
    mode : "textareas",
    theme : "advanced",
    plugins :
"autolink,lists,spellchecker,pagebreak,style,layer,table,save,advhr,advimage,
advlink,emotions,iespell,inlinepopups,insertdatetime,preview,media,search
replace,print,contextmenu,paste,directionality,fullscreen,noneditable,visualchars,nonbreaking,xhtmlxtras,template",

    // Theme options
```

```

        theme_advanced_buttons1 :
"save, newdocument, |, bold, italic, underline, strikethrough, |, justifyleft, justifycenter, justifyright, justifyfull, |, styleselect, formatselect, fontselect, fontsizeselect",
        theme_advanced_buttons2 :
"cut, copy, paste, pastetext, pasteword, |, search, replace, |, bullist, numlist, |, outdent, indent, blockquote, |, undo, redo, |, link, unlink, anchor, image, cleanup, help, code, |, insertdate, inserttime, preview, |, forecolor, backcolor",
        theme_advanced_buttons3 :
"tablecontrols, |, hr, removeformat, visualaid, |, sub, sup, |, charmap, emotions, iespell, media, advhr, |, print, |, ltr, rtl, |, fullscreen",
        theme_advanced_buttons4 :
"insertlayer, moveforward, movebackward, absolute, |, styleprops, spellchecker, |, cite, abbr, acronym, del, ins, attribs, |, visualchars, nonbreaking, template, blockquote, pagebreak, |, insertfile, insertimage",
        theme_advanced_toolbar_location : "top",
        theme_advanced_toolbar_align : "left",
        theme_advanced_statusbar_location : "bottom",
        theme_advanced_resizing : true,

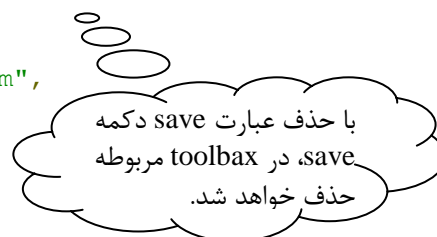
// Skin options
skin : "o2k7",
skin_variant : "silver",

// Example content CSS (should be your site CSS)
content_css : "css/example.css",

// Drop lists for link/image/media/template dialogs
template_external_list_url : "js/template_list.js",
external_link_list_url : "js/link_list.js",
external_image_list_url : "js/image_list.js",
media_external_list_url : "js/media_list.js",

// Replace values for the template plugin
template_replace_values : {
    username : "Some User",
    staffid : "991234"
}
});
</script>

```



می‌توانید configuration این موجود را تغییر دهد و بعضی از دکمه‌ها را بردارید و جای toolbox ها را عوض کنید. Mode "textareas" یا جای toolbox ها را عوض کنید. و ... می‌توانید دو configuration مختلف برای مد admin و مد client تعریف کنید.

در نهایت کد ذیل را در این صفحه می‌نویسیم:

```

public partial class EditPageContent : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        divErrorMessage.Visible = false;
    }
    protected void btnOpen_Click(object sender, System.EventArgs e)
    {
        string strRootRelativePath = "~/App_Data/PageContents";
        string strRootRelativePathName = strRootRelativePath + "/" + txtFileName.Text;
    }
}

```



```

string strPathName = Server.MapPath(strRootRelativePathName);

txtFileName.Text = txtFileName.Text.Trim();
if (txtFileName.Text == string.Empty)
{
    string strErrorMessage = "you did not specify filename for opening";
    DisplayErrorMessage(strErrorMessage);
    return;
}

if (System.IO.File.Exists(strPathName) == false)
{
    string strErrorMessage = "file specified does not exists!";
    DisplayErrorMessage(strErrorMessage);
    return;
}

if (System.IO.File.Exists(strPathName))
{
    System.IO.StreamReader oStream = null;
    try
    {
        oStream = new System.IO.StreamReader(strPathName,
System.Text.Encoding.UTF8);
        txtPageContent.Text = oStream.ReadToEnd();
        string strInformationMessage = "file is opened!";
        DisplayInformationMessage(strInformationMessage);
    }

    catch (System.Exception ex)
    {
        txtPageContent.Text = ex.Message;
    }

    finally
    {
        if (oStream != null)
        {
            oStream.Close();
            oStream.Dispose();
            oStream = null;
        }
    }
}
}

protected void btnSave_Click(object sender, System.EventArgs e)
{
    string strRootRelativePath = "~/App_Data/PageContents";
    string strRootRelativePathName = strRootRelativePath + "/" + txtFileName.Text;

    string strPathName = Server.MapPath(strRootRelativePathName);

    if (System.IO.File.Exists(strPathName))
    {
        System.IO.StreamWriter oStream = null;

        try
        {
            oStream = new System.IO.StreamWriter(strPathName, false,
System.Text.Encoding.UTF8);

```

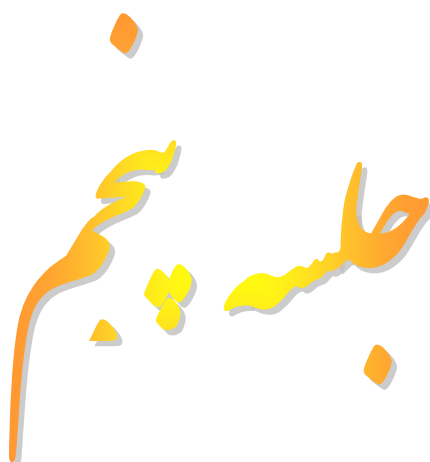
```
        oStream.Write(txtPageContent.Text);
    }

    catch (System.Exception ex)
    {
        // میتوان config تست برای
        txtPageContent.Text = ex.Message;
    }

    finally
    {
        if (oStream != null)
        {
            oStream.Close();
            oStream.Dispose();
            oStream = null;
        }
    }
}

private void DisplayErrorMessage(string strErrorMessage)
{
    divErrorMessage.Visible = true;
    lblErrorMessage.Text = strErrorMessage;
    lblErrorMessage.CssClass = "error";
}

private void DisplayinformationMessage(string strInformationMessage)
{
    divErrorMessage.Visible = true;
    lblErrorMessage.Text = strInformationMessage;
    lblErrorMessage.CssClass = "information";
}
}
```



امکان اضافه نمودن صفحات به صورت داینامیک در CMS

ما در جلسه قبل توانستیم محتویات صفحات موجود را update کنیم. الان می‌خواهیم به صورت Run time صفحه ایجاد کنیم. Behind code یک صفحه می‌تواند در یک صفحه جدا باشد، یا در همان صفحه باشد. ما نمی‌توانیم به صورت run time صفحه با پسوند CS را هم تولید کنیم. بنابراین به صورت run time صفحه ای تولید می‌کنیم که behind code آن در همان صفحه باشد.

بنابراین من یک صفحه خالی ایجاد می‌کنم و نام آن را template می‌گذارم ولی نمی‌خواهم این صفحه Behind code را در فایل جدا داشته باشد ولی فقط از master استفاده کند. بنابراین تیک مربوط به Place Code in Separate File آن را در زمان ایجاد بر می‌دارم و فقط تیک مربوط به Select Master Page را نگاه می‌داریم.

برای اینکه Content این صفحه هم بتواند تغییر کند باید در داخل ContentPlaceHolder مربوط به Body آن یک تگ لیترال بگذارم و بعد در Page-Load یک تابع initialize بنویسم تا این Literal را بخواند.

پس برای درست کردن فایل template به ترتیب مراحل ذیل را انجام می‌دهیم:

- 1 - فولدری با عنوان App_Data\PageTemplates درست می‌کنیم.
- 2 - در این فولدر یک صفحه جدید درست می‌کنیم و نام آن را Template.aspx می‌گذاریم. (تیک مربوط به Place Code in Separate File را بر می‌داریم و تیک مربوط به Select Master Page را می‌زنیم).
- 3 - در ContentPlaceHolder مربوط به قسمت Body نیز یک تگ literal می‌گذاریم.
- 4 - در قسمت Script ، تابع initialize را در Page_Load صفحه صدا می‌زنیم.
- 5 - در تابع initialize کد مربوط به خواندن محتویات یک صفحه در یک تگ literal را می‌نویسیم.
- 6 - در خط اول که صفات صفحه را مشخص می‌کنیم، title صفحه را مساوی [TITLE] قرار می‌دهیم. در تابع initialize نیز مقدار strFileName را مساوی [NAME].htm قرار می‌دهیم.
- 7 - پسوند این فایل را از.aspx به.txt تغییر می‌دهیم.

کد این صفحه به صورت ذیل می‌باشد:

```
<%@ page title="[TITLE]" language="C#" masterpagefile="~/Site.master" %>
```

```
<script runat="server">
```

```

protected void Page_Load(object sender, System.EventArgs e)
{
    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
    string strFileName = "[NAME].htm";
    string strRootRelativePath = "~/App_Data/PageContents";
    string strRootRelativePathName = strRootRelativePath + "/" + strFileName;

    string strPathName = Server.MapPath(strRootRelativePathName);

    if (System.IO.File.Exists(strPathName))
    {
        System.IO.StreamReader oStream = null;

        try
        {
            oStream =
                new System.IO.StreamReader(strPathName,
                    System.Text.Encoding.UTF8);

            litContent.Text = oStream.ReadToEnd();
        }
        catch (System.Exception ex)
        {
            litContent.Text = ex.Message;
        }
        finally
        {
            if (oStream != null)
            {
                oStream.Close();
                oStream.Dispose();
                oStream = null;
            }
        }
    }
}
</script>

<asp:content id="cntHead" contentplaceholderid="cphHead" runat="Server">
</asp:content>

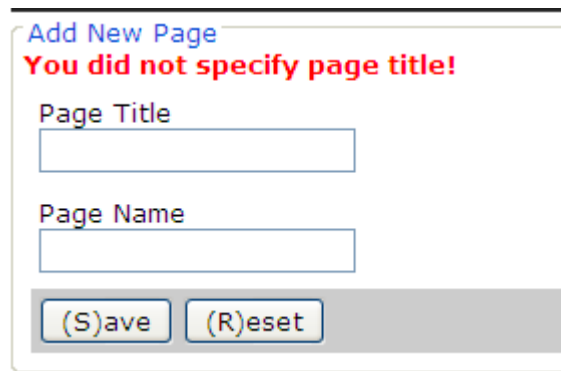
<asp:content id="cntMain" contentplaceholderid="cphMain" runat="Server">
    <asp:literal id="litContent" runat="server" />
</asp:content>

```

در قسمت Admin یک صفحه به نام AddNewPage درست می‌کنم. در Add new page باید یک صفحه باز شود که بپرسد، این صفحه جدید که می‌خواهید باز کنید، title آن چیست و از چه صفحه ای محتویات آن را بخواند. یک فولدر به نام Pages درست می‌کنم و کلیه صفحاتی را که به صورت dynamic درست می‌کنیم را در آن می‌گذاریم. پس چه چیزهایی در Add new page باید داشته باشم. یک Legend که با Title یکی است و یک divMessage دارم که پیامهای error و information را از طریق آن اعلام می‌کنم و یک Textbox , Label برای PageTitle و یک Textbox

label برای نام صفحه می‌گذارم. 100 کارکتر برای نام صفحه کافی است. دو دکمه هم داریم، یکی btnSave و یکی هم btnReset.

```
<fieldset>
    <legend>
        Add New Page
    </legend>
    <div id="divMessage" runat="server" class="message">
        <asp:label id="lblMessage" runat="server" />
    </div>
    <div class="field">
        <asp:label id="lblPageTitle" runat="server" text="Page Title" />
        <br />
        <asp:textbox id="txtPageTitle" runat="server" maxlength="100"
            accesskey="1" />
    </div>
    <div class="field">
        <asp:label id="lblPageName" runat="server" text="Page Name" />
        <br />
        <asp:textbox id="txtPageName" runat="server" maxlength="100"
            accesskey="2" />
    </div>
    <div class="buttons">
        <asp:button id="btnSave" runat="server" text="(S)ave"
            accesskey="S" onclick="btnSave_Click" />
        <input type="reset" value="(R)eset" accesskey="R" />
    </div>
</fieldset>
```



موقع save کردن ابتدا باید یک سری business checking انجام دهیم. مثلاً title خالی باشد، error بدهد یا ندهد و ...

```
txtPageTitle.Text = txtPageTitle.Text.Trim();
if (txtPageTitle.Text == string.Empty)
{
    string strErrorMessage =
        "You did not specify page title!";

    DisplayErrorMessage(strErrorMessage);
    return;
}
```

```

txtPageName.Text = txtPageName.Text.Trim();
if (txtPageName.Text == string.Empty)
{
    string strErrorMessage =
        "You did not specify page name!";

    DisplayErrorMessage(strErrorMessage);
    return;
}

```

اگر نام صفحه و title درست داده شده باشد و هیچ خطایی در این زمینه وجود نداشته باشد، باید یک صفحه جدید منطبق با template و با نام داده شده، ایجاد شود. یعنی در فولدر Pages یک صفحه با نام مشخص شده در textbox نام فایل و با پسوند aspx، کپی شود. پس ابتدا مسیر فیزیکی فایل Template را پیدا می‌کنیم.

```

string strFileName = "Template.txt";
string strRootRelativePath = "~/App_Data/PageTemplates";
string strRootRelativePathName = strRootRelativePath + "/" + strFileName;

string strPathName = Server.MapPath(strRootRelativePathName);

```

باید چک کند اگر فایل Template وجود نداشته خطا دهد.

```

if (System.IO.File.Exists(strPathName) == false)
{
    string strErrorMessage =
        string.Format("There is not the [{0}.txt] file!",
            strFileName);

    DisplayErrorMessage(strErrorMessage);
    return;
}

```

بعد از اینکه فایل مسیر فایل Template را پیدا کردیم و از وجود داشتن آن نیز مطمئن شدیم، فایل Template را از مسیر تعیین شده با کمک متد StreamReader می‌خوانیم.

```

string strTemplate = string.Empty;

System.IO.StreamReader oStreamReader = null;

try
{
    oStreamReader =
        new System.IO.StreamReader(strPathName,
            System.Text.Encoding.UTF8);

    strTemplate = oStreamReader.ReadToEnd();
}
catch (System.Exception ex)
{
    DisplayErrorMessage(ex.Message);
    return;
}
finally
{
    if (oStreamReader != null)

```

```

        {
            oStreamReader.Close();
            oStreamReader.Dispose();
            oStreamReader = null;
        }
    }
}

```

در فایل خوانده شده که در رشته strTemplate قرار گرفته است، مقدار [NAME] را با محتویات Textbox نام فایل و مقدار [TITLE] را با محتویات TextBox عنوان فایل جایگزین می‌کنیم.
روش اول:

```

strTemplate = strTemplate.Replace("[NAME]", txtPageName.Text);
strTemplate = strTemplate.Replace("[TITLE]", txtPageTitle.Text);

```

روش دوم:

```

strTemplate =
    strTemplate.Replace("[NAME]", txtPageName.Text).Replace("[TITLE]",
    txtPageTitle.Text);

```

سپس مسیری را که می‌خواهیم صفحه داینامیک را در آن ایجاد کنیم، پیدا می‌کنیم.

```

strRootRelativePath = "~/Pages";
strRootRelativePathName = strRootRelativePath + "/" + txtPageName.Text + ".aspx";

```

```

strPathName = Server.MapPath(strRootRelativePathName);

```

در مرحله آخر نیز متغییر strTemplate را که محتوی template می‌باشد و عملیات جایگزینی [Title] و [Name] نیز در آن صورت گرفته است، در مسیر مورد نظر ایجاد می‌کنیم. (از طریق تابع StreamWriter).

```

System.IO.StreamWriter oStreamWriter = null;

try
{
    oStreamWriter =
        new System.IO.StreamWriter(strPathName, false,
        System.Text.Encoding.UTF8);

    oStreamWriter.Write(strTemplate);
}
catch (System.Exception ex)
{
    DisplayErrorMessage(ex.Message);
    return;
}
finally
{
    if (oStreamWriter != null)
    {
        oStreamWriter.Close();
        oStreamWriter.Dispose();
        oStreamWriter = null;
    }
}

string strInformation =
    string.Format("The [{0}.aspx] page created successfully...",
    txtPageName.Text);

```

```
DisplayInformationMessage(strInformation);
```

امکان مدیریت نمودن صفحات به صورت داینامیک در CMS

در مرحله بعد یک صفحه Manage Pages که لیست صفحات را نشان می‌دهد درست می‌کنیم. صاحب سایت می‌تواند از طریق این صفحه، یک سری صفحات را delete و یا Rename کند. این صفحه شامل مولفه‌های ذیل می‌باشد.

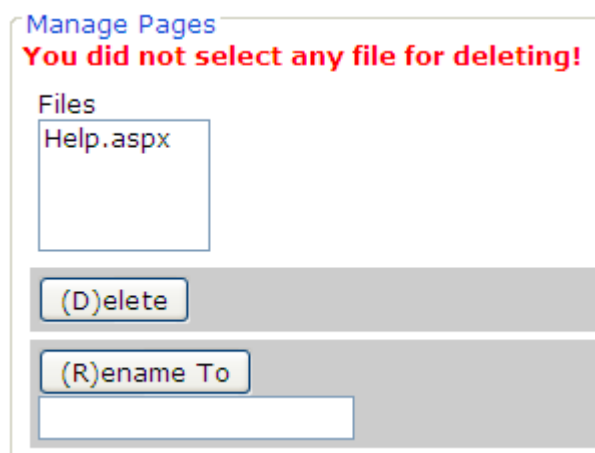
```
<fieldset>
  <legend>
    Manage Pages
  </legend>

  <div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
  </div>

  <div class="field">
    <asp:label id="lblFiles" runat="server" text="Files" />
    <br />
    <asp:listbox id="lstFiles" runat="server" accesskey="1"
      selectionmode="Single" />
  </div>

  <div class="buttons">
    <asp:button id="btnDelete" runat="server" text="(D)elete"
      accesskey="D" onclick="btnDelete_Click" />
  </div>

  <div class="buttons">
    <asp:button id="btnRename" runat="server" text="(R)ename To"
      accesskey="R" onclick="btnRename_Click" />
    <br />
    <asp:textbox id="txtNewPageName" runat="server" maxlength="100"
      accesskey="2" />
  </div>
</fieldset>
```



یکی از مهمترین name space ها، System.IO است.

ما نیاز به یک متدی داریم که فهرست فایل‌های موجود در یک مسیر را نشان دهد. یک تابع به نام FillListBox داریم و باید از جانوری استفاده کنیم که اشاره کند به ابتدای sub directory و من باید داخل فایلها loop بزنم و اسم آنها را در Listbox بیاورم. یک شی directoryInfo دارم. که در system.io است و آدرس آن نسبی است. برای loop زدن در آرایه دو راه داریم یا استفاده از index و یا استفاده از دستور for each به صورت ذیل که بسیار شیک تر است.

```

Foreach( )
{
}

```

باید به ازای هر فایل در این مسیر، Loop بزند. از طرف دیگر، چک می‌کنیم اگر extention فایل .aspx بود، نام فایل را به لیست اضافه می‌کنیم.

هر وقت یک لیست را پر کردید، حتماً قبل از پر کردن، آن را clear کنید.

تابع Selectedindex اگر که لیست خالی باشد، عدد 1- را بر می‌گرداند و اگر هم که هیچ آیتمی انتخاب نشده بود، باز هم 1- را بر می‌گرداند و در غیر اینصورت، اندیس آیتم انتخاب شده را بر می‌گرداند.

در خصوص delete کردن، باید مسیر فایل را که می‌خواهم پاک کنم را به دست آورم. پس ایندفعه rootrelativpath من می‌شود Pages و RootRelativePathName می‌شود، rootrelativePath + "/" + lstfiles.selectedValue.

همیشه هنگام کار بر روی فایل (ویرایش، حذف و تغییر نام) از cache , throw , try استفاده کنید، چون ممکن است به خطا بخورید.

از دستور move به جای rename استفاده می‌کنیم. وقتیکه می‌گوییم که یک فایل را از یک مسیر به همان مسیر با نام دیگر move کن یعنی rename انجام گرفته است.

کد مربوط به این صفحه به صورت ذیل می‌باشد:

```

protected void Page_Load(object sender, System.EventArgs e)
{
    divMessage.Visible = false;

    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
    FillListBox();
}

private void FilllListBox()
{
    string strRootRelativePath = "~/Pages";
    string strPath = Server.MapPath(strRootRelativePath);

    System.IO.DirectoryInfo oDirectoryInfo =
        new System.IO.DirectoryInfo(strPath);

    lstFiles.Items.Clear();
}

```

```

        foreach (System.IO.FileInfo oFileInfo in oDirectoryInfo.GetFiles())
        {
            if (oFileInfo.Extension.ToUpper().EndsWith(".ASPX"))
            {
                lstFiles.Items.Add(oFileInfo.Name);
            }
        }
    }

protected void btnDelete_Click(object sender, System.EventArgs e)
{
    if (lstFiles.SelectedIndex == -1)
    {
        string strErrorMessage =
            "You did not select any file for deleting!";

        DisplayErrorMessage(strErrorMessage);
        return;
    }

    string strRootRelativePath = "~/Pages";
    string strRootRelativePathName = strRootRelativePath + "/" +
        lstFiles.SelectedValue;
    string strPathName = Server.MapPath(strRootRelativePathName);

    //string strRootRelativePathContent = "~/App_Data";
    //string strRootRelativePathNameContent = strRootRelativePath + "/" +
        lstFiles.SelectedValue;
    //string strPathName = Server.MapPath(strRootRelativePathName);

    try
    {
        System.IO.File.Delete(strPathName);

        FillListBox();

        string strInformation =
            string.Format("The [{0}] page deleted successfully...",
                lstFiles.SelectedValue);

        DisplayInformationMessage(strInformation);
    }
    catch (System.Exception ex)
    {
        DisplayErrorMessage(ex.Message);
    }
}

protected void btnRename_Click(object sender, System.EventArgs e)
{
    if (lstFiles.SelectedIndex == -1)
    {
        string strErrorMessage =
            "You did not select any file for renaming!";

        DisplayErrorMessage(strErrorMessage);
        return;
    }

    txtNewPageName.Text = txtNewPageName.Text.Trim();
    if (txtNewPageName.Text == string.Empty)
    {

```

```

        string strErrorMessage =
            "You did not specify new page name!";

        DisplayErrorMessage(strErrorMessage);
        return;
    }

    string strRootRelativePath = "~/Pages";
    string strPath = Server.MapPath(strRootRelativePath);

    string strOldPathName = strPath + "\\\" + lstFiles.SelectedValue;
    string strNewPathName = strPath + "\\\" + txtNewPageName.Text + ".aspx";

    try
    {
        System.IO.File.Move(strOldPathName, strNewPathName);

        FillListBox();

        string strMessage =
            string.Format("You page renamed to {0}.aspx",
                txtNewPageName.Text);

        DisplayInformationMessage(strMessage);
    }
    catch (System.Exception ex)
    {
        DisplayErrorMessage(ex.Message);
    }
}

private void DisplayErrorMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "error";
}

private void DisplayInformationMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "information";
}

```

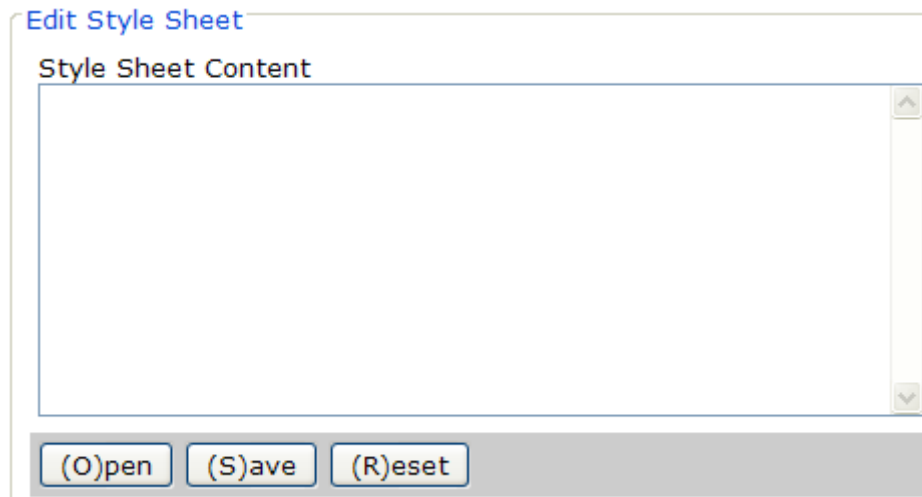
در Visual Studio 2010 اگر **Ctrl** و **scrol** ماوس را بگیرید، می‌توانید صفحه را **Zoom in** و **Zoom out** کنید.

امکان ویرایش فایل CSS به صورت داینامیک در CMS

یک سطح دسترسی به نام `account programmer` درست کنید که از `admin` نیز بالاتر است و این را به مشتری تحویل هم ندهید و یک سری از کارها را که نباید به مشتری تحویل دهید، بر عهده سطح دسترسی این `account` بگذارید. مثلاً `Edit` کردن خود `CSS` کار مشتری نیست، باید یک `account` به نام `Programmer` داشته باشید، تا اینکارها را انجام دهد. فایل‌های `Critical` را وقتیکه می‌خواهید `Save` کنید، مطمئن شوید که یک `backup` از آن تهیه کرده‌اید. در خصوص

ایجاد تغییر در فایل CSS اول می بینید، که الان site.back وجود دارد، اگر وجود داشت آن را update می کند و روی آن می نویسد وگرنه اول از نسخه قبلی backup می گیرد و بعد روی آن می نویسد. یک دکمه reset هم می گذارید که کار این دکمه این است که اگر اشتباهی مرتکب شدید، site.back را به جای site.css کپی کند.

ظاهر صفحه ایجاد تغییرات در Style Sheet :



آیتمهای موجود در صفحه ایجاد تغییرات در Style Sheet :

```
<fieldset>
  <legend>
    Edit Style Sheet
  </legend>

  <div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
  </div>

  <div class="field">
    <asp:label id="lblContent" runat="server" text="Style Sheet
      Content" />
    <br />
    <asp:textbox id="txtContent" runat="server" textmode="Multiline"
      rows="10" columns="60" accesskey="2" />
  </div>

  <div class="buttons">
    <asp:button id="btnOpen" runat="server" text="(O)pen"
      accesskey="O" onclick="btnOpen_Click" />
    <asp:button id="btnSave" runat="server" text="(S)ave"
      accesskey="S" onclick="btnSave_Click" />
    <asp:button id="btnReset" runat="server" text="(R)eset"
      accesskey="R" onclick="btnReset_Click" />
  </div>
</fieldset>
```

برنامه ایجاد تغییرات در صفحه Style Sheet :

```
protected void Page_Load(object sender, System.EventArgs e)
{
    if (Page.IsPostBack == false)
```

```
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnOpen_Click(object sender, System.EventArgs e)
    {
        string strRootRelativePath = "~/CSS";
        string strRootRelativePathName = strRootRelativePath + "/Site.css";

        string strPathName = Server.MapPath(strRootRelativePathName);

        System.IO.StreamReader oStream = null;

        try
        {
            oStream =
                new System.IO.StreamReader(strPathName,
                    System.Text.Encoding.UTF8);

            txtContent.Text = oStream.ReadToEnd();

            string strInformationMessage =
                "The style sheet file opened successfully...";

            DisplayInformationMessage(strInformationMessage);
        }
        catch (System.Exception ex)
        {
            DisplayErrorMessage(ex.Message);
        }
        finally
        {
            if (oStream != null)
            {
                oStream.Close();
                oStream.Dispose();
                oStream = null;
            }
        }
    }

    protected void btnSave_Click(object sender, System.EventArgs e)
    {
        string strRootRelativePath = "~/CSS";
        string strPath = Server.MapPath(strRootRelativePath);

        string strPathName = strPath + "\\Site.css";
        string strBackUpPathName = strPath + "\\Site.bak";

        if (System.IO.File.Exists(strBackUpPathName) == false)
        {
            try
            {
                System.IO.File.Copy(strPathName, strBackUpPathName);
            }
            catch (System.Exception ex)
            {
            }
        }
    }
}
```

```

        DisplayErrorMessage(ex.Message);
        return;
    }
}

System.IO.StreamWriter oStream = null;

try
{
    oStream =
        new System.IO.StreamWriter(strPathName, true,
            System.Text.Encoding.UTF8);

    oStream.Write(txtContent.Text);

    string strInformation =
        "Your style sheet file updated successfully...";

    DisplayInformationMessage(strInformation);
}
catch (System.Exception ex)
{
    DisplayErrorMessage(ex.Message);
}
finally
{
    if (oStream != null)
    {
        oStream.Close();
        oStream.Dispose();
        oStream = null;
    }
}
}

protected void btnReset_Click(object sender, System.EventArgs e)
{
}

private void DisplayErrorMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "error";
}

private void DisplayInformationMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "information";
}
}

```

مفاهیم Object Oriented و کلاس


Object Oriented یا شی گزایی خاص برنامه نویسی و کدنویسی نیست، در حوزه‌های مختلف مثلاً الکترونیک و مدیریت هم کاربرد دارد. Object oriented با entity هایی سر و کار دارد که در هر دنیایی معنی دارد. OO چندین مزیت دارد. که مهمترین آن این است که باعث می‌شود، برنامه‌های خیلی بزرگ را راحت‌تر و سریعتر بنویسید و به دنیای واقعی نزدیک‌تر است.

Object Oriented چیست؟

در OO ما مفهوم کلاس و object داریم، که هر کدام از آنها دارای یک سری اعضاء و member هستند. این اعضا عبارتند از property و method و Event .

کلاس چیست؟

کلاس مفهومی است انتزاعی که هنوز از آن هیچ نمونه، شی یا instance درست نشده است. دو تعبیر برای آن وجود دارد. زمانیکه به چاپخانه می‌روید، می‌گویید که کاغذ سربگردار می‌خواهم. این کاغذ هنوز وجود ندارد و شما کلاستان را تعریف می‌کنید و وقتی که چاپخانه کاغذها را تحویل داد، نمونه یا شی را به شما تحویل داده است. تعبیر دیگر اینکه وجه مشترک تمام اشیاء را بگیریم و کلاس را درست کنیم. مثلاً همه انسانهای روی زمین را در نظر می‌گیریم و یک مفهوم کلی و انتزاعی به نام کلاس انسان درست می‌کنیم. منتقدین نسبت به تعبیر دوم اظهار می‌کنند که پس با این تعریف هیچوقت نمی‌توان کلاس اژدها داشته باشیم، چون موجودی به نام اژدها واقعاً وجود ندارد. خداوند وقتیکه انسان را خلق نمود، یک تفکری از انسان در ذهن خدا بود. حضرت انسان، حوا و ما یک نمونه یا شی ای از آن مفهوم هستیم. هر کدام از ما دارای یک سری ویژگی field یا Property هستیم. مثل رنگ چشم، قد و رنگ مو و ...

یک سری متد و روش داریم، بدو و بنشین و پاشو و حرف بزن و ... ما تا این لحظه کلاس طراحی نکرده‌ایم و از کلاسهایی که دیگران تعریف کرده‌اند استفاده کرده‌ایم. آیکن یک دست کنار کاغذ ، همان Property ها هستند، پس همان صفاتی است که تا الان هم استفاده کردید. و متدها همان توابع هستند، متدها کارهایی است که باید انجام شود، و بهتر است با do ، let ، و Get ... شروع شود (یعنی به صورت دستوری باشد). event در واقع نصیحتهایی است که انجام گرفته است، اگر کسی صدات زد، باید این کار را کنی، اگر کسی نگاهت کرد، باید این کار را بکنی، اگر کسی ... باید این کار را بکنی. گاهی می‌گوییم خانم یا آقا باید این کار را بکنی، ولی گاهی می‌گوییم، اگر این اتفاق افتاد، باید این کار را انجام دهیم. پس اینها همان event ها هستند. Event هایی برای اشیاء وجود دارد. مثلاً اگر به این خودکار بیش از این فشار بیاورید، می‌شکند.

در بازی Need for speed ، می‌گوید این ماشینها هست، کدام را می‌خواهی. در ابتدای کار ماشینهایی که نشان می‌دهد، کلاس هستند و وقتی در طول بازی از یک ماشین خاص استفاده می‌کنی، یک شی از کلاس مورد نظر خود را می‌گیری و به آن می‌گویی چه کاری انجام بده، مثلاً گاز بده، شروع کن، حرکت کن ، راست برو و ... ، پس اینها method های آن است و event ها نیز عکس‌العملها است، مثلاً اگر به دیوار بخوری، این اتفاق خواهد افتاد و

گاهی اوقات در سیستم‌های موجود، event درست raised نشده است، که به آن باگ می‌گویند. مثلاً اگر ماشین به دیوار رسید و آن را رد کرد، یک event درست raised نشده است، که به صورت عامیانه به آن error یا باگ می‌گوییم. در دنیای کامپیوتر می‌گویید که یک asp.button می‌خواهید، پس یک شی از کلاس button می‌گیرید و آن را مقداردهی می‌کنید.

از کلاس StreamReader یک شی new می‌کردی. پس new یک شی درست می‌کند، بعد می‌گویی read یا write کن، پس بهش دستور می‌دهی. یک سری event هم دارد که می‌توانید استفاده کنید. مثلاً در صفحه دکمه save و open گذاشتیم.

بدون انجام کاری (یعنی بدون raised کردن event) هر چه روی دکمه کلیک می کردید، اتفاقی نمی افتاد، پس باید دکمه را نصیحت می کردید، یعنی یک evnet را بر روی آن شی raised می کردید.



کلاسها را به صورت reusable درست کنید و بعد از روی آن یک یا چند شی درست کنید.

Name Space چیست؟

در .net framework 2 54000 کلاس وجود دارد. فرض کنید، 54000 فایل دارید، از عکس و متن و آیا شما همه 54000 فایل را در هم نگاهداری می کنید، نه مسلماً دسته بندی می کنید. موزیکها جدا، در دسته موزیکها هم موزیک کلاسیک جدا، موزیک پاپ جدا و ...





پس ماکروسافت name space تعریف کرده تا کلاسها را دسته بندی کند. مثلاً همه خرت و پرتهای مربوط به کار کردن با فایلها را در system.IO گذاشته. وقتی با SQL Server کار می کنید. System.Data را استفاده می کنید و ...

پس کلاسهای ما به عنوان موجودیتهایی با وظایف مختلف که می توان به آنها امر کرد و آنها را نصیحت نمود، در namespace های مختلف قرار گرفته اند.

خود microsoft برای شناسایی اینها به ما intellisense می دهد. اگر آیکن به شکل آکولاد باز و بسته  را دیدید، پس namespace است. اگر آیکن به شکل سه مکعب دیدید که با خط به هم وصلند , پس کلاس است. هر وقت که بخواهید از کلاس استفاده کنید، باید از روی آن شی درست کنید، نمی توانید بگویید که انسان باید راه رود، باید از انسان یک شی مثلاً آقای x را درست کنید و بعد دستور دهید که راه برود.

Class , Field , Property , Method چیست؟

هر وقت که یک متغیر را از یک کلاس درست می کنید، باید آن را new کنید تا بتوانید با آن کار کنید.

به عنوان مثال Ostream دارد به یک شی اشاره می کند. اگر نقطه بزنی دارای یک سری member است، آنهایی که یک کاغذ و دست  است، پس property است و اگر یک مکعب صورتی  دیدید، method است و اگر علامت رعد و برق  بود، event است. مکعب آبی  یعنی Field است.

بعضی وقتها به صراحت یک چیز را می نویسید. در کد بگویید، که می خواهم به مجموعه کنترل های صفحه من دکمه را اضافه کنید. مثلاً در قسمت کد asp:button را اضافه می کنید و ...

زمانیکه یک دکمه می اندازید، از کلاس دکمه اضافه کردید و آن شی ای از این کلاس است با Property F4 بگیرد. بعد event آن را ببینید. وقتی که event ای را raised می کنید، یعنی آقای دکمه وقتی که کسی تو را کلیک کرد، این تابع را صدا بزن.

در صفحه editPageContent ، هر چی asp: بندازید، همه شی هستند. اگر کسی بپرسد چند شی دارید، می گوئید یک شی از کلاس Label داریم به نام lbl... و ...

در مورد دو دکمه ای که در مثال بالا در صفحه داشتیم دکمه event آن را raise کردیم، تا آن را نصیحت کنیم، تا عملیات save و open را انجام دهیم.

شی Txtfilename جنس آن از **System.Web.UI.WebControls.TextBox** ، در این عبارت، textbox نام کلاس و همه قسمتهای قبل از آن name space است. با زدن نقطه بعد از TextBox ، Property و method های آن را می بینید.

Property ها را باید با گذاشتن = مقدارهی کنید ولی در خصوص متدها باید آن را به صورت (f) صدا کنید.

مثلاً می گفتیم، text این textbox را بگیر و بعد trim کن. یعنی می تواند چند مرحله زیر جز داشته باشد.

مثلاً textbox.text خودش از جنس string است و بنابراین می توان متد Trim را روی آن صدا نمود.

Member استاتیک و غیر استاتیک چیست؟

گاهی اوقات از بعضی از کلاسها شی درست می‌کنم و گاهی درست نمی‌کنم و مستقیم از آن کلاس استفاده می‌کنیم. Member ها یا استاتیک هستند یا غیر استاتیک، اگر استاتیک باشند، مالکیت کلاس مربوط به خود کلاس است وگرنه مربوط به اشیای آن کلاس است.

مثلاً در مورد کلاس File یک متد Exist دارد که استاتیک است، پس مالکیت این متد مربوط به خود کلاس است و نه شی، پس مستقیم با کلاس صدا زده می‌شود.

این مسئله الهام شدنی نیست و باید تست کنید. پس برای همین است که وقتی می‌خواهید چک کنید، ببینید، یک فایل وجود دارد یا خیر، شی درست نکردیم و از همان کلاس متد را صدا زدیم ولی برای streamReader شی درست کردم و سپس متد استفاده کردیم. با MSDN و HELP و سعی و خطا باید به نتیجه برسید.

Method Overloading چیست؟

همانطور که بیان شد با new کردن کلاس، شی بوجود می‌آید. زمانیکه ما new می‌کنیم، چه زمان new کردن و چه زمان صدا زدن متد، یک سری شماره نشان می‌دهد، به آن Method overloading می‌گویند.

▲ 1 of 5 ▼ string string.Format(**string format**, object arg0)
Replaces one or more format items in a specified string with the string representation of a specified object.
format: A composite format string.

با arrow key بین متدها می‌توانید حرکت کنید.

مثال: یک کلاس به نام Utility درست کنیم و از کلاس خودمان استفاده کنیم.

در این کلاس یک تابع دارید که میانگین دو عدد را محاسبه می‌کند. بعد یک تابع می‌خواهید که میانگین سه عدد، چهار عدد را حساب کند. پس:

Class utility

```
{
    Float GetAverage (float x, float y)
    {
        Return ((x+y)/2);
    }

    Float GetAverage (float x, float y, float z)
    {
        Return ((x+y+z)/3);
    }

    Public float GetAverage (float x, float y, float z, float t)
    {
        Return ((x+y+z+t)/4);
    }
}
```

در زبانهای functional باید، نام تابع را عوض کنید. ولی در OO اجازه تعریف چند تابع با نام یکسان و با signature های مختلف را فراهم می‌سازد و به این کار که به دنیای واقعی نزدیک است، می‌گویند، Method Overloading. در دنیای واقعی نیز اینگونه عمل می‌کنیم. یعنی اگر بخواهیم به یک نفر بگویم که یک لیوان آب بیار و یا بگویم یک لیوان آب و یک لیوان نوشابه بیار، در هر دو صورت دستور بیار است و فقط آیتمها یا همان signature را انتخاب می‌کند.

حالا می بینیم که GetAverage بسیار پر کاربرد است، پس آنها را static می کنیم تا دیگر نیاز به new کردن نداشته باشد. این متد به تنهایی پر کاربرد است و دیگر نیاز به new و تشریفات و ... ندارد، پس این متد را static کنید.

Public static float GetAverage ()

هیچ وقت مکعب آبی یا field را بعد از کلاس نمی بینید، مکعب آبی یعنی فیلد و دلیل آن این است که ماکروسافت ، فیلدها را به صورت Private در نظر گرفته و بعد یک Property برای آن نوشته است.

Error Handling

انواع خطاها و Error هایی که در برنامه ممکن است با آن مواجه شوید، عبارتند از:

- 1 - Syntac Error – اگر Syntac Error داشته باشید، برنامه اجرا نمی شود.
 - 2 - گاهی در منطق برنامه اشکال دارید، Logical Error، برنامه اجرا می شود ولی جواب درستی نمی دهد. مثلاً Stream را تا close نکنید، فایل نوشته نمی شود، فایل شما مثل یک ابر در هوا است و با Close کردن ایجاد می شود. اگر فراموش کنید که stream را close کنید برنامه شما از نظر منطقی درست عمل نمی کند. یعنی عبارت مورد نظر شما را در فایل نمی نویسد.
 - 3 - یک سری error داریم که به صورت Run time Error خود را نشان می دهد. که باید با iff جلوی آن را بگیرید.
 - 4 - گاهی با خطاهایی سر و کار داریم، که اتفاق می افتد ولی if ای برای آن وجود ندارد. Drive خراب باشد، floppy disk خراب شده و خطا داشته باشد. احتمال می دهیم که دستور شما با خطا raise شود، می خواهید، خطا را مدیریت کنید، نمی توانید پیش گیری کنید، باید در اینجا try throw catch بگذارید.
- پس: هر وقت که کدی می نویسید و احتمال می دهید که خطایی بوجود بیاید، که نمی توانید آن را با if کنترل کنید، از try catch استفاده می کنید. اگر از طریق try catch آن را مدیریت نکنیم، یک صفحه خطرناک باز خواهد شد.

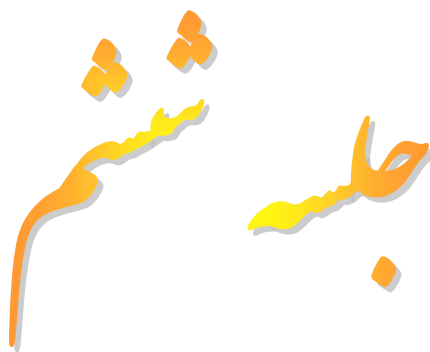
به عنوان مثال، ما می خواهیم در root این سایت یک فایل با نام Alaki.txt بسازد. این کار ممکن است به دلایل مختلف با خطا مواجه شود. پس از try catch finally استفاده کنید. قسمتی از کد که ممکن است باعث بروز خطا شود، داخل try نوشته می شود. اگر به هر خطایی بخورد، داخل catch می رود و کار گفته شده در این قسمت را انجام می دهد. اگر در try به خطایی نخورد، به finally می رود و ادامه راه را می دهد. اگر خطا هم دهد باز هم به finally می رود.

در finally کار تر و تمیز کردن حافظه، مثل close کردن و Dispose کردن انجام می گردد.

شی هایی که درست کردید باید Dispose شود، بعضی از موجودات مثل connection و stream writer , stream reader نیاز به Close دارد. حتماً باید close شود، ولی Dispose اجباری نیست و اگر dispose کنید، سیستم سر حال تر می شود ، ولی اگر یادتان رفت مشکلی پیش نمی آید.

پس: Close مهم است ولی dispose کردن و null کردن مهم نیست و فقط برای تمیز کردن است و مشکلی نیست. در وب در زمان اجرا با رسیدن به خطای run time سر سطر خطا می ایستد و باید با زدن F5 بقیه برنامه را run کنید، تا خطا دیده شود.

صفات می تواند simplex یا Complex باشد. مثلاً رنگ یک ویژگی است، یک خصلت است، ولی Simplex است ولی گاهی می گویی، چراغ خودرو و بعد لامپش، به این صفت Complex می گویند.



دسته‌بندی Property های کنترلها

Property های هر کنترل به 5 دسته تقسیم می‌شود.

- 1 - دسته اول همه کنترلها دارند ولی نباید به عنوان آدم حرفه‌ای آنها را تغییر دهیم.
- 2 - دسته دوم بین دو دسته 1 و 3 هستند.
- 3 - دسته سوم همه دارند اگر لازم شد تغییر می‌دهیم.
- 4 - دسته چهارم بین دو دسته 3 و 5 است.
- 5 - دسته پنجم خاص یک کنترل است و به صورت تخصصی مربوط به همان کنترل است. تکلیف 4 دسته از صفات این کنترل را مشخص می‌کنیم.

Property های نوع اول

همه کنترلها این Property ها را دارند و این Property ها بین تمام کنترلها مشترک است و ما نباید تحت هیچ شرایطی به عنوان یک آدم حرفه‌ای، مقدار آنها را عوض کنیم. (البته نه اینکه هیچ وقت عوض نشود، بلکه باید در CSS تنظیم شود)

Back Color	- 1
Border Color	- 2
Border Style	- 3
Border Width	- 4
Font	- 5
Fore Color	- 6

Property های نوع سوم

همه کنترلها آن را دارند و اگر بخواهیم می‌توانیم مقدار آن را تغییر دهیم:

ID	- 1
Access Key	- 2
Css Class	- 3
Enabled (با false کردن این property کنترل دیده می‌شود ولی کار نمی‌کند)	- 4
EnabledTheming	- 5
SkinID (این Property به property شماره 5 مرتبط است)	- 6
Enabled View State (هنوز دلیلی پیدا نشده است که این Property به صورت False تنظیم شود. چون تنها مزیت کنترلهای Server Side داشتن View State است که اگر با false کردن این property آن را از بین ببریم، دیگر چه لزومی به استفاده از کنترل server side وجود داشت. در ضمن	- 7

- این صفت کار نمی‌کند مگر اینکه View State صفحه را هم false بگذارید که این هم اصلاً به درد نمی‌خورد.
- 8 - Tab index (به صورت پیش فرض Tabindex همه کنترلها 0 است، بنابراین ترتیب اجرای آنها مطابق HTML آن است و اگر بخواهید شماره می‌دهید. اگر tab index یک کنترل را 1- بگذارید، دیگر با زدن Tab داخل آن نمی‌رود. در زبانهای قبلی یک صفت به نام Tab Stop داشتیم که همین کار را می‌کرد و معادل 1- کردن Tab index است.
- 9 - Tooltip
- 10 - Visible (اگر false کنید در asp اصلاً render نمی‌شود. در Windows وقتی این صفت را false می‌کنید کنترل وجود دارد ولی در web اصلاً کد HTML آن Generate نمی‌شود.

Property های نوع دوم

دسته دوم بین دو دسته اول و سوم است، باید با درایت تشخیص دهید که باید تغییر یابد یا خیر.

- 1 - width
- 2 - Hight (گاهی با CSS به همه دکمه ها طول و عرض می‌دهید ولی گاهی در یک صفحه یک عکس دارید که می‌خواهید به آن طول و عرض دهید، دیگر لازم نیست که در CSS این کار را انجام دهید و در خود صفحه Property کنترل را تغییر می‌دهیم. پس تشخیص آن با توجه به درایت برنامه‌نویس می‌باشد.

Property های نوع چهارم

این Property حدوداً بین 98٪ کنترلها مشترک است و این 98٪ کنترل این Property را دارند.

- 1 - Text (اکثراً دارند بنابراین نه داخل 5 می‌رود و نه 3 چون 98٪ کنترلها دارند)

Property های نوع پنجم

هر چیزی که باقی مانده است، دسته پنجم است، پس ما هر کنترلی که درس می‌دهیم فقط دسته پنجم را می‌گوییم.

کنترل Validator

در مبحث قبلی CMS بسیاری از Validation ها را در سمت سرور انجام دادیم. Validation یعنی که یک سری کنترلها را قبل از انجام عمل مورد نظر انجام دهیم. به عنوان نمونه قبل از اینکه یک فایل را کپی کنیم، کنترل اینکه آن فایل خالی نباشد و موجود باشد یک validation است.

می‌توان به صورت دستی عمل Validation را انجام داد با کد نویسی در سمت Sever بگوییم اگر این شد، این کار را بکن و ... ولی می‌توان بسیاری از Validation ها را به صورت javascript در سمت Client Side انجام دهیم، با این کار بار سایت را کم کردیم.

JavaScript بسیار زبان سخت و غیر استاندارد است، بنابراین ما به سراغ JQuery می‌رویم البته خود Visual Studio یک زحمتی کشیده و یک سری Validator تولید نموده است.

کنترل Required Field Validator

یک Lable در صفحه می‌گذاریم به نام lblMessage با runat سمت Server
یک Lable و یک Textbox هم با عنوان SomeField می‌گذارم.

یک دکمه Submit هم می‌گذاریم.

به طور سنتی بر روی دکمه دستور می‌نوشتید که خالی بودن را چک کن و ... را چک کن و ... در این روش فقط پشت دکمه کارهای قطعی را انجام می‌دهید.

```
lblMessage.text = "Your information update successfully"
```

اولین مورد اینکه می‌خواهیم textbox خالی نباشد. که ماکروسافت کنترل Required Field Validator را برای این کار پیشنهاد داده است.

Some Field

(S)ubmit

```
<form id="frmMain" runat="server">
  <asp:label id="lblMessage" runat="server" />
  <br />
  <br />
  <asp:label id="lblSomeField" runat="server" text="Some Field" />
  <asp:textbox id="txtSomeField" runat="server" />
  <asp:requiredfieldvalidator runat="server" id="rfvSomeField"
    controltovalidate="txtSomeField" display="Dynamic"
    enableclientscript="False" setfocusonerror="True" >
    Some field is required!</asp:requiredfieldvalidator>
  <br />
  <br />
  <asp:button id="btnSubmit" runat="server" text="(S)ubmit"
    accesskey="S" onclick="btnSubmit_Click" />
</form>
```

حال صفتهای این کنترل را معرفی می‌کنیم.

- 1 - **Controltovalidate** (این صفت باید حتماً دارای یک مقدار باشد و حتماً هم نام یک کنترل در صفحه باشد. در غیر این صورت error می‌دهد. خودش یک DropDown list دارد که می‌توانید نام کنترل را انتخاب کنید. هر Validator ای باید حتماً یک و فقط یک کنترل را Validate کند. هر Validator ای ممکن است قوانین متفاوتی از custom Validator داشته باشد و قوانین اینچنینی که در مورد هر validator بیان می‌کنیم، ممکن است در مورد custom Validator صادق نباشد. به عنوان مثال اگر 50 تا فیلد داریم و بخواهیم هر 50 تا را کنترل کنیم که پر شده باشد، باید 50 تا Validator control بگذاریم. البته برعکس این قانون صادق نیست و هر کنترلی می‌تواند چندین Validator control داشته باشد که هر کدام یک کاری بر روی آن کنترل انجام دهد.
- 2 - ID (هر Validator مربوط به یک کنترل است و هر کنترل هم که یک ID داشت، Required Field Validator به صورت مخفف rfv است، پس ID این کنترل با rfv شروع می‌شود و سپس نام کنترل بدون گذاشتن abbreviation خود کنترل (مثلاً بدون txt و btn) قرار می‌گیرد).
- 3 - دو تا صفت دیگر هم validator ها دارند که بعداً می‌گوییم. یکی **error message** و دیگری **Validation group** می‌باشد.

کار این Validation کنترل کردن خالی نبودن textbox بود، اینکه در صورت خالی بودن چه خطایی بدهد، باید **خطا را در Text بنویسید**.

پس برای کنترل خالی بودن یا نه، کد server side و c# ای نوشته نشد و فقط از java script استفاده شد و اصلاً هیچ کدی نوشته نشده است. اگر کنترل خالی باشد اصلاً صفحه Post back هم نمی‌شود. البته این کنترل یک باگ دارد. این کنترل به طور طبیعی عمل نمی‌کند. یعنی همان اول هر چی Tab بزنید و بالا و پایین کنید، خطا نمی‌دهد. وقتی عمل می‌کند که یک بار بر روی دکمه submit کلیک کنید و یا اینکه یک بار داخل آن یک چیزی بنویسید و پاک کنید و خارج شوید، تازه با این کار این کنترل راه می‌افتد.

الان focus روی textbox است، وقتی دکمه submit را می‌زنم، focus بر روی دکمه منتقل می‌شود. اگر صفت **set focus on error property** را true کنید، focus بر روی textbox باقی می‌ماند.

از اول پیدایش .net یک سری تغییرات ایجاد شد.

- 1 - یک سری کنترلها کنار گذاشته شد و کنترلهای جدید آمد. قبلاً DataGrid داشتیم و الان DataGridView داریم.
- 2 - یک سری کنترلها نام آنها عوض شد مثل Menu Strip.
- 3 - یک سری کنترلها یک سری Property به آن اضافه شد. مثل set focus on error Property این از 2003 به بعد اضافه شد.

صفت **InitialValue**: یک صفت است برای ست کردن مقدار اولیه. البته این را استفاده نکنید، چون به صورت ثابت می‌نویسد، اگر خواستید در صورت خالی بودن چیزی بنویسد، حتماً آن را با javascript ست کنید.

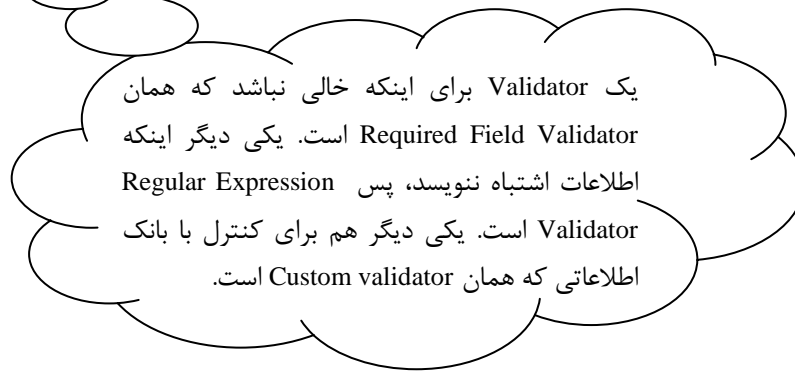
صفت **Display** است که اگر none بگذارید، متن خطا را نشان نمی‌دهد، یعنی **Validation** کار می‌کند ولی خطا را نمایش نمی‌دهد. غیر none آن یا static است یا Dynamic. یک کنترل می‌تواند چندین Validator داشته باشد. اگر به صورت استاتیک باشد، خطاها پشت سر هم گذاشته می‌شود و بین آنها بر اساس طول خطا فاصله می‌گذارد و قشنگ نیست. ولی اگر dynamic بگذارید، روی هم می‌اندازد و این درست است. سوال: اگر چند تا خطا بر بخورد، چه کار باید کرد. این هم اشتباه است. Validator ها باید به صورت جامع، کامل و مانع استفاده شود.

property بعدی، **EnableClientScript** است. یعنی می‌خواهم سمت Client side و script ای عمل کنم. اگر false باشد، یعنی فقط می‌خواهم Post back شود و کنترل در سمت سرور و با کد C# ای انجام گیرد. اگر true باشد، هم در Client و هم در Server صورت می‌گیرد.

یعنی اگر true باشد و صد بار خطا داشته باشد، صد بار هم سمت client و هم سمت server اجرا می‌شود؟ نه اینطور نیست، تا خطا دارد فقط سمت Client اجرا می‌شود و بعد که درست شد، برای check نهایی سمت سرور می‌رود.

False کردن این property هیچ فایده‌ای ندارد. اینکه این صفت true باشد و javascript را disable کنید، معادل این است که این صفت را false کنید و javascript را بگذارید enable باشد. پس برای بررسی نکات امنیتی است.

حالا این صفت را false کنید تا فقط سمت سرور، اجرا شود. پس با زدن submit صفحه Post Back می‌شود.

username: 

با وجود مشکل در صفحه، عمل دارد انجام میشود. هدف این بوده که مقدار خالی در بانک اطلاعاتی ذخیره نشود، ولی حالا شده است.

سه نکته مهم در .NET.

تا به حال 3 چیز مهم یاد گرفته اید

1. View State
 2. Post Back
 3. دکمه submit همیشه اتفاق می افتد. تابع Submit اگر نوشته شود، همیشه اتفاق می افتد.
- If page.isvalid این در واقع دستوری است که همیشه در تابع submit بنویسید.

Server side validation بعد از Page load اتفاق می افتد.

پس فرآیندی که اتفاق می افتد.

1. Client Validation
2. Post Back
3. Page load
4. Server side Validation
5. function (آن چیزی که باعث Post Back شده است)

PageValid یعنی سمت سرور هیچ خطایی را detect نکند.

تمام چیزهایی که باعث Post back می شود، با چشم بسته بنویسید. هیچ فرقی نمی کند. پس همیشه در توابعی که بعد ازPostBack صورت می گیرد، این if را بنویسید.
لزوماً دکمه و event click باعث Post Back نمی شود، چیزهای دیگری هم هستند که باعث Post Back می شوند.

چیزهایی که باعث Post Back می شوند

1. وقتی که بر روی Button رویداد click صورت گیرد.
2. وقتی که بر روی LinkButton رویداد Click صورت گیرد. (در مورد LinkButton عملیات open in New Window معنی ندارد)
3. وقتی که بر روی Image Button رویداد click صورت گیرد.
4. وقتی که بر روی DropDownList ، ListBox ، CheckBox List و Radio Button List ، رویداد Selectedindex Change اتفاق بیافتد.

یک ListBox دارید وقتی item آن را تغییر می‌دهد که نباید Post Back شود چون اتفاقی نیافتاده است. ولی اگر IsPostBack آن را True کنیم، با تغییر item ، Post Back می‌شود. Selectedindex Change به شرط true بودن AutoPostBack عمل می‌کند.

TextBox اگر autoPostBack آن true باشد با TextChanged عمل PostBack بر روی آن صورت می‌گیرد.

در خصوص check Box نیز با ValueChange آن به شرط درست بودن AutoPostBack ، PostBack انجام می‌شود.

صفحه Login و Registration با کنترل Field Required Validator

برنامه نمونه: صفحه login را با کنترل‌های مربوطه بنویسید.

```
public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        divMessage.Visible = false;

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
        }
    }

    private void DisplayInformationMessage(string message)
    {
        divMessage.Visible = true;

        lblMessage.Text = message;
        lblMessage.CssClass = "information";
    }
}
```


}

```

<div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
</div>

<fieldset>
    <legend>
        Login
    </legend>

    <div class="field">
        <div class="caption">
            <asp:label id="lblUsername" runat="server" cssclass="label"
                text="Username" />
        </div>
        <div class="controls">
            <asp:textbox id="txtUsername" runat="server"
                cssclass="textBox" maxlength="20" accesskey="1"
                autocompletetype="Disabled" />
            <asp:requiredfieldvalidator id="rfvUserName" runat="server"
                cssclass="validator" controltovalidate="txtUserName"
                display="Dynamic" setfocusonerror="true"
                text="Username is required!" />
        </div>
    </div>

    <div class="field">
        <div class="caption">
            <asp:label id="lblPassword" runat="server" cssclass="label"
                text="Password" />
        </div>
        <div class="controls">
            <asp:textbox id="txtPassword" runat="server"
                cssclass="textBox" maxlength="20"
                accesskey="2" textmode="Password"
                autocompletetype="Disabled" />
            <asp:requiredfieldvalidator id="rfvPassword" runat="server"
                cssclass="validator" controltovalidate="txtPassword"
                display="Dynamic" setfocusonerror="true"
                text="Password is required!" />
        </div>
    </div>

    <div class="buttons">
        <asp:button id="btnSubmit" runat="server" cssclass="button"
            text="(L)ogin" accesskey="L" onclick="btnSubmit_Click" />
        <input type="reset" value="(R)eset" accesskey="R" class="button"
            />
    </div>
</fieldset>

```

برنامه نمونه: صفحه registration را با کنترل‌های مربوطه بنویسید.

Registration	Registration
Username _____	Username _____ Username is required!
Password _____	Password _____ Password is required!
Confirm Password _____	Confirm Password _____ Confirm password is required!
Email _____	Email _____ Email is required!
Confirm Email _____	Confirm Email _____ Confirm email is required!
Regi(s)ter (R)eset	Regi(s)ter (R)eset

```

<div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
</div>

<fieldset>
    <legend>
        Registration
    </legend>

    <div class="field">
        <div class="caption">
            <asp:label id="lblUsername" runat="server" cssclass="label"
                text="Username" />
        </div>
        <div class="controls">
            <asp:textbox id="txtUsername" runat="server"
                cssclass="textBox" maxlength="20" accesskey="1"
                autocomplete="Disabled" />
            <asp:requiredfieldvalidator id="rfvUserName" runat="server"
                cssclass="validator" controltovalidate="txtUserName"
                display="Dynamic" setfocusonerror="true"
                text="Username is required!" />
        </div>
    </div>

    <div class="field">
        <div class="caption">
            <asp:label id="lblPassword" runat="server" cssclass="label"
                text="Password" />
        </div>
        <div class="controls">
            <asp:textbox id="txtPassword" runat="server"
                cssclass="textBox" maxlength="20" accesskey="2"
                textmode="Password" autocomplete="Disabled" />
            <asp:requiredfieldvalidator id="rfvPassword" runat="server"
                cssclass="validator" controltovalidate="txtPassword"
                display="Dynamic" setfocusonerror="true"
                text="Password is required!" />
        </div>
    </div>

    <div class="field">
        <div class="caption">
            <asp:label id="lblConfirmPassword" runat="server"
                cssclass="label" text="Confirm Password" />

```

```

</div>
<div class="controls">
    <asp:textbox id="txtConfirmPassword" runat="server"
        cssclass="textBox" maxlength="20" accesskey="3"
        textmode="Password" autocompletetype="Disabled" />
    <asp:requiredfieldvalidator id="rfvConfirmPassword"
        runat="server" cssclass="validator"
        controltovalidate="txtConfirmPassword"
        display="Dynamic" setfocusonerror="true"
        text="Confirm password is required!" />
</div>
</div>
<div class="field">
    <div class="caption">
        <asp:label id="lblEmail" runat="server" cssclass="label"
            text="Email" />
    </div>
    <div class="controls">
        <asp:textbox id="txtEmail" runat="server"
            cssclass="textBox" maxlength="100" accesskey="4"
            autocompletetype="Disabled" />
        <asp:requiredfieldvalidator id="rfvEmail" runat="server"
            cssclass="validator" controltovalidate="txtEmail"
            display="Dynamic" setfocusonerror="true" text="Email
            is required!" />
    </div>
</div>
<div class="field">
    <div class="caption">
        <asp:label id="lblConfirmEmail" runat="server"
            cssclass="label" text="Confirm Email" />
    </div>
    <div class="controls">
        <asp:textbox id="txtConfirmEmail" runat="server"
            cssclass="textBox" maxlength="100" accesskey="5"
            autocompletetype="Disabled" />
        <asp:requiredfieldvalidator id="rfvConfirmEmail"
            runat="server" cssclass="validator"
            controltovalidate="txtConfirmEmail"
            display="Dynamic" setfocusonerror="true"
            text="Confirm email is required!" />
    </div>
</div>
<div class="buttons">
    <asp:button id="btnSubmit" runat="server" cssclass="button"
        text="Regi(s)ter" accesskey="S" onclick="btnSubmit_Click"
        />
    <input type="reset" value="(R)eset" accesskey="R" class="button"
        />
</div>
</fieldset>

```

```

public partial class Registration : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        divMessage.Visible = false;
    }
}

```

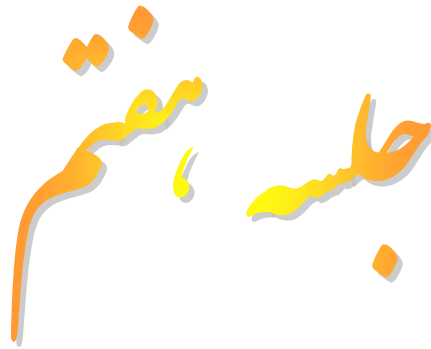
```
        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
        }
    }

    private void DisplayInformationMessage(string message)
    {
        divMessage.Visible = true;

        lblMessage.Text = message;
        lblMessage.CssClass = "information";
    }
}
```



کنترل Regular Expression Validator

وظیفه این کنترل

این validator تست می‌کند که آیا متنی یا عبارتی که وارد نموده‌اید از فرمت درستی طبیعت می‌کند یا خیر. این validator فقط از نظر شکلی تست می‌کند و نه از نظر مفهومی.

چند مورد مثالی مختلف را تست می‌کنیم:

فرض کنید که می‌پرسید شماره موبایل شما چنده؟ می‌گوید 123 آیا قبول می‌کنید؟ خیر. ولی 09122788745 را قبول می‌کنید؟ بله

ولی اصلاً کاری نداریم که آیا این شماره موبایل واقعاً وجود دارد یا خیر و متعلق به آن فرد است یا خیر.

در خصوص آدرس سایت و ایمیل و غیره هم همینطور است. یعنی مثلاً `x@y.z` یک آدرس پست الکترونیک معتبر است، ولی ممکن است این آدرس ایمیل متعلق به فرد مورد نظر نباشد و یا اصلاً وجود نداشته باشد، اصلاً مهم نیست از نظر این validator این یک آدرس ایمیل معتبر است.

Property های این کنترل

- تمام کنترل‌های validator یک صفت به نام `Control ToValidate` دارند و مشخص می‌کند که کدام کنترل را validate می‌کنند.
- `Text` شامل پیغام خطا: مثلاً `You did not specify Valid Format`
- `Validation Group` و `ErrorMessage`: مهم هستند و بعداً می‌گوییم.
- برای `ID` (اسم کنترل) یک مخفف `rev` را به نام کنترلی که می‌خواهیم Validate کنیم بدون `txt, btn` و ... اضافه می‌کنیم.

دو کار در برنامه‌نویسی تحت وب معادل ریاضت کشیدن است،

1 - استفاده از javascript

2 - نوشتن یا تولید Regular Expression

در خصوص نوشتن Regular Expression خیلی کم مواردی پیش می‌آید که بخواهید بنویسید، بخاطر اینکه خود محیط Visual Studio بسیاری از موارد را به صورت template برای ما گذاشته است و در اینترنت هم پیدا می‌شود و بعضی از سایتها هم هستند که regular Expression را ارائه می‌کنند. یا در موارد ساده می‌توانید با ایجاد تغییر در template های موجود template مربوط به خود را بسازید.

در یک پروژه بزرگ تعداد Regular Expression بیش از 15 تا نیست.

راهکارهای پیشنهادی جهت نوشتن Regular Expression

- 1- استفاده از خود template های از پیش تعریف شده microsoft
- 2- جستجو در اینترنت
- 3- استفاده از سایت ها یا برنامه هایی در اینترنت که قدم به قدم می پرسد و در نهایت expression را تحویل می دهند.
- 4- استفاده از expression های موجود و ایده گرفتن از آنها

نکته: دانستن یا داشتن یک سری از expression برای سرچ کردن در یک متن 5000 صفحه ای مثلاً یافتن هر چی ایمیل است مفید است.

نکته: خیلی از این expression ها ساده است. مثلاً `\d{5}` یعنی یک عدد 5 رقمی. فرض کنید که یک فایل word دارید که 5 صفحه است. اگر در فایل word بگوییم هر چه email است را پیدا کنید، چگونه اینکار را انجام می دهید. مفهوم expression یک مفهوم استاندارد است. می پرسد می خواهید فلان چیز را پیدا کنید و یا expression را پیدا کنید، اگر در قسمت Expression این خط را بنویسید، می رود و آن را پیدا می کند. اگر بخواهم در متن آدرس هر چی سایت است را پیدا کنید یا هر چی عدد 5 رقمی است را پیدا کنید، می توانید از همین عبارات برای search در محیط های استاندارد Microsoft استفاده کنید.

`/d{3,5}` عدد 3 یا 4 یا 5 رقمی را می پذیرد.

به عنوان مثال validator ای برای چک کردن نام و email را تعریف می کنیم.

:UserName

```
<asp:regularexpressionvalidator id="revUserName" runat="server" cssclass="validator"
controltovalidate="txtUserName" display="Dynamic" setfocusonerror="true"
text="Username is not valid!" validationexpression="([a-zA-Z0-9]{6,14})" />
```

:email

```
<asp:regularexpressionvalidator id="revEmail" runat="server" cssclass="validator"
controltovalidate="txtEmail" display="Dynamic" setfocusonerror="true" text="Email is
not valid!" validationexpression="\w+([-+.' ]\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*" />
<asp:customvalidator id="csvEmail" runat="server"
cssclass="validator" controltovalidate="txtEmail" display="Dynamic"
setfocusonerror="true" text="Email is already exist! Please choose another one..."
validateemptytext="false" onservervalidate="csvEmail_ServerValidate" />
```

وقتی که textbox را خالی بگذارید و روی دکمه کلیک submit را بزنید، error می گیرد یا خیر؟ خطا نمی دهد، این کنترل می گوید وقتی که طرف چیزی را نوشت خطا بدهد یا خیر. تا چیزی نوشته که نباید error بگیرد. بجز Required Field Validator بقیه کنترلها تا چیزی درون آن نوشته نشده باشد، خطا نمی دهد. تمامی Validator ها به جز Required Field Validator تا وقتی که چیزی در آن کنترل که آن را validate می کنیم، نوشته باشیم، خطا نمی دهد.

برای پیدا کردن Regular Expression مثلا در مورد username در Google عبارت username regular expression را جستجو کند، 7، 8 مورد اول، جواب را برای شما مشخص می کند.

در خصوص username , password به صورت معمول باید بین 8 تا 14 کارکتر باشد. سایت Citi Bank یا doich Bank سایت های کاملاً امن هستند. در این سایتها تا لحظه ای که وارد سایت شوید تا بخواهید مقدار حساب خود را ببینید، نیم ساعت طول می کشد، چون چندین بار login و سوال امنیتی و ... را می پرسد.

همیشه بین امن کردن سایت و کاربرد پسند بودن سایت یک trade off وجود دارد. هر چه سایت بیشتر امن باشد، کار با سایت را سخت تر می کند. باید یاد بگیریم که در هر سایت تا چه حد باید امنیت را رعایت کنیم. مثلاً اگر یک سایت جوک به نام جوکستان راه بیندازید، نبایده تا سوال امنیتی بپرسید. بسته به جدی بودن پروژه و سایز پروژه که در سطح شرکت، وزارت خانه و یا کشور است، میزان امنیت را تعیین کنید.

مقدار مجاز username , password بین 8 تا 14 تا است و بیشتر از آن بودن نیز امنیت زیاد را نشان نمی دهد. باید حرف کوچک و بزرگ و عدد و wild card ها مثل under score و ... باید استفاده شده باشد.

کنترل Compare Validator

وظیفه این کنترل

برای مقایسه محتویات دو Text Box است.

Property های این کنترل

- ID: برای نام گذاری مخفف آن cmv است.
- controltovalidate: نام کنترلی را که می خواهیم با controltovalidate مقایسه کنیم.
- ValueToControl: یک صفت اضافه به نام ValueToControl دارد، که می توان بجای کنترل با Value ثابت مقایسه کرد.
- type: یک صفت دیگر به نام type وجود دارد. در مقایسه string ای 11 کوچکتر از 2 است ولی در مقایسه عددی برعکس است. پس باید type تعیین شود.
- operator: صفت operator دارد و می پرسد که می خواهد عمل مقایسه چه باشد، Greater , equal و ...

هر کدام validator مستقل هستند و نیاز به حضور دیگری ندارد.

در خصوص این کنترل نیز هم اگر در کنترلی که به عنوان controlToValidate مشخص نموده اید، چیزی ننویسید دیگر خطا نمی دهد یعنی در خصوص password , ConfirmPassword اگر Password را پر کنید و confirm آن را خالی بگذارید، خطا نمی دهد. چون قرار است confirm را اعتبار سنجی کند و اگر این کنترل متفاوت از pass بود خطا دهد ولی وقتی چیزی ننوشتید که دیگر خطا نمی دهد.

```
<asp:comparevalidator id="cmvConfirmPassword" runat="server" cssclass="validator"
controltovalidate="txtConfirmPassword" display="Dynamic" setfocusonerror="true"
text="Confirm password is not the same as password!" controltocompare="txtPassword"
operator="Equal" type="String" cultureinvariantvalues="true" />
```

کنترل Range Validator

وظیفه این کنترل

فرض کنید که می‌خواهید سن فرد را برای استخدام بگیرید. و محدودیت سنی داشته باشید

Property های این کنترل

- ID: مخفف این کنترل است.
- Text: پیام خطای آن است. مثلاً Range Check Error.
- چند صفت خاص دارد. Min آن مثلاً 25 و Max آن 35 باشد.
- در مورد اعداد می‌تواند integer , double باشد. مثلاً فرد می‌تواند 25.5 بزند.
- type یا چگونگی مقایسه
- CultureInvariant در windows Application وقتی که کنترل را right to left می‌کنیم. و 25 می‌نویسیم، 25 رافارسی نشان می‌دهد. این صفت را وقتی که true می‌کنید، یعنی اینکه اگر کد حروف متفاوت است، اول آن را به انگلیسی تبدیل کند و بعد آن را در range مناسب مقایسه کند.
- Compare validator هم بهتر است این Property را ست نمایید به true.

Validator ها را جامع و مانع و کامل استفاده کنید. برای Confirm ها باید Regular Expression ها را حذف کنید. چون در صورت نامعتبر بودن مثلاً Password ، دو بار error می‌دهد.

برای Confirm ها Regular Expression را ننویسید. اصولاً سعی کنید، به گونه‌ای از validator ها استفاده کنید، که هیچوقت دو خطا با هم ظاهر نشود.

کنترل Custom Validator

وظیفه این کنترل

فرض کنید می‌خواهید Registration بنویسید و با database مقایسه کنید که درست است یا خیر. کلیه validator هایی که Microsoft نوشته را الان درس دادیم. هر کدام را نتوان با این Validator ها انجام داد باید با custom Validator بنویسیم.

99٪ اوقات Custom validator های ما چون با database کار می‌کنیم، Server Side است.

Property های این کنترل

- ID را با مخفف csv نام‌گذاری کنید.
- Set focus on error را به true مقدار دهیم و ...
- ترجیحاً صفت ValidateEmptyText را false نگاه دارید تا سیاستهای ماکروسافت را ادامه دهید.
- قسمت EnableClientScript را هم False بگذارید. چون اولاً javascript بلد نیستیم و ثانیاً بیشتر کار دیتابیس است و باید سمت سرور باشد. ولی اگر بخواهید true بگذارید باید یک تابع javascript بنویسید و نام آن را در اینجا بنویسید.

- در خصوص validator custom می‌توانید **control To Validate** را بدهید یا ندهید. عرف این است که اگر به یک کنترل مربوط می‌شود، نام ببرید و اگر به بیش از یک کنترل مربوط می‌شود، نام نبرید.

پس اگر خواستید username , password را با هم چک کنید ، چون دو تا است، نباید Control to validate را مقدار دهید ولی اگر فقط username را کنترل می‌کنید، بنویسید. این یک تاکتیک است ولی اجبار نیست.

❖ در خصوص custom ها شما باید event سمت سرور آن را raised کنید. onservervalidate نام این event است.

```
<asp:customvalidator id="csvEmail" runat="server" cssclass="validator"
controltovalidate="txtEmail" display="Dynamic" setfocusonerror="true"
text="Email is already exist! Please choose another one..."
validateemptytext="false" onservervalidate="csvEmail_ServerValidate" />
```

در وب یک سری احمق داریم. شما در دوره سوم Xml Web Service خلق Event را یاد می‌گیرید. در دوره visual studio فقط استفاده از آن را یاد می‌گیرید.

آنجایی که خلق event را یاد می‌گیرید، می‌بینید که لازم نیست که برای هر Event دو پارامتر بگیرید، می‌توانید یک پارامتره، شش پارامتره و 7 پارامتره البته همانجا یاد می‌گیرید که اگر دو پارامتره باشد، خیلی استاندارد است.

اگر یک نفر از شما پرسید که signature این تابع چیست؟ غلط است که نام متغیرها را بکار ببرید، آنها فقط دکور هستند. مثلاً اگر کسی از شما پرسید که signature پشت submit چیه؟ باید بگویید object , system , sender نه e .
غلط:

```
protected void csvPassword_ServerValidate(object source,
System.Web.UI.WebControls.ServerValidateEventArgs args)
```

درست:

```
protected void csvPassword_ServerValidate (object sender,
System.Web.UI.WebControls.ServerValidateEventArgs e)
```

پس از عرفیات خارج نشوید، اینها عرف است. به هر دوره وب و یا ویندوز برگردید. اسم این دو متغیر sender و e است.

استاندارد نویسی و بحث تاکتیکی گاهی از تکنیک بهتر است.

آیا این e به درد بخور است یا خیر؟ در اینجا به هیچ درد نمی‌خورد. خداییش دکور است و به هیچ درد نمی‌خورد. اگر از نوع EventArgs باشد به هیچ درد نمی‌خورد و اگر پیشوند و پسوند داشت به یک دردهایی می‌خورد و e. بزنید یک چیزهایی در ارتباط با آن می‌آورد.

اگر control to validate نام کنترل مورد نظر گذاشته باشید. e.Value معادل نام(محتوی) کنترل می‌شود.

```
String strUsername = e.value;
```

ولی اگر نگذاشته باشید هم مهم نیست. به این صورت بنویسید

```
String strUsername = txtUsername.text;
```

پس مقدار را از خود Textbox خنودم و دیگر نیازی به E ندارم. پس برای همین بود که مهم نبود.

مهم: به هیچ عنوان e.isvalid را نباید true کنید، فقط صرفاً اگر مشکلی وجود داشت آن را False کنید. صفحه‌ای را در نظر بگیرید که 5 تا validator دارد. از 5 تا 2 تا آن به مشکل خورده است و آخرین validator یک custom است. و در if آن نوشته بودم که اگر این validator به مشکل نخورد true کن و به این ترتیب کل validator های صفحه و کارهایی که کرده بودند را از بین بردیم و همه چیز کشک شد. پس با true کردن آن نتیجه کلیه validator ها را True می‌کنید.

یک تور بر روی مباحث مطرح شده:

Custom یک صفت داشت که تا خالی است کنترل نکنم. و ما آن را انتخاب نمودیم.

صفت validation group

❖ اگر در صفحه ای دو سری validation group داشته باشید، به هم گیر می‌دهد. در Visual Studio 2005 به بعد این باگ بر طرف شد. در قبل از آن اگر قرار بود صفحه ارسال یا reset پسورد را بگذارند آن را در یک صفحه دیگر می‌گذاشتند.

زمانی که بیش از یک ناحیه در صفحه داشته باشید، validation group به ما کمک می‌کند. برای این کار برای validation ها و دکمه ها validation group تعریف کنید. حواستان جمع باشد که textbox ها هم Validation group دارند ولی آن را به هیچ عنوان set نکنید چون دچار مشکل می‌ترکد.

اول در ناحیه validation group عملیات client side validation صورت می‌گیرد و بعد Post back و بعد Page Load و بعد server side validation در ناحیه validation group صورت می‌گیرد و بعد خود تابعی که باعث post back شده است.

اگر نخواهید پروژه را چند لایه بنویسید، شیک‌تر است که Validation را به این صورت که امروز درس داده شده است، عمل کنید. ولی در مدل چند لایه که business جداست اصلاً معنی Custom Validation حذف می‌شود. اگر لایه business ندارید بهتر است که یک کاسه کنید.

The screenshot shows a web browser window with the title 'Login3.aspx'. The page content includes a login form with the following elements:

- A header bar with 'cphMain (Custom)' and 'div#divMessage.message'.
- A message box containing '[!blMessage]'. Below it, the text 'Login' is displayed.
- A 'Username' input field with a red error message: 'Username is required!Username is not valid!Username and/or password is not correct! Please try again...'.
- A 'Password' input field with a red error message: 'Password is required!Password is not valid!'.
- A 'Login' button.
- A 'Forgot Password' section with an 'Email' input field and a red error message: 'Email is required!Email is not valid!Email does not exist! Please try again...'.
- A 'Send Me New Password' button.

Login3.aspx X

```

<%@ page title="Login" language="C#" masterpagefile="~/Site.master"
autoeventwireup="true" codefile="Login3.aspx.cs" inherits="Login3" %>

<asp:content id="cntHead" contentplaceholderid="cphHead" runat="Server">
</asp:content>

<asp:content id="cntMain" contentplaceholderid="cphMain" runat="Server">

<div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
</div>

<fieldset>
    <legend>
        Login
    </legend>

<div class="field">
    <div class="caption">
        <asp:label id="lblUsername" runat="server" cssclass="label" text="Username" />
    </div>
    <div class="controls">
        <asp:textbox id="txtUsername" runat="server" cssclass="textBox" maxlength="14"
            autocomplete="Disabled" />
        <asp:requiredfieldvalidator id="rfvUserName" runat="server"
            cssclass="validator" controltovalidate="txtUserName" display="Dynamic"
            setfocusonerror="true" text="Username is required!" validationgroup="Login" />
        <asp:regularexpressionvalidator id="revUserName" runat="server"
            cssclass="validator" controltovalidate="txtUserName" display="Dynamic"
            setfocusonerror="true" text="Username is not valid!" validationexpression="([a-
            zA-Z0-9]{6,14})" validationgroup="Login" />
        <asp:customvalidator id="csvUsernamePassword" runat="server"
            cssclass="validator" controltovalidate="" display="Dynamic"
            setfocusonerror="true" text="Username and/or password is not correct! Please
            try again..." validateemptytext="false"
            onservervalidate="csvUsername_ServerValidate" validationgroup="Login" />
    </div>
</div>

<div class="field">
    <div class="caption">
        <asp:label id="lblPassword" runat="server" cssclass="label" text="Password" />
    </div>
    <div class="controls">
        <asp:textbox id="txtPassword" runat="server" cssclass="textBox" maxlength="14"
            textmode="Password" autocomplete="Disabled" />
        <asp:requiredfieldvalidator id="rfvPassword" runat="server"
            cssclass="validator" controltovalidate="txtPassword" display="Dynamic"
            setfocusonerror="true" text="Password is required!" validationgroup="Login" />
        <asp:regularexpressionvalidator id="revPassword" runat="server"
            cssclass="validator" controltovalidate="txtPassword" display="Dynamic"
            setfocusonerror="true" text="Password is not valid!" validationexpression="([a-
            zA-Z0-9]{8,14})" validationgroup="Login" />
    </div>
</div>

<div class="buttons">
    <asp:button id="btnLogin" runat="server" cssclass="button" text="Login"
        onclick="btnLogin_Click" validationgroup="Login" />

```

```

        </div>
    </fieldset>

    <br />
    <br />

    <fieldset>
    <legend>
        Forgot Password
    </legend>

    <div class="field">
        <div class="caption">
            <asp:label id="lblEmail" runat="server" cssclass="label" text="Email" />
        </div>
        <div class="controls">
            <asp:textbox id="txtEmail" runat="server" cssclass="textBox"
            maxlength="100" accesskey="3" autocompletetype="Disabled" />
            <asp:requiredfieldvalidator id="rfvEmail" runat="server" cssclass="validator"
            controltovalidate="txtEmail" display="Dynamic" setfocusonerror="true"
            text="Email is required!" validationgroup="ForgotPassword" />
            <asp:regularexpressionvalidator id="revEmail" runat="server"
            cssclass="validator" controltovalidate="txtEmail" display="Dynamic"
            setfocusonerror="true" text="Email is not valid!" validationexpression="([a-zA-
            Z0-9]{8,14})" validationgroup="ForgotPassword" />
            <asp:customvalidator id="csvEmail" runat="server" cssclass="validator"
            controltovalidate="txtEmail" display="Dynamic" setfocusonerror="true"
            text="Email does not exist! Please try again..." validateemptytext="false"
            onservervalidate="csvEmail_ServerValidate" validationgroup="ForgotPassword" />
        </div>
    </div>

    <div class="buttons">
        <asp:button id="btnSendMeNewPassword" runat="server" cssclass="button"
        text="Send Me New Password" onclick="btnSendMeNewPassword_Click"
        validationgroup="ForgotPassword" />
    </div>
</fieldset>

</asp:content>

```

```

Login3.aspx.cs

```

```

public partial class Login3 : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        divMessage.Visible = false;

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void csvUsername_ServerValidate(object sender,
    System.Web.UI.WebControls.ServerValidateEventArgs e)
    {

```

```

        if((string.Compare(txtUsername.Text, "Dariush",
System.StringComparison.InvariantCultureIgnoreCase) != 0) ||
(string.Compare(txtPassword.Text, "1234512345",
System.StringComparison.InvariantCultureIgnoreCase) != 0))
        {
            e.IsValid = false;
        }
    }

    protected void csvEmail_ServerValidate(object sender,
System.Web.UI.WebControls.ServerValidateEventArgs e)
    {
        if (string.Compare(txtUsername.Text, "DariushT@Gmail.com",
System.StringComparison.InvariantCultureIgnoreCase) != 0)
        {
            e.IsValid = false;
        }
    }

    protected void btnLogin_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            string strInformationMessage =
                string.Format("Welcome {0}!", txtUsername.Text.ToLower());

            DisplayInformationMessage(strInformationMessage);
        }
    }

    protected void btnSendMeNewPassword_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            string strInformationMessage =
                "New password sent to your mailbox.";

            DisplayInformationMessage(strInformationMessage);
        }
    }

    private void DisplayInformationMessage(string message)
    {
        divMessage.Visible = true;

        lblMessage.Text = message;
        lblMessage.CssClass = "information";
    }
}

```

کنترل Validation Summery

وظیفه این کنترل

Validation Summary پیامهای خطاهای (errormessage) مربوط به Validation را در هنگام بروز خطا را دسته‌بندی و به همراه یک متن نمایش می‌دهد. می‌توان به جای text برای خطاها تصویر گذاشت. Error message پراپرتی در validation summary استفاده می‌شود و کاربرد آن این است که خطاهایی که در کل اتفاق افتاده را نشان می‌دهد. بهتر است validation summary را در ابتدا و بالای صفحه بگذارید.

Property های این کنترل

❖ DisplayMode property

یک property دارد که نوع نمایش خطاها است که سه نوع است :

- list
- singleparagraph
- bulletlist

که bullet خوب است که هر خطا را در یک پاراگراف نشان می‌دهد و با یک شکل زیبا نیز نمایش می‌دهد.

❖ Header Text

معمولاً برای validation summary یک صفت به نام title وجود دارد. که آن را ست می‌کنند.

❖ show message box

showmessagebox را false کنید.

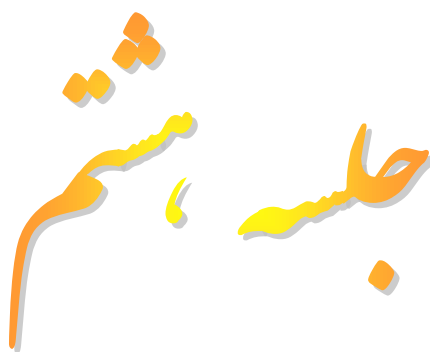
نکته: Alert و Tooltip را فارسی ننویسید

دو چیز را هیچ وقت فارسی نکنید، alert مربوط به javascript و Tooltip فارسی.

می‌توان یک div گذاشت و با position و ... کاری نمود که به صورت مجازی درست نمایش دهد.

Tooltip فقط روی windows فارسی جواب می‌دهد. ولی tooltip و alert مربوط به javascript در سایر سیستم‌ها فارسی دیده نمی‌شود و به صورت مربع و علامت سوال می‌شود.

```
<asp:ValidationSummary ID="RegisterUserValidationSummary" runat="server"
  CssClass="failureNotification" ValidationGroup="RegisterUserValidationGroup"
  headertext="notice" showmessagebox="False" />
```



مفهوم User Control

از قرار گرفتن چند کنترل در کنار هم که به اصطلاح مفهوم جدیدی را تولید کند به آن user control می‌گویند. مثلاً می‌خواهید تاریخ شمسی داشته باشید. پس می‌توانید یک textbox و بعد از آن یک lable اسلش و بعد دوباره textbox و بعد lable و در انتها combobox و .. اگر بلد نباشید user control درست کنید هر بار که بخواهید تاریخ شمسی داشته باشید باید تمام این کارها را تکرار کنید. ولی اگر همه این کارها را در کنار هم بگذارید و یک موجود جدید درست می‌کنید و آن را صدا می‌کنید.

یک سایت بورس درست می‌کنید و یک کنترل به نام Currency Control

یک بحث به نام inheret control ها دارید و به یک کنترل یک سری ویژگی می‌دهید و در این درس کاری با آن نداریم در اینجا با کنار هم گذاشتن کنترلها یک کنترل جدید یا مفهوم جدید ایجاد می‌کنیم.

عرف بر این است که اول یک فولدر به نام UserControlس درست کنید و در داخل آن یک عدد صفحه از نوع web user control اضافه کنید نام آن را CurrencyConvertor می‌گذاریم.

User control تحت وب بسیار شبیه به User control در windows application است و خالی است و اجازه می‌دهد که هر چه بخواهید در آن بگذارید.

در این صفحه یک textbox به نام txtDollars با طول 10 و بعد از آن یک textbox دیگر با نام txtRials درست کنید و سپس یک دکمه با نام btnSubmit اضافه کنید.

در textbox دوم نباید چیزی تایپ شود، پس یا باید read only آن را ست کنید و یا enabled آن را false کنید، ولی به دلیل زیبایی الان enabled آن را false می‌کنیم.

❖ container

در ویندوز 5 based نوع container بیشتر نداشتیم یعنی کنترلهایی که درونشان کنترلهای دیگری می‌شود انداخت و 5 تا بیشتر نیستند:

❖ Form
❖ panel

- ❖ user Control
- ❖ tab control
- ❖ group box

❖ span و div

در وب ما container زیاد داریم. تگی که در آن می توان تگهای دیگر نوشت، container است، هم div , p , h1..h6 , span, .. container هستند ولی دو تگ هستند که اصالت container آن بیشتر است و این دو تا div و span هستند که معمولا وقتی چیزهایی را بخواهیم مدیریت کنیم یا زیبا کنیم از این دو استفاده می کنیم.

○ فرق div , spam چیست؟

تگها از نگاه دیگر به دو دسته تقسیم می شوند یا **in line** هستند این تگها شما را سر خط نمی برد و پشت سر هم هستند، مثل i,u, b ,span هم همینطور است.

تگهای **block** شما را به سر خط می برد. مثل div

در خصوص user control ها بهتر است از span استفاده کنید اگر در یک خط چند تا تاریخ شمسی داشته باشید، با div نمی توانید ولی با span می توانید. برعکس آن می توانید یک span را در یک خط نگاه دارید و دنبال آن چیزی ننویسید.

```
<span style="background-color: Yellow; color: Blue; border: 1px outset Gray; margin: 5px; padding: 5px;">
    <asp:textbox id="txtDollars" runat="server" maxlength="10" />
    <asp:textbox id="txtRials" runat="server" enabled="false" />
    <asp:button id="btnSubmit" runat="server" text="..." onclick="btnSubmit_Click"
validationgroup="CurrnecyConvertor" />
</span>
```

○ **تذکر آیین نامه ای اول:** اینکه به تگ style دادم بسیار غیر حرفه ای و زشت است و این کار را فقط برای پیش بردن سریعتر درس انجام دادم و شما در فایل style یک کلاس باید اضافه کنید. و شما بهتر است در site.css این **span.CurrencyConvertor{}** را اضافه کنید:

○ تذکر آیین نامه ای دوم:

تصور کنید که شما یک user Control درست کردید و در چند صفحه گذاشتید، و مشتری می گوید می شود در تمام صفحات بگذارید، می گوییم بله و آن را در master page می گذارم. در ضمن چند تا validator در این صفحه گذاشتم و در صفحه login هم چند تا validator داشتم. و حالا تا دکمه یک قسمت را می زنم، validator دسته دیگر خطا می دهد. پس یک تاکتیک این است که همیشه برای validator هایتان و برای دکمه ای که در کنترل خود دارید یک validation group بگذارید تا اگر بعداً validation group جدید آمد با هم conflict نکند، بهترین حالت برای unique شدن نام validation group این است که آن را هم نام کنترل بگیرید.

در کد نویسی: اولین شرط می گوییم که اگر space داریم به string.empty تبدیل کن. برای این کار از تابع replace استفاده می کنیم.

برای تبدیل یک چیز غیر عددی به عددی:

```
System.Convert.ToInt64(txtDollars.Text);
```

برای تبدیل یک غیر string را به string تبدیل کنید:

```
txtRials.Text = (lngDollars * 123450).ToString("#,##0.Rials");
```

تازه علاوه بر string شدن، برای آن به صورت بالا می توان فرمت تعریف کرد. فرمت تعریف شده در بالا یعنی از separator برای جدا کردن سه عدد، سه عدد یک عدد استفاده کن.

❖ نحوه یافتن استاندارد های فرمت دادن در string :

microsoft office->format cells->custom

```
protected void btnSubmit_Click(object sender, System.EventArgs e)
{
    if (Page.IsValid)
    {
        //txtDollars.Text = txtDollars.Text.Trim();
        1 → txtDollars.Text = txtDollars.Text.Replace(" ", string.Empty);
        if (txtDollars.Text == string.Empty)
        {
            txtRials.Text = string.Empty;
        }
        else
        {
            try
            {
                2 → long lngDollars =
                    System.Convert.ToInt64(txtDollars.Text);
                3 → txtRials.Text = (lngDollars * 12450).ToString("#,##0
Rials");
            }
            catch
            {
                txtRials.Text = string.Empty;
                txtDollars.Text = string.Empty;
            }
        }
    }
}
```

❖ کارتان با user control تمام شد، یک rebuild کنید و بعد برگردید. اول rebuild می کنید و اگر error نداشت ادامه می دهید.

یک صفحه به نام user Control Tester درست کنید. برای اینکه، این user Control را بتوانید در صفحه درست کنید. می توان به design جایی که می خواهید user Control آنجا باشد و بعد drag & drop کنید.

در 2003 بعد از drag & drop هیچ چیز جز یک صفحه طوسی زشت نمی‌دیدید ولی user interface مربوط به user control از 2005 درست شد. با انجام این کار دو عمل صورت می‌گیرد:

drag & drop a UserControl in design of a page

- اولاً user control رجیستر می‌شود
- خطی را برای استفاده از آن ایجاد می‌کند. در واقع یک عدد از آن را در صفحه می‌اندازد.

اگر بخواهید می‌توانید (uc1) tagprefix را در قسمت register از user control به هر چیزی که می‌خواهید عوض کنید، مثلاً dtsx و یا abfa یا هر چیز دیگری که بخواهید.

create property

Property در #C با زدن نقطه فهرست آن می‌آید.

حال می‌خواهیم یک property برای user control بنویسیم. یک property برای آن تعریف کنیم که بتوانیم در کد نویسی به آن مقدار اولیه دهیم و یکی که فقط از آن مقدار در یافت کنیم.

❖ تعریف Property bubling

باید Property یا get یا set داشته باشد. set برای گرفتن یک مقدار از دنیای بیرون و get برای دادن یک مقدار به دنیای بیرون است.

```
public long Dollars
{
    set
    {
        txtDollars.Text = value.ToString();
    }
}
```

مقداری که از دنیای بیرون به آن می‌دهیم

```
public long Rials
{
    get
    {
        if (txtRials.Text == string.Empty)
        {
            return (0);
        }
        else
        {
            string strRials =
                txtRials.Text.Replace(",", string.Empty).Replace("
Rials", string.Empty);

            return (System.Convert.ToInt64(strRials));
        }
    }
}
```

مقداری که به دنیای بیرون می‌دهیم

در کد نویسی هر صفحه ای:

```
cc1.Dollars = 100;
```

```
cc3.Dollars = 2500;
```

Ajax

آموزش ajax سه level دارد، 1 و 2 مال Microsoft است که الان آن را درس می‌دهد و نوع 3 آن مربوط به jQuery Ajax است.

روش اول راه‌انداختن آن ساده است ولی یک مقدار زیادی over head دارد نوع سه سخت راه‌اندازی می‌شود ولی بسیار تکنیکی و خوب است و بعضی از کارها را اصلاً نمی‌توان با ajax Microsoft انجام داد.

Ajax Microsoft بسیار خوب است حداقل اینکه در ساخت سایت به شما سرعت می‌دهد و در واقع اگر سایت ajax Microsoft داشته باشد، خیلی بهتر از سایتی است که نداشته باشد. ضمن اینکه بدترین نوع استفاده از ajax این است که یک صفحه ajax نداشته باشد بهتر است. در حالت ایده آل ما سراغ ajax jquery می‌رویم.

تمام روالی که در حالت post back عادی گفته بودیم در حالت ajax هم همینطور است. فقط یک data gathering از طریق soap و سپس مراحل دیگر صورت می‌گیرد و نتیجه دوباره در غالب xml به صفحه بر می‌گردد و تنها فرقی که با صفحه بدون ajax دارد این است که کل صفحه نمی‌رود (سفید نمی‌شود). فرق درس امروز با جلسات قبل فقط در همین است. اصلاً با ajax اتفاق خاصی نمی‌افتد. هرگز نگویید در ajax دیگر post back صورت نمی‌گیرد و فقط در ajax pure jquery است که واقعاً post back صورت نمی‌گیرد.

مراحل کار برای ajax کردن یک صفحه :

1. اول سایت را تمام می‌کنند و بعد ajax ای می‌کنند، شاید اگر از اول ajax ای شود ممکن است به مشکل برخورد و بعد از تکمیل سایت به آن ajax اضافه می‌کنید. برای مثال از صفحه ای که نام و نام خانوادگی را می‌گیرد :

نام + نام خانوادگی = نام و نام خانوادگی

Enable مربوط به نام خانوادگی را false می‌کنیم.

2. وقتی از ajax استفاده می‌کنید، باید یک و فقط یک **scriptmanager** وجود داشته باشد، چون یکی است می‌تواند id نداشته باشد ولی به دلیل رعایت استاندارد id می‌گذاریم.

`id="myScriptManager"`

3. آن ناحیه ای را که می‌خواهید update شود به صورت ajax ای را، در داخل چیزی به نام `update panel` می‌گذاریم.

اگر صفحه ای چندین ناحیه `update` ای داشته باشد و نیاز به چندین `update panel` داشته باشد در این حالت ها بهتر است سراغ `jquery` بروید.

در سایت بزرگ باید تشخیص دهید کجا از ajax مربوط به Microsoft و کجا از ajax مربوط به jQuery استفاده کنید. اگر فقط در صفحه یک grid دارید، اگر با ajax jQuery برید فکتان زمین می‌خورد و آخر هم هیچ `performance` ای بدست نیآورده‌ایم.

از آنجایی که یک صفحه محدود به یک `update panel` نمی‌باشد پس باید یک `abbreviation` به آن بدهیم.

باید از `abbreviation` استفاده شود که با `udp` شروع می‌شود.

4. انداختن بخشهایی که می خواهیم ajaxی باشد در `contenttemplate` از `update panel` می اندازیم.

کل صفحه را در `update panel` نمی گذاریم و همان `Content place holder` را داخل آن می گذاریم. نواحی که باید `update` خود این قسمت را می توان درون چند `update panel` قرار داد.

برای اضافه کردن `Microsoft ajax` ابتدا یک `scriptmanager` اضافه کردم و بعد هم بقیه کد را در قسمت `template` مربوط به `update panel` می گذارم.

```
<form id="frmMain" runat="server">
  <asp:scriptmanager id="myScriptManager" runat="server" />
  <asp:updatepanel id="udpFullName" runat="server">
    <contenttemplate>
      <asp:label id="lblFirstName" runat="server" text="First Name" />
      <br />
      <asp:textbox id="txtFirstName" runat="server" />
      <br /><br />
      <asp:label id="lblLastName" runat="server" text="Last Name" />
      <br />
      <asp:textbox id="txtLastName" runat="server" />
      <br />
      <br />
      <asp:button id="btnSubmit" runat="server" text="(S)bmit" accesskey="S"
        onclick="btnSubmit_Click" />
      <br />
      <br />
      <asp:label id="lblFullName" runat="server" text="Full Name" />
      <br />
      <asp:textbox id="txtFullName" runat="server" enabled="false" />
    </contenttemplate>
  </asp:updatepanel>
</form>
```

ایجاد لختی و کندی در اجرای سیستم

برای تشخیص اینکه `ajax` چه کاری انجام می دهد باید یک لختی در سایت ایجاد کنید. این لختی را با دستور ذیل درست کنید، الان انگار که کندی اینترنت را شبیه سازی کرده اید.

```
System.Threading.Thread.Sleep(10000);
```

ایجاد `pageLoader` با استفاده از `asp:updateProgress`

حالا یک مشکل دیگر پیش می آید. اولاً عوام با فرهنگ `ajax` آشنا نشده اند و ثانیاً باید به طرف بفهمانیم که صفحه `ajax` ای است، چون در غیر اینصورت ممکن است فکر کند که صفحه کار نمی کند و مشکل دارد. پس باید در صفحه `ajax` ای یک چیزی بگردد و بچرخد و ...

در دنیا ایده بسیار مهم است. اگر ایده داشته باشید، یک صفحه می تواند شما را میلیاردر نماید. یک صفحه داریم که اصلاً قشنگ نیست و قیمت آن بالغ بر 500000 دلار است. این سایت، سایتی است که یک `drop down list` ای دارد از انواع

pageLoaderها و می گوید که این گوی که می خواهد بچرخد، چه شکلی است و در همان زمان آن را برای شما درست می کند. اگر این را با ie باز کنید فقط عنوان ها می آید ولی در mozilla می توانید دموی آن را در combo box ببینید. در همانجا رنگ و رنگ زمینه و ... را انتخاب و download می کنید. اسم این سایت. ajaxLoad.info می باشد.

برای اینکه به user بگوییم که صبر کن تا اتفاق بیفتد. یک تگ داریم به نام asp:updateProgress این را هم می توان استفاده نمود ولی اگر استفاده کردید، فقط یکدانه می توانید استفاده کنید. زیر خط scriptmanager این خطوط را اضافه می کنیم.

```
<asp:updateprogress id="myUpdateProgress" runat="server">
<progresstemplate>
<div style="background-color: Yellow; color: Blue; border: 1px outset Gray; margin: 4px;
padding: 4px;">
<asp:image id="imgLoading" runat="server" imageUrl="~/Images/Loading.gif"
imagealign="AbsMiddle" />
Loading... Please Wait!
</div>
<br /><br />
</progresstemplate>
</asp:updateprogress>
```

این تگ ناحیه ای است که سمت client به صورت visible می شود و برای آن هیچ کدی نوشته نمی شود و خودش به محض اینکه صفحه سمت ajax می رود متوجه می شود و visible می شود و پس از اتمام کار invisible می شود.

در بعضی از سایتها کل کنترلها قفل می شود و مثل یک تلق روی عبارات را می گیرد، این کار از طریق javascript صورت می گیرد و به درس ما مربوط نمی شود.

در سمت کد نویسی :

```
protected void Page_Load(object sender, System.EventArgs e)
{
    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
}

protected void btnSubmit_Click(object sender, System.EventArgs e)
{
    System.Threading.Thread.Sleep(10000);

    if (Page.IsValid)
    {
        txtLastName.Text = txtLastName.Text.Trim();
        txtFirstName.Text = txtFirstName.Text.Trim();

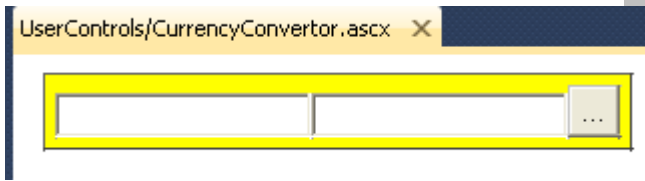
        txtFullName.Text = (txtFirstName.Text + " " + txtLastName.Text).Trim();
    }
}
```

}

System.globalization.persian

فارسی کردن تاریخ میلادی به شمسی

مال خود ماکروسافت است و خود شما یک سری convertor خوشگل و خوش دست درست کنید و در همه پروژه استفاده کنید ، در این user control هم استفاده کنید.

مثال درست کردن PersianCalendar UserControl**PersianCalendar.ascx**

```
<% @ Control Language="C#" AutoEventWireup="true" CodeFile="PersianCalendar.ascx.cs"
Inherits="UserControls_PersianCalendar" %>

<span style="background-color: Yellow; color: Blue; border: 1px outset Gray; margin: 5px; padding:
5px;">
    <asp:dropdownlist id="ddlY" runat="server" />
    /
    <asp:dropdownlist id="ddlM" runat="server" />
    /
    <asp:dropdownlist id="ddlD" runat="server" />
</span>
```

```
public partial class UserControls_PersianCalendar : System.Web.UI.UserControl
{
    public UserControls_PersianCalendar()
    {
        MinYear = 1385;
        MaxYear = 1390;
    }

    private int _minYear;
    public int MinYear
    {
        get
        {
            return (_minYear);
        }
        set
        {
            _minYear = value;
        }
    }

    private int _maxYear;
    public int MaxYear
    {
        get
```

ایجاد یک constructor که
اسمش با نام کلاس یکی است
معروف به ایجاد کننده کلاس.
مقدار اولیه دادن به property

```
        {
            return (_maxYear);
        }
        set
        {
            _maxYear = value;
        }
    }

    protected void Page_Load(object sender, System.EventArgs e)
    {
        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
        System.Web.UI.WebControls.ListItem oListItem = null;

        for (int intIndex = 1; intIndex <= 31; intIndex++)
        {
            oListItem =
                new System.Web.UI.WebControls.ListItem(intIndex.ToString(),
intIndex.ToString());

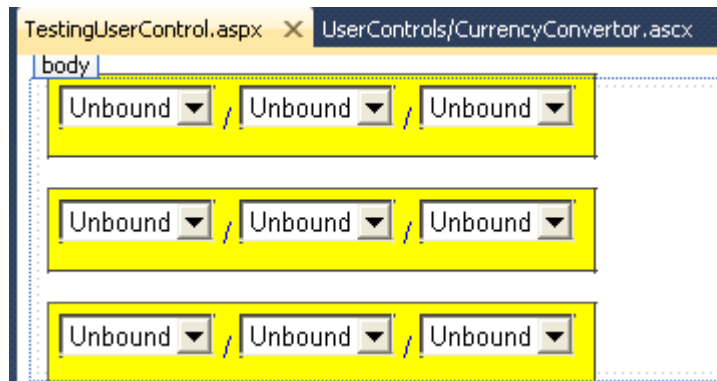
            ddlID.Items.Add(oListItem);
        }

        for (int intIndex = 1; intIndex <= 12; intIndex++)
        {
            oListItem =
                new System.Web.UI.WebControls.ListItem(intIndex.ToString(),
intIndex.ToString());

            ddlM.Items.Add(oListItem);
        }

        for (int intIndex = MinYear; intIndex <= MaxYear; intIndex++)
        {
            oListItem =
                new System.Web.UI.WebControls.ListItem(intIndex.ToString(),
intIndex.ToString());

            ddlY.Items.Add(oListItem);
        }
    }
}
```

ایجاد صفحه ای برای استفاده از usercontrol Persian calendar

TestingUserControl.aspx

```

<%@ Register src="UserControls/PersianCalendar.ascx" tagname="PersianCalendar" tagprefix="uc1"
%>

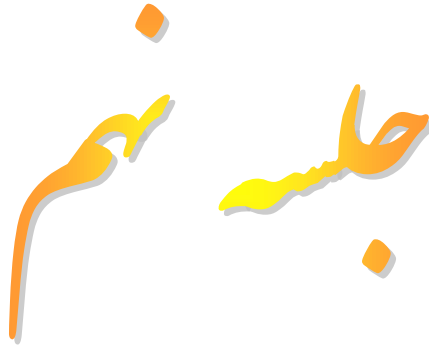
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>
  </head>
  <body>
    <form id="frmMain" runat="server">
      <div>
        <uc1:persiancalendar id="PersianCalendar1" runat="server" />
        <br />
        <br />
        <uc1:persiancalendar id="PersianCalendar2" runat="server"
minyear="1300" maxyear="1370" />
        <br />
        <br />
        <uc1:persiancalendar id="PersianCalendar3" runat="server" />
      </div>
    </form>
  </body>
</html>

```

جلسه بعد آموزش ado.net

Ado.net version

- 1 و 1.1 سنتی
- 2 مناسب
- 3 و 3.5 link to sql
- 4 Lint to entity
 - Database first
 - Model first
- 4.1 Link to entity code first



روش کار با ADO.Net

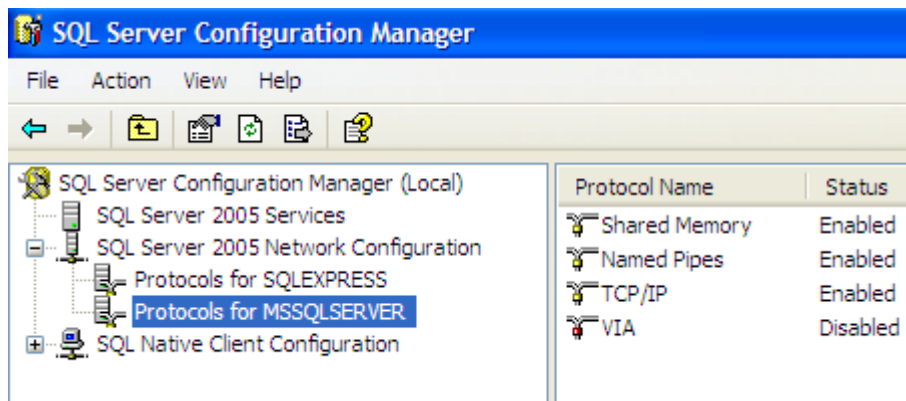
قسمت بانک اطلاعاتی

تنظیمات اولیه بانک اطلاعاتی – SQL Configuration & SQL Services

ابتدا باید بانک اطلاعاتی SQL Server را برای برنامه‌نویسی آماده نمایید. مراحل ذیل را به ترتیب انجام دهید:
1 - ابتدا باید تنظیمات لازم در قسمت SQL configuration را انجام دهید.

برای این کار از طریق مسیر **start** → **Microsoft SQL Server** → **Configuration Tools** → **SQL Server Configuration Manager** وارد صفحه تنظیمات شوید.

در قسمت **protocol for MSSQLSERVER** سه پروتکل **Named Pipes** و **shard memory** و **TCP/IP** را **enable** نمایید و سپس سرویس را مجدداً **restart** کنید.



2 - برای **restart** کردن سرویس SQL وارد قسمت **services** شوید و سرویس **SQL Server** را یک بار **Stop** و مجدداً **Start** نمایید.

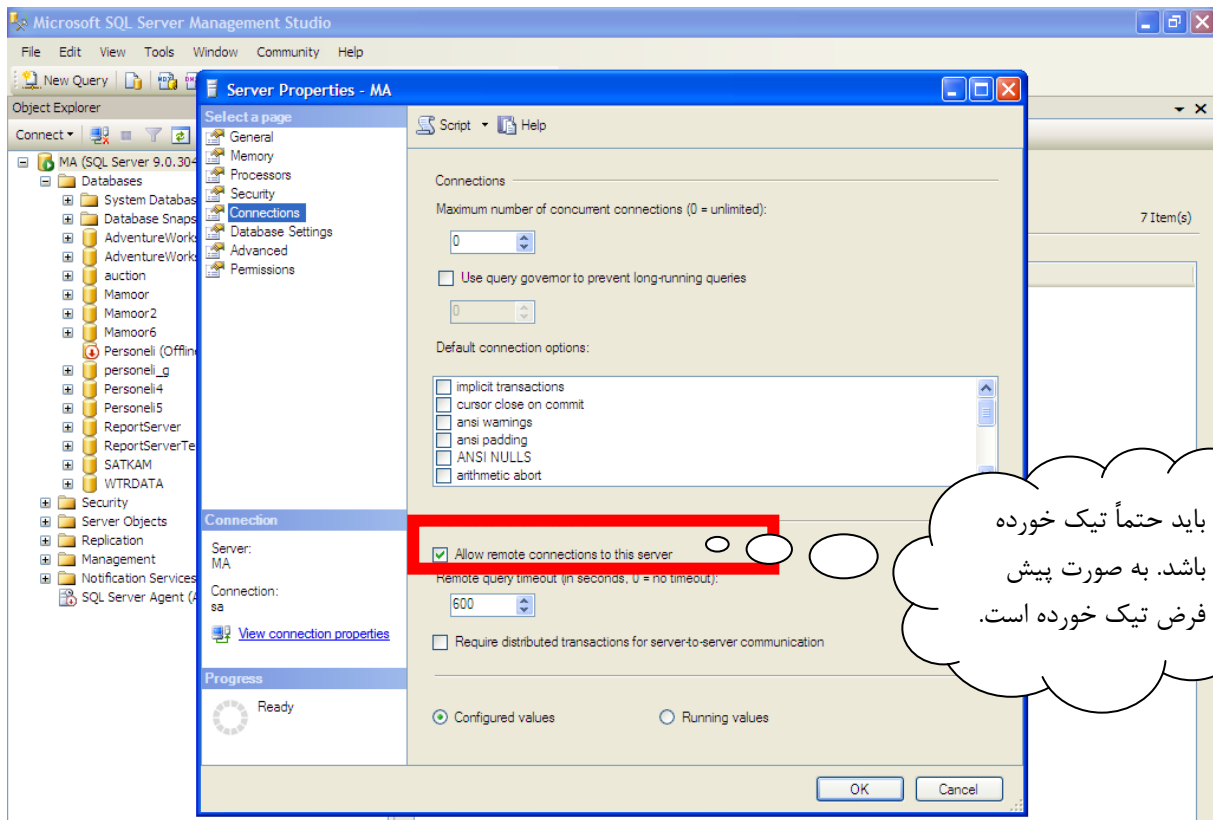
نکته: لیست سرویس‌های مربوط به **SQL Server** را می‌توان در قسمت **SQL Server Services** نیز مشاهده و در صورت نیاز سرویس مورد نظر را **start**, **stop** نمایید.

نکته: مسئله **sql engine** و محیطی برای وصل شدن به **SQL (SQL Management)** با هم فرق دارد. مثلاً ما در اینجا **Management Studio 2005** را نصب کردیم و این می‌تواند به **SQL Server 2008 engine** وصل شود.

هنگام ورود به محیط SQL Server و باز شدن پنجره Connect to Server در قسمت server name اگر بر روی دستگاه شما نسخه SQL Server standard یا enterprise بود، در قسمت sever type فقط یک نقطه بگذارید. و اگر نسخه express بود، باید SQLExpress.\ بگذارید.



بعد از انجام تنظیمات و ورود به محیط SQL Server Management باید یک سری تنظیمات را انجام دهیم. وارد قسمت Properties و بعد به قسمت security بروید و mixed mode را انتخاب نمایید. بعد از این کار نیز باید یک بار سرویس SQL را stop و start کنید. باید در tab connection هم تیک ذیل حتماً تیک خورده باشد.



تعریف user

در هنگام برنامه‌نویسی نباید از user مربوط به مدیریت SQL یا همان (SQL Administrator) sa استفاده نمود، بلکه باید یک user جدید تعریف نمود.

از لحاظ امنیتی، لو رفتن کلمه عبور sa بسیار خطرناک است. از طریق sa مربوط به sql server به file system آن کامپیوتر هم می‌توان دسترسی یافت.

پس اول یک user بر روی بانک می‌سازید و بعد یک بانک اطلاعاتی می‌سازیم و بعد owner آن بانک اطلاعاتی را آن user می‌گذارید.

پس وارد new login می‌شویم و یک user به نام abfa می‌سازیم.

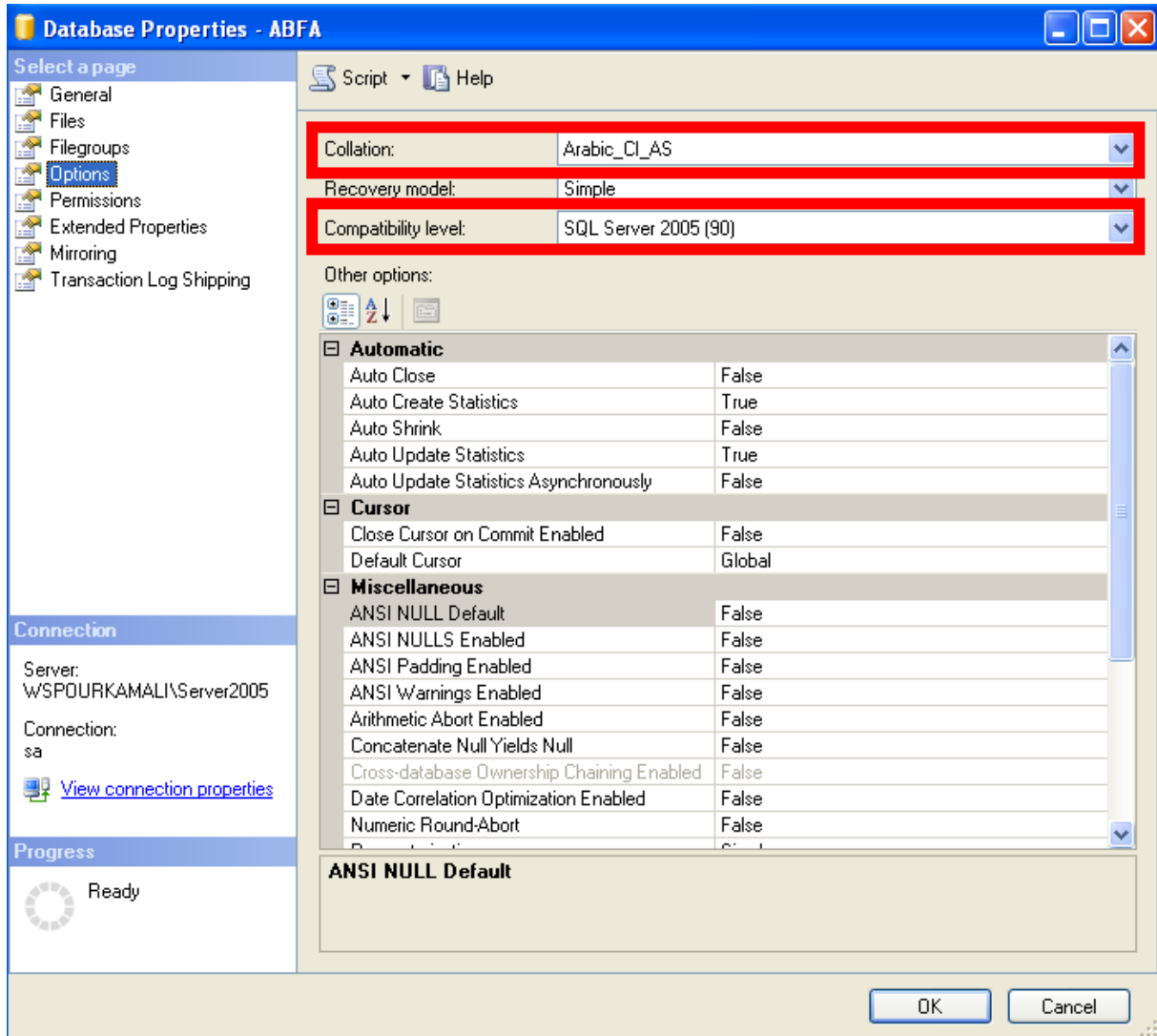
The screenshot shows the 'Login - New' dialog box in SQL Server Enterprise Manager. The 'Login name' field contains 'ABFA'. Under 'Authentication', 'SQL Server authentication' is selected. The 'Password' and 'Confirm password' fields are filled with masked characters. There are checkboxes for 'Enforce password policy', 'Enforce password expiration', and 'User must change password at next login', all of which are currently unchecked. Under 'Mapped to certificate', the 'Certificate name' field is empty. Under 'Mapped to asymmetric key', the 'Key name' field is empty. The 'Default database' dropdown is set to 'master' and the 'Default language' dropdown is set to '<default>'. The 'Connection' section shows the server 'WSPOURKAMALI\Server2005' and connection 'sa'. The 'Progress' section shows a 'Ready' status with a circular progress indicator.

تعریف بانک اطلاعاتی (Database)

سپس یک بانک اطلاعاتی به نام ABFA درست کنید. باید collation را CI_AS انتخاب نمایید. در SQL 2008 collation برای فارسی‌ها داریم به نام Persian100 ولی آن را استفاده نکنید، چون مشکل compatibility دارد. فقط با پروژه‌های .net framework و آن هم 3.5 با sp1 کار می‌کند، می‌توانید از persian100 استفاده کنید ولی باید حواستان باشد که با سیستم دیگری مثل Delphi, ... به آن وصل نشوید، در غیر اینصورت از persian100 استفاده نکنید.

هیچ بانک اطلاعاتی که با 2005 بسازید با 2000 قابل برگرداندن نیست و ... بنابراین بدانید که خاصیت Compatibility level که در هنگام ایجاد بانک اطلاعاتی آورده شده است، عملاً به درد نخور است و کاری نمی‌کند و بنابراین روی همان 2008 یعنی بالاترین قرار دهید.

سپس باید سه جدول درست کنیم. فهرست کشورها، استانها و کاربران را در سه جدول درست می‌کنیم. اول جدول کشورها را درست می‌کنیم.



تعریف جدول (Table)

اولین جدول Countries را ایجاد می‌کنیم. باید در نامگذاریها دقت نمایید. چون قرار است لیست کشورها در این جدول قرار گیرد، پس از نام جمع استفاده کنید.

برای نامگذاری فیلدهای جداول باید نامهایی را که حدس می‌زنید در جداول دیگر نیز تکرار می‌شود، همراه با اسم جدول به کار ببرید، مثلاً برای ستون ID کشورها نباید نام فیلد را ID خالی بگذارید و باید CountryID بگذارید.

سوال:

شش type مختلفی را که برای متن وجود دارد (char, nchar, varchar , nvarchar) را تفاوت یابی کنید و وقتی که می‌خواهید Table درست کنید، هوشمندانه تشخیص دهید که چه type متنی را استفاده کنید. text , ntext وجود دارد که دیگر استفاده نمی‌شود.

پاسخ:

1 - char : نوع داده‌ای char طول ثابتی دارد. یعنی اگر ستونی به طول 20 کارکتر تعریف کنید، آنگاه 20 کارکتر ذخیره خواهد شد و اگر تعداد کمتر از تعداد کارکترهای تعریف شده وارد نمایید، بقیه طول از سمت راست با فضای خالی پر می‌شود. از این نوع داده هنگامی استفاده کنید که قرار است داده‌های ستون طول ثابتی داشته باشد. مانند کد ملی یا شناسه حساب بانکی و ...

2 - nchar : دقیقاً شبیه char می‌باشد، با این تفاوت که کارکترها به صورت یونیکد (Unicode) و نه ANSI نگهداری خواهد شد. قالب یونیکد در مقایسه با ANSI محدوده کارکتری وسیعتری دارد. مجموعه کارکترهای ANSI تنها 256 کارکتر را نگهداری می‌کند و اما انواه داده‌ای Unicode قادر به نگهداری 65536 کاراکتر مختلف است. انواع داده‌ای Unicode فضای بیشتری در SQL Server می‌گیرند. در حقیقت SQL Server دو برابر فضای داخلی را به آن اختصاص می‌دهد بجز در مواردی که ناچار به ذخیره‌سازی این نوع کاراکتر در بانک اطلاعاتیتان هستید، ساده‌تر است با ANSI کار کنید.

3 - Varchar : نوع داده ای varchar دقیقاً مانند char داده‌های حرفی - عددی را نگهداری می‌کند، تفاوت در این است که هر ردیف می‌تواند تعداد متغییر کاراکتر را حداکثر تا طول تعریف شده نگهداری کند. حداکثر اندازه یک ستون varchar ، 8000 کاراکتر است. اگر بخواهید این ستون را بدون اندازه یعنی به شکل varchar() تعریف کنید، به صورت پیش‌فرض طول 1 خواهد شد. برای فراتر رفتن از مرز 8000 کاراکتر، می‌توانید نوع داده‌ای را با ثابت max مورد استفاده قرار دهید.

4 - Nvarchar : این نوع داده مشابه با varchar می‌باشد با این استثناء که از Unicode استفاده می‌کند و بنابراین میزان فضای مورد نیاز جهت ذخیره داده‌ها دو برابر است.

تعریف Primary key ، Foreign key و Alternate key

به هیچ عنوان جدول بدون Primary Key ایجاد نکنید و این خیلی اشتباه است. سوال: چرا امکان ذخیره کردن جدول بدون primary key وجود دارد؟ مثلاً در بانک کار می‌کنید و یک table فعال در بانک اطلاعاتی دارید که داخل آن یک میلیون رکورد است و شما می‌خواهید این یک میلیون رکورد را در یک جدول دیگر import کنید. primary key , foreign key و ... یک سری constraint است و قید و بند هستند و سرعت را پایین می‌آورند و یک بار اضافی برای سیستم است. چون مثلاً برای primary key به ازای هر رکورد چک می‌کند که تکراری نباشد، پس در جدول جدید، تمام قیدها را حذف می‌کنم و bulk insert می‌کنیم و بعد قیدها را می‌گذاریم.

هر جدول فقط باید یک کلید اصلی داشته باشد، اگر چند تا فیلد کنارشان کلید آمده، این چند فیلد یک کلید را می‌سازند و ما چند کلید اصلی نداریم.

سعی کنید، primary key بر روی یک فیلد باشد، البته در جداول واسط نمی‌شود و این را باید فقط سعی کنید. اگر چندین فیلد می‌تواند کاندید primary key باشند، آن یکی را انتخاب کنید که عددی باشد، چون سربار آن کمتر است.

به سه دلیل برای فیلدها index می‌گذاریم.

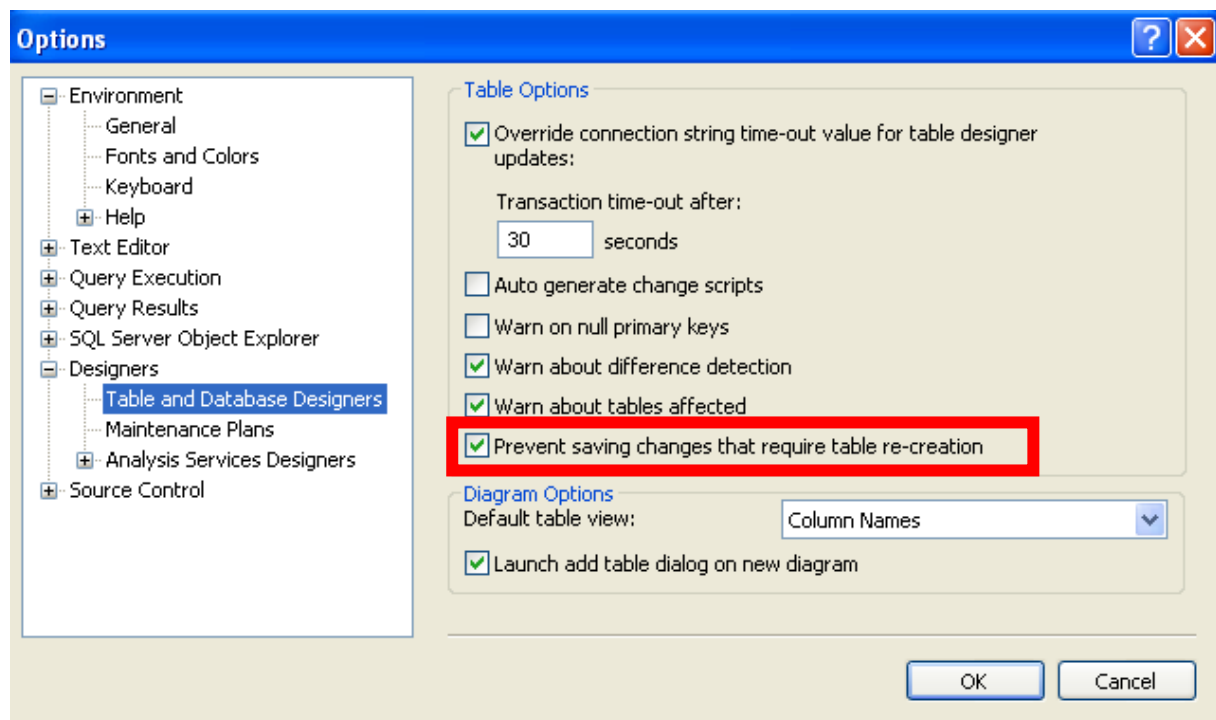
- 1- زمانی که می‌خواهیم بر روی آن فیلد order by داشته باشیم .
- 2- وقتی که می‌خواهیم روی آن جستجو (search) داشته باشیم.
- 3- جایی که می‌خواهیم بگوییم که unique یا منحصر به فرد است.

فیلد ContryID را به صورت Primary key تعریف می‌کنیم. فیلد ContryName هر سه حالت را دارد، پس حتماً باید index شود. چون حتماً می‌خواهیم بر اساس آن مرتب کنیم. و ثانیاً باید روی آن جستجو کنیم و ثالثاً حتماً باید unique باشد. معنی ندارد نام کشور تکراری داشته باشیم.

Alternate Key: فیلدهایی که روی آن index تعریف می‌کنیم اگر index به صورت unique باشد به آن Alternate key گفته می‌شود.

نکته: اگر روی فیلدی unique بودن مهم نیست ولی search , sort مهم است. اگر امکان می‌دهید که تعداد رکوردهای آن جدول از 1000 تا تجاوز نمی‌کند، اتفاقاً نباید روی آن جدول، ایندکس بگذارید، چون performance را پایین می‌آورد. Sort بدون index برای تعداد رکورد کمتر از 1000 تا سریعتر است. پس فقط در جدول زیر 1000 رکورد در صورتی که می‌خواهید فیلدی unique باشد، بر روی آن index تعریف کنید، و در غیر اینصورت index نگذارید.

نکته: در Sql server 2008 اگر امکان ذخیره کردن تغییرات ایجاد شده در ساختار یک جدول موجود، وجود ندارد، باید تیک Tools → option → designer → prevent saving changes that require table re-creation را بردارید.



Primary key باید unique باشد.

نکته: برای Null کردن یک فیلد در جدول، می‌توانید، کلمه Null را تایپ کنید و یا Ctrl+0 را تایپ کنید.

استانها در هر کشور unique هستند.

از ID به صورت auto increment استفاده نکنید. در جداول اصلی مثل user و ... می‌توان از auto increment استفاده نمود.

GoId یک id است که در کل دنیا unique است و عملاً string است.

بحث Data Access Layer داریم. در این دوره وارد بحث Business Access Layer نمی‌شویم و این لایه با UI ما یکی است، ما نام لایه Data Access Layer را dataset می‌گذاریم.

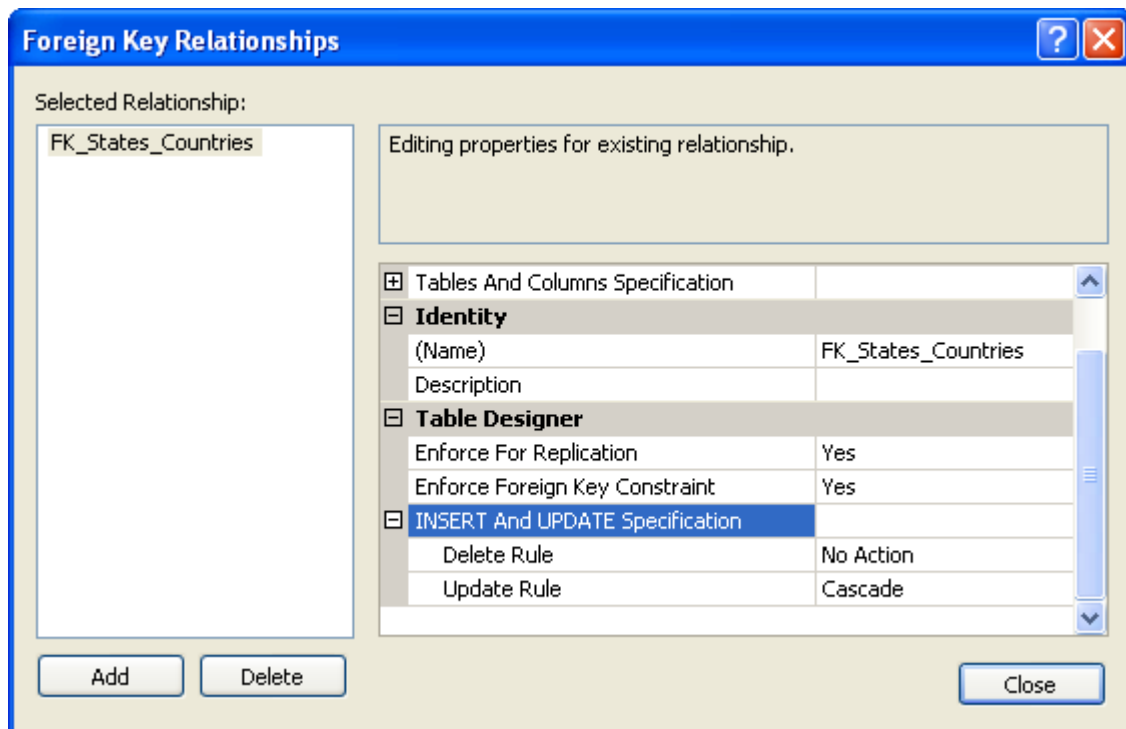
سعی کنید از همان ابتدا کل تنظیمات جدول و فیلدها را درست انجام دهید، اگر بعد از درست کردن dataset یادتان بیاید که این تیک auto increment را در مورد فیلد ID در جدول نزدیک، باید تمام کارها را مجدداً انجام دهید. پس مثلاً در جدول Users، فیلد userid را سریعاً identity آن را yes بگذارید.

Username به صورت Alternate key است. البته در دنیای واقعی Email هم به همین صورت است.

نکته: در فیلدی که به صورت identity معرفی شود، نمی‌توان تایپ کرد.

شاید یک relation بین دو جدول state, countries وجود نداشته باشد، ولی باید یک foreign key بین آنها برقرار کنید، برای اینکار وارد قسمت relationship شوید و بعد در قسمت Tables and Columns specification نام دو جدول و کلید اصلی و کلید خارجی از جدول دیگر را انتخاب نمایید.

در قسمت Insert and Update specification دو تا rule داریم، Delete Rule را اصولاً No Action می‌گذارند، مگر اینکه خیلی دلیل موجه‌ای داشته باشیم. ولی Update Rule را به صورت Cascade تعریف کنید، چیز بدی نیست.



به طور کلی ساختار سه جدول به صورت ذیل می‌باشد:

جدول کشورها:

Table - dbo.Countries		
Column Name	Data Type	Allow Nulls
CountryId	int	<input type="checkbox"/>
CountryName	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Selected Primary/Unique Key or Index:
IX_Countries
PK_Countries

Editing properties for existing primary/unique key or index.

(General)

Columns	CountryName (ASC)
Is Unique	Yes
Type	Index

Identity

(Name)	IX_Countries
Description	

No
PRIMARY

Add

نام ستونی که می‌خواهیم به صورت index و یا Alternate key تعریف شود.

Yes بودن این ستون به معنی Unique بودن این ستون است و بنابراین index را تبدیل به Alternate key می‌نماید.

جدول استانها:

Table - dbo.States		
Column Name	Data Type	Allow Nulls
StateId	int	<input type="checkbox"/>
CountryId	int	<input type="checkbox"/>
Statename	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Indexes/Keys

Selected Primary/Unique Key or Index:
IX_States
PK_States

Editing properties for existing primary/unique key or index.

(General)	
Columns	Statename (ASC), CountryId (ASC)
Unique	Yes
Index	IX_States
Clustered	No
Space Specification	PRIMARY
Fill Specification	

Buttons: Add, Delete

Yes بودن این ستون به معنی Unique بودن این ستون است و بنابراین index را تبدیل به Alternate key می نماید.

نام ستونهایی است که می خواهیم به صورت index و یا Alternate key تعریف شود.

Foreign Key Relationships

Selected Relationship:
FK_States_Countries

Editing properties for existing relationship.

(General)	
Check Existing Data On Create	Yes
Tables And Columns Specific	
Identity	
(Name)	FK_States_Countries
Description	
Table Designer	
Enforce For Replication	Yes
Enforce Foreign Key Constraints	Yes
Enforce DELETE And UPDATE Specific	

Buttons: Close

نام دو فیلد از دو جدول که قرار است با هم در ارتباط باشند، مشخص می شود.

نام Foreign key

Tables and Columns

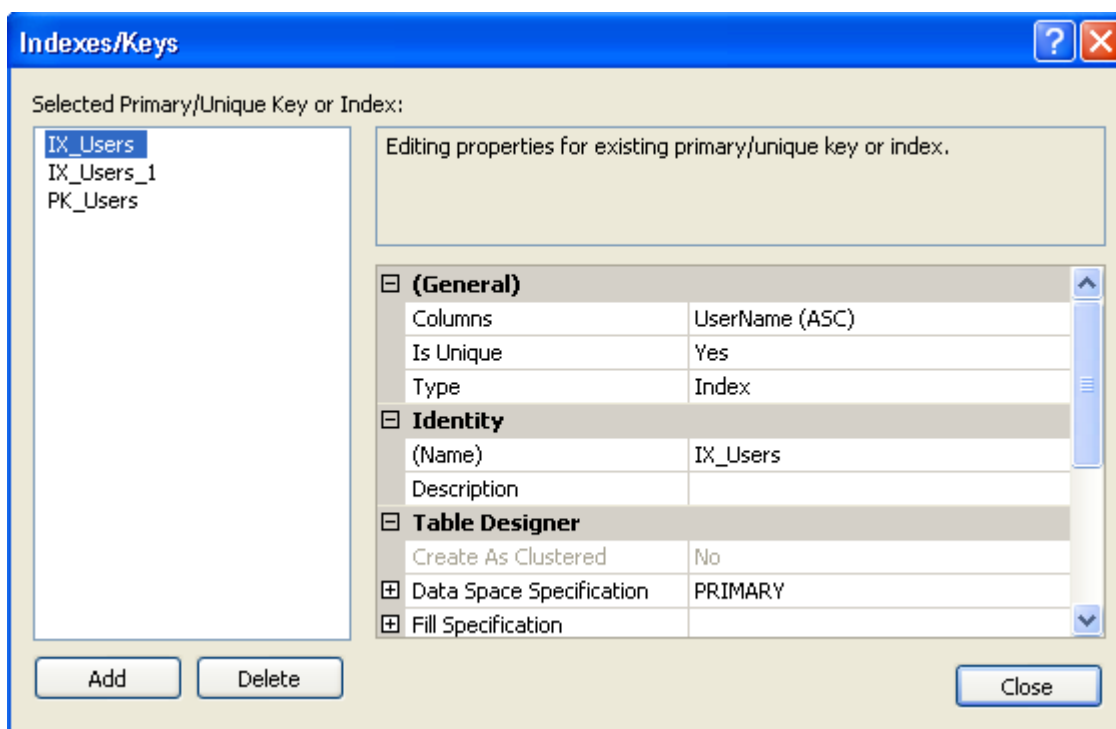
Relationship name:
FK_States_Countries

Primary key table: Countries Foreign key table: States

CountryId	CountryId
-----------	-----------

جدول کاربران:

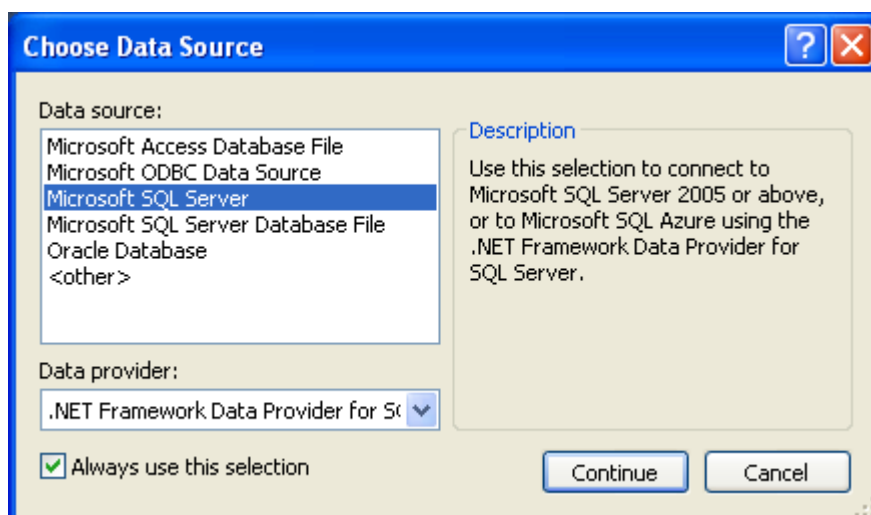
Table - dbo.Users	Table - dbo.States	Table - dbo.Countries
Column Name	Data Type	Allow Nulls
UserId	int	<input type="checkbox"/>
UserName	varchar(14)	<input checked="" type="checkbox"/>
Password	char(10)	<input checked="" type="checkbox"/>
IsActive	bit	<input checked="" type="checkbox"/>
FullName	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>



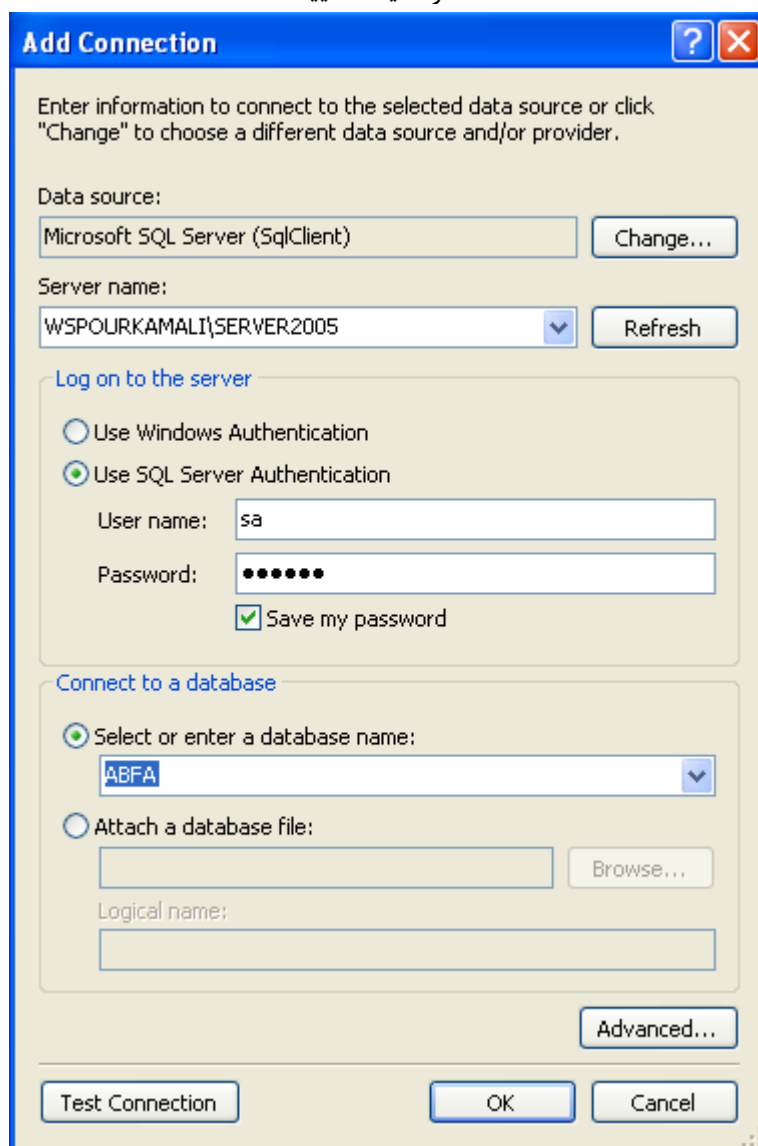
قسمت برنامه نویسی

برای ایجاد کلاس باید ابتدا یک فولدر از نوع فولدرهای باکلاس App_code ایجاد نمایید و سپس داخل آن کلاسها را ایجاد کنید. در این مسیر نیز یک فولدر به نام DAL که مخفف Data Access Layer است ایجاد کنید و لایه Data layer را در اینجا ایجاد می‌کنیم. اسم دیگر DataSet، InMemoryDatabase است و این شی تناظر یک به یک با DataSet دارد. به جای Column، DataColumn، و به جای Relation، DataRelation، و ... داریم. این دو کاملاً با هم Synch است.

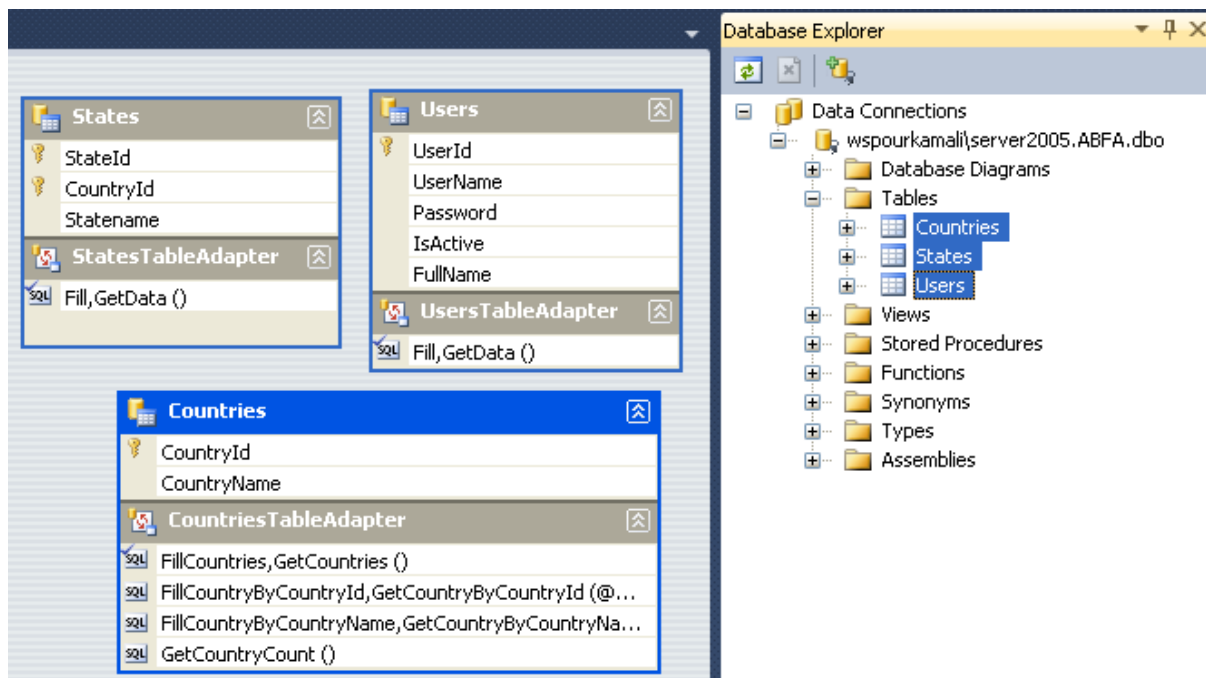
به قسمت connecto to Database → Tools رفته و Microsoft SQL Server را انتخاب نمایید.



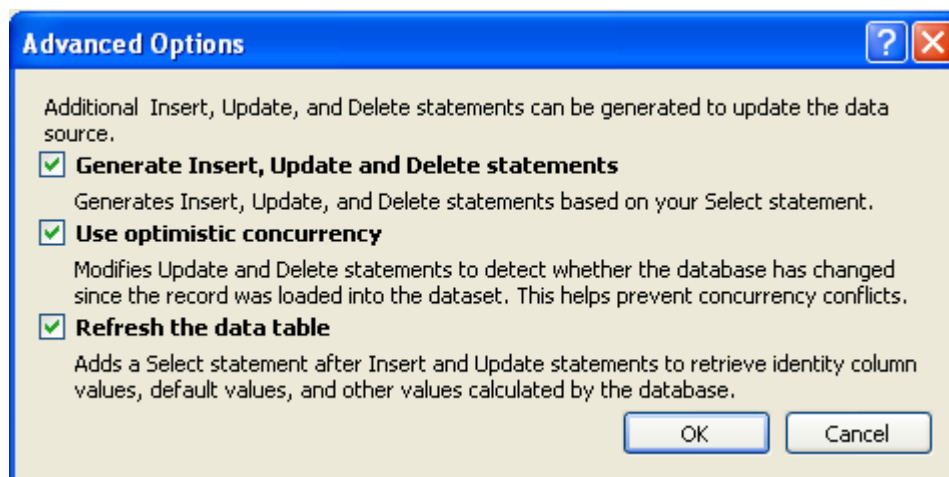
در صفحه ذیل نام سرور و user name و Password و ... را وارد نمایید. حتماً تیک مربوط به save my password را انتخاب نمایید و در نهایت حتماً دکمه test connection را کلیک نمایید.



سپس در قسمت Server Explorer و یا database explorer در نسخه express مشخصات connection تعریف شده را می بینید و بعد با drag & drop نام جداول و View هایی را که می خواهیم استفاده کنیم را انتخاب می کنیم و به صفحه data set می کشیم. و اگر relation ای ایجاد شد، آن را حذف می کنیم.



به ازای هر Table ، یک DataTable و یک TableAdapter ساخته می شود. در DataTable به ازای هر فیلد، یک Property با همان نام درست می شود. در Microsoft TableAdapter دو تابع به صورت اشانسیون به ما می دهد. این دو تابع به صورت خلاصه تمام فیلدها و رکوردها را بر می گرداند و هر دو دقیقاً یک کار می کند. به اولین متد که خود ماکروسافت به ما می دهد، متد Master می گویند و به متدهایی که ما اضافه می کنیم، متد Slave می گویند و بعداً فرق آنها را خواهیم گفت. در لایه Business و UI درگیر query نخواهید شد و آن را در لایه dataset مطرح می کنید. گفتیم تابع اول که در TableAdapter می آید، Master است. پس در آن checkbox های باحال در قسمت Advanced option را انتخاب می کنیم (چک می زنیم).



اسم توابع را در پروژه‌های بزرگ عوض نکنید ولی در مثال ما که کوچک است آن را به FillCountries و GetCountries تغییر نام دهید.

فرق توابع FillData و GetData این است که : FillData این دو پارامتر را به صورت ورودی می‌گیرد و پر می‌کند و GetData این دو پارامتر را به عنوان پارامتر خروجی می‌گیرد و پر می‌کند.

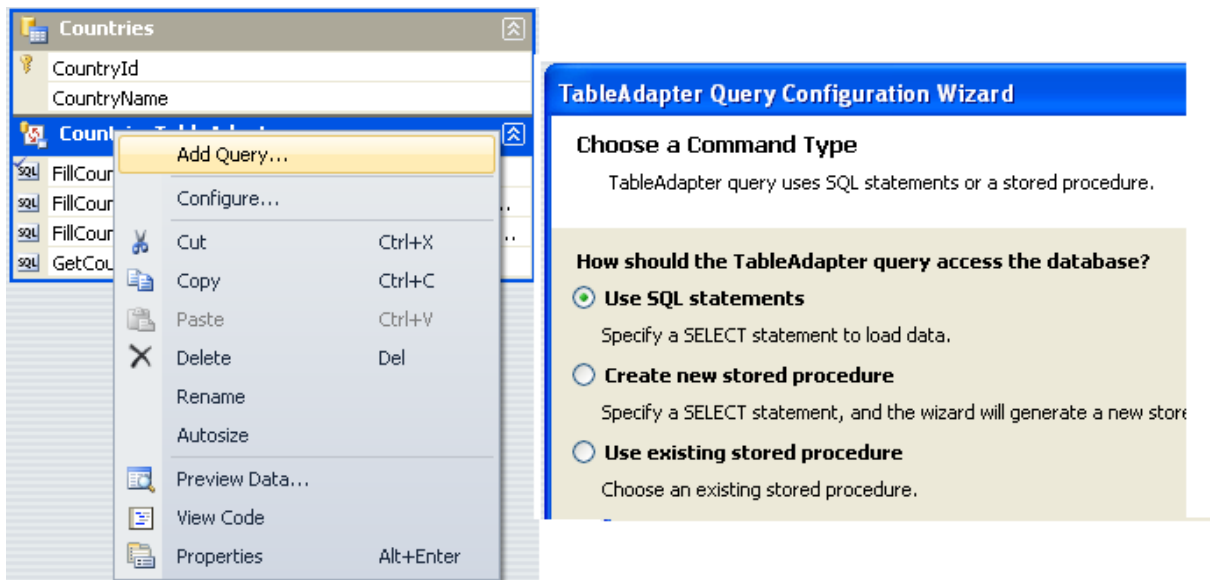
Next را می‌زنید باید تمام تیکها بخورد وگرنه مشکلی پیش آمده است.

جدول Primary key نداشته باشد و یا اینکه فیلدی که Primary key است را در مجموعه فیلدهای خود نیاورده باشید. (Visual Studio که خودش همه فیلدها را می‌آورد، پس کی این اتفاق می‌افتد. ما که روی هر table فقط یک TableAdapter و datatable نداریم، شاید به دلیل مسائل امنیتی و ... ما بعضی از فیلدها را انتخاب کنیم.

مرحله بعد به ازای همه primary key ها و alternate key ها متد می‌نویسیم. چون لازم می‌شود. یعنی پیش می‌آید در این پروژه موردی پیش بیاید، که نام کشور را بدهند و من بخواهم رکورد کشور را به دست آورم و یا اینکه کد کشور را بدهند و من رکورد کشور را پیدا کنم. بقیه متدها را هم بنا به نیاز بعداً اضافه می‌کنم.

می‌خواهم یک تابع بنویسم که در آن کد کشور را بدهم و رکورد آن کشور را برگردانم.

Add query را می‌زنم و می‌توانم query عادی را انتخاب کنم و یا stored procedure و ... که من همان query عادی را انتخاب می‌کنم. چون می‌خواهم همینجا query را بنویسم.



بعد هم بین نوع select return value و select return records و insert , delete , update ، مورد select return records را انتخاب می‌کنم، چون تابع من یک عدد یا یک رشته را بر نمی‌گرداند و بلکه یک رکورد را بر می‌گرداند..

TableAdapter Query Configuration Wizard

Choose a Query Type

Choose the type of query to be generated

What type of SQL query would you like to use?

SELECT which returns rows

Returns one or many rows or columns.

SELECT which returns a single value

Returns a single value (for example, Sum, Count, or any other aggregate function).

UPDATE

Changes existing data in a table.

DELETE

Removes rows from a table.

INSERT

Adds a new row to a table.

اگر تابع اصلی شما روی 17 تا آیتم است، باید حتماً بقیه query ها را حتماً روی همه 17 آیتم بگیرید، هر چه می خواهید شرط بگذارید و ... ولی تعداد آیتمها را کم و یا زیاد نکنید.

Where CompanyId = @X

این شرط query است.

نکته: روش چاق کردن query مشکل sql injection دارد. روش sql parametric که الان نشان می دهیم و در 2 net. به بعد آمد اصلاً خطر sql injection ندارد و sql injection free است.

نکته: در این حالت و استفاده از روش پارامتریک یکی از توسعه هایی که می کنند این است که اسم پارامتر را سعی کنید، با نام فیلد یکسان بدهید. این حتمی نیست و بهتر است چون باعث می شود بهتر detect شود.

نکته: نام متدها را به صورت استاندارد تعریف کنید تا بعداً قابل شناسایی باشد، مثلاً نام این تابع را FillCountryByCountryID انتخاب می کنیم.

در مورد select ها دست به ترکیب نکنید، ولی در مورد delete , update , insert یا آنهایی که scalar بر می گرداند، باید تعداد فیلدها را کم و یا زیاد نکنید.

در خصوص استفاده از دستور count باید یک alias تعریف کنید، بهتر است از نام RecordCount استفاده کنید.

به عنوان نمونه TableAdapter های تعریف شده بر روی جدول Countries به شرح ذیل می باشد:

Commant Type:	SQL Statement
Query Type:	Select with Return Rows
Method or Function Name :	FillCountries - GetCountries
SQL Select Statement:	<u><i>SELECT CountryId, CountryName</i></u> <u><i>FROM Countries</i></u> <u><i>ORDER BY CountryName</i></u>

Commant Type:	SQL Statement
Query Type:	Select with Return Rows
Method or Function Name :	FillCountryByCountryId - GetCountryByCountryId
SQL Select Statement:	<u><i>SELECT CountryId, CountryName</i></u> <u><i>FROM Countries</i></u> <u><i>WHERE (CountryId = @CountryId)</i></u>

Commant Type:	SQL Statement
Query Type:	Select with Return Rows
Method or Function Name :	FillCountryByCountryName - GetCountryByCountryName
SQL Select Statement:	<u><i>SELECT CountryId, CountryName</i></u> <u><i>FROM Countries</i></u> <u><i>WHERE (CountryName = @CountryName)</i></u>

Commant Type:	SQL Statement
Query Type:	Select which returns a single value
Method or Function Name :	GetCountryCount
SQL Select Statement:	<u><i>SELECT COUNT(*) AS RecordCount</i></u> <u><i>FROM Countries</i></u>

هر وقت کارتان با MyDataSet تمام شد، باید سه کار انجام دهید:

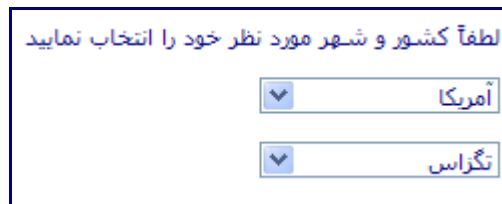
- save - 1
- close - 2
- rebuild - 3

جلسه دهم

با استفاده از ASP.net و ADO.NET 2 دو روش برای کار با بانک اطلاعاتی داریم. یا به صورت wizard است که برای پر کردن combo Box و ... است و یک روش دستی است.

کار با بانک اطلاعاتی با کمک wizard

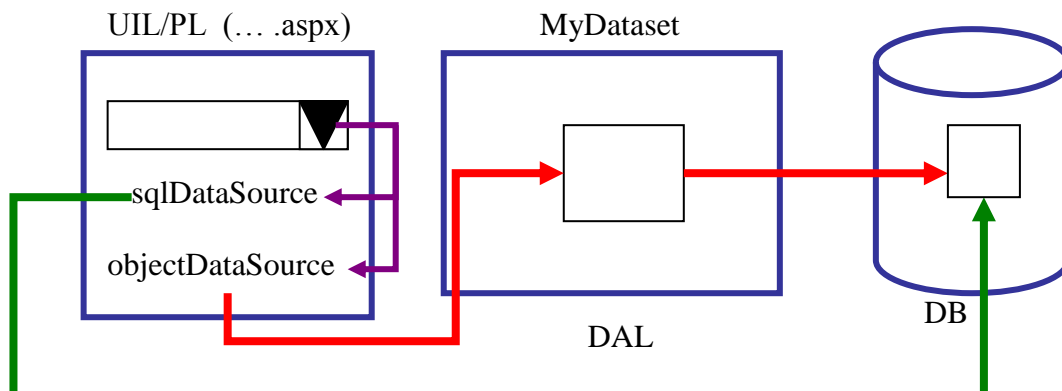
فرض کنید که می‌خواهیم یک فرم داشته‌باشیم که در آن فرم دو عدد combo box وجود داشته باشد، در اولی نام کشورها موجود باشد و در دومی نام شهرهای مربوط به آن کشور. با انتخاب هر کشور باید نام شهرهای موجود در آن کشور در combo box دوم لیست شود.



پس یک فرم با نام frmMain درست می‌کنم و دو عدد combo box یا همان drop down list می‌گذارم. Combo box اولی را ddlCountries می‌گذارم و دومی را ddlStates می‌گذارم.

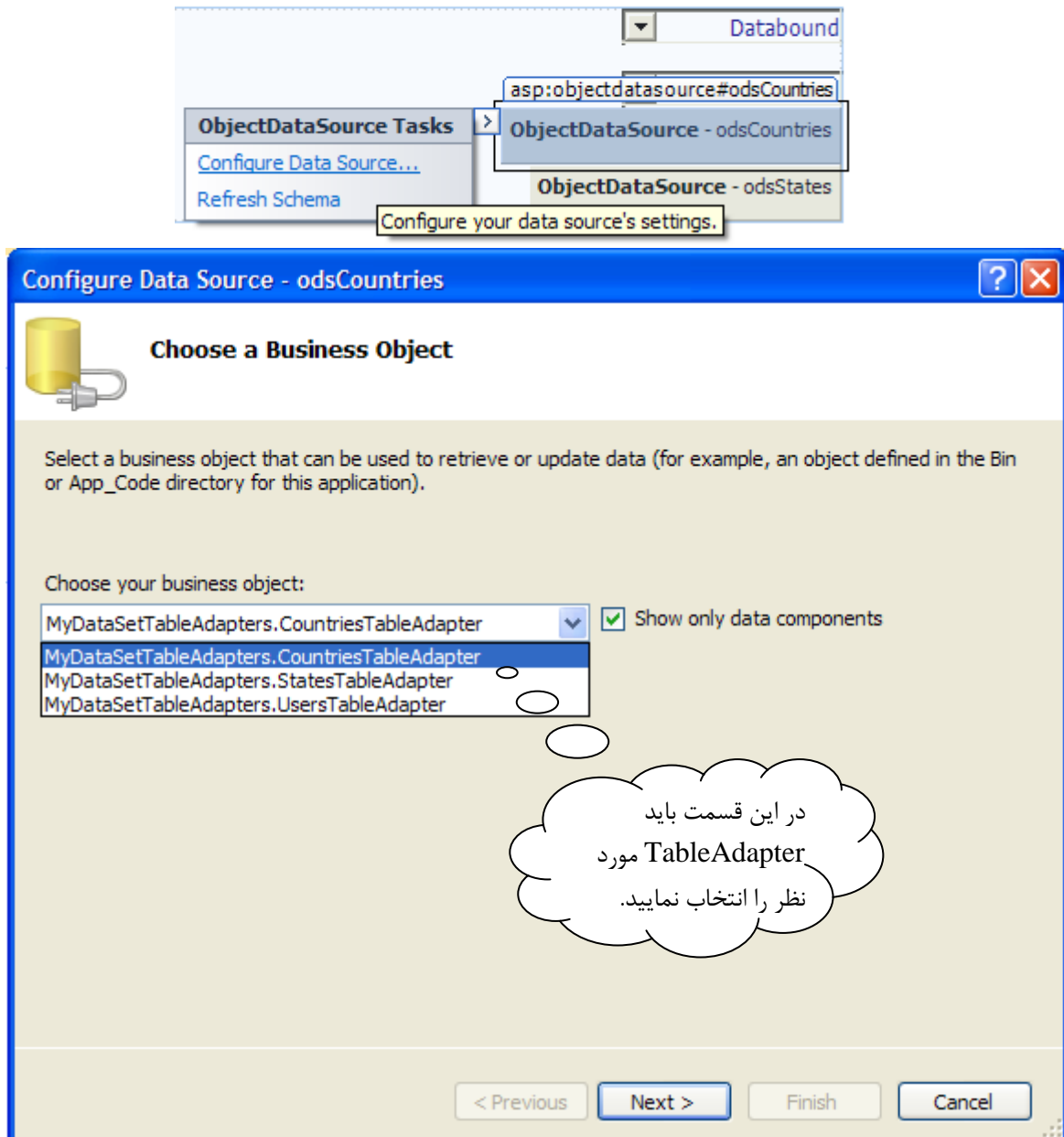
در ضمن می‌خواهم، وقتی که طرف combo box کشور را باز می‌کند، شهرهای آن استان در combo box دوم ظاهر شود، پس باید بر روی این combo box (ddlCountries) عمل post back اتفاق بیافتد، پس Auto postback ان را true می‌کنم.

پس الان در پشت صحنه چه چیزهایی داریم.



اگر من از sqlDataSource استفاده کنم اصلاً به لایه Data Access Layer من کاری ندارد، پس تمام کارهای من و تعریف DAL کشک می‌شود، پس به جای آن از objectDataSource استفاده می‌کنم که با DAL ارتباط دارد. پس دو عدد ObjectDataSource در صفحه قرار می‌دهم و نام یکی را odsCountries و نام دیگری را odsStates می‌گذارم و آنها را configure می‌کنیم.

لطفاً کشور و شهر مورد نظر خود را انتخاب نمایید



در مرحله بعد باید method های مربوط به Select, Insert, Delete, Update را تعریف نمایید. از dataSource بهتر است برای insert , delete , update استفاده نکنید و آنها را none کنید. فقط برای select استفاده کنید.

Configure Data Source - odsCountries

Define Data Methods

SELECT UPDATE INSERT DELETE

Choose a method of the business object that returns data to associate with the SELECT operation. The method can return a DataSet, DataReader, or strongly-typed collection.

Example: GetProducts(Int32 categoryId), returns a DataSet.

Choose a method:

- GetCountries(), returns CountriesDataTable
- GetCountryByCountryId(Int32 CountryId), returns CountriesDataTable
- GetCountryByCountryName(String CountryName), returns CountriesDataTable

< Previous Next > Finish Cancel

Configure Data Source - odsCountries

Define Data Methods

SELECT UPDATE INSERT DELETE

Choose a method of the business object to associate with the UPDATE operation. The method should accept a parameter for each property of the data object, or a single parameter which is the data object to update.

Examples: UpdateProduct(Product p), or UpdateProduct(Int32 productID, String name, Double price)

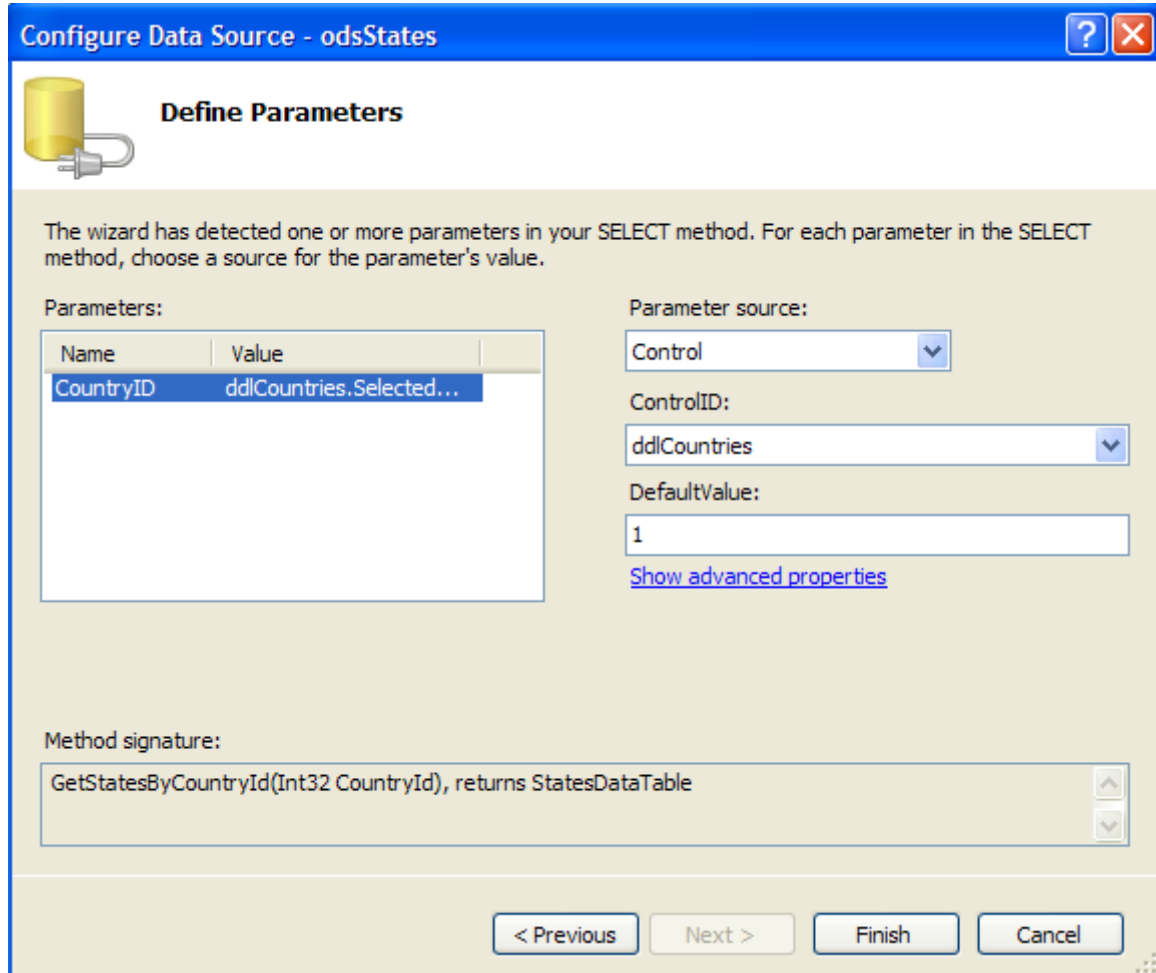
Choose a method:

- (None)

Method signature:

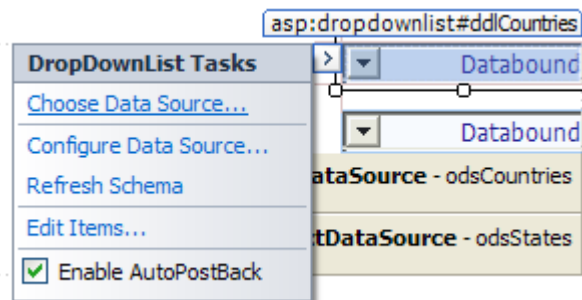
< Previous Next > Finish Cancel

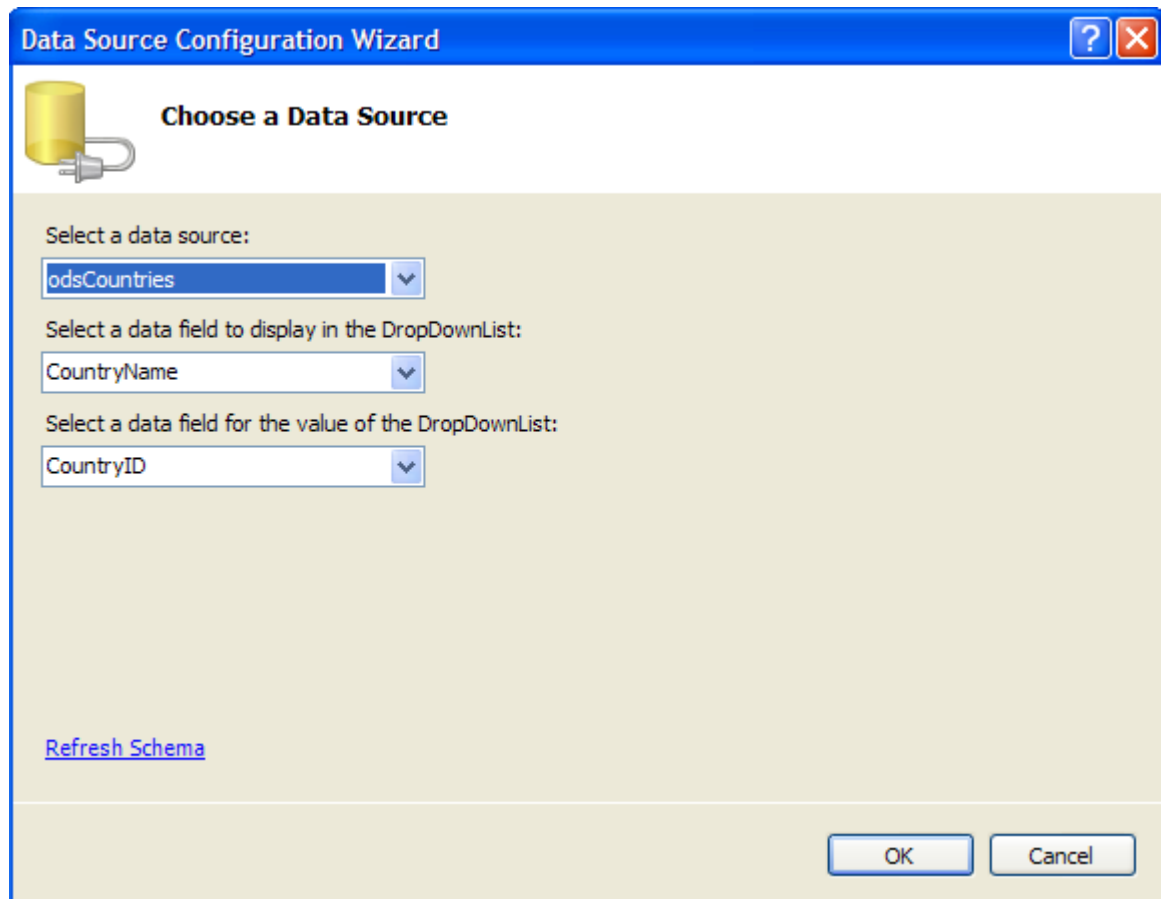
موقعی که query مربوط به select را انتخاب کردید، اگر query پارامتر نداشته باشد، Finish ظاهر می‌شود ولی اگر پارامتریک باشد، Next ظاهر می‌شود و باید پارامتر خود را هم انتخاب کنید. مثلاً در خصوص odsStates وقتی که query مربوطه به Select یا همان GetStatesByCountryID را انتخاب می‌کنید، چون این متد، پارامتریک می‌باشد، در مرحله بعد در خصوص تعریف پارامتر سوال می‌نماید:



در این مثال، چون می‌خواهیم ContryID را از combo box دیگر بخوانیم، پس Parameter Source را control انتخاب می‌کنیم و ControlID را هم ddlCountries انتخاب می‌کنیم.

بعد باید کنترلها را به data source ها bind نمایید.





نکته: در فرم یک تگ div قرار دهید. و در آن یک متن بنویسید، مثلاً «لطفاً کشور و شهر مورد نظر خود را انتخاب نمایید». برای body نیز فرمت و style مورد نظر خود را بنویسید.

```
<style type="text/css" >
  body
  {
    font-family:Tahoma;
    font-size:12px;
    color:Navy;
    text-align:right;
    direction:rtl;
  }
</style>
```

CSS به معنی Cascade Style sheet است. معنی آن است که وقتی یک style تعریف کنید، در سطوح زیری نیز منتقل می‌شود. ولی این style تا یک حدی جلو می‌رود و در یک جایی می‌ایستد، یعنی چه. مثلاً اگر رنگ فونت را در body، رنگ سورمه‌ای قرار دهید، div، table و ... رنگش قرمز می‌شود. ولی کجا متوقف می‌شود. در تگ select متوقف می‌شود. و همانطور که می‌دانید، checkbox، drop down، combo box و ... به تگ select، render می‌شوند، پس باید جداگانه برای آنها style تعریف کنید.

```
select
{
  font-family:Tahoma;
  font-size:12px;
  color:Navy;
  text-align:right;
  direction:rtl;
}
```

حالا فرض کنید، که یک صفحه سنگینی است و وقتی combo box را کلیک می کنید، محتویات صفحه می آید و می رود. این خوشایند نیست. پس Ajax می نویسیم. پس باید ناحیه مورد نظر را در update panel بگذارید و می توانید فقط تکیه ای را که می خواهید در این تگ بگذارید. می توان تگ مربوط به update panel را در هر جایی قرار داد.

روش وصل شدن به بانک اطلاعاتی و کار کردن با بانک اطلاعاتی به صورت دستی

دوباره به صفحه Login بر می گردیم. در این صفحه اگر شرایط درست است، اجازه Login می دهد.

در این دوره چهار بار وارد صفحه Login می شویم.

1 - موقع Validator

2 - موقع ADO.Net

3 - موقع Session , Application

4 - موقع Authorization , Authentication

وقتی که می خواهیم Login کنیم، username و Password می دهیم. برای Login نیاز به متد FillUserByUsername داریم و این متد متعلق به UsersTableAdapter است. و آن هم مربوط به UsersDataTable است.

پس با این حساب باید مراحل ذیل را انجام دهیم، تا بتوانیم با بانک اطلاعاتی کار کنیم.

1 - اول یک UsersDataTable را new کنید.

2 - بعد TableAdapter را باید new کنید.

3 - باید متد FillUserByUsername را صدا کنید.

معادل این دستورات به صورت ذیل می باشد:

```
1- MyDataSet.UsersDataTable oUsersDataTable =
    new MyDataSet.UsersDataTable();

2- MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
    new MyDataSetTableAdapters.UsersTableAdapter();

3- oUsersTableAdapter.FillUserByUsername(oUsersDataTable,
    txtUserName.Text);
```

پس از آن باید کنترلرها و مقایسه های لازم صورت گیرد:

```
if (oUsersDataTable.Count != 1)
//if (oUsersDataTable.Count == 0)
{
    DisplayErrorMessage("Username and/or password is not correct!");
    return;
}

MyDataSet.UsersRow oUsersRow =
    oUsersDataTable[0];

//if (oUsersRow.Password.Trim() != txtPassword.Text.Trim())
if(string.Compare(oUsersRow.Password.Trim(), txtPassword.Text.Trim(), false) != 0)
```

```

{
    DisplayErrorMessage("Username and/or password is not correct!");
    return;
}

if (oUsersRow.IsUserActive == false)
{
    DisplayErrorMessage("You should not login at this time, Please contact support!");
    return;
}

```

اول بررسی می‌کنیم که آیا رکوردی با username مورد نظر وجود دارد یا خیر؟

نکته: همیشه در برنامه‌نویسی موقع نوشتن شرط if نقیض آن چیزی که هست، را بنویسید. مثلاً وجود یک username برای ما منطقی است، پس می‌گوییم اگر تعداد userName مورد نظر مخالف 1 بود.

نکته: همیشه باید پیغام گنگ به user دهید، تا از پیغام ما برای hack کردن استفاده نکند. پس در صورتی که تعداد رکورد یافت شده مخالف یک بود، باید پیغام «خطا در شناسه کاربری یا کلمه عبور» بدهد.

در مرحله بعد باید کلمه عبور را مقایسه کنیم. برای مقایسه کردن از operation مساوی استفاده نکنید. ما مبحثی به نام code analyzer داریم. نباید برای مقایسه password از operatoion مساوی استفاده کنید، بلکه باید از دستور Compare استفاده کنید. در این دستور هم اگر پارامتر سوم false باشد، به صورت Case Sensitive مقایسه می‌کند.

در مرحله بعد نیز باید active بودن کاربر را چک کنیم. پس باید سه شرط ذیل را به ترتیب چک کنیم و در هر مرحله در صورت درست نبودن شرط باید یک پیغام خطای گنگ به کاربر داده شود و با استفاده از دستور return تابع خاتمه یابد.

```

<%@ page language="C#" autoeventwireup="true" codefile="Login.aspx.cs"
inherits="Login" %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title>Login</title>
  </head>
  <body>
    <form id="frmData" runat="server">
      <asp:label id="lblMessage" runat="server" />

      <fieldset>
        <legend>Login</legend>

        <asp:label id="lblUserName" runat="server" text="Username"
          />
        <br />
        <asp:textbox id="txtUserName" runat="server" maxlength="20"
          autocompletetype="Disabled" accesskey="U" />
        <br />
        <br />
        <asp:label id="lblPassword" runat="server" text="Password"
          />
        <br />
      </fieldset>
    </form>
  </body>
</html>

```

```

        <asp:textbox id="txtPassword" runat="server"
            textmode="Password" maxlength="20"
            autocomplete="Disabled" accesskey="P" />
        <br />
        <br />
        <asp:button id="btnSubmit" runat="server" text="Login"
            accesskey="L" onclick="btnSubmit_Click" />
        <input type="reset" value="Reset" accesskey="R" />
    </fieldset>
</form>
</body>
</html>

```

```

public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        lblMessage.Visible = false;

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            MyDataSet.UsersDataTable oUsersDataTable =
                new MyDataSet.UsersDataTable();

            MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
                new MyDataSetTableAdapters.UsersTableAdapter();

            oUsersTableAdapter.FillUserByUserName(oUsersDataTable,
                txtUserName.Text);

            if (oUsersDataTable.Count != 1)
            //if (oUsersDataTable.Count == 0)
            {
                DisplayErrorMessage("Username and/or password is not
                    correct!");
                return;
            }

            MyDataSet.UsersRow oUsersRow =
                oUsersDataTable[0];

            //if (oUsersRow.Password.Trim() != txtPassword.Text.Trim())
            if(string.Compare(oUsersRow.Password.Trim(),
                txtPassword.Text.Trim(), false) != 0)
            {
                DisplayErrorMessage("Username and/or password is not
                    correct!");
                return;
            }
        }
    }
}

```

```

    }

    if (oUsersRow.IsUserActive == false)
    {
        DisplayErrorMessage("You should not login at this time,
            Please contact support!");
        return;
    }

    DisplayInformationMessage(string.Format("Welcome {0}!",
        txtUserName.Text));
}

private void DisplayErrorMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Red;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}

private void DisplayInformationMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Blue;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}
}

```

در صفحه registration هم باید، از متد FillUserByUserName استفاده کنم. ولی در اینجا در صورت پیدا کردن رکورد، مشکل دارد، چون نباید user name تکراری وجود داشته باشد، پس باید در صورت وجود داشتن رکورد پیغام دهد.

```

<%@ page language="C#" autoeventwireup="true" codefile="Registration.aspx.cs"
inherits="Registration" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title>Registration</title>
  </head>
  <body>
    <form id="frmData" runat="server">
      <asp:label id="lblMessage" runat="server" />

      <fieldset>
        <legend>Registration</legend>

        <asp:label id="lblUserName" runat="server" text="Username"

        <br />
        <asp:textbox id="txtUserName" runat="server" maxlength="20"
          autocomplete="Disabled" accesskey="U" />
        <br />

```



```

        <br />
        <asp:label id="lblPassword" runat="server" text="Password"
/>
        <br />
        <asp:textbox id="txtPassword" runat="server"
            textmode="Password" maxlength="20"
            autocomplete="Disabled" accesskey="P" />
        <br />
        <br />
        <input type="reset" value="Reset" accesskey="R" />
        <asp:button id="btnSubmit" runat="server" text="Submit"
            accesskey="S" onclick="btnSubmit_Click" />
    </fieldset>
</form>
</body>
</html>

```

```

public partial class Registration : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        lblMessage.Visible = false;

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            MyDataSet.UsersDataTable oUsersDataTable =
                new MyDataSet.UsersDataTable();

            MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
                new MyDataSetTableAdapters.UsersTableAdapter();

            oUsersTableAdapter.FillUserByUserName(oUsersDataTable,
                txtUserName.Text);

            if (oUsersDataTable.Count != 0)
            {
                DisplayErrorMessage("Username is already exist!, please
                    choose another one...");
                return;
            }

            MyDataSet.UsersRow oUsersRow =
                oUsersDataTable.NewUsersRow();

            // oUsersRow.UserID; //AUTO INCREMENT!
            oUsersRow.IsUserActive = true;
            oUsersRow.UserName = txtUserName.Text;
            oUsersRow.Password = txtPassword.Text;

```

```

        oUsersDataTable.AddUsersRow(oUsersRow);

        oUsersTableAdapter.Update(oUsersDataTable);

        DisplayInformationMessage("Your registraion was completed
            successfully...");
    }
}

private void DisplayErrorMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Red;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}

private void DisplayInformationMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Blue;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}
}

```

صفحه update profile به صورت ذیل می باشد:

```

<%@ page language="C#" autoeventwireup="true" codefile="UpdateProfile.aspx.cs"
inherits="UpdateProfile" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title>Update Profile</title>
  </head>
  <body>
    <form id="frmData" runat="server">
      <asp:label id="lblMessage" runat="server" />

      <fieldset>
        <legend>Update Profile</legend>

        <asp:label id="lblUserName" runat="server" text="Username"
          />
        <br />
        <asp:textbox id="txtUserName" runat="server" maxlength="20"
          autocompletetype="Disabled" accesskey="U" />
        <br />
        <br />
        <asp:label id="lblPassword" runat="server" text="Password"
          />
        <br />
        <asp:textbox id="txtPassword" runat="server" maxlength="20"
          autocompletetype="Disabled" accesskey="P" />
        <br />
        <br />
      </fieldset>
    </form>
  </body>
</html>

```

```

        <input type="reset" value="Reset" accesskey="R" />
        <asp:button id="btnSubmit" runat="server" text="Submit"
            accesskey="S" onclick="btnSubmit_Click" />
    </fieldset>
</form>
</body>
</html>

```

```

public partial class UpdateProfile : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        lblMessage.Visible = false;

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
        MyDataSet.UsersDataTable oUsersDataTable =
            new MyDataSet.UsersDataTable();

        MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
            new MyDataSetTableAdapters.UsersTableAdapter();

        oUsersTableAdapter.FillUserByUserId(oUsersDataTable, 1);
        // oUsersTableAdapter.FillUserByUserId(oUsersDataTable,
        (int)Session["UserID"]);

        if (oUsersDataTable.Count != 1)
        {
            DisplayErrorMessage("There is no such a user any more!");
            return;
        }

        MyDataSet.UsersRow oUsersRow =
            oUsersDataTable[0];

        txtUserName.Text = oUsersRow.UserName;
        txtPassword.Text = oUsersRow.Password.Trim();
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            MyDataSet.UsersDataTable oUsersDataTable =
                new MyDataSet.UsersDataTable();

            MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
                new MyDataSetTableAdapters.UsersTableAdapter();

            oUsersTableAdapter.FillUserByUserId(oUsersDataTable, 1);
            // oUsersTableAdapter.FillUserByUserId(oUsersDataTable,
            (int)Session["UserId"]);

            if (oUsersDataTable.Count != 1)

```

```

    {
        DisplayErrorMessage("There is no such a user any more!");
        return;
    }

    MyDataSet.UsersRow oUsersRow =
        oUsersDataTable[0];

    oUsersRow.UserName = txtUserName.Text;
    oUsersRow.Password = txtPassword.Text;

    oUsersTableAdapter.Update(oUsersDataTable);

    DisplayInformationMessage("Your profile was update
        successfully...");
}
}

private void DisplayErrorMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Red;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}

private void DisplayInformationMessage(string message)
{
    lblMessage.Visible = true;
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Blue;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}
}

```

صفحه حذف کاربر به صورت ذیل می باشد:

```

<%@ page language="C#" autoeventwireup="true" codefile="DeleteUser.aspx.cs"
inherits="DeleteUser" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title>Delete User</title>
  </head>
  <body>
    <form id="frmData" runat="server">
      <asp:label id="lblMessage" runat="server" />

      <fieldset>
        <legend>Delete User</legend>

        <asp:label id="lblUserName" runat="server" text="Username"
          />
        <br />
        <asp:textbox id="txtUserName" runat="server" maxlength="20"
          autocomplete="Disabled" accesskey="U" />
      </fieldset>
    </form>
  </body>
</html>

```

```

        <br />
        <br />
        <input type="reset" value="Reset" accesskey="R" />
        <asp:button id="btnSubmit" runat="server" text="Delete
            User" accesskey="D" onclick="btnSubmit_Click" />
    </fieldset>
</form>
</body>
</html>

```

```

public partial class DeleteUser : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }

    protected void btnSubmit_Click(object sender, System.EventArgs e)
    {
        if (Page.IsValid)
        {
            // Solution (1)
            MyDataSet.UsersDataTable oUsersDataTable =
                new MyDataSet.UsersDataTable();

            MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
                new MyDataSetTableAdapters.UsersTableAdapter();

            oUsersTableAdapter.FillUserByUserName(oUsersDataTable,
                txtUserName.Text);

            if (oUsersDataTable.Count != 1)
            {
                DisplayErrorMessage("There is no such a user with this
                    username!");
                return;
            }

            MyDataSet.UsersRow oUsersRow = oUsersDataTable[0];

            oUsersRow.Delete();

            oUsersTableAdapter.Update(oUsersDataTable);

            string strMessage =
                string.Format("The user with this username [{0}] was
                    deleted successfully...", txtUserName.Text);

            DisplayInformationMessage(strMessage);
            // /Solution (1)

            // Solution (2)
            // Note: It's better!

```

```

        //MyDataSetTableAdapters.UsersTableAdapter oUsersTableAdapter =
        //    new MyDataSetTableAdapters.UsersTableAdapter();

        //int intAffectedRows =
        //    oUsersTableAdapter.DeleteUserByUserName(txtUserName.Text);

        //if (intAffectedRows == 0)
        //{
        //    DisplayErrorMessage("There is no such a user with this
        //        username!");
        //    return;
        //}

        //string strMessage =
        //    string.Format("The user with this username [{0}] was deleted
        //        successfully...", txtUserName.Text);

        //DisplayInformationMessage(strMessage);
        // /Solution (2)
    }
}

private void DisplayErrorMessage(string message)
{
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Red;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}

private void DisplayInformationMessage(string message)
{
    lblMessage.Font.Bold = true;
    lblMessage.ForeColor = System.Drawing.Color.Blue;

    lblMessage.Text = string.Format("{0}<br /><br />", message);
}
}

```

برای delete کردن تعداد زیاد user باید یک tableAdapter جدا بنویسید.
دستور insert و update و delete یک عدد به نام affected row بر می گرداند.

نکته: در قسمت Tools و execution manager یک سری Component به صورت live و on-line در اختیار شما قرار

می دهد. بعضی از این ابزارها عبارتند از:

- Productivity power tools - 1
- Indent guides - 2
- Highlight all occurrence of selected - 3

جلسه یازدهم

بحث Web Application

در web application با یک فایل با نام Global.asax مواجه هستید. اسم و پسوند این فایل رزرو است. سه فایل در کل visual studio وجود دارد که نام و پسوند آن رزرو است:

- Web.config
- Global.asax
- Web.site

خصوصیات Global.asax

- 1 - اولاً وقتی که در پروژه ایجاد نمودید، دیگر نمی توان یکی دیگر اضافه کرد.
- 2 - 7 تا event دارد که پنج event آن را ما آموزش می بینیم که چهار تای آن را امروز بررسی می کنیم:

- a. Application Start
- b. Session Start
- c. Application Error
- d. Session End
- e. Application End

3 - چند متغییر نیز برای آن تعریف شده است.

Session Start & Applicatio Start

زمانی که یک سایت را در جایی upload می کنید، اولین فردی که به هر صفحه ای از سایت وصل شود، application start می شود و به ازای این درخواست user پشت سر آن session نیز start می شود. اگر فرد دیگر وارد شود، یک session دیگر start می شود.

Session End & Applicatio End

: Session End

به دو روش صورت می گیرد:

- اگر user 20 دقیقه کار نکند، session خودکار قطع می شود.
- به طور دستی clear شود.

:Application End

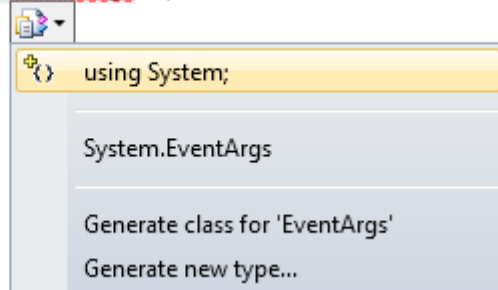
آخر شب است و افراد یکی یکی خارج می شوند، نفر آخر که خارج شد، 20 دقیقه بعد session end می شود، و بعد از end شدن آخرین session، application نیز end می شود.

در واقع پروسه start و end شدن session و application به این صورت انجام می گیرد.

یادآوری و مرور ایجاد یک برنامه تحت web

1. create folder CSS
 - 1.1 create StyleSheet Site.css
2. create folder MasterPages for all master pages in project
 - 2.1 name of first master page that creat must be Site.master
 - 2.1.1 format hole document
 - 2.1.2 rename ContentPlaceHolder id="head" to id="cphHead"
 - rename ContentPlaceHolder id="ContentPlaceHolder1" to id="cphMain"
 - rename form id="form1" to id="frmMain"
3. create folder Images for our images
 - 3.1 create folder Layout for our layout images
4. create a page Default.aspx that uses of Site.master
 - 4.1 give a titlle to this page : title="Home"
 - 4.2 rename ID="Content1" to id="cntHead"
 - 4.3 rename ID="Content2" to id="cntMain"
 - 4.4 in behind code of master page and all page do these works:
 - 4.4.1 delete all of using System in the top of page code
 - 4.4.2 use CTRL+ on the word EventArgs and then select System.EventArgs

Page_Load(object sender, EventArgs e)



add this code: 4.4.3

```

1. protected void Page_Load(object sender, System.EventArgs e)
2.     {
3.         if (Page.IsPostBack == false)
4.         {
5.             Initialize();
6.         }
7.     }
8.
9. private void Initialize()
10.    {
11.    }

```

use CTRL+ .

initialize();

Generate method stub for 'initialize' in

add new item : Global Application Class 5

6. معمولاً یک فولدر به نام images ایجاد می‌کنند و در داخل آن نیز یک sub folder با نام Layout و تصاویر مربوط به Layout را در آن قرار می‌دهند، تا در صورتی که بخواهند تغییری در Layout سایت ایجاد نمایند، به دنبال تصاویر آن نگردند.

نکته: وقتی که Global Application Class را به پروژه خود add کنید و اگر بار دیگر بروید و یک new item را add کنید، دیگر حتی نام آن را هم نمی‌بینید.

در این کلاس چهار تابع وجود دارد. همان چهار تابعی که در بالا بیان کردیم. البته اکثر افراد اینها را به نام event می‌شناسند ولی در واقع تابعی است که قبل از event ... , start application صدا می‌شود.

ترتیب این پنج تابع مهم نیست ولی اگر ترتیب اینها را به صورت منطقی بنویسید، قابلیت فهم آن بالا می‌رود. همان ترتیبی که در بالا برای این پنج تابع بیان شد.

متغیرهایی در سطح Application , session, ViewState , cookie

ما متغیرهایی در سطح Application , session, ViewState , cookie داریم. معماری این متغیرها کاملاً متفاوت از محیط .net است و در واقع از ASP کلاسیک به ارث رسیده است.

متغیرهای Application

متغیرهای Application همه global هستند (برای همه user ها) و در کل start شدن تا end شدن Application مقدار آن در تمام صفحات موجود خواهد بود.


✓ مثل تعداد کاربران online در application(site)


تعریف متغیر onlineUserCount در Application_Start(object sender, EventArgs e):

```
Application["OnlineUserCount"] = 0;
```

اصولاً متغیرهای Application از نوع object هستند و نمی‌شود آنها را ++ کرد
//Application["onlineUserCount"]++;

که سه روش برای این کار وجود دارد :

cast : Application["OnlineUserCount"]=(int)Application["OnlineUserCount"] + 1; 

parse : System.Int32.Parse(Application["OnlineUserCount"].ToString()) + 1; 

convert : System.Convert.ToInt32(Application["OnlineUserCount"]) + 1; 

که در تابع Session_start به صورت ذکر شده این متغیر را اضافه می‌کنید.

توصیه: توصیه ماکروسافت این است که هر وقت که می‌خواهید متغیر Application را تغییر دهید، یک بار آن را Lock و بعد onLock کنید تا سرویس صف راه اندازی شود. دلیل آن این است که شاید در یک لحظه تعداد زیادی آدم در سراسر دنیا، به سایت وصل شود و آن موقع عملیات اضافه کردن دچار مشکل می‌شود.

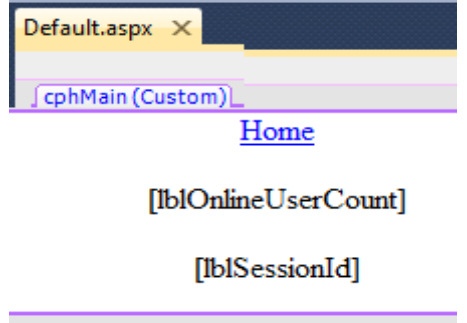
این توصیه غلط نیست ولی ناقص است. هر وقت که می‌خواهید با Content کار کنید و تغییر دهید امکان concurrent بودن برای آن وجود دارد، و آن content توانایی مدیریت concurrent را ندارد، حتماً قبل از دستکاری کردن، آن را Lock کنید. بعنوان مثال فایل های text توانایی concurrent ای را ندارند برای همین از lock و unlock استفاده می‌کنیم.

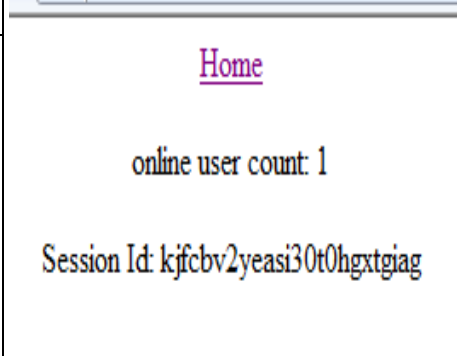
بطور خلاصه : متغیرهای Application ، متغیر Static و فایل‌های text و xml و متنی ... را حتماً باید قبل از استفاده Lock کنید.

در صفحه default، یک HyperLink با نام home بگذارید که navigation url آن "~/default.aspx" باشد و یک Lable با نام lblOnlineUserCount و یک Lable با نام lblSessionId تعریف کنید.

<pre><asp:hyperlink id="HyperLink1" runat="server" navigateurl="~/Default.aspx">Home</asp:hyperlink>

 <asp:label id="lblOnlineUserCount" runat="server"/ >

 <asp:label id="lblSessionId" runat="server"/ ></pre>	
--	--

<p>Default.aspx.cs</p> <pre>protected void Page_Load(object sender, System.EventArgs e) { int intOnlineUserCount =System.Convert.ToInt32 (Application["onlineUserCount"]); lblOnlineUserCount.Text =string.Format ("online user count: {0:##,###0} ", intOnlineUserCount); lblSessionId.Text =string.Format("Session Id: {0} " , Session.SessionID); }</pre>	
--	--

<p>Global.asax</p> <pre>void Application_Start(object sender, EventArgs e) { Application.Lock(); Application["onlineUserCount"] = 0; Application.Unlock(); } void Session_Start(object sender, EventArgs e) { Application.Lock(); Application["onlineUserCount"] = System.Convert.ToInt32(Application["onlineUserCount"])+1; Application.Unlock(); } void Session_End(object sender, EventArgs e) { Application.Lock(); Application["onlineUserCount"] = System.Convert.ToInt32(Application["onlineUserCount"]) - 1; Application.Unlock(); }</pre>

ماجرای session id چیست؟

Session.SessionID

در asp کلاسیک وقتی وارد صفحه‌ای می‌شدید، در Ram server یک عدد درست می‌شد و این عدد به صورت یک Cooki لطیف که هم لطیف است و هم یک خط کد برای آن نمی‌نوشتید، بر روی دستگاه شما می‌نشست و وقتی که شما از یک صفحه به صفحه دیگر می‌رفتید، خود سرور می‌رفت بررسی می‌کرد که آیا Cooki لطیفه هست یا خیر؟ اگر فایل وجود نداشت و یا expire شده بود، دوباره یک Session_start دیگر صدا می‌زد و گرنه در صفحات دیگر هم استفاده می‌شد. این مسئله باعث حک شدن سایتها می‌شد.

یک تکنیک به نام session Hi jacking وجود دارد. وقتی که عدد 15 در cooki لطیف را به 16 تبدیل کنید، شناسایی user دچار مشکل می‌شد. در asp کلاسیک یک عدد Long بود که یکی یکی اضافه می‌شد. پس در .net اولاً sessionID را از عدد Long ترتیبی به Guid تبدیل کرد.

تعریف Session Time out

Session time out را از 20 دقیقه به یک دقیقه کاهش می‌دهیم.

برای اینکار ، وارد فایل web.config شوید. داخل system.web یک تگ به نام sessionState داریم که یک پارامتر آن timeout بود که می‌توانید آن را به 1 دقیقه تغییر دهید.

web.config

```
<system.web>
  <sessionState timeout="20" ></sessionState>
```

تعداد User Count واقعاً دقیق نیست، چون اگر Browser دیگر باز کند، یک session جدید، باز می‌شود و userOnline هم تغییر می‌کند .

معماری متغیر application , session با معماری سایر متغیرهای .net فرق دارد. در این متغیرها عملیات declare و Initialize و ... وجود ندارد.

session فقط مربوط به بحث login,logout نیست.

یک سایت را در نظر بگیرید که log in و log out ندارد. ولی در صفحه اول زبان فارسی را انتخاب می‌کنید و بعد وارد صفحات بعدی هم که می‌شوید، باز هم فارسی خواهد بود. ولی اگر در یک صفحه آن را انگلیسی کنید، در صفحات بعدی هم انگلیسی خواهد بود. پس یک session برای culture در سطح user تعریف شده است.

متغیر session

می‌خواهیم متغیرهایی در سطح session تعریف کنیم مثل sessionId,userId,userRole.

به متغیر application می‌گویند end متغیرهای global یعنی متغیرهایی برای همه userها و همه صفحات.

متغیر session فقط برای یک کاربر و برای همه صفحات است.

در صفحه login.aspx.cs:

In page : Login.aspx.cs

```
protected void btnSubmit_Click(object sender, System.EventArgs e)
{
    if (Page.IsValid)
    {
        Session["UserId"] = 1;
        // Session["UserId"] = oUsersRow.UserId;
        Session["Username"] = "Dariush";
    }
}
```

```

// Session["Username"] = oUsersRow.Username;
//Response.Redirect("~/My.aspx", false);
Response.Redirect("~/Default.aspx", false);
//Response.Redirect("~/Dashboard.aspx", false);
//Response.Redirect("~/ControlPanel.aspx", false);
Return;
    }
}

```

در طول دوره 5 مرتبه وارد صفحه login شدیم.

- 1 - موقع تعریف layout صفحه login
- 2 - موقعی که validator یاد گرفتید
- 3 - بحث ADO.Net
- 4 - بحث session
- 5 - بحث authentication , authorization

که در پروژه‌های واقعی باید با هم Mix شود.

نکته :

Response.redirect یک پارامتر دوم دارد که default آن true است. true یعنی اینکه user را به آن صفحه ببر و بی خیال tread صفحه ای که داخل آن هستی بشو. واقعیت این است که چون شما یک هو می‌خواهید در گوش tread ، گاهی قاطی می‌کند.

در جلسه بعد Centralized Error Handling یاد می‌گیرید. در جلسه بعد می‌بینید که در این مواقع یک سری Log ایجاد می‌شود که شما thread را ترک‌نکردید. البته چیز مهمی نیست ولی بهتر است که پارامتر دوم را false بگذارید و بعد از آن return بنویسید تا به صورت طبیعی tread شما قطع شود.

کاربر را بعد از login به چه صفحه‌ای هدایت کنیم:

- 1 - my
- 2 - Default
- 3 - Dashboard
- 4 - controlPanel

نکته :

از دستور string.compare برای مقایسه دو رشته استفاده کنید با پارامتر true = ignore case (یعنی بی خیال بزرگی و کوچکی حروف بشو)

```

(string.Compare(txtUsername.Text, "Dariush", true) != 0)
(string.Compare(txtPassword.Text, "1234512345", false) != 0)

```

در صفحه login ما دو تا متغییر

```
Session["UserId"] = 1;    Session["Username"] = "Dariush";
```

را در سطح session ایجاد کردیم و سپس در usercontrol دو تا link می‌گذاریم:

- login
- logout
- و یک متغییر wellcome

UserLoginStatus.ascx

```
<%@ Control Language="C#" CodeFile="UserLoginStatus.ascx.cs"
Inherits="UserControls_UserLoginStatus" %>

<span class="userLoginStatus">
    <asp:hyperlink id="lnkLogin" runat="server" text="Login"
navigateurl="~/Login.aspx" />
    <asp:hyperlink id="lnkLogout" runat="server" text="Logout"
navigateurl="~/Logout.aspx" />

    <asp:label id="lblWelcome" runat="server" />
</span>
```

UserLoginStatus.ascx.cs

```
public partial class UserControls_UserLoginStatus : System.Web.UI.UserControl
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        // یعنی کاربر لاگین نکرده است
        if (Session["UserId"] == null)
        {
            lnkLogin.Visible = true;

            lnkLogout.Visible = false;
            lblWelcome.Visible = false;
            lblWelcome.Text = string.Empty;
        }
        // یعنی کاربر لاگین کرده است
        else
        {
            lnkLogin.Visible = false;

            lnkLogout.Visible = true;
            lblWelcome.Visible = true;
            lblWelcome.Text =
                string.Format(" | Welcome {0}!", Ses "Username"]);
        }

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }
}
```

در خصوص دستورات امنیتی هر دفعه
که Post Back می شود، باید چک
شود. بیرون از page.IsPostBack

در صفحه logout.aspx:

Logout.aspx X

Client Objects & Events

```

4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6   <head runat="server">
7     <title></title>
8   </head>
9   <body>
10    <form id="frmMain" runat="server">
11      <div>
12        این صفحه هیچگاه دیده نخواهد شد
13      </div>
14    </form>
15  </body>
16 </html>

```

```

public partial class Logout : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        // Solution (1)
        Session.Clear();

        // Solution (2)
        //Session.Abandon();

        // Solution (3)
        //Session.Clear();
        //Session.Abandon();

        Response.Redirect("~/Login.aspx", false);
        //Response.Redirect("~/Default.aspx", false);
    }
}

```

void HttpSessionState.Clear()
 Removes all keys and values from the session-state collection.

void HttpSessionState.Abandon()
 Cancels the current session.

three solution in
page logout

تعریف Session Clear , Session Abandon

Session.clear : کلیه متغیرهای session را خالی می‌کند.
 Session.abandon : Session را می‌بندد و یک session جدید باز می‌شود.

صفحه Log out صفحه ای است که هیچ وقت توسط کاربر دیده نمی‌شود. کارهای لازم برای log out در پشت این صفحه انجام می‌شود و به صفحه login redirect می‌شود.

نکته :

یک سری از کلاسها را اگر داشته باشید، یا موقع develop به مشکل بر می خورید و یا در جای دیگر.

مثلاً اگر یک صفحه به نام x داشته باشید.

یک سری از صفحات اگر default اسم آنها Login و Default , profile باشد، دچار مشکل می شوید.

بنابر این این خطوط از برنامه ها را به این صورت بنویسید:

```
in x.aspx.cs :
public partial class LoginPage : System.Web.UI.Page
public partial class DefaultPage : System.Web.UI.Page
in x.aspx :
codefile="Default.aspx.cs" inherits="DefaultPage"
codefile="Default.aspx.cs" inherits="LoginPage"
```

Files → C# learning → _Visual_Studio_2010 → ASP.NET (web form)

نکته : در css در divMain (در div) از صفت Min Height استفاده می شود و این معادل معنی height در جدول است،

چون در جدول خود به خود اگر محتویات آن زیاد شود، ارتفاع آن را زیاد می کند ولی td به صورت پیش فرض ارتفاع خود را زیاد می کند.

تعریف cookie

نکته : Accept All Cookies یا Deny All Cookies اگر فایروال شما جلوی تمام cooki ها را بگیرد، چه می شود؟ اگر

user صد بار وارد سایت شود، 100 بار session ایجاد می شود.

در قدیم، می گفتند که اگر از سایت ما استفاده می کنید، باید Cookies را enable کنید.

در سایتهایی که در آدرس بار در می آید، دیده اید. در این سایتها Cookie را به جای اینکه روی کامپیوتر کاربر ایجاد

کنند. بر روی آدرس بار می گذارند و حمل می کنند.

روش آن:

in web.config is a CookieLess property that it's default is UseCookie and we must change it to

UseUrl

for having two state that is the best way :

```
cookieless="AutoDetect"
```

if it can use cookie then write to cookie and if it can not, it use of url

اگر cookie support بود، که بر روی کامپیوتر کاربر ایجاد می کند وگرنه در address bar می نویسد.

اگر سایت شما بر روی کامپیوتر کاربر Cookie درست کند، سایت دیگر نیز می تواند آن cookie را بخواند. خیلیها سایت برای

مصارف مختلف درست می کنند. حواستان نیست و CookieName را set نمی کنید و default آن cookieName است و

سایتهای مختلف از cookie name استفاده می کنند. پس بهتر است آن را تغییر دهید و از اسامی unique استفاده کنید:

یک برنامه به نام Guid Generator دارید. بنابراین CookiName را در sessionState با این guid پر کنید.

```
<sessionState timeout="1" cookieless="AutoDetect"
cookieName="0d79836357b64eeefa46e83500b2d4e38" mode="InProc"/>
```

یک عبارت mode هم وجود دارد. این از آن چیزهایی است که اگر راه حل آن را پیدا کنید، 10 تا 15 میلیون می‌ارزد. سازمان سنجش به ما می‌گوید، سایت ما موقع اعلام نتایج تا یک ماه بالا نمی‌آید. Blogfa , club سازمان سنجش و مناقصه گذاری که concurrent بالا دارد، نمی‌شود روی یک سرور گذاشت، هر چقدر هم که آن سایت خفن باشد. شما یک IP و یک domain می‌بینید و از طریق Load Blancing انجام می‌شود، این یک موضوع Hi tech است و مشاوره آن خیلی می‌ارزد.

در خصوص digintal sign library هم همینطور است اگر آن را یاد بگیرید و نمونه کد آن را بنویسید، ارزی بالایی دارد.

جلسه دوازدهم

این درس یک درس مهم، ساده و بدون نمونه مشابه در windows Application می‌باشد. اگر شما یک سایت با asp کلاسیک یا php تولید کنید، حتی اگر 7 و 8 سال هم با این سایت کار کنند، بعد از این همه مدت اگر کسی از شما سوال کند که واقعاً سایت باگ یا error دارد یا خیر و stable شده است یا خیر؟ شما نمی‌توانید پاسخ دهید. چون جایی نیست که log شود و کاربران هم نمی‌آیند برای شما داستان و انشاء بنویسند، که مثلاً در این صفحه وقتی این دکمه را کلیک کردم، اینگونه شد و ...

Centralized Error Handling

هدف این مبحث این است که هر خطایی که در پروژه وب شما رخ دهد بتواند هندل شود. در برنامه‌های windows based فقط یک نوع خطا و آن هم از نوع سیستمی وجود دارد.

انواع خطاها در برنامه‌های مبتنی بر web

در برنامه‌های مبتنی بر web سه نوع خطا داریم

- خطاهای system ای (مانند برنامه‌های windows application)
- خطاهای .net ای
- خطای iis ای

مثال خطای .net ای:

در بحث simple CMS اگر در Textbox تگ html می‌نوشتید error می‌داد که این خطا سیستمی نبود، بلکه .net ای بود.

مثال خطای iis ای:

File not found یا authentication error یا forbidden خطای وب سرویسی و iis ای است.

ما می‌خواهیم هر خطایی از هر نوعی که رخ می‌دهد Log شود.

این log کردن اولین کاری است که باید در هر سیستمی بنویسید حتی اگر یک صفحه داشته باشد.

آن فایل Log ای که ایجاد می‌کنید، اوایل باید زیاد به آن مراجعه کنید (روزی 3 تا 4 بار)، و به مرور این سرزدن‌ها به صورت هفتگی و ماهیانه خواهد بود و به این ترتیب مطمئن می‌شوید که دیگر سایت خطا ندارد چون هر خطایی log شده و در version بعدی می‌توانید آن را حذف کنید.

اهداف Centralized Error Handling

1 - کلیه خطاها Log شود.

2 - بعضی وقتها صفحات خطرناکی دیده می شود که هم cheap و زشت است و هم خطرناک (چون در آن صفحه زشت اطلاعات حساسی نشان داده می شود. مثلاً اگر خطا در باز کردن یک Connection رخ دهد، Connection String در این صفحه نشان داده می شود و هر کسی می تواند نام و IP سرور و همچنین نام بانک اطلاعاتی و حتی Username , Password را هم مشاهده نماید و این خیلی خطرناک است). پس دومین هدف ما این است که یک صفحه شیک برای error ها تولید کنیم و نه یک صفحه بد و زشت.

Try throw catch در کنار Centralized Error Handling یا در کنار آن

شاید در اینجا سوالی مطرح شود که آیا استفاده از Centralized Error Handling جای try throw catch را می گیرد و دیگر نیازی به نوشتن try Catch نداریم؟ پاسخ این است که خیر. هدف از try catch جلوگیری کردن از خطاهای سیستمی است. ولی بعضی اوقات خطاهای iis ای و .net ای رخ می دهد و یا اینکه سوتی دادیم و خطای سیستمی را از طریق try , catch , try هندل نکرده ایم، در واقع CEH فقط به منظور کنترل نهایی و جلوگیری از خطاهای ناخواسته است و تا جای ممکن دوست نداریم که کار ما به آن ختم شود. با کمک try catch می توانید جلوی خطای سیستمی را بگیرید. پس این درس بیشتر به جاهایی بر می گردد که سوتی دادید و try catch را ننوشتید و می خواهید مشکل را برطرف کنید و یا اینکه اصلاً خطای سیستمی نیست.

انواع خطاها

- در خطاهای سیستمی و غیر سیستمی که هم در ویندوز و هم در وب داریم سه نوع خطا داریم.
- 1 - **compile error یا syntac error** : این خطا تا برطرف نشود، برنامه اجرا نمی شود. (مثلاً آکولاد و پرانتز باز شده و بسته نشده و ...)
 - 2 - **خطای Logic error**: برنامه اجرا می شود و **run time error** را نمی دهد، ولی کار اشتباه را انجام می دهد و منطق آن درست نمی باشد. (مثلاً شما می خواستید 1 تا 9 را چاپ کند، ولی 1 تا 11 را چاپ می کند و برنامه به خطا برخورد ولی عملکرد اشتباهی داشت).
 - 3 - **Run time error** که تا امروز هر چه می شناختیم سیستمی بوده است.

بعضی از خطاهای سیستمی را می توان با if جلوگیری نمود. مثلاً تقسیم بر صفر را با if کنترل کنیم و بگوییم که اگر مخرج صفر است، پیغام بدهد.

گاهی اصلاً برای خطای ما if ای وجود ندارد یا صرف نمی کند که بنویسید. مثلاً می گوییم برو از floppy drive فلان فایل را برو بخوان؟ چه خطاهایی ممکن است رخ دهد. اصلاً فلاپی نداشته باشد یا باشد، drive آن وجود ندارد یا drive آن سالم نیست و یا جلوی آن گرفته و بسته شده است و ...

پس برو رو drive مربوط به A بنویس، باید 20 تا if بنویسید و در آخر هم مشخص نیست که همه کنترل شده یا نه؟ پس نمی دانیم و نمی توانیم با if کنترل کنیم پس خطا را catch می کنیم. پس خطاهایی که نمی ارزد که آن را بررسی کنیم و یا نمی توانیم بررسی کنیم با try catch آن را کنترل می کنیم.

درس امروز جایگزین try catch است؟ اگر در جایی باید try catch برای خطاهای سیستمی می نوشتید و سوتی دادید و ننوشتید، یا try catch ننوشتید ولی خطا .net ای یا iis ای است و try catch نمی تواند جلوی آن را بگیرد، در این گونه مواقع این درس به کمک شما می آید.

برای اینکه صفحه خطرناک چه در windows و چه در web مشاهده نشود، try catch می نویسیم، ولی کافی نیست و در نهایت centralized error handling را هم می گذاریم.

به عنوان مثال فرض کنید که می خواهید عملیات ذیل را در یک برنامه انجام دهید:

```
int A = 5;
int B = 0;
int C = A / B;
```

پس باید با try catch کنترل کنیم. حال مدل‌های مختلف را بررسی می کنیم:

مدل 1:

برای catch هیچ چیز نمی نویسد و این مزخرفترین حالت ممکن است و اصلاً برنامه نویسی نمی فهمد مشکل چیست.

```
// *****
int A = 5;
int B = 0;
try
{
    int C = A / B;
}
catch
{
    Response.Write("کاربر گرامی خطای ناشناخته‌ای رخ داده است، یک فکری به حال خودت بکن");
}
// *****
```

مدل 2:

برای catch یک پارامتر exception می گیرند و این از این جهت بهتر است که می توان حداقل خود e یا Exception را به کاربر نشان داد.

```
//*****
int A = 5;
int B = 0;
try
{
    int C = A / B;
}
catch (System.Exception ex)
{
    Response.Write(ex.Message);
}
// *****
```

نکته:

نوشتن response.write در net. منسوخ شده است. response.write کل layout را شیفتمی دهد پایین و آن بالا خطا را نشان می دهد و اصلاً شکل نیست. ما خطاها را دوست داریم در label بنویسیم. البته هنوز هم در asp کلاسیک و php این دستور بسیار کاربرد دارد. حتی در روش MVC دستور Response.write بسیار کاربرد دارد.

همیشه سعی کنید که از قسمت finally استفاده کنید. در دستور try چه به خطا بخورد و چه نخورد finally اتفاق می افتد.

```
// *****
int A = 5;
int B = 0;
try
```

```

{
    int C = A / B;
}
catch (System.Exception ex)
{
    Response.Write(ex.Message);
}
finally
{
}
// *****

```

به دور روش می توان از تابع خارج شد:

1 - استفاده از دستور return

2 - استفاده از روش throw. در این روش خطا یا مشکل را خودمان بوجود می آوریم و پرتابش می کنیم داخل سیستم.

```

// *****
// امکانپذیر روش دو به کلی طور به تابع از خروج
1. return();
2. throw();
// *****

```

وقتی که می گوییم خطا را به لایه های دیگر پرتاب می کنیم، دیوانه نیستیم، بلکه کاربرد دارد. کسانی که برنامه نویسی چندلایه می کنند، به عنوان message passing خطا پرتاب می کنند به لایه های دیگر، وقتی که در لایه Business مشکل پیش بیاید، خطا به لایه user interface پرتاب می شود.

همانطور که گفتیم ما دو روش داریم که می توان با استفاده از آن از دل تابع خارج شد. اگر دستور throw , return هر کدام داخل try نوشته شود این کار را انجام می دهند. می توان در finally کار تر و تمیز کردن حافظه و ... را انجام داد.

سعی کنید در برنامه نویسی ASP.NET از response.write استفاده نکنید و به جای آن از Label استفاده کنید.

Exceptions

System.exception یک کلاس است و خطاها را در خود نگه می دارد. وقتی که یک کلاس داریم، این کلاس یک کلاس عمومی است و می توان از آن یک کلاس تخصصی درست کنیم.

در خصوص کلاسها تفکر از بالا به پایین را specialization و از پایین به بالا را Generalization می گویند. اگر ما یک کلاس به نام system. Exception داریم که فقط دارای یک متد است، کلاس sqlexception وجود دارد که یک سری آیتم اضافه دارد. در واقع تمام متدهای کلاس Exception را دارد و یک سری متد اضافه دارد.

پس ما کلاسهای تخصصی برای خطاها داریم. ماکروسافت توسعه می کند که در حوزه ای که دارید برنامه نویسی می کنید، Exception همان حوزه را انتخاب کنید، مثلاً برای SQL از SQLException استفاده کنید تا پیغام دهد که رکورد قابلیت ذخیره یا حذف ندارد و ... یعنی دوست داریم که خطاها را به صورت اختصاصی و به ریز داشته باشیم و نه به صورت کلی.

```

// *****
int A = 5;
int B = 0;
try

```

```

    {
        int C = A / B;
    }
    catch (System.DivideByZeroException ex)
    {
        DisplayErrorMessage(ex.Message);
    }
    finally
    {
    }
}
// *****

```

مثلاً در مثال بالا، چون عملیات تقسیم بر صفر داریم، باید از DivideByZeroException استفاده کنیم. برای مدیریت خطاهای مدیریت فایل نیز ماکروسافت یک کلاس ایجاد نموده است. که با استفاده از آن می‌توان خطاهایی که در هنگام مدیریت فایلها، مانند ایجاد، ویرایش و حذف و ... ایجاد می‌شود را به صورت دقیق پیغام دهیم.

علاوه بر اینکه در catch ، exception همان حوزه را بهتر است بنویسید، کلاس کلی system.exception را بنویسید، تا اگر بی‌دقتی کردید و exception تخصصی را ننوشتید، خطا از دستتون در نره. مثلاً 200 خط کار بانک اطلاعاتی داشتید و وسط آن دو خط I/O هم داشتید، حالا فراموش کردید که IO exception را بنویسید، پس Exception کلی را بنویسید تا خطا از دستتون در نرود.

```

// *****
int A = 5;
int B = 0;
try
{
    int C = A / B;
}
catch (System.DivideByZeroException ex)
{
    DisplayErrorMessage(ex.Message);
}
catch (System.Exception ex)
{
    DisplayErrorMessage(ex.Message);
}
finally
{
}
// *****

```

هر دستوری که ممکن است، خطایی پیش بیاورد و قابل کنترل نیست، حتماً در try می‌نویسید، و در finally کارهای فاینال کردن را می‌نویسید و در catch هم عملیات را می‌نویسید.

به عنوان آدم حرفه‌ای باید در تمام توابعمان اینگونه کد بنویسیم. بحث امروز از جایی شروع می‌شود که سوتی دادیم و ننوشتیم و یا اینکه نوشتیم و خطا iis , net. است و نتوانسته هندل شود.

در پروژه های web based , windows application وقتی که یک دستور می‌خواهد اجرا شود، که ممکن است باعث بروز خطا شود، باید جلوی آن را به هر شکلی است بگیرد. اگر سایتی جلوی آن را نگرفته باشد و اگر صفحه error سیستمی ظاهر شد، دو ریال هم نمی‌ارزد.

1 - cheap است و اعتماد کاربر را از بین می‌برد.

2 - Source code را لو می‌دهد. مثلاً دستور `connection . open` گذاشتید و اگر آن را هندل نکنید، عالم و آدم IP سرور و نام `connection` و ... را نشان می‌دهد.

کاربرد صفحه `Offline.htm`

نکته: گاهی صفحه خطایی در برنامه‌های وب ظاهر می‌شود و پیغام می‌دهد که `custom error` را در `web config` تغییر بدهید. آن را نمی‌شود کاملاً اجتناب کرد. آن خیلی مهم نیست. مثلاً وقتی که سایت را به دلیل تغییر ورژن جایگزین می‌کنید، ممکن است با این خطا روبه‌رو شوید، ولی این مسئله هم راه حل دارد: می‌توان قبل از اینکه سایت را `delete` کنید، کلیک بزنید. برای اینکار، یک فایل به نام `offline.htm` داریم. اگر آن را در سایت بگذارید، سایت `down` می‌شود و کلیه فایل‌های `in use` هم از کار می‌افتد و در این زمان شما سایت را `delete` و ورژن جدید آن را جایگزین نمایید.

خوب تا اینجا در خصوص مکانیزم `try` و `catch` توضیح دادیم، حالا می‌رویم سراغ خطاهایی که شما آن را هندل نکرده‌اید یا `iis` ای و یا `.net`. ای است و می‌خواهیم با برخورد با این خطاها یک صفحه با کلاس نشان داده شود. با `centralized error handling` صفحه با کلاس نشان می‌دهیم.

صفحات یک صفت `auto event wireup` داشتند. مزیت آن این بود که وقتی آن را `true` کنید، یعنی اگر صفحه شما `auto wire up` باشد، دو تابع را به صورت اتوماتیک صدا می‌کند. این دو تابع، `Page_load` , `page_error` می‌باشند.

ما در `Page error` دنبال چه هستیم؟ دنبال اینکه از طریق توابع `server.getlasterror().getBaseException()` خطا و `base` آن را به دست آوریم.

چون دنبال این زنجیره هستیم و در `Page error` هستیم، باید حواسمان را جمع کنیم. باید حواستان را جمع کنید که در خود `Page error` خطایی پیش نیاید. چون دیگر آن را نمی‌شود `handle` کرد.

پس کد مورد نظر خود را در تابع `Page_Error` می‌نویسیم.

```
protected void Page_Error(object sender, EventArgs e)
```

کد این قسمت را به صورت ذیل می‌نویسیم.

```
if (Server != null)
{
    if (Server.GetLastError() != null)
    {
        if (Server.GetLastError().GetBaseException() != null)
        {
            // *****
            System.Exception ex =
                Server.GetLastError().GetBaseException();
            // *****

            // *****
            string strErrorMessage =
                string.Format(
                    "{0} Time: {1}",
                    ex.Message,
                    System.DateTime.Now.ToString("yyyy/MM/dd - HH:mm:ss")
                );
            // *****
        }
    }
}
```

```

// *****
// LogHandler.Log(strErrorMessage);
// *****

// *****
Response.Write
    ("  


```

نکته: چندین دوره پیش گفتیم که هیچوقت متغیر `server` ، `null` نمی‌شود. ولی یکجا `server` ، `Null` می‌شود. در `Application_end` ، `Server` به صورت `null` می‌باشد. حواستان به این مسئله جمع باشد. برای اینکه عادت کنید که این شیوه نگارش را داشته باشید، قانون را کامل بنویسید.

پیغام را می‌سازم و می‌گویم فلان خطا در فلان زمان رخ داده است و خطا را در فایلی `log` می‌کنم. و بعد پیغام خطا را به `user` نشان می‌دهیم. باید یک خطای گنگ به کاربر بگوییم. مثلاً بگوییم یک خطا پیش آمده و باید با مدیریت تماس بگیری. بعد که خطا را در فایل نوشتیم باید خطا را پاک کنید، تا سیستم به کار خود ادامه دهد. این کار را از طریق `Server.clearError` انجام می‌دهید.

در زبان `C#` می‌توان سه تا `if` را یکی کنید و بعد با `and` جدا کنید. در زبان `vb` سنتی وقتی همچنین `if` ای دارید، باید تک تک اجرا کنید و بعد `and` می‌کند ولی `C#` باهوش است و اول می‌رود سراغ پرانتز اول و اگر اولی `null` باشد، اصلاً سراغ بقیه نمی‌رود و به مشکل نمی‌خوریم.

```

// *****
if ((Server != null) &&
    (Server.GetLastError() != null) &&
    (Server.GetLastError().GetBaseException() != null))
{
    // *****
    System.Exception ex =
        Server.GetLastError().GetBaseException();
    // *****
    // *****
    string strErrorMessage =
        string.Format
        (
            "{0} Time: {1:yyyy/MM/dd - HH:mm:ss}",
            ex.Message,
            System.DateTime.Now
        );
    // *****

    // *****
    LogHandler.Log(strErrorMessage);
    // *****

    // *****
    Response.Write
        ("  

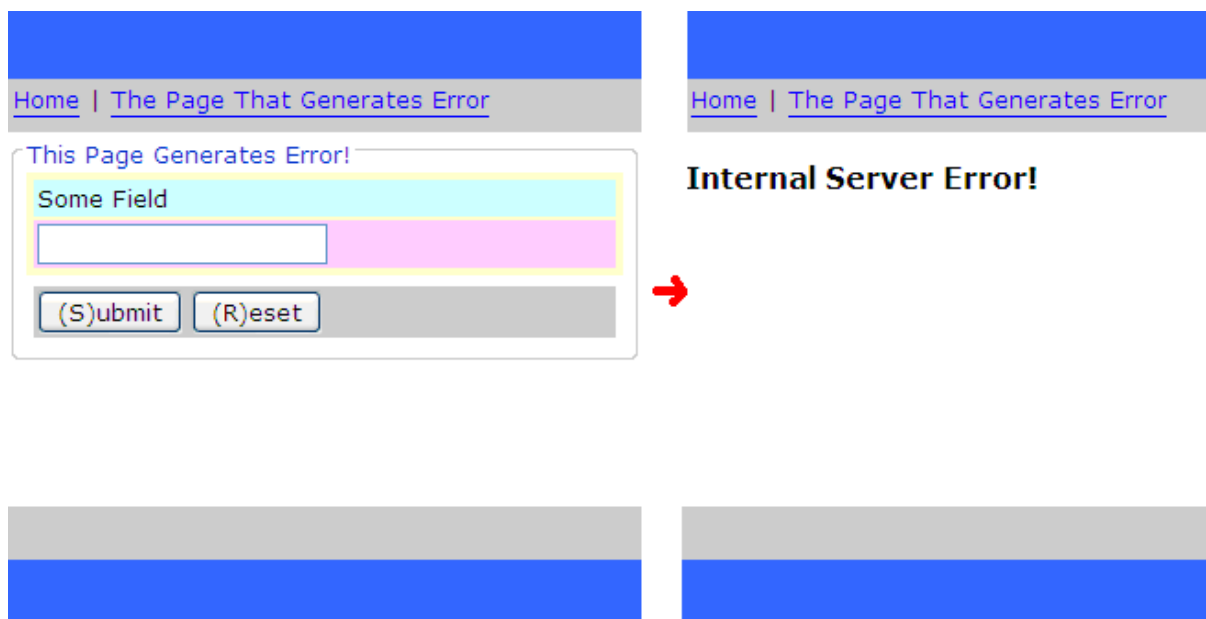

```

```

Server.ClearError();
// *****
}
// *****

```

کد و تصویر مربوط به صفحه ای که قرار است در آن خطای هندل نشده از طریق Try Catch ، وجود داشته باشد را به صورت ذیل تعریف می کنیم.



```

<%@ page title="This Page Generates Error!" language="C#"
masterpagefile="~/MasterPages/Site.master" codefile="ThisPageGeneratesError.aspx.cs"
inherits="ThisPageGeneratesError" %>

```

```

<asp:content id="cntHead" contentplaceholderid="cphHead" runat="Server">
</asp:content>
<asp:content id="cntMain" contentplaceholderid="cphMain" runat="Server">

    <div id="divMessage" runat="server" class="message">
        <asp:label id="lblMessage" runat="server" />
    </div>

    <fieldset>
        <legend>
            This Page Generates Error!
        </legend>

        <div class="field">
            <div class="caption">
                <asp:label id="lblSomeField" runat="server" text="Some
                Field" />
            </div>
            <div class="controls">
                <asp:textbox id="txtSomeField" runat="server"
                maxlength="100" accesskey="1" />
            </div>
        </div>
    </fieldset>

```



```

<div class="buttons">
    <asp:button id="btnSubmit" runat="server" text="(S)ubmit"
        accesskey="S" onclick="btnSubmit_Click" />
    <input type="reset" value="(R)eset" accesskey="R" />
</div>
</fieldset>
</asp:content>

```

حالا این کد را trace می‌کنیم. چون در این صفحه auto event wire up به صورت true مقداردهی شده است و تابع Page error هم وجود دارد، وارد Page Error می‌باشد. وارد این تابع شده و پیغام error را می‌سازد و در فایل می‌نویسد یا email می‌کند و ... و یک پیغام unexpected به کاربر نشان می‌دهد و بعد خطا را clear می‌کند. یعنی خطا حل شد.

Centralized error باعث می‌شود که Layout از بین برود. بنابراین همانطور که بارها و بارها گفتیم، جایگزین Try catch نیست. چون دیگر Layout ندارد و زیبا نیست.

اگر Page error داشته باشید و Clear error کنید، خطا ادامه پیدا نمی‌کند. اگر Page error داشته باشید و Clear error نکنید، خطا ادامه پیدا می‌کند و به Application Error می‌رود. اگر هم اصلاً Page Error نداشته باشید، باز هم ادامه پیدا می‌کند و به Application Error می‌رود. ولی اگر Page error داشته باشید و Clear Error کنید، اصلاً وارد Application Error نمی‌شود.

بنابراین Page Error را برای حفظ سیلابس گفتم و لازم نیست که آن را بنویسید. پس در قسمت کد مربوط به Page Error خط Clear Error را بردارید. وارد فایل global.asa شوید. در جلسه قبل در خصوص Application Error صحبتی نکردیم. الان وارد این قسمت می‌شویم.

در فایل global.asa کارهای ذیل را انجام می‌دهیم:

دوباره همان شرط را می‌نویسیم و خطا را sense می‌کنیم، خطا را دوباره ایجاد می‌کنیم، می‌گوییم خطا در چه زمانی رخ داده است. دستور آن به صورت ذیل می‌باشد:

```
System.DateTime.Now
```

قبلاً که خطا را در Page Error جستجو می‌کردید و کد آن را در آنجا می‌نوشتید، می‌دانستید که در کدام صفحه خطا اتفاق افتاده، ولی الان باید آن صفحه‌ای که خطا در آن رخ داده است را پیدا کنید. دستور آن این است:

```
Request.PhysicalPath
```

حال می‌خواهیم IP مربوط به user ای که خطا برایش رخ داده را هم به دست آوریم. این دستور IP کاربری که الان وصل شده است را پیدا می‌کند:

```
Request.UserHostAddress
```

پیغام خطا را بوجود آوردیم. ولی نباید آن را به کاربر نشان دهیم و باید یک پیام General به نام Unexpected Error را به کاربر نشان دهیم. حالا باید خطا را یکجا Log کنم. کجا خطا را Log کنم؟ باید در App_Data بگذارم. چون این فولدر با شخصیت یک فولدر Secure است و hacker نمی‌تواند به آن دسترسی پیدا کند. حواستان باشد که خطا را به آن append کنید.

فولدر Logs را به فولدر App_Data اضافه کنید و به آن یک فایل به نام Application.log اضافه کنید. تمام خطاهایی که اتفاق می‌افتد را به انتهای این فایل append کنید. می‌خواهیم به یک content دسترسی داشته باشیم و آن را دستکاری کنیم.

امکان concurrency وجود دارد. فایل text قابلیت مدیریت Concurrency را ندارد، بنابراین باید قبل از نوشتن در آن، آن را Lock و دوباره unlock کنیم.

```
Application.Lock();
Application.Unlock();
```

برای نوشتن در فایل آدرس فیزیکی لازم داریم. ولی گفتیم که در برنامه هیچوقت آدرس فیزیکی ننویسید. پس با استفاده از دستوراتی که قبلاً هم گفته بودیم، آدرس فیزیکی را به دست می‌آوریم. البته به صورت Unicode ای هم می‌خواهم بنویسم. خطای مفصلی را که درست کردیم، داخل این فایل Write می‌کنیم. اگر فایل وجود ندارد، بساز وگرنه به ته آن append کن.

```
string strApplicationLogRootRelativePathName =
    "~/App_Data/Logs/Application.log";

string strApplicationLogPathName =
    Server.MapPath(strApplicationLogRootRelativePathName);
oStream =
    new System.IO.StreamWriter
        (strApplicationLogPathName, true, System.Text.Encoding.UTF8);

oStream.WriteLine(strErrorMessage);
```

کار Finally هم که گفتیم finalize کردن بود. یعنی close و dispose کردن و ... در این مثال هم همینکار را کردیم.

```
finally
{
    if (oStream != null)
    {
        //oStream.Close();
        oStream.Dispose();
        oStream = null;
    }
}
```

Close , dispose چه فرقی دارند. Stream یا خیلی از جانورها را باید اول Close کرد و بعد dispose. وقتی که Stream دارید، تا Close یا dispose نکردیم، ایجاد نمی‌شود. پس حتماً یادتان باشد که close یا dispose یا هر دو را صدا بزنید.

حالا پروژه را rebuild کنید. از روز اول گفتیم که یک مفهوم به نام code analyzer داریم و اگر آن را فعال کنیم. ممکن است به ما بگوید، چرا کدتان را اینگونه نوشتید. Code Analyser به ما error نمی‌دهد بلکه warning می‌دهد. سعی کنید که تاجایی که ممکن است برنامه تان را به گونه‌ای بنویسید که Code Analyser یا اصلاً warning ندهد یا حداقل warning ندهد.

حالا ببینیم که چگونه باید Code Analyser را فعال کنیم. فقط کافی است که دکمه راست را روی پروژه بزنید و برید پایین و بر روی run code Analsis on web site کلیک نمایید و بعد پروژه را rebuild کنید. Code Analyst برای قرتی‌بازی نیست. مثلاً در اینجا Code anylyser می‌گوید، close stream به صورت داینامیک dispose هم می‌کند، لازم نیست دو کار را انجام دهید، و یکیش را باید حذف کنید. در واقع Dispose هم اگر بیاورید، اول close و بعد dispose می‌کند.

حالا برنامه را trace می‌کنم.

```
void Application_Error(object sender, EventArgs e)
{
```

```

if ((Server != null) &&
    (Server.GetLastError() != null) &&
    (Server.GetLastError().GetBaseException() != null))
{
    // *****
    System.Exception ex =
        Server.GetLastError().GetBaseException();
    // *****

    // *****
    string strErrorMessage =
        string.Format
        (
            "{0} Time: {1:yyyy/MM/dd - HH:mm:ss}, Path:
            {2}, User IP: {3}",
            ex.Message,
            System.DateTime.Now,
            Request.PhysicalPath,
            Request.UserHostAddress
        );
    // *****

    // *****
    //Response.Write
    // ("<br />Unexpected Error! Please Contact Support!");

    //Server.ClearError();
    // *****

    // *****
    Application.Lock();
    System.IO.StreamWriter oStream = null;

    try
    {
        string strApplicationLogRootRelativePathName =
            "~/App_Data/Logs/Application.log";

        string strApplicationLogPathName =
            Server.MapPath(strApplicationLogRootRelativePathName);

        oStream =
            new System.IO.StreamWriter
            (strApplicationLogPathName, true,
            System.Text.Encoding.UTF8);

        oStream.WriteLine(strErrorMessage);
    }
    // Note: در این قسمت نیازی به اجرای دستور ذیل نمی باشد.
    //catch
    //{
    //}
    finally
    {
        if (oStream != null)
        {
            //oStream.Close();
            oStream.Dispose();
            oStream = null;
        }
    }
    Application.Unlock();
}

```

```
// *****
```

```
}
}
```

یک نکته جالب در اینجا می بینید، این است که اصلاً در اینجا Catch وجود ندارد، این یعنی اینکه هر خطایی اتفاق افتاد فدای سرم. یعنی قرار است خطاهای دیگر را در اینجا هندل کنید، اما اگر اینجا خطایی رخ داد، دیگه کاری نمی توانم بکنم و بنابراین فدای سرم.

فرض کنید، من Catch , try هم داشتم. می خواهم کاری کنم که خطای .net ای رخ دهد. پس tag html در این text box می نویسم. اینجا با اینکه try catch دارم، باز هم وارد Application Error می شود و خطای این قسمت را تولید می کند.

این دفعه یک خطای IIS ای درست می کنم و آن را هندل می کنیم.

مثلاً صفحه a.aspx وجود ندارد. چون این صفحه اصلاً وجود ندارد، مستقیماً وارد application error می شود و وارد error general می شود و unexpected error می دهد.

شاید به ازای این خطای خاص بخواهم یک صفحه دیگر به کاربر نشان دهد. وارد web.config بشوید و تگ Custom Error را به صورت زوجی و سه تا خطا برای آن می نویسم. این خطاها دارای status code هستند و معادل خطاهای iis ای یا وب سروری است. این خطاها است. مثلاً 403 همان forbidden است. 404 معادل file not found و 500 معادل سرور ترکیده است. پس می خواهیم بگوییم، که اگر خطا از نوع پیدا نکردن فایل بود به 404 برو و اگر این بود. ولی اگر هیچکدام نبود برو به general .

```
<!--
  <customErrors mode="Off"/>
-->
<!--
  <customErrors mode="On" defaultRedirect="~/Errors/ErrorGeneral.aspx"/>
-->

<customErrors mode="On" defaultRedirect="~/Errors/ErrorGeneral.aspx">
  <error statusCode="403" redirect="~/Errors/Error403.aspx"/>
  <error statusCode="404" redirect="~/Errors/Error404.aspx"/>
  <error statusCode="500" redirect="~/Errors/Error500.aspx"/>
</customErrors>
```

در ادامه یک فولدر به نام Errors درست می کنیم و هر کدام از صفحات Error403.aspx و Error404.aspx و ... را در آن درست می کنیم.

```
public partial class Errors_Error403 : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if ((Server != null) &&
            (Server.GetLastError() != null) &&
            (Server.GetLastError().GetBaseException() != null))
        {
            // افتد نمی اتفاق هیچگاه شرط، داخل دستورات
        }
    }
}
```

```
        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }
}

public partial class Errors_Error404 : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if ((Server != null) &&
            (Server.GetLastError() != null) &&
            (Server.GetLastError().GetBaseException() != null))
        {
            // افتد نمی اتفاق هیچگاه شرط، داخل دستورات
        }

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }
}
```

```
public partial class Errors_Error500 : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if ((Server != null) &&
            (Server.GetLastError() != null) &&
            (Server.GetLastError().GetBaseException() != null))
        {
            // افتد نمی اتفاق هیچگاه شرط، داخل دستورات
        }

        if (Page.IsPostBack == false)
        {
            Initialize();
        }
    }

    private void Initialize()
    {
    }
}
```

تعداد خطاها بسیار زیاد است، 100، 200، 300 و ... برای همه نباید در سایت خطا بنویسید، فقط برای آنهایی بنویسید که .net آن را تشخیص می‌دهد. مثلاً خطای 503 یعنی سرور منفجر شده است، وقتی که سرور منفجر شده است، دیگر نمی‌تواند خطا را تشخیص دهد. پس همین سه تا اهمیت دارد و باید بنویسید. بقیه خیلی کم کاربرد هستند.

انواع mode های مربوط به Custom Error

در خصوص mode مربوط به custom error یا on است یا off یا remote only .
وقتی کسی از روی خود Visual Studio روی سایت، سایت را باز می‌کند local است ولی بقیه که از روی کامپیوترها به سایت وصل شود، remote هستند. وقتی که مد custom error را off می‌گذارید، یعنی همه چه کسی که local است و چه کسانی که remote هستند، همان صفحه خطرناک را در هنگام خطاهای هندل نشده ببینند. ولی اگر مد آن را on بگذارید، یعنی هم کسی که local است و هم کسانی که remote هستند، صفحه خوشگله که طراحی کرده‌ایم را ببینند ولی اگر مد آن را remote only بگذارید، یعنی کسی که local وصل می‌شود، صفحه خطرناکه را ببیند و بقیه کسانی که remote وصل می‌شوند، صفحه خوشگله را ببینند. این بهترین حالت است. من که پشت کامپیوتر نشستم و دارم سایت را ویرایش می‌کنم که نمی‌خواهم برای خودم کلاس بگذارم، باید همان صفحه خطرناکه را با شماره خط را ببینم.

آخرین نکته:

این به همان تاکتیکهای وب بر می‌گردد و ابتکارانی است که وب سایت شما را valuable می‌کند

404 Error را که استاد شدید و می‌دانید که در صورتیکه فایل پیدا نشد، به آن بر می‌خورید. در 404 error یکی از کارهایی که انجام می‌دهید این است که یک table درست می‌کنید، و نام صفحات درخواست شده و IP فرد و تعداد دفعات درخواست را در آن قرار می‌دهید. اگر کسی (IP ای) برای اولین بار به سایت ما متصل شده بود و یک صفحه خاص را زده بود، IP و نام صفحه را با شماره 1 در قسمت تعداد درخواست در جدول ثبت می‌کنیم و اگر دفعات بعدی همان IP همان صفحه را درخواست کرد، counter آن را افزایش دهید.

حسن آن دو چیز است:

- 1- بعد از یک سال، 6 ماه توقعات افراد را از سایت شما حدس می‌زنید. سایت بازهای حرفه‌ای به دنبال صفحات نمی‌گردند و صفحات استاندارد را خود به خود می‌شناسند. وقتی که شما در سایتتان می‌بینید که 1000 بار صفحه help را جستجو کردند، پس توقعات کاربر را می‌فهمید. و اگر help سایت شما نامش rahnama است، آن را برای کاربران رdirect کنید.
- 2- کاربرد دوم این است که می‌بینید که چقدر مردم کلمه مثلاً default را اشتباه تایپ می‌کنند و به چه شکلهایی آن را اشتباه تایپ می‌کنند. پس نیازهای کاربران را شکار کنید. یعنی غلطهای دیکته‌ای آنها را خودتان درست کنید و هندل کنید و کاربر را به صفحه page not found نبرید، بلکه به صفحه درست redirect کنید.

```
public partial class Errors_Error404 : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        if ((Server != null) &&
            (Server.GetLastError() != null) &&
            (Server.GetLastError().GetBaseException() != null))
        {
            // افند نمی اتفاق هیچگاه شرط، داخل دستورات
        }
    }
}
```

```
}

string strAspxErrorPath =
    Request.Params["aspxerrorpath"];

//string strAspxErrorPath =
//    Request.QueryString["aspxerrorpath"];

string strFileName =
    System.IO.Path.GetFileName(strAspxErrorPath);

switch (strFileName.ToUpper())
{
    case "DEFAULT.ASPX":
    case "DIFALT.ASPX":
    case "DIFAULT.ASPX":
    case "DEFUALT.ASPX":
    {
        Response.Redirect("~/Default.aspx", false);
        break;
    }

    default:
    {
        // Note: Send Email to admin and/or Log in Database!
        break;
    }
}

if (Page.IsPostBack == false)
{
    Initialize();
}

private void Initialize()
{
}
}
```

جلسه پنجم

مبحث Authentication & Authorization

این مبحث را باید با مبحث session میکس (mix) کنیم .
یک پروژه بکر درست می کنید :

Create folder MasterPages - 1

Create site.master .a

در این master page لینک به همه صفحات می گذاریم و فائدا لینک به خود صفحه home (Default.aspx) هم خواهیم داشت.

Create folder CSS - 2

Create site.css .a

:Create - 3

Default.aspx .a

Logout.aspx .b

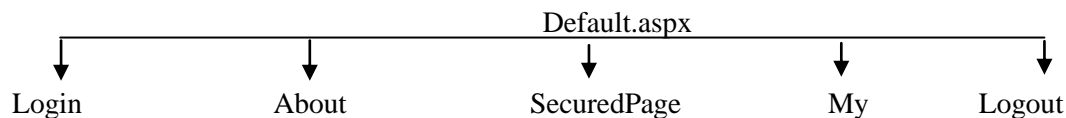
My.aspx .c

Login.aspx .d

securedPage.aspx .e

همین که این صفحات را درخواست کنیم واردشان می شویم اتم از اینکه link به این صفحات داشته باشیم یا مستقیما در address bar آدرس این صفحات را بزنیم .

حالا کاری که انجام می دهیم در ابتدا این است که site map پروژه را روی صفحه وایت برد می کشیم: یک سری صفحه داریم که سپس ارتباط این صفحات را با هم مشخص می کنیم link to each other که این را با فلش نشان می دهیم که اگر این فلش را نگذاریم یعنی دو طرفه است.



هدف این است که سه تا از این صفحات را وقتی user درخواست می کند حتما باید قبلش Login کرده باشد:

SecuredPage.aspx ○

My (another name for My page is : home or dashboard or my or controlpanel) ○

Logout.aspx ○

اول در master page لینک به همه این صفحات را می گذاریم.

نکته خارج از مطلب: یک jquery است به نام slide show : که وقتی می خواهید یک عکس را با یک متن شعاری نمایش بدهید از این استفاده کنید که این را برایتان share می کنیم.

Authentication:

وارد web.config می شویم و در زیر مجموعه <system.web> یک tag ای داریم به نام Authentication که به معنی نحوه ی شناسایی کاربر است که اول تشخیص دهیم چه کسی است و سپس نسبت به آن سطح دسترسی قائل شویم:

○ یک mode ای دارد به نام none یعنی همه صفحات این سایت آزاد و امن است و هیچ صفحه ای secured و نامن نیست.

```
<!--
<authentication mode="None"/>
-->
```

یک mode ای دارد به نام windows یعنی widows کاربران را شناسایی کند که به درد پروژه هایی می خورد که intranet هستند و یا active directory یا domain در آنها فعال است و به درد پروژه های web نمی خورد.

```
<!--
<authentication mode="Windows"/>
-->
```

○ یک mode ای دارد به نام passport که برای ماکرو سافت است که registration را یکبار انجام دهد برای مشتریان خودش و سایتی درست کرد به نام passport.com

```
<!--
<authentication mode="Passport"/>
-->
```

ولی از وقتی که شیپنت های ماکروسافت زیاد شد و با اطلاعات مردم زیاد ور میرفت علاقه به سایت password کم شد و خیلی از مردم دیگر از این سرویس سایت استفاده نکردند.

○ یک mode ای دارد به نام forms یعنی user را از طریق فرمهای aspx شناسایی کنیم. یعنی همان user name , password , ... از طریق فرمهای aspx

```
<!--
<authentication mode="Forms"/>
-->
```

حالا بر می گردیم به web.config تا مقداری خورده ریز یاد بگیریم به طور پیش فرض وقتی کاربر وارد صفحه login شود و user , password را درست بزند دستور redirect به طور اتوماتیک آن را به سمت صفحه default می برد ولی اگر user بخواهد بدون login کردن وارد صفحه secure بشود اتوماتیک سیستم user را redirect می کند به صفحه login و اگر درست زد به همان صفحه بر میگردداند .

پس می بینید که دو تا صفحه به صورت پیش فرض اتفاق می افتد و لی این قانون نیست و می توانیم عوضش کنیم. می توانید دستور authentication را به صورت tag زوجی بنویسید

```
<authentication mode="Forms">
  <forms loginUrl="~/Login.aspx" returnUrl="~/My.aspx" />
</authentication>
```

```
defaultUrl="~/Default.aspx"
```

معمولا به جایش می توان این صفحات گذارد

```
~/My.aspx
~/Home.aspx
~/Dashboard.aspx
~/ControlPanel.aspx
```

مطلب بعد: در بحث session ما یک مفهومی داشتیم به نام cookieless که یا به صورت cookie ذخیره کند یا در url ذخیره کند سپس گفتیم بهتر است که Autodetect را انتخاب کنید و cookiename آن unique باشد.

```
<sessionState timeout="20" cookieless="AutoDetect"
cookieName=".75aa0f1024ca45ba8e67bd8333a32afb"/>
```

که البته تگ forms در authentication هم همین ها را مخصوص به خود دارد که البته time out آن باید همیشه 10 تا بیشتر از sessionstate باشد.

```
<authentication mode="Forms">
  <forms loginUrl="~/Login.aspx" returnUrl="~/My.aspx"
  timeout="30" cookieless="AutoDetect"
  name=".0278f9b47dca453eba7c2337b274c19c"/>
</authentication>
```

Autorization

یعنی دسترسی کاربر را مشخص کنیم. دو دستور داریم:

Allow

○ Roles در کار ما نیست

○ Users روی این حالت کار می کنیم

در مقابل users یک سری user name می توانیم بدهیم که یعنی دسترسی دارند

Value="*" یعنی اتم از اینکه همه کاربران login کرده اند یا نکرده اند یا عبارتی یعنی اتم از اینکه سایت آنها را می شناسد یا نمی شناسد.

Value="?" یعنی اتم از اینکه سایت آنها را نمی شناسد یا کاربرانی که هنوز login نکرده اند

Deny

○ Roles در کار ما نیست

○ Users روی این حالت کار می کنیم

در مقابل users یک سری user name می توانیم بدهیم که یعنی دسترسی ندارند خواهیم دید که وقتی صفحه ای را secure می کنیم اتم از اینکه لینک به آن صفحه گذاشته باشیم یا نگذاشته باشیم و مستقیم در address bar نام صفحه را صدا کنیم بدون اینکه هیچ کدی نوشته باشیم اگر login نکرده باشد مستقیم مارا به سمت صفحه login می برد و ما را مجبور می کند user & password درست را بزنیم لذا ما می خواهیم عمل اعلان امن بودن یا نبودن صفحات را با این دستورات مشخص کنیم.
دو تا سیاست داریم :

1 اولین اینکه بگوییم همه صفحات اتم از اینکه کاربر login کرده یا نکرده آزاد است بغیر از استثنا هایی که مشخص می کنیم

```
<system.web>
```

```
    <authorization>
      <allow users="*" />
    </authorization>
```

```
</system.web>
```

<pre><location path="My.aspx"> <system.web> <authorization> <deny users="?" /> </authorization> </system.web> </location></pre>	<pre><location path="Logout.aspx"> <system.web> <authorization> <deny users="?" /> </authorization> </system.web> </location></pre>
---	---

```
<location path="SecuredPage.aspx">
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</location>
```

2 اینکه بگوییم همه صفحات اتم از اینکه کاربر login کرده یا نکرده secure یا امن است و تا وقتی که هیچ user ای login نکرده حق دیدن ندارد بغیر از استثنا هایی که مشخص می کنیم
نکته : هرچه در داخل system.web بنویسید یعنی قوانین و مقرراتی که برای همه صفحات وزیر دایرکتوری صادق است.

و هر چه در location زیر <system.web> بنویسیم یعنی این قوانینی که اکنون می نویسیم استثنائاً برای این صفحه یا sub directory صادق است.

```
<system.web>
  <authorization>
<deny users="?" />
  </authorization>
</system.web>
```

```

<location path="Login.aspx">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>

<location path="About.aspx">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>

<location path="Default.aspx">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>

```

نکته : احمق های web

ما برای کدهای #c و آدرس دهی هایمان همیشه از آدرس دهی نسبت به root استفاده می کنیم و حتی من در web.net از آدرس دهی نسبت به root حتی در web.config استفاده کردیم و آن احمقی که این location را طراحی کرده است حواسش به این قضیه نبوده است بدین صورت که شما باید از آدرس دهی نسبت به root استفاده بکنید ولی /~ را نباید بگذارید که اگر این کار را بکنید خطا می دهد لذا نباید بگذارید ولی معنی اش همان است.

نکته : path در location می تواند به Folder ختم شود یا به فایل ، اگر به Folder شود یعنی به تمام اطلاعات داخل آن فولدر این خصوصیت را بگیرد.

دستور یا کد برای Login کردن

در Login.aspx.cs

1 - اگر user name , password درست بود session ["UserId"], session ["UserName"] و هر چی را که دلمان بخواهد ست می کنیم.

2 - دستوری که باعث می شود ما به authentication , authorization sign in بشویم این دستور است که پارامتر اول username است و پارامتر دوم را می توانید false بگذارید یا برابر با محتوی متغییر checked remember me (مرا بیاد داشته باش) که توسط کاربر انتخاب می شود.

```

protected void btnSubmit_Click(object sender, EventArgs e)
{
    if (Page.IsValid)
    {
        Session["UserId"] = 12345;
        Session["Username"] = txtUsername.Text;
    }
}

```

```
//Session["UserId"] = oUsersRow.UserId;
//Session["Username"] = oUsersRow.Username;

// For Sign In
```

```
System.Web.Security.FormsAuthentication.RedirectFromLoginPage
(txtUsername.Text, chkRememberMe.Checked);
```

```
}
}
```

در برنامه UserLoginStatus.aspx.cs

وقتی می خواستیم بفهمیم user ای Login کرده است چک می کردیم که session[userid] != null باشد و حالا باید این مبحث جدید را با این تلفیق کنیم. در کنترل [UserLoginStatus.aspx.cs](#) ما این را انجام می دهیم:

```
Context.User.Identity.IsAuthenticated=true
```

این یعنی user login کرده است.

```
Context.User.Identity.Name این همان user name است
```

نکته : حتما باید session را با Authentication تلفیق کنید. session یک timeout ای دارد که 20 دقیقه است و authentication یک timeout ای دارد که 30 دقیقه است و این دو را باید به صورت خطی ببینید نه هندسی، که در هندسی یعنی باید 1.5 برابر آن باشد و خطی یعنی همیشه 10 دقیقه بیشتر از آن باشد، پس باید هر دوی اینها را mix چک بکنید در غیر این صورت کاربر رفتار طبیعی به از گذشت زمان session نمی بیند

```
public partial class UserControl1_UserLoginStatus :
System.Web.UI.UserControl
{
protected void Page_Load(object sender, System.EventArgs e)
{
//یعنی کاربر وارد پایگاه شده است

if ((Context != null) &&
(Context.User != null) &&
(Context.User.Identity != null) &&
(Context.User.Identity.IsAuthenticated) &&
Session["UserId"] != null)
{
lnkLogin.Visible = false;
lnkLogout.Visible = true;
lblWelcome.Visible = true;

//string strUsername = Context.User.Identity.Name;

string strUsername =
Session["Username"].ToString();
lblWelcome.Text =
string.Format(" | Welcome {0}!", strUsername);
}
}
```

//یعنی کاربر هنوز وارد پایگاه نشده است

```

else
{
    lnkLogin.Visible = true;
    lnkLogout.Visible = false;
    lblWelcome.Visible = false;
    lblWelcome.Text = string.Empty;
}
if (Page.IsPostBack == false)
{
    Initialize();
}
}

private void Initialize()
{
}
}

```

در برنامه Logout

در logout ما کار خاصی نمی کردیم فقط session را `Session.Clear()` یا `Session.Abandon()` می کردیم. چون ما مفهوم authentication & authorization داریم هم باید اولی اتفاق بیفتد و هم مفهومی به نام sign out داریم که باید انجام شود.

```

public partial class Logout : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        // ترتیب نوشتن دستورات ذیل اهمیت دارد

        // For Sign Out
        System.Web.Security.FormsAuthentication.SignOut();
        // Solution (1)
        Session.Clear();
        // /Solution (1)

        // Solution (2)
        //Session.Abandon();
        // /Solution (2)

        // Solution (3)
        //Session.Clear();
        //Session.Abandon();
        // /Solution (3)

        Response.Redirect("~/Login.aspx", false);
        //Response.Redirect("~/Default.aspx", false);
    }
}

```

یک سری مطلب:

بعضی از سایت ها تحت یک شرایطی https می شوند یعنی secure می شوند که باید certificate بخیریم و روی server نصب کنید.

مطلب بعدی اینکه :

در مورد check box مربوط به remember me در فرم login اگر tick بخورد آن cookie ای که روی دستگاه ساخته می شود به آن permanentcookie گفته می شود.

مطلب بعدی stylesheet: برای اینکه همه ی label های پروژه را قابل کنترل داشته باشیم از آنجایی که label تبدیل به span می شود در CSS از این دستور استفاده و یک class برای label ها درست می کنیم.

```
span.label
```

```
{
    font-weight: bold;
}
```

```
span.validator
```

```
{
    font-weight: bold;
}
```

ما برای دو textbox مربوط به password , username کلاس متفاوت تعریف می کنیم برای اینکه بتوانیم از شکلکهای زیبای آدمک به عنوان کاربر و شکلک کلید به عنوان password استفاده کنیم و با layout متفاوتی ، از آنجایی که textbox در html تبدیل به input می شود

```
input.username
```

```
{
    color: Red;
    font-weight: bold;

    border-top: 0px solid;
    border-left: 0px solid;
    border-right: 0px solid;
    border-bottom: 1px solid Gray;
    background-color: Transparent;

    padding-left: 17pt;
    background-repeat: no-repeat;
    background-attachment: scroll;
    background-position: left center;
    background-image: url(../Images/Username.png);
}
```

```
input.password
```

```
{
    color: Red;
    font-weight: bold;

    border-top: 0px solid;
    border-left: 0px solid;
    border-right: 0px solid;
    border-bottom: 1px solid Gray;
    background-color: Transparent;

    padding-left: 17pt;
    background-repeat: no-repeat;
```

```
background-attachment: scroll;
background-position: left center;
background-image: url(../Images/Password.png);
}
```

این کلاس هم برای فرمهای استاندارد ورود اطلاعات مناسب است مثل فرم مرخصی

```
input.textBox
{
    color: #333333;
    font-weight: bold;

    border-top: 0px solid;
    border-left: 0px solid;
    border-right: 0px solid;
    border-bottom: 1px solid Gray;
    background-color: Transparent;
}
```

مطلب بعدی: disable کردن ویژگی browser به نام autoCompleteType (ذخیره اطلاعات قبلی تایپ شده و نمایش آن بمانند یک لیست به کاربر هنگام تایپ) و ما نباید این خصوصیت را در صفحه change, registration, login, password این خصوصیت را داشته باشیم

```
<asp:textbox id="txtUsername" runat="server" maxlength="20"
cssclass="username" accesskey="1" autocomplete="Disabled" />
```

تعریف menu

اگر منوهای شما fix است و طبق شرایط در زمانهای مختلف لازم نیست تغییر کند از هر منویی که بخواهید می توانید استفاده کنید و یک سایتی هست که منوهای کلاسیک زیبایی برای این کار دارد به نام templatedl.com اگر منوهایتان دینامیک باشد باید به طور قطع از منوهای ماکرو سافت استفاده کنید.

درس 1: چگونه می توانیم منو داشته باشیم؟

خیلی ساده است از دستور <asp: menu> و به اختصار نام آن هم mnu... می گذاریم و داخلش یک سری items داریم و می توانیم یک سری menuItem تعریف کنیم که اینها هم می توانند یکسری بچه داشته باشند.

نکته: IIS نمی تواند بیش از یک پروژه را در mode run debug اجرا کنید

```
<asp:menu id="mnuMain" runat="server">
<items>
    <asp:menuItem text="Menu Item 1" navigateurl="~/Default.aspx" />
    <asp:menuItem text="Menu Item 2" navigateurl="~/Default.aspx" />
</items>
</asp:menu>
```

درس 2: چگونه می توانیم submenu داشته باشیم؟ اگر در داخل یک item شما menuItem ای بگذارید که داخلش menuItem است تبدیل به submenu می شود. در مثال زیر 3 menu item دارای بچه خواهد بود.


```

<asp:menu id="mnuMain" runat="server" >
<items>
<asp:menuitem text=" Menu Item 1" navigateurl="~/Default.aspx" />
<asp:menuitem text=" Menu Item 2" navigateurl="~/Default.aspx" />
<asp:menuitem text=" Menu Item 3" navigateurl="#">
<asp:menuitem text=" Item 31" navigateurl="~/Default.aspx" />
<asp:menuitem text=" Item 32" navigateurl="#">
<asp:menuitem text=" Item 321" navigateurl="~/Defau.aspx"/>
<asp:menuitem text=" Item 322" navigateurl="~/Defau.aspx"/>
<asp:menuitem text=" Item 323" navigateurl="#">
<asp:menuitem text=" Item 3231" navigateurl="#">
<asp:menuitem text=" Item 32311" navigateurl="~/Defau.aspx"/>
</asp:menuitem>
</asp:menuitem>
</asp:menuitem>
<asp:menuitem text="Menu Item 33" navigateurl="~/Default.aspx" />
</asp:menuitem>

<asp:menuitem text="Menu Item 4" />
<asp:menuitem text="Menu Item 5" />
</items>
</asp:menu>

```

درس 3 و 4: ادبیات ماکروسافت از منوها به این شکل است اولین سری منوهایی که می بیند static است افقی یا عمودی ، از این به بعد submenu هایی که می بیند dynamic است. مثال بالا یک لایه استاتیک دارد و 4 لایه دینامیک دارد حالا ببینیم آیا همه لایه ها را نشان میدهد یا خیر؟ لایه چهارم را نشان نمی دهد و علتش این است که منوها یک property به نام maximumdynamicdisplaylevels دارند که default آن 3 است و باید آن را به تعداد دلخواه لایه هایمان افزایش دهیم.

```

<asp:menu id="mnuMain" runat="server" maximumdynamicdisplaylevels="4">

```

تا 6 لایه بیشتر را نشان نمی دهد.

درس 5: جهت منو است که به صورت default جهت آن عمودی است و ما می توانیم آن را افقی کنیم و این فقط مربوط به بخش static menu دارد.

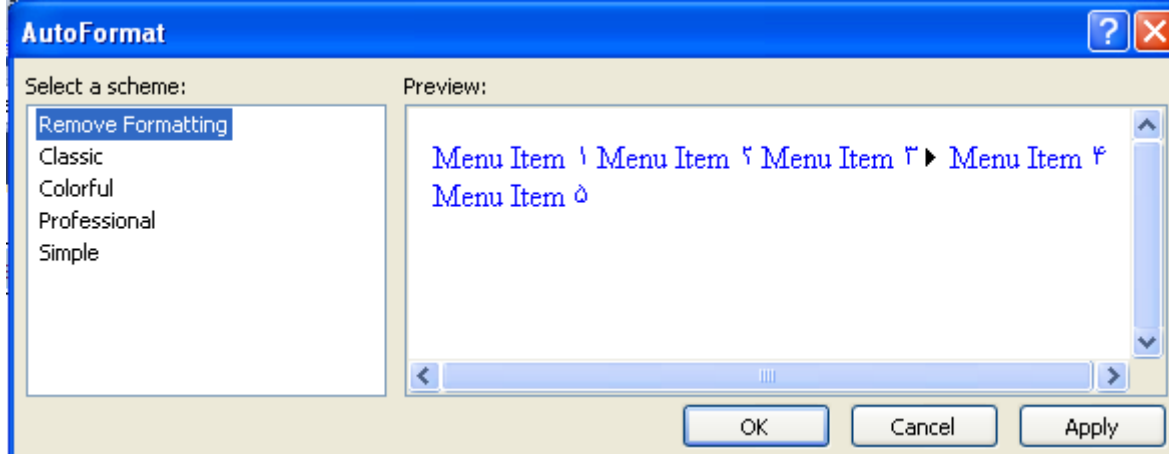
```

<asp:menu id="mnuMain" runat="server" maximumdynamicdisplaylevels="4"
orientation="Horizontal">

```

درس 6: منو بغیر از آیتم یک چیزهایی برای گوگوری مگوری کردنش دارد که گاهی به TEMPLATE و گاهی به STYLES ختم می شود و یک چیزی هم دارد به نام DATABINDING که TEMPLATE و DATABINDING جزو برنامه مان نیست.

برای style بهتر است که وارد design شویم و بعد روی منویمان برویم و روی فلش آن کلیک بکنیم و روی گزینه AutoFormat برویم و می بینید که ماکروسافت زحمت کشیده و یک سری چیزها را برایتان گذاشته است. مدل : simple colorful,,professional... که ما مدل colorful را انتخاب میکنیم



درس 7 و 8: فرض کنید که آمده اید روی منو آبتم 3 و می خواهید منو آبتم 35 را انتخاب کنید که معمولا ما می خواهیم با خط راست راست به آن برسیم ولی به محض اینکه از روی منو آبتم 3 خارج می شویم زیر منو ناپدید میشود و مجبور می شویم دنده ای به سمت آن برویم که برای این حالت منو یک property به نام disappearafter دارد که واحد آن میلی second است. و با این پراپرتی ما می گوییم که لطفا به اندازه 400 ms آن submenu را باز بگذار که اگر من خواستم رویش بروم ، بتوانم. ما چیزی داریم به نام DynamicHorizontalUpset که فاصله بین submenu و menu ها است.

```
<asp:menu id="mnuMain" runat="server" maximumdynamicdisplaylevels="4"
disappearafter="400" dynamichorizontaloffset="2"
font-size="0.8em" orientation="Horizontal">
```

اگر عدد disappear را صفر بگذارید و با هر سرعتی که حرکت کنید نمی توانید روی زیر منو مربوطه بروید و فقط در firefox می توانید رویش بروید. بین 400 تا 600 بگذارید.

درس 9: این درس خاص منو نیست، هر کدام از منو ایتم ها در html تبدیل به anchor می شوند هر وقت asp:hyperlink دیدید یا تگ a را دیدید یا هر وقت menu item دیدید این درس برایش صدق می کند. ما در windows application ، submenu همان هیچوقت هیچ کاری نمی کند یعنی item هایی که بچه دارند اگر رویشان کلیک کنید هیچ اتفاقی نمی افتد و فقط روی بچه هایشان اتفاق می افتد. و ما برای همین آنها را که بچه ندارند اسمشان را itm... و آنها را که بچه دارند نامشان را mnu... می گذاریم و ما فقط برای itm...ها لینک می گذاشتیم ولی در web این صدق نمی کند مثلا ما submenu ای به نام محصولات داریم که اگر روی خودش برویم همه محصولات مربوط به زیر منوهایش را درهم می آورد و اگر ما نخواهیم submenu هایمان جایی بروند نباید آن را خالی بگذاریم و بهتر است از number sign استفاده کنیم.

```
<asp:menuitem text="Menu Item .." navigateurl="#">
```

درس آخر:

چند تا خورده ریز :

دیگر پراپرتی های منوها:


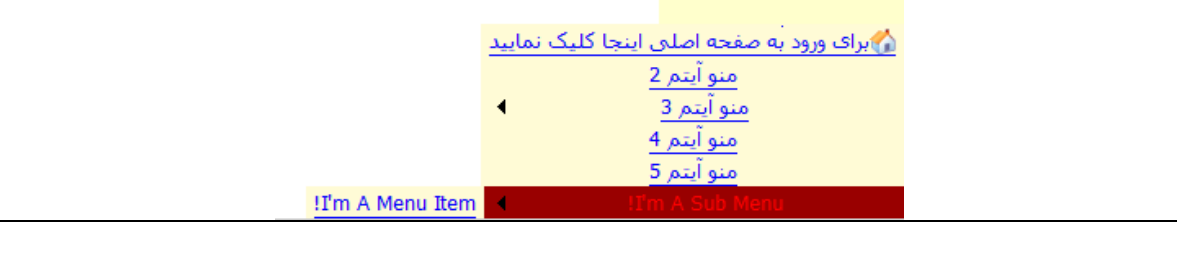
- text :
- چیزی که تویش داده می شود.
- Navigate Url :
- آدرس جایی که باید برود.
- Enable :
- اگر false شود دیده می شود ولی کار نمی کند.
- ImageUrl :
- دیدید که بعضی از سایتها کار انتخاب home از menu عکس خانه است این همان است.
- Popup image url :
- یک عکس برجسته از imageurl در نظر بگیرید که وقتی با موس رویش بروید عکس برجسته را نشان بدهد.
- Separated image url :
- برای sumbemuهایی در نظر بگیرید که خطی بین آنها ایجاد می کند که بین منوها فاصله ای ایجاد شود .
- Target :
- دو مقدار دارد :
- 1 -Self_ صفحه می رود و صفحه جدید می آید
- 2 -Blank_ صفحه جدید در یک پنجره جدید باز می شود (new tab or window).
- Tooltip :
- با موس رویش بروی یک توضیحی را نشان می دهد.
- Selectable :
- اگر منوها را selectable بکنید:
- اگر true کنید وارد هر صفحه ای که بشوید آیتم مربوط به آن صفحه را highlight شده نشان می دهد.
- Value :
- اگر بخواهیم منوها را run time اضافه و کم کنیم آنوقت VALUE به درد می خورد

ADD RUN TIME MENU

چگونه RUN TIME منو اضافه کنیم .

ما یک event PreRender داریم که قبل از تبدیل به html شدن اتفاق می افتد. که ما با این در تفکر 2 منوها با آن آشنا می شویم ولی در تفکر اول ما نیاز به prerender نداریم و در page Load در بخش initialize تغییر را اعمال کنیم .

Main menu یعنی منو main یک collection ای دارد که ما می توانیم به آن منو addAT کنیم.

	<p>منو قبل از اضافه شدن یک منو آیتم و زیر منو بصورت آنلاین در کد نویسی</p>
<pre>private void Initialize() { System.Web.UI.WebControls.MenuItem oMenuItem =new System.Web.UI.WebControls.MenuItem ("I'm A Sub Menu!", "10", string.Empty, "#", string.Empty); mnuMain.Items.Add(oMenuItem); //mnuMain.Items.AddAt(0, oMenuItem); System.Web.UI.WebControls.MenuItem oSubMenuItem =new System.Web.UI.WebControls.MenuItem ("I'm A Menu Item!", "34", string.Empty, "http://www.y.com", "_blank"); oMenuItem.ChildItems.Add(oSubMenuItem); }</pre>	
	

جلسه چهارم

ایجاد منو با استفاده از Web.sitemap

منوی شماره 2 که الان گفته می‌شود، یک حسن و سه ایراد دارد:

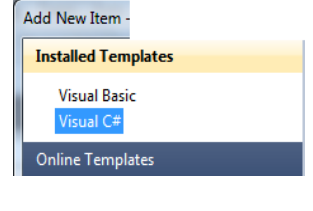
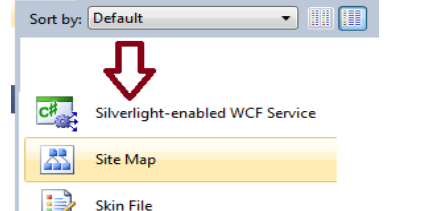
حسن این menu

برای چند زبانگی بسیار خوش دست است.

ایرادات این menu

- 1 - اولین ایراد اینکه از یک نقطه شروع می‌شود. یعنی اول از یک sub menu شروع می‌شود و بعد منوی اصلی ظاهر می‌شود.
- 2 - دومین ایراد آن اینست که قرتی بازی ندارد. این منویی که داریم، با ویژگیهایی که داریم، نمی‌توانیم بگوییم، عکس آن و pop up image آن چیست؟
- 3 - سومین ایراد اینکه نمی‌توانیم دو تا آیتم داشته باشیم، که به یکجا اشاره کند. در درس قبل تمام آیتمهای منو این خاصیت را داشت، ولی این منو این اجازه را نمی‌دهد. مثلاً در این حالت منوی سبک 2 یک زیر منو داریم مثلاً تنظیمات که برای admin سیستم است و 50 کار می‌کند. مثلاً یکی از آنها backup گیری از بانک اطلاعاتی است. یک زیرمنوی دیگر به نام تنظیمات داریم. که آن مال admin نیست و سبک تر است. پس یک منویی داریم که دو تا زیرمنو به یک صفحه اشاره می‌کند و اینکه این منو این خاصیت را ندارد، خطا ایجاد می‌کند.
در اکثر کتابها به آن اشاره نمی‌شود و اگر اشاره هم شود، راه حلی برای آن ارائه نمی‌شود. منوی سبک دو، سه ایراد بزرگ دارد، و ما می‌خواهیم حل کنیم، که چی؟ چه قصدی داریم؟ این امکان را در workshop web در قالب یک library ارائه می‌دهم که به صورت اتوماتیک منو ها را visible و invisible کنیم. اگر این library را بنویسیم، این نوع منو جذاب می‌شود.

مراحل:

		<p>Add new item - 1 web.sitemap این ایتم هم نامش و هم پسوندش مهم است</p>
---	---	--

ما سه تا فایل داریم که اسم و پسوند هر سه رزرو است
Web.config ○
Global.asax ○
Web.sitemap ○

web.sitemap را اضافه می‌کنید، داخل آن که می‌روید، با یک فایل xml ای مواجه می‌شوید. که داخلش یک root نام siteMap است که آن هم المنت هایی به نام siteMapNode دارد، منوهای ما چیزی به نام menu item داشتند ولی اینجا siteMapNode داریم. وقتی که منوی خود را به یک شکلی به این فایل bind می‌کنیم، در مرحله اول sitemap node ها تبدیل به منوآیتم و سپس آنها تبدیل به HTML می‌شود.

معماری به این شکل است که اول یک siteMapNode می‌گذاریم و سپس node ها را تویش می‌گذاریم برای همین است که می‌گویند منو از یک نقطه یعنی از یک submenu شروع می‌شود. پس sitemapnode که بچه داشته باشد، تبدیل به submenu می‌شود.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Main Menu" description="Main Menu">
    <siteMapNode roles="" url="~/Default.aspx" title="خانه" description="Home"/>
    <siteMapNode url="~/About.aspx" title="ما درباره" description="About"/>
    <siteMapNode url="~/Login.aspx" title="پایگاه به ورود" description="Login"/>
    <siteMapNode title="مرتبط های پایگاه">
      <siteMapNode url="http://www.DBX.com" title="DBX!"/>
      <siteMapNode url="http://www.IranianExperts.com"
        title="IranianExperts"/>
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

2 - یک menu در صفحه می‌اندازیم و سپس از sitemapdatasource استفاده می‌کنیم

قبلاً وقتی که menu به صفحه اضافه می‌کردیم، منو را داخل صفحه می‌انداختیم و سپس آیتمهای آن را ست می‌کردید. ولی آیتم های آن را set نمی‌کنیم و در اینجا یک site map datasource به صفحه اضافه کنید و چون یک sitemapdatasource بیشتر نداریم اسمش را mySiteMapDataSource می‌گذاریم و این sitemapdatasource همیشه اطلاعاتش را از Web.sitemap می‌خواند، و منو باید خوراک خود را از این بگیرد. باید برای اینکار DataSourceID منو را مساوی با نام sitemap Datasource بگذارید.

```
<div id="divMenu" dir="rtl">
  <asp:menu id="mnuMain" runat="server" datasourceid="mySiteMapDataSource"
    orientation="Vertical" maximumdynamicdisplaylevels="4" disappearafter="500"
    onprerender="mnuMain_PreRender">
  </asp:menu>

  <asp:sitemapdatasource id="mySiteMapDataSource" runat="server"
    showstartingnode="false" />
</div>
```

دوباره وارد web.sitemap شوید. یک property به نام title دارد که معادل text است و description معادل tooltip است و url معادل navigateurl است و دیگر هیچ اثری از قرتی بازی نیست حالا ایرادهای آن را بررسی می‌کنیم:

اول باید یک sitenode بندازیم و سپس داخل آن آیتم درست کنید. یعنی از یک submenu شروع می شود و وقتی با موس رویش می روید آنوقت منوی اصلی دیده می شود.

ایراد دوم آن، اینکه نمی توان دو تا site map node داشته باشید که به یک جا می روند و به یک صفحه اشاره می کنند. اصلاً مهم نیست که همتراز باشد، یا خیر. در هر صورت اگر دو آیتم به یک صفحه اشاره کند، خطا می دهد.

3 - اولین error را حل کنیم: گزینه showstartingnode را در تگ sitemapdatasource برابر false قرار دهید.

```
<asp:sitemapdatasource id="mySiteMapDataSource" runat="server"
showstartingnode="false" />
```

4 - مشکل دوم: می توانید Site map های متفرقه درست کنید. می توانید داخل sitemap اصلی آن اعلام کنید که از کدام site map بخواند.

اگر هریک از دو تا site map node ای که به یک جا می روند و به یک صفحه اشاره می کنند داخل یک فایل sitemap دیگر بگذارید و آن را در sitemap اصلی include کنید، دیگر خطا نمی دهد و کار می کند.

```
<!--
<siteMapNode url="" title="مدیر تنظیمات">
<siteMapNode url="~/DropDataBase.aspx" title="اطلاعاتی بانک حذف" description="About"/>
<siteMapNode url="~/BackupDataBase.aspx" title="اطلاعاتی بانک از گیری پشتیبان"
description="About"/>
</siteMapNode>
-->

<!--
<siteMapNode url="" title="سرپرست تنظیمات">
<siteMapNode url="~/BackupDataBase.aspx" title="اطلاعاتی بانک از گیری پشتیبان"
description="About" />
</siteMapNode>
-->
```

دستور include آن به این صورت است:

```
<!--
Note: In below line never write [url], [title], [description] and [roles] value!
-->
<siteMapNode siteMapFile="~/Administrator.sitemap"/>
<!--
Note: In below line never write [url], [title], [description] and [roles] value!
-->
<siteMapNode siteMapFile="~/Supervisor.sitemap"/>
```

یعنی می توانید در همانجا roles را مشخص کنید و با visible و invisible شدن آن را کنترل کنید.

```
Administrator.sitemap X
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
4
5 <siteMapNode url="" title="تنظیمات مدیر">
6 <siteMapNode url="~/DropDataBase.aspx" title="حذف بانک اطلاعاتی" description="About" />
7 <siteMapNode url="~/BackupDataBase.aspx" title="پشتیبان گیری از بانک اطلاعاتی" description="About" />
8 </siteMapNode>
9
10 </siteMap>
11
```

```

1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
4
5 <siteMapNode url="" title="تنظیمات سرپرست">
6 <siteMapNode url="About.aspx" title="پشتیبان گیری از بانک اطلاعاتی" description="About" />
7 </siteMapNode>
8
9 </siteMap>
10

```

5 - مشکل سوم (نداشتن قرتی بازی) ما یک Event به نام pre Render داریم. Pre Render دقیقاً مرحله قبل از تبدیل شدن به HTML است. وقتی که فقط منو سبک شماره یک دارید، فقط menu item داریم که در مرحله Pre render و بعدش تبدیل به HTML می شود.

در خصوص site map اول باید نودها تبدیل به Menu item شوند و بعد Pre render و تبدیل به HTML شود. در واقع یک مرحله اضافه تر دارد. باید در Event pre render یک عدد loop ایجاد کنیم و با دیدن نام مورد نظر، Negative URL مورد نظر، تنظیمات را انجام دهیم.

در Learn12.aspx در رویداد prerender نشان می دهیم که چگونه می توانیم روی item های menu حرکت کنیم و اگر خواستیم خصوصیت آنها را عوض کنیم:

```

protected void mnuMain_PreRender(object sender, System.EventArgs e)
{
    int intIndex = 0;
    while (intIndex <= mnuMain.Items.Count - 1)
    {
        System.Web.UI.WebControls.MenuItem oMenuItem =
            mnuMain.Items[intIndex];

        if (oMenuItem.NavigateUrl.IndexOf("~/Default.aspx") >= 0)
        {
            oMenuItem.ImageUrl = "~/Images/Home.gif";
            break;
        }
        intIndex++;
    }
}

```

مبحث چند زبانی

تصورات غلطی که در مورد چند زبانی وجود دارد، این است که خیلیها فکر می کنند که یک سایت را با یک زبان درست می کنند و کار تمام شد، همان را کپی و paste می کنند و با زبانهای دیگر ایجاد می نمایند. پس عملاً چندین سایت وجود دارد و مدیریت آن بسیار دشوار است.

وقتی که زبان تغییر می کند، باید همه چیز از Layout، ظاهر، کلیه پیغامها و ... فارسی شود.

1- یک پروژه جدید درست نمایید.

2- در داخل default یک منو بیندازید. در اولین قدم می‌خواهیم منو ما دو زبانه شود و بعد خرد خرد بقیه چیزها را نیز یاد می‌گیریم.

```
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Menu ID="MainMenu" runat="server" BackColor="#FFFBD6"
        DynamicHorizontalOffset="2" Font-Names="Verdana" Font-Size="0.8em"
        ForeColor="#990000" Orientation="Horizontal"
        StaticSubMenuIndent="10px" DataSourceID="mySiteMapDataSource">
        <DynamicHoverStyle BackColor="#990000" ForeColor="White" />
        <DynamicMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
        <DynamicMenuStyle BackColor="#FFFBD6" />
        <DynamicSelectedStyle BackColor="#FFCC66" />
        <StaticHoverStyle BackColor="#990000" ForeColor="White" />
        <StaticMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
        <StaticSelectedStyle BackColor="#FFCC66" />
      </asp:Menu>
      <asp:SiteMapDataSource ID="mySiteMapDataSource" runat="server"
        ShowStartingNode="false" />
    </div>
    <br />
    <asp:HyperLink ID="lnkFarsi" runat="server"
      NavigateUrl="~/ChangeCulture.aspx?Culture=fa-IR"
      Text="فارسی"></asp:HyperLink> |
    <asp:HyperLink ID="lnkEnglish" runat="server"
      NavigateUrl="~/ChangeCulture.aspx?Culture=en-US"
      Text="English"></asp:HyperLink>
  </form>
</body>
```

استفاده از فایل‌های Resource

در web.siteMap در siteMapNode ها title ها را به صورت عجیب می‌بینید. یعنی چه؟ به جای اینکه، عنوان منو را به صورت fix بنویسید، برو از فایل منبع ... مقدار این متغییر را پیدا کن و به جای عنوان منو بنویس. دستور آن به صورت ذیل می‌باشد:

```
<siteMap enableLocalization="true" xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url=""
    title="$Resources:SiteMap,MainMenuTitle"
    description="$Resources:SiteMap,MainMenuDescription">
    <siteMapNode url="~/Default.aspx"
      title="$Resources:SiteMap,HomeTitle"
      description="$Resources:SiteMap,HomeDescription"/>
    <siteMapNode url="~/Login.aspx"
      title="$Resources:SiteMap,LoginTitle"
      description="$Resources:SiteMap,LoginDescription" />
    <siteMapNode url="~/About.aspx"
      title="$Resources:SiteMap,AboutTitle"
      description="$Resources:SiteMap,AboutDescription" />
  </siteMapNode>
</siteMap>
```

ما یک سری فولدر با شخصیت داریم که به مرور بعضی از آنها را یاد گرفتیم. این فولدرها 8 مورد هستند. خیلی از این فولدرها را وقتی که انتخاب می کنید، دیگر از لیست حذف می شود، چون فقط یک مورد از آن در کل پروژه می تواند وجود داشته باشد. یکی از فولدرهای با شخصیت App_GlobalResources است که فایل های Resource در آن قرار می گیرد. یک Resource file با پسوند rc انتخاب می کنید. چون این فایل resource را برای site map درست می کنیم، پس بهتر است که اسم آن را sitemap.rc می گذاریم. به هیچ عنوان در ستون Name حق نوشتن space و نقطه ندارید ولی در value هر چیزی خواستید بنویسید. Comment به هیچ دردی نمی خورد. ولی می توانید بعداً اینها را در excel وارد کنید و به مترجم دهید، تا Value های مختلف را به زبانهای مختلف استخراج کند، پس بهتر است که قسمت comment را پر نمایید.

Name	Value	Comment
AboutDescription	About Page	
AboutTitle	About Page	
HomeDescription	Home Page	
HomeTitle	Home Page	
LoginDescription	Login Page	
LoginTitle	Login Page	
MainMenuDescription	Main Menu	
MainMenuTitle	MainMenu	
*		

Resource file ها به دو دسته slave , Master تقسیم می شوند، آنهایی که Culture دارند slave و آنهایی که ندارند Master هستند. Culture یا دو حرفی است یا پنج حرفی. اگر دو حرفی باشد، فقط زبان است. و اگر 5 حرفی باشد، مثل Fa-IR یا en-Uk و ... ، هم زبان در آن مشخص است و هم کشور. دو حرفی فقط زبان است و 5 حرفی زبان و کشور است. همیشه دو حرف اول 5 حرفی با کوچک است و بعد از آن یک علامت - و دو حرف بعدی که country است، با حروف بزرگ است. فایل resource ای که culture ندارد، Master است، بهتر است که Master به زبان انگلیسی باشد ولی در Slave از زبانهای دیگر استفاده کنید. باید Master را حتماً داشته باشید. باید در Master همه کلیدهایی که دارید به درستی تعریف شود. همه کلیدهای منو به درستی باید در Master تعریف شود. اگر کم بدهید و غلط بدهید، خطا می دهد. پس تمام کلیدها به درستی و بدون غلط دیکته ای آورده شود. در slave می توانید بعضی از کلیدها را نداشته باشید وقتی که Culture شما از انگلیسی به فارسی تغییر می کند، سراغ site map فارسی می رود، دنبال تک تک کلیدها می گردد، ولی مثلاً یک سری از کلیدها را نمی بیند و آن کلیدهایی را که می بیند از slave می خواند و ما بقی را از Master می خواند.

Name	Value	Comment
AboutTitle	صفحه درباره	
HomeTitle	صفحه اصلی	
LoginTitle	صفحه ورود	
MainMenuTitle	منوی اصلی	
*		

ولی از کجا بفهمد که هر چه را که در Title می بیند، عیناً بر روی منو بنویسد، یعنی از کجا بفهمد که این دستورات معنی دارد. باید enable localization را true نماید.

```
<siteMap enableLocalization="true" xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
```

ما می خواهیم وقتی که کاربر وارد سایت شد، اول سایت را به زبان فارسی ببیند، بعد که دکمه انگلیسی را زد، انگلیسی ببیند و ...

برای این کار باید کلاس مربوطه را درست نماییم. در AppCode یک فولدر با عنوان MyWebApplication درست می کنیم و بعد کلاسها را داخل آن می نویسیم. سعی کنید برای کلاسهای از نامهای General استفاده کنید، که در همه جا کاربرد داشته باشد.

برخلاف windows application در web application وقتی که یک class را به web site اضافه می کنید، برای آن name space نمی گذارد. لذا خودتان Name Space درست کنید و کلاستان را داخل آن قرار دهید.

نکته: اگر قرار باشد تمام صفحات از یک قانون طبیعت کنند، یک کلاس به نام Base Class درست کنید و آن را از Page ، Inherit کنید، و داخل آن هر چیزی می خواهید بنویسید. سایتی که اجازه دسترسی ندارد و به کشور ما سرویس نمی دهند، در تک تک صفحات سایت دستور چک کردن IP و عدم اجازه دسترسی رنج آدرس IP ایرانیها را تعریف نمی کنند، بلکه یک کلاس برای خود درست می کنند و آن را از Page ، Inherit می کنند و تمام صفحات را از آن Inherit می کنند.

برای چند زبانی، ابتدا یک کلاس به نام Base Page درست می کنیم و این کلاس از کلاس Page به ارث می برد. همانطور که قبلاً گفتیم، یک namespace با نام myWebApplication تعریف می کنیم.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for BasePage
/// </summary>
namespace myWebApplication
{
    public class BasePage : System.Web.UI.Page
    {
        public BasePage()
        {
            //
            // TODO: Add constructor logic here
            //
        }
    }
}
```

یک کلاس با نام LocalizePage تعریف کنید و در این کلاس یک تابع به نام Initialize Culture را over write کنید . نام این تابع را تایپ نکنید، بلکه باید آن را بیاورید.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

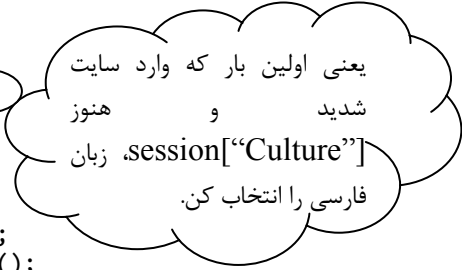
```

/// <summary>
/// Summary description for LocalizePage
/// </summary>
namespace myWebApplication
{
    public class LocalizePage : BasePage
    {
        public LocalizePage()
        {
            //
            // TODO: Add constructor logic here
            //
        }

        protected override void InitializeCulture()
        {
            base.InitializeCulture();
            if (Session["Culture"] == null)
            {
                Session["Culture"] = "fa-IR";
            }

            Culture = Session["Culture"].ToString();
            UICulture = Session["Culture"].ToString();
        }
    }
}

```



یعنی اولین بار که وارد سایت شدید و هنوز session["Culture"] زبان فارسی را انتخاب کن.

تا به امروز گفتید که اولین تابع، تابع Page Load است که صدا زده می‌شود. ولی بعضی از توابع قبل از Page Load صدا زده می‌شوند، یکی از این توابع initialize Culture است که قبل از Page load صدا زده می‌شود. اولین بار که کاربر وارد این سایت می‌شود، متغیر session مربوط به Culture آن خالی است و در اینجا است که باید مشخص کنید که سایت به صورت پیش‌فرض فارسی باشد، این کار را از طریق fa-iR تعیین می‌کند.

```

if (Session["Culture"] == null)
{
    Session["Culture"] = "fa-IR";
}

```

ولی برای عوض کردن culture، چه باید کرد؟ تنها جایی که حق دارید، Culture و UICulture را تغییر دهید، فقط در قسمت initialize Culture است. دستورات تغییر Culture به صورت ذیل می‌باشد، پس به هیچ عنوان حق ندارید در جایی غیر از این تابع، مقدار این دو متغیر را تغییر دهید، البته در جاهای دیگر می‌توانید بخوانید:

```

Culture = Session["Culture"].ToString();
UICulture = Session["Culture"].ToString();

```

پس نمی‌توانید در پشت link button مربوط به فارس و انگلیسی در صفحه default، مستقیماً culture را عوض کنید. خوب شاید بگویید که در پشت دکمه‌ها Session["Culture"] را تغییر می‌دهیم. ولی این هم امکان‌پذیر نیست، چون در این صورت با یک مرحله تأخیر، تغییر انجام می‌شود.

ببینیم سناریوی آن چگونه است، فرض کنید دکمه فارسی را کلیک می‌کنید (فرض کنید که در پشت این دکمه session["culture"] فارسی شود). اول client side validation و بعد Post Back ، بعد initialize culture ، قبل از Page Load صدا زده می‌شود و بنابراین دوباره Culture انگلیسی می‌شود، در حالیکه ما می‌خواستیم، culture فارسی شود. البته اگر یک بار دیگر کلیک کنیم درست می‌شود ولی این فایده ندارد. برای اینکه باید صبر کند تا load بعدی تابع initialize culture. بنابراین یک صفحه به نام Change Culture درست می‌کنیم و کاربر را به آن صفحه می‌فرستیم و دوباره او را به صفحه اولیه برمیگردانیم. برای همین از دستور HTTPReferere استفاده می‌کنید تا بفهمید که کاربر کجا بوده است و کاربر را به صفحه‌ای که بوده برگردانیم.

صفحه ChangeCulture.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class ChangeCulture : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string strHttp = Request.ServerVariables["HTTP_REFERER"];
        string strCulture = Request.Params["Culture"];
        Session["Culture"] = strCulture;
        Response.Redirect(strHttp);
    }
}
```

برای گرفتن پارامتر از آدرس URL از دستور [" "] Request.Param استفاده می‌کنیم. البته اگر به صورت حرفه‌ای می‌نویسید بهتر است که یک سری کنترل‌های اضافه، مثلاً کوچک و بزرگ بودن حروف مربوط به Culture و ... را هم بهتر است انجام دهید. کد آن به صورت ذیل می‌باشد:

```
public partial class ChangeCulture : System.Web.UI.Page
{
    protected void Page_Load(object sender, System.EventArgs e)
    {
        // *****
        string strCulture = Request.Params["Culture"];
        if (string.IsNullOrEmpty(strCulture))
        {
            Response.Redirect("~/Default.aspx", false);
            return;
        }
        // *****

        // *****
        string strReferer =
            Request.ServerVariables["HTTP_REFERER"];

        if (string.IsNullOrEmpty(strReferer))
```

```
{
    Response.Redirect("~/Default.aspx", false);
    return;
}
// *****

// *****
switch (strCulture.ToUpper())
{
    case "FA-IR":
    case "EN-US":
    {
        // *****
        strCulture =
            string.Format("{0}-{1}",
                strCulture.Substring(0, 2).ToLower(),
                strCulture.Substring(3, 2).ToUpper());
        // *****

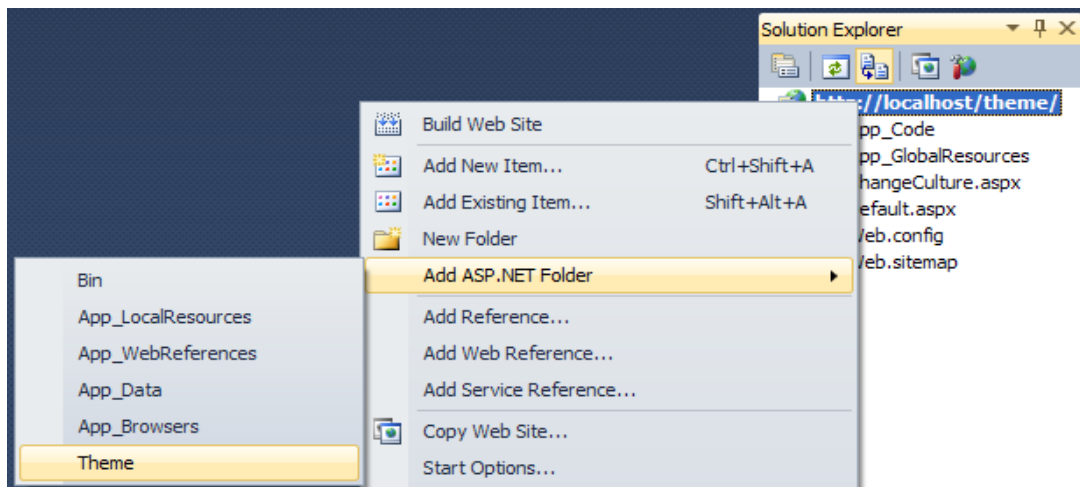
        // *****
        Session["Culture"] = strCulture;
        // *****

        // *****
        Response.Redirect(strReferer, false);
        return;
        // *****
    }

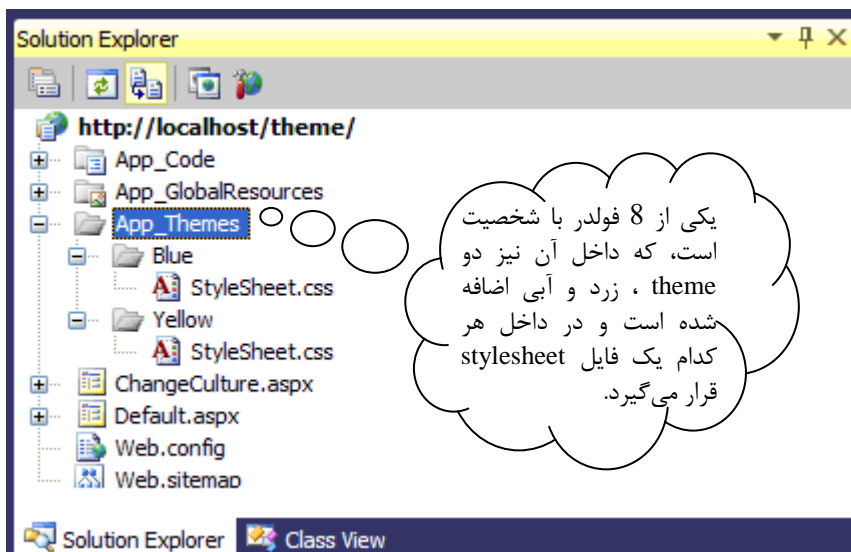
    default:
    {
        // *****
        Response.Redirect("~/Default.aspx", false);
        return;
        // *****
    }
}
// *****
}
```

جلسه پانزدهم

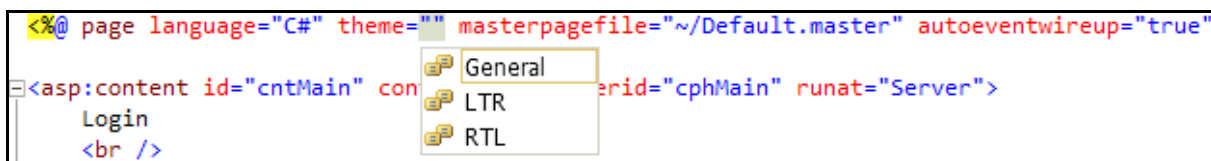
در خصوص stylesheet یک صفحه Css درست می کردیم و بعد link به صفحات می کردیم. الان این روش منسوخ شده است و از روش theme استفاده می شود. فولدر theme هم جزء فولدرهای باشخصیت است.



یک فولدر با شخصیت به نام theme ایجاد می کنیم و یک فولدر به نام App_Themes درست می شود. در حقیقت theme چیز خارق العاده ای نیست و فقط یک فولدر است. داخل آن نیز یک theme به نام Blue درست کنید و داخل آن یک فایل CSS و در قسمت فرمت body رنگ فونت را سیاه و رنگ پیش زمینه را آبی انتخاب کنید. بعد یک theme دیگر به نام yellow اضافه کنید و داخل آن یک فایل StyleSheet تعریف کنید و در آن رنگ زمینه را مثلاً زرد بگذارید.



Theme را نمی‌توانید به Master Page اختصاص دهید و باید به تمام صفحات (per page) اختصاص دهید. در Property صفحه باید کلمه theme را تایپ کنید و لیست theme هایی که تعریف کردید، می‌آید و شما می‌توانید انتخاب کنید.



اگر قرار باشد، به صفحه به صورت hard code یک theme اختصاص دهید، اصلاً جالب نیست، این جالب است که بتوانید بسته به شرایط و در داخل کد theme را تغییر دهید و یا اینکه خودتان به عنوان admin سایت theme را تغییر دهید. یعنی run time بسته به شرایط theme را تغییر دهید.

صفحه یا می‌توانید انگلیسی یا فارسی باشد. پس دو عدد theme درست کنید و یکی را right to left (RTL) و دیگری را left to right (LTR) بگذارید. حالا می‌خواهیم این theme را در چندزبانگی به کار ببریم.

وارد کلاس Initialized culture شوید، همان کلاس معروف خودمان. این تابع اولین تابعی است که اتفاق می‌افتد و اگر session ("Culture") به صورت Null بود، به آن مقداری یا زبانی را به صورت پیش‌فرض اختصاص می‌دهید. یک تابع به نام On Pre init دارید که بعد از initialized culture و قبل از Load اتفاق می‌افتد. لذا در اینجا مقدار session ("culture") را می‌خواهیم و می‌گوییم اگر فارسی یا عربی بود، theme راست به چپ را ست کن و در غیر اینصورت LTR را به عنوان theme ست کن.

در right to left و left to right باید همه چیز شامل direction و text-align و font و ... را ست نمایید.
کد مربوط به فایل stylesheet تعریف شده در theme LTR :

```
body
{
    direction: ltr;
    font-size: 10pt;
    text-align: left;
    font-family: Verdana, Sans-Serif;
}

#DTMenu
{
    float: left;
    clear: left;
}

#DTMain
{
    float: right;
    clear: right;
}
```

کد مربوط به فایل stylesheet تعریف شده در theme RTL :

```
body
{
    direction: rtl;
    font-size: 10pt;
    text-align: justify;
}
```



```

        font-family: Tahoma, Sans-Serif;
    }
#DTMenu
{
    float: right;
    clear: right;
}
#DTMain
{
    float: left;
    clear: left;
}

```

کد مربوط به کلاس LocalizedPage.cs و دو تابع initial Culture و on pre init مشهور:

```

namespace MyWebApplication
{
    public class LocalizedPage : BasePage
    {
        public LocalizedPage()
        {
        }

        protected override void InitializeCulture()
        {
            base.InitializeCulture();

            if (Session["Culture"] == null)
            {
                //Session["Culture"] = "en-US";
                Session["Culture"] = "fa-IR";
            }

            // نابع دقت کنند که دو دستور یل خارج از دستور فوق نوشته می شود.
            Culture = Session["Culture"].ToString();
            UICulture = Session["Culture"].ToString();
        }

        protected override void OnPreInit(System.EventArgs e)
        {
            base.OnPreInit(e);

            string strLanguage =
                Session["Culture"].ToString().ToUpper().Substring(0, 2);

            switch (strLanguage)
            {
                case "FA":
                case "AR":
                {
                    Theme = "RTL";
                    break;
                }

                //case "EN":
                //case "FR":
                default:
            }
        }
    }
}

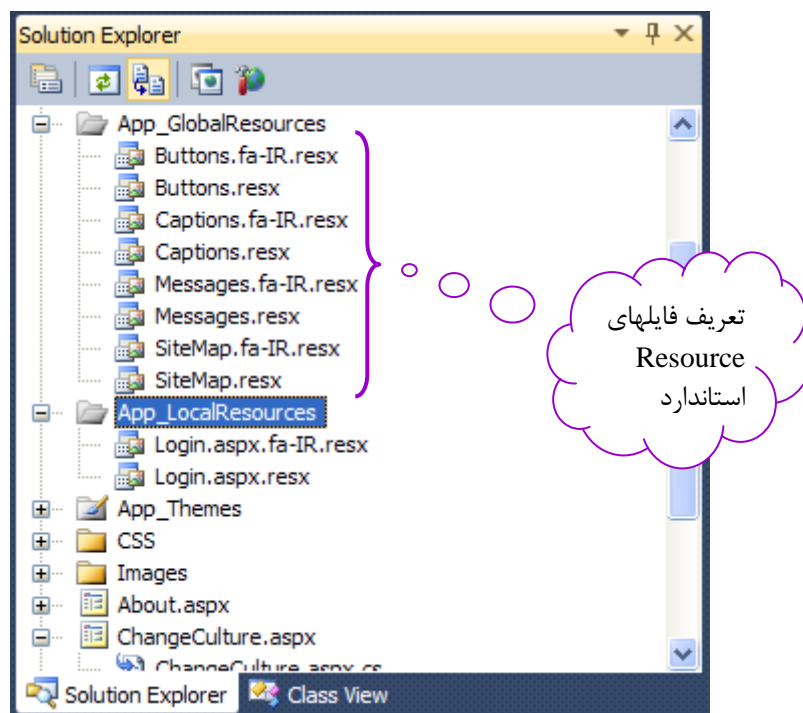
```

```

    {
        Theme = "LTR";
        break;
    }
}
}
}
}

```

در چندزبانگی Layout و منو را تغییر دادید و با صفحه Change Culture کار کردید و تغییرات را کلاً انجام دادید. حالا سراغ تغییر بقیه چیزها می‌رویم. برای کلیه سرستون grid ها و Label ها یک Resource به نام Captions و برای دکمه‌ها یک Resource به نام Buttons و برای Hyper Link ها یک Resource به نام HyperLinks و برای همه پیغامها یک Resource به نام Messages ایجاد کنید. این باعث می‌شود که پروژه شما استاندارد شود.



نکته: خیلی از چیزهایی که در چندزبانگی یاد می‌گیرید، حتی در پروژه‌های غیرچندزبانه هم کاربرد دارد.

مسئله اول: برای کلیه دکمه‌ها، Label ها، پیغامها، حتی اگر دو زبانه و چندزبانه هم ندارید، عنوان آنها را در resource بگذارید و از Resource بخوانید، تا با قوانین UML و دستورات RUP هم همخوانی داشته باشد. در واقع Resource ها همان Data Dictionary شما در RUP را تشکیل می‌دهد.

1.

مسئله دوم: مثلاً در ده تا صفحه کلمه شناسه کاربر وجود دارد. و در این صفحات Label ای دارید که username یا همان شناسه کاربر را در آن آورده‌اید و همه را شناسه کاربر گذاشته‌اید، و حالا کارفرما می‌گوید، شناسه کاربر غلط است و آن را شناسه کاربری تبدیل کن، در حالتی که از Resource استفاده نکرده‌باشید، باید در همه صفحات (همه 10 جا یا بیشتر) تغییر دهید ولی در صورتیکه Label ها را از Resource File خوانده باشید، کافی است که Resource را اصلاح کنید و کل پروژه اصلاح خواهد شد.

نکته: به عنوان آدم حرفه‌ای هر وقت می‌خواهید فیلدی را برای DataBase یا برای Global Resource در نظر بگیرید، از روی دیکشنری کلمه مناسب پیدا کنید. این خیلی زشت است که بعضی از فیلدهای بانک اطلاعاتی یا resource به صورت Penglish و یا با غلط دیکته‌ای نوشته شده است.

نکته: در مورد پیغامها، ما نام پیغامها را با معنی نمی‌گذارید. مثلاً نام کلید را WhatAreYourName نمی‌گذارم. به دو دلیل:
1- شاید خیلی طولانی شود.
2- شاید انگلیسی من اینقدر خوب نباشد که جمله مناسبی را به عنوان کلید پیدا کنم.

نکته: در پروژه‌های ویندوز و وب چهار نوع پیغام بیشتر نداریم.

1. error
2. warning
3. information
4. question

پس پیغامها را به صورت ذیل کلید گذاری می‌کنیم:

1. ErrorMessage001 , ErrorMessage002 , ...
2. WarningMessage001 , WarningMessage002 , ...
3. InfoMessage001 , InfoMessage002 , ...
4. QuestionMessage001 , QuestionMessage002 , ...

نکته: شاید پروژه‌ای داشته باشیم، که تعداد پیامها زیاد باشد. با توجه به نوع کدگذاری تعریف شده ما حداکثر 999 تا پیغام از هر نوع می‌توانیم داشته باشیم و شاید در پروژه‌های خیلی بزرگ کم باشد. برای رفع این مشکل و همچنین به دلیل قشنگ‌تر شدن پروژه، پیغامهای پارامتریک درست کنید. در تمام پیامهایی که یک فیلد لازم است که پر شود، یک پیام تعریف کنید و به آن پارامتر پاس کنید. بنابراین برای پر شدن یک فرم که 10 فیلد دارد که باید حتماً پر شود، 10 تا پیغام تعریف نکنید و یک پیغام تعریف کنید و نام فیلد را به صورت پارامتر به آن پاس کنید. شما می‌توانید به زیبایی پیامهای پارامتریک درست کنید.

نکته: با فرض اینکه پروژه شما چند زبانه هم نباشد و حتی همه پروژه را هم خودتان نوشته باشید، ممکن است بسته به شرایط پیغامهای متفاوتی به کار برده باشید و این باعث می‌شود که یکنواختی پروژه شما از بین برود. بنابراین می‌توانید پروژه که تمام شد، فایل Messages Resource را پرینت بگیرید و همه را با هم درست و یکنواخت کنید.

خواندن از فایل Resource

اگر بخواهید در قسمتی از تگ HTML و در کد HTML از یک فایل Resource بخوانید، به صورت ذیل می‌باشد:

```
<siteMapNode
    url=""
    title="$Resources:SiteMap,MainMenuTitle"
    description="$Resources:SiteMap,MainMenuDescription">

<asp:Label ID="lblUsername" runat="server" Text="<%$Resources:Captions,UserName
%>"></asp:Label>
```

در واقع در site map به صورت "\$Resources: Resource Name , Key Name" می‌باشد و در جاهای دیگر نیاز به علامت %> هم می‌باشد.

اگر بخواهید در قسمت کد Asp.net از فایل Resource بخوانید به صورت ذیل می‌باشد:

```
lblMessageInfo.Text = string.Format(Resources.Messages.ErrorMessage001 ,
Resources.Captions.UserName);
```

در واقع به صورت `string.Format(Resources.Resource Name.Key Name , Resources.Resource Name.KeyName)` می‌باشد. البته در صورت داشتن پارامتر، وگرنه فقط قسمت صورتی رنگ کافی است.

ارسال Email

معمولاً وقتیکه می‌خواهید از طرف سایت email ارسال کنید، چه برای admin و چه برای سایر کاربران با یک account به نام no reply انجام دهید. ما یک domain داریم و روی آن account ای به نام No reply تعریف می‌کنند و معنی آن این است که این Email مال یک روبات است و نه یک آدم و کسی تشکر و یا پیغامی را به آن reply نکند. یکی از مهمترین قسمت‌های سایت صفحه Contact us می‌باشد. صفحه contact us برای ارسال و دریافت Email می‌باشد. تکنیک آن به چه صورت است؟

ما یک صفحه به نام Contact درست می‌کنم که در آن از user یک سری اطلاعات (نام، نام خانوادگی، موضوع، نشان پست الکترونیک و توضیحات و ...) را می‌گیرم و یک دکمه ارسال می‌گذارم که با زدن آن Email ارسال می‌شود.

```
<div id="divMessage" runat="server" class="message">
    <asp:label id="lblMessage" runat="server" />
</div>
```

```

<fieldset>
  <legend>
    ما با تماس
  </legend>

  <div class="field">
    <asp:label id="lblFullName" runat="server" text="نام و نام خانوادگی" />
    <br />
    <asp:textbox id="txtFullName" runat="server" maxlength="50"
      accesskey="1" />
  </div>

  <div class="field">
    <asp:label id="lblEmail" runat="server" text="الکترونیکی پست نشانی" />
    <br />
    <asp:textbox id="txtEmail" runat="server" maxlength="100"
      accesskey="2" />
  </div>

  <div class="field">
    <asp:label id="lblSubject" runat="server" text="موضوع" />
    <br />
    <asp:textbox id="txtSubject" runat="server" maxlength="100"
      accesskey="3" />
  </div>

  <div class="field">
    <asp:label id="lblDescription" runat="server" text="توضیحات/شرح" />
    <br />
    <asp:textbox id="txtDescription" runat="server"
      textmode="Multiline" rows="10" columns="60" accesskey="4" />
  </div>

  <div class="buttons">
    <asp:button id="btnSubmit" runat="server" text="ارسال"
      accesskey="S" onclick="btnSubmit_Click" />
    <input type="reset" value="اولیه حالت" accesskey="R" />
  </div>
</fieldset>

```

تکنیک ارسال email

ابتدا یک شی Message می‌سازم، که این شی ... Subject , To , BCC , CC , Body دارد و سپس یک شی SMTP درست می‌کنم. و بعد Message را می‌گذارم داخل آن و می‌گویم ارسال کن.

یک سری تاکتیک داریم:

وقیکه از user این اطلاعات را می‌گیریم. در صفحه تماس با ما، یک متن و یا Body ایجاد می‌کنم، که بگویم یک بابایی با این نام، با این آدرس و با این شرح و توضیح و ... این نامه را ایجاد و به شما ارسال نموده است. ابتدا Body را درست کردیم. فلانی با این نام و نام خانوادگی و با این آدرس و با این شرح این email را ایجاد کرده است و فقط یک ابتکار به خرج دادم، چون ممکن است فرد در وسط mail از enter استفاده کرده است، پس می‌گویم هر چه n \ می‌بینی آن را تبدیل به
 کن که در موقع نمایش هم Enter ها مشخص باشد.

بازی از اینجا شروع شد که من یک شی از نوع Message درست کردم و نام آن را MainMessage می‌گذارم.

Property تا Main message سه تا Property دارد، به نام From و to و Reply to که Reply to منسوخ شده است و هر سه تا Property نیز ای یک جنس و از نوع Message هستند. بعضی از این Property ها بی معنی هستند، ولی چون بعضیها از تکنیک email برای نوشتن اتوماسیون استفاده می کنند، در اتوماسیون معنی دارد که sender با فرستنده اصلی فرق داشته باشد.

نکته: در asp3 یا کلاسیک نمی توانستید Body را به صورت Unicode ارسال کنید و بنابراین PHP این قابلیت را داشت و در این زمینه برتری داشت. ماکروسافت بعد از asp3 این قابلیت را اضافه کرد.

از .net2 به بعد نه تنها Body بلکه cc , bcc , subject و ... را هم می توانید Unicode ای کنید و این خیلی خوب است.

Mail و sender از جنس mail address است، پس یک Mail Address را declare و New می کنیم. موقع New کردن سه تا overload یا پارامتر می گیرد، اولی همان email است و دومی Display name است که می تواند فارسی باشد و یا اصلاً آن را خالی بگذارید و می تواند UTF8 باشد. اگر می خواهید display name متفاوت از آدرس باشد، آن را مقداردهی کنید. چونکه می تواند فارسی باشد، می توان آن را به عنوان UTF8 تعیین کنید. ضمناً آن بحث را رعایت کردم، که email هایی که از طرف سایت ارسال می شود، باید sender و as آنها no reply باشد. و این no reply یک عرف است.

چهار Collection (To, CC, BCC, Attachment) یک message دارد که بهتر است اول کار خالی شود. و نوشتن clear کار اضافی است ولی همیشه هر collection را اول بسم الله خالی کنید. To یعنی گیرندگان، BCC, CC هم گیرندگان است ولی به نوع رونوشت و رونوشت مخفی. Attachment هم یعنی فایل های الصاقی.

فرق عمده بین From , Sender , Reply to و bcc , cc , to , attachment این است که:

■ BCC, CC, To, Attachment از نوع Mail Address Collection است.

■ From , Sender , Reply to از جنس Mail address می باشد.

نامه همیشه یک فرستنده دارد ولی گیرندگان متفاوت می تواند داشته باشد.

To , CC , BCC از نظر تکنیکی فرق ندارند، بلکه فرق فلسفی و مفهومی دارند و شما باید مفاهیم آن را بدانید. نامه هایی که در BCC می زنید، هیچ احدی جز خودشان نمی فهمند.

ما الان می خواهیم صفحه Contact Us را درست کنیم، پس گیرنده آن admin سایت است. چون to به صورت collection است، اول mail address آن را تعریف می کنم و بعد به To اضافه می کنیم.

مسئله بعد body نامه است، عرف نیست که subject ای که کاربر تعریف می کند، را به عنوان subject نامه بگذارید و معمولاً یک subject تیپ درست می کنند. بنابراین ما subject را در دل body گذاشتیم.

معمولاً یک subject prefix را به نامه ها اضافه می کنند، تا مشخص شود که نامه از یک سایت به فرد ارسال شده است. چون شما Email کل افراد گروه را نمی شناسید و با آن prefix تشخیص می دهید.

Is body html true چیست؟ گاهی اوقات بجایی اینکه در متن نامه تگها باز شود و به صورت render شده ببینید، تگها را ببینید، معلوم است که این صفت را یکی اشتباهی false گذاشته است.

اولویت یا Priority نامه را مشخص کنید.

تا به حال شده است که Eail برای کسی بفرستید و پاسخ از email server بیاید که این email معتبر نیست و یا Expire شده. گاهی ایمیل می فرستید و از طریق Email server پاسخ دریافت می کنید، که ایمیل معتبر نیست و یا ... و یا پیغام می دهد که من delay دارم و تا 4 و 5 ساعت دیگر نمی رسم، بفرستم. می توان گفت که mail server شما را notify کنید.

Time out که واحد آن mili second است و enable ssl که true یا false است. برای attachment باید آدرس فیزیکی را به دست آورید و تعیین کنید.

```
protected void Page_Load(object sender, System.EventArgs e)
{
    divMessage.Visible = false;

    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
}

protected void btnSubmit_Click(object sender, System.EventArgs e)
{
    if (Page.IsValid)
    {
        try
        {
            // *****
            string strBody = string.Empty;

            strBody += string.Format("<b>Full Name</b>: {0}<br />",
                txtFullName.Text);
            strBody += string.Format("<b>E-Mail</b>: <a
                href='mailto:{0}'>{0}</a><br />", txtEmail.Text);
            strBody += string.Format("<b>Subject</b>: {0}<br />",
                txtSubject.Text);
            strBody += string.Format("<b>Description</b>: {0}<br />",
                txtDescription.Text.Replace("\n", "<br />"));
            // *****

            // *****
            System.Net.Mail.MailMessage oMailMessage =
                new System.Net.Mail.MailMessage();
            // *****

            // *****
            System.Net.Mail.MailAddress oMailAddress = null;
            // *****

            // *****
            oMailAddress =
                new System.Net.Mail.MailAddress
                (
                    "noreply@IranianExperts.com",
                    "Auto Response Email Sender",

```

```

        System.Text.Encoding.UTF8
    );

    oMailMessage.From = oMailAddress;
    oMailMessage.Sender = oMailAddress;
    //oMailMessage.ReplyTo = oMailAddress;
    // *****

    // *****
    oMailMessage.To.Clear();
    oMailMessage.CC.Clear();
    oMailMessage.Bcc.Clear();
    oMailMessage.Attachments.Clear();
    // *****

    // *****
    oMailAddress =
        new System.Net.Mail.MailAddress
        (
            "Dariush@IranianExperts.com",
            "Mr. Dariush Tasdighi",
            System.Text.Encoding.UTF8
        );

    oMailMessage.To.Add(oMailAddress);
    // *****

    // *****
    oMailMessage.BodyEncoding = System.Text.Encoding.UTF8;
    oMailMessage.Body = strBody;
    // *****

    // *****
    oMailMessage.SubjectEncoding = System.Text.Encoding.UTF8;
    oMailMessage.Subject = "[IranianExperts.com] - Somebody
        sent some email from contact page to you..";
    // *****

    // *****
    oMailMessage.IsBodyHtml = true;

    oMailMessage.Priority =
        System.Net.Mail.MailPriority.Normal;

    oMailMessage.DeliveryNotificationOptions =
        System.Net.Mail.DeliveryNotificationOptions.Never;
    // *****

    // *****
    string strRootRelativePathName = "~/Banners/Banner.gif";

    string strPathName =
        Server.MapPath(strRootRelativePathName);

    if (System.IO.File.Exists(strPathName))
    {
        System.Net.Mail.Attachment oAttachment =
            new System.Net.Mail.Attachment(strPathName);

        oMailMessage.Attachments.Add(oAttachment);
    }
    // *****

```



```

// *****
System.Net.Mail.SmtpClient oSmtpClient =
    new System.Net.Mail.SmtpClient();

oSmtpClient.Timeout = 100000;
oSmtpClient.EnableSsl = true;

oSmtpClient.Send(oMailMessage);
// *****

// *****
string strInformationMessage =
    ".گردید ارسال پایگاه مسوولین به موفقیت، با الکترونیکی نامه";

DisplayInformationMessage(strInformationMessage);
// *****
}
catch (System.Exception ex)
{
    DisplayErrorMessage(ex.Message);
}
}
}

protected virtual void DisplayErrorMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "error";
}

protected virtual void DisplayInformationMessage(string message)
{
    divMessage.Visible = true;

    lblMessage.Text = message;
    lblMessage.CssClass = "information";
}
}

```

واقعیت این است که SMTP خیلی property دارد. و همه اینها را می‌توانید hard code تعیین کنید ولی اگر تغییری ایجاد شود، باید مجدداً hard code را تغییر دهید، بنابراین این تنظیمات را روی خود SMTP server انجام دهید. فقط 2 تا از پارامترها را نمی‌توانید بر روی خود SMTP Server ست کنید. که باید در WebConfig بنویسید. این دو عبارتند از deliveryMethod و ...

```

<configuration>
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="Network" from="sematec.class19@gmail.com">
        <!-- Default Port is [25] -->
        <!-- Specific Port for Gmail is [587] -->
        <!-- Specific Port for Yahoo is [465] -->
        <!-- Specific Port for Hotmail is [587] -->
        <network
          defaultCredentials="false"
          port="587"
          host="smtp.gmail.com"
          password="sematec12345"
        </network>
      </smtp>
    </mailSettings>
  </system.net>
</configuration>

```

```

        userName=sematec.class19@gmail.com />
    </smtp>
</mailSettings>
</system.net>

<system.web>
    <compilation debug="true" targetFramework="4.0"/>
</system.web>
</configuration>

```

5 چیز باید یکسان باشد، وگرنه ممکن است که نامه به قسمت Spam برود. این 5 آیتم عبارت است از :

from -
sender -
reply to -
username -
From -

نکته: برای Publish کردن سایت، روی پروژه یک کلیک راست کنید و مسیر publish را بزنید و فقط checkbox اول را انتخاب کنید. Debug را باید false کنید.

یکی از بهترین و رایگان برنامه upload برنامه File zilla است. خیلی شبیه NC است. سمت راست مال دستگاه و سمت چپ مال سایت و ... امکان drag & drop دارد. این برنامه، resume دارد. اگر وسط کار قطع شود، دوباره از ادامه کار شروع می کند. در واقع از Thread استفاده می کند. فایل های upload شده را با Client چک می کند. File Zilla را روی سرور نصب کنید. Ftps ایجاد می کند و ارسال ftp را به صورت secure انجام می دهد.