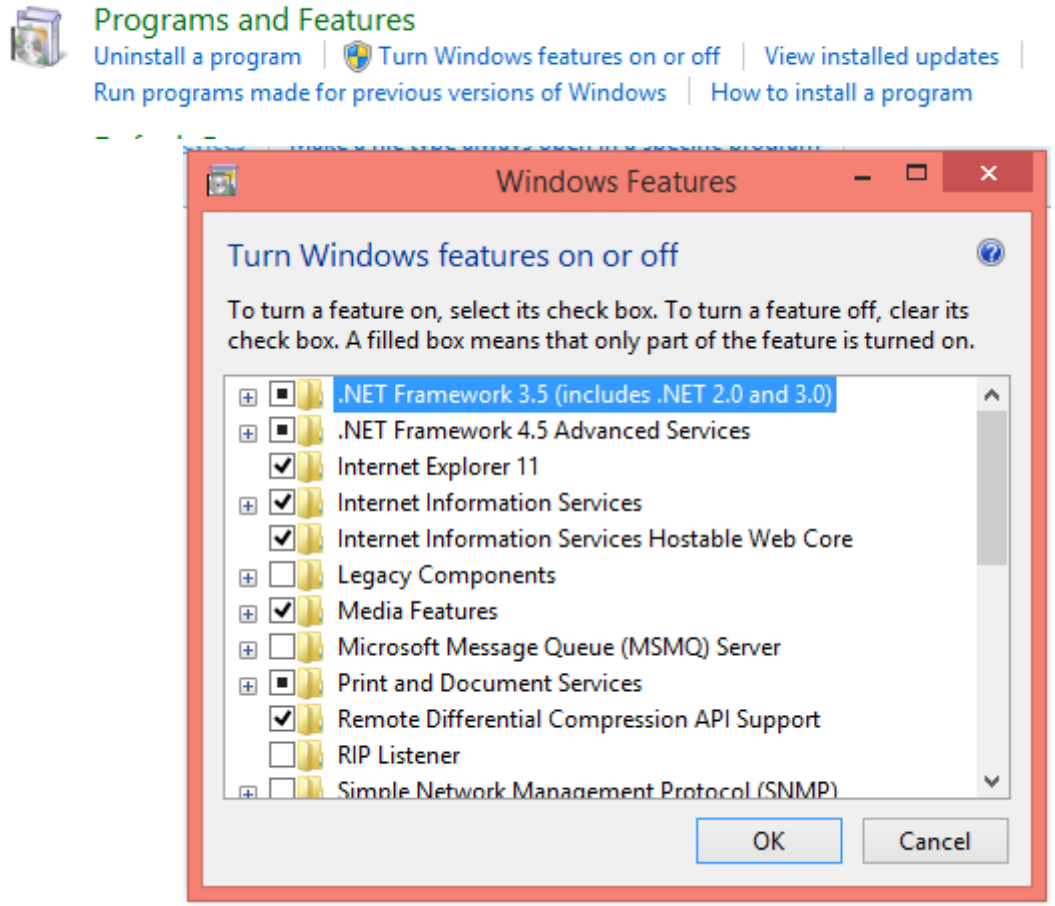
s.ali.taher@gmail.com

IIS:Internet information services

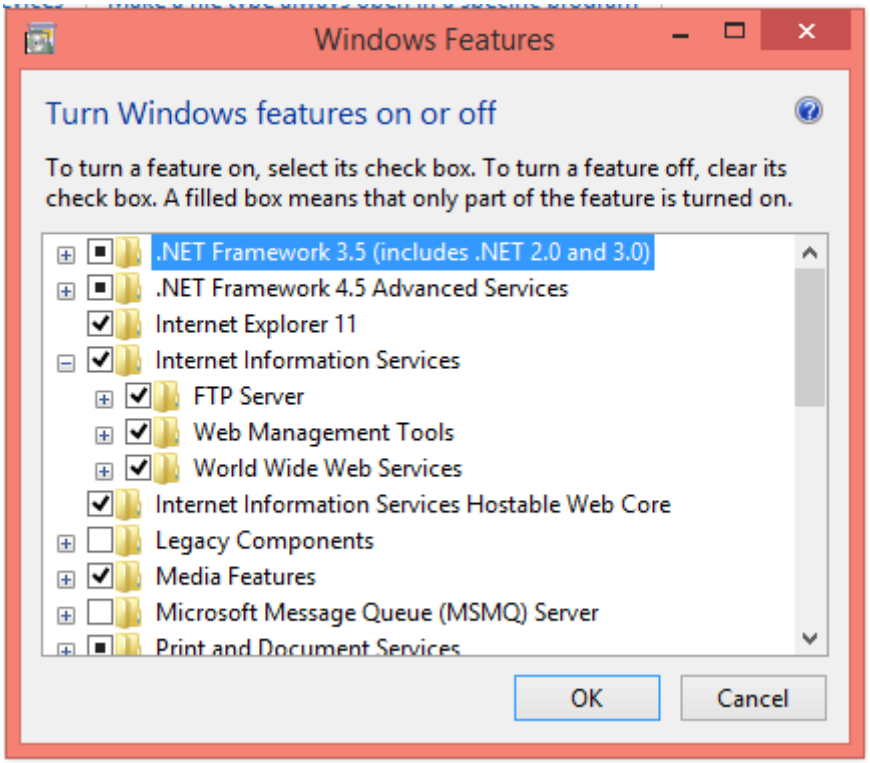
برای نصب iis مراحل زیر را انجام می دهیم :

۱ -با کلیک بر روی گزینه control panel و سپس کلیک روی آیکن program

تا پنجره زیر بازگردد

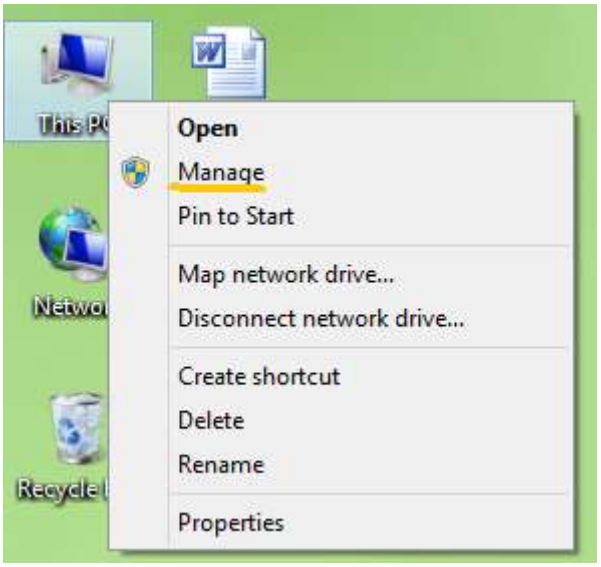


سپس با تیک زدن گزینه های مربوطه

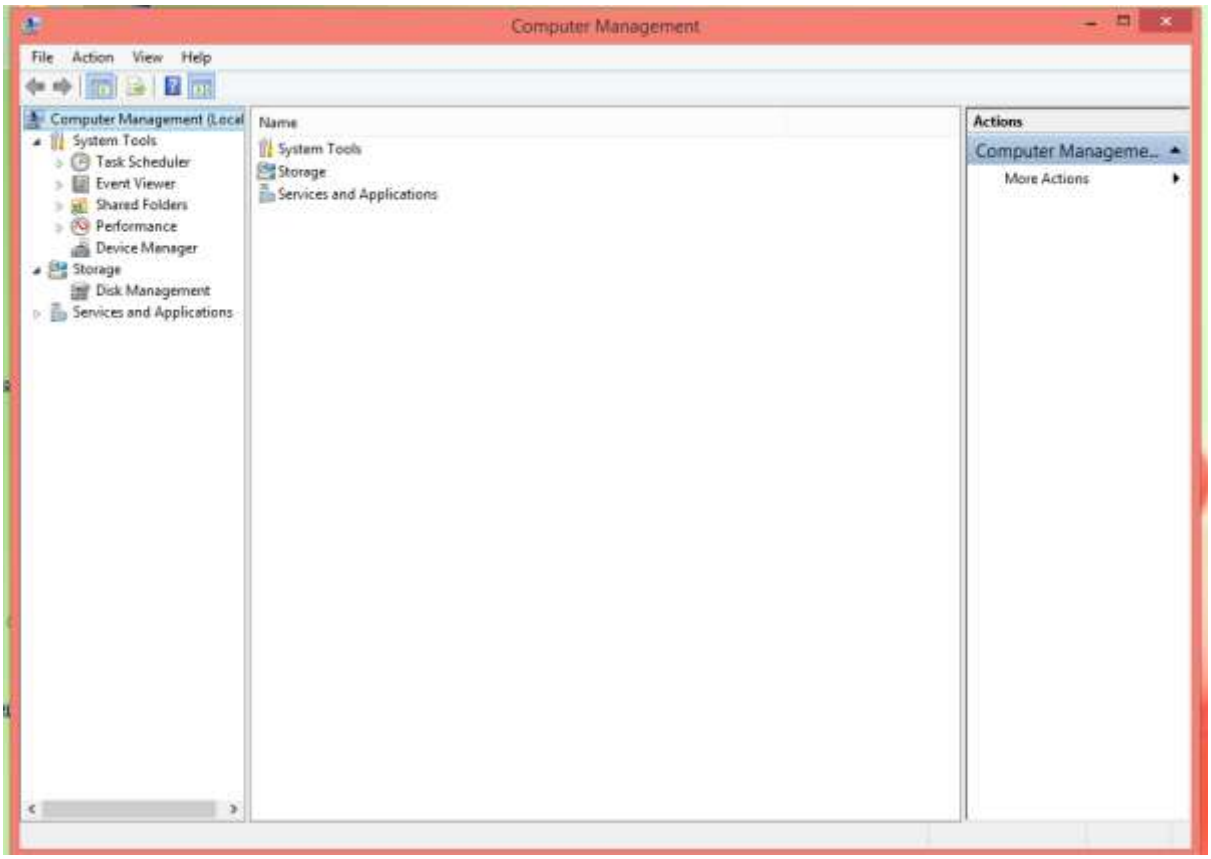


برای مشاهده اینکه آیا سرویس iis فعال هست با خیر موارد زیر عمل می کنیم :

۱-با راست کلبک بر روی my computer گزینه manage را کلیک می کنیم

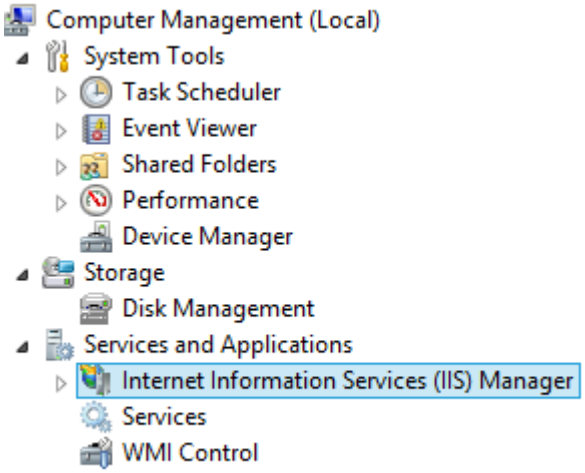


سپس از پنجره زیر



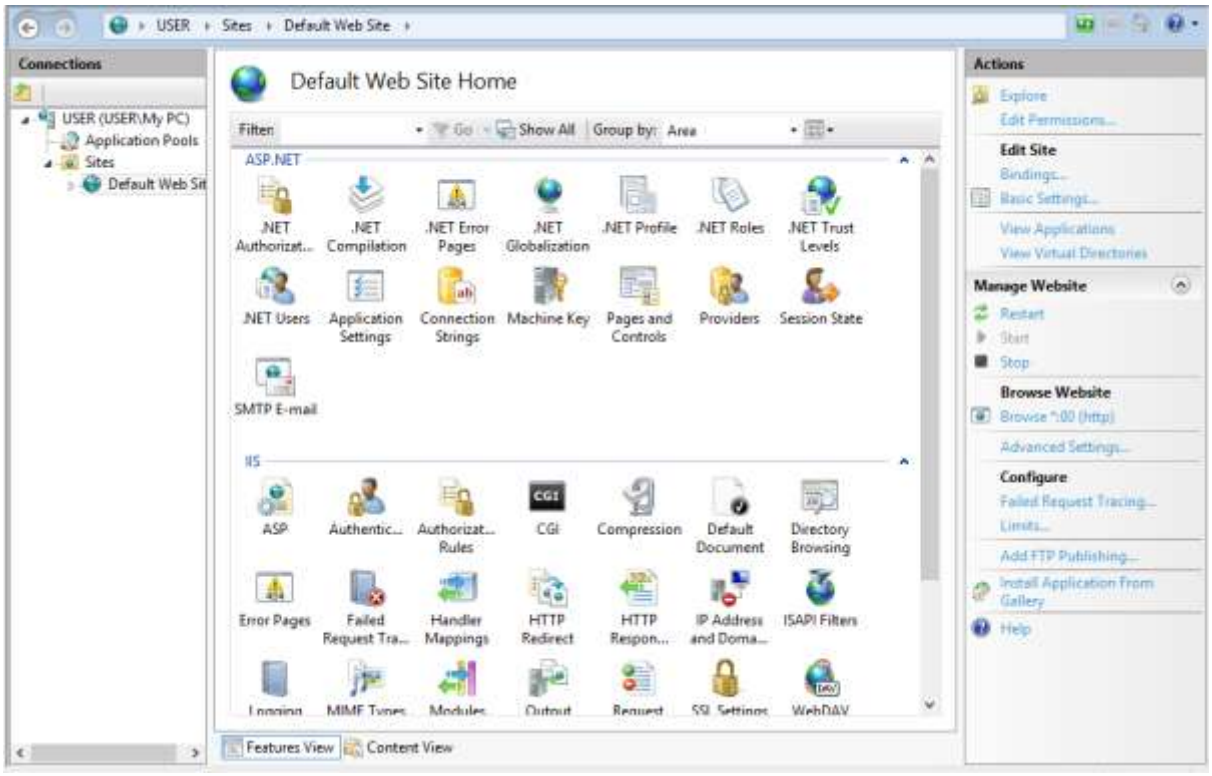
روی گزینه services and Applications کلیک می کنیم

- روی گزینه

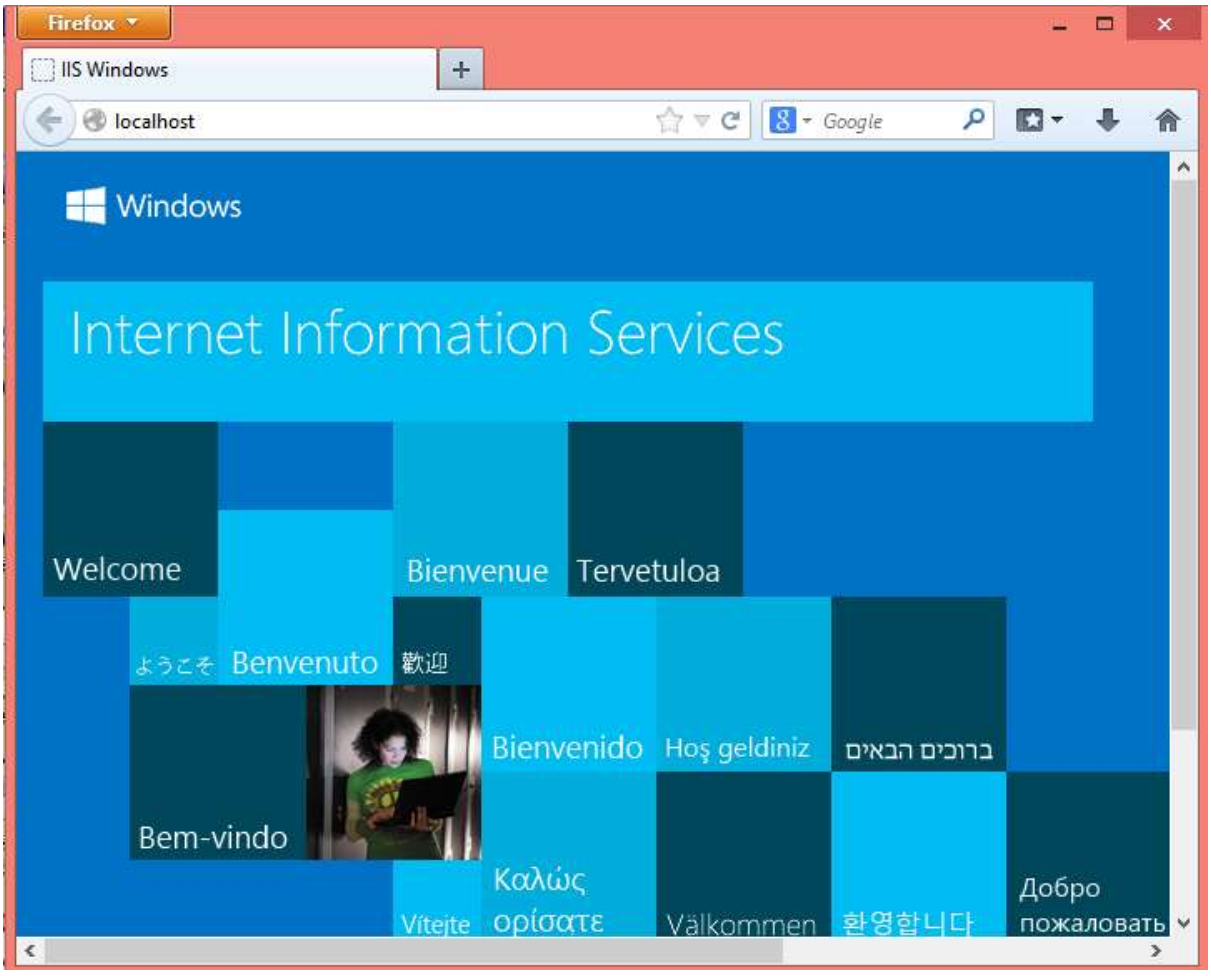


-

- منوی زیر ظاهر می گردد:

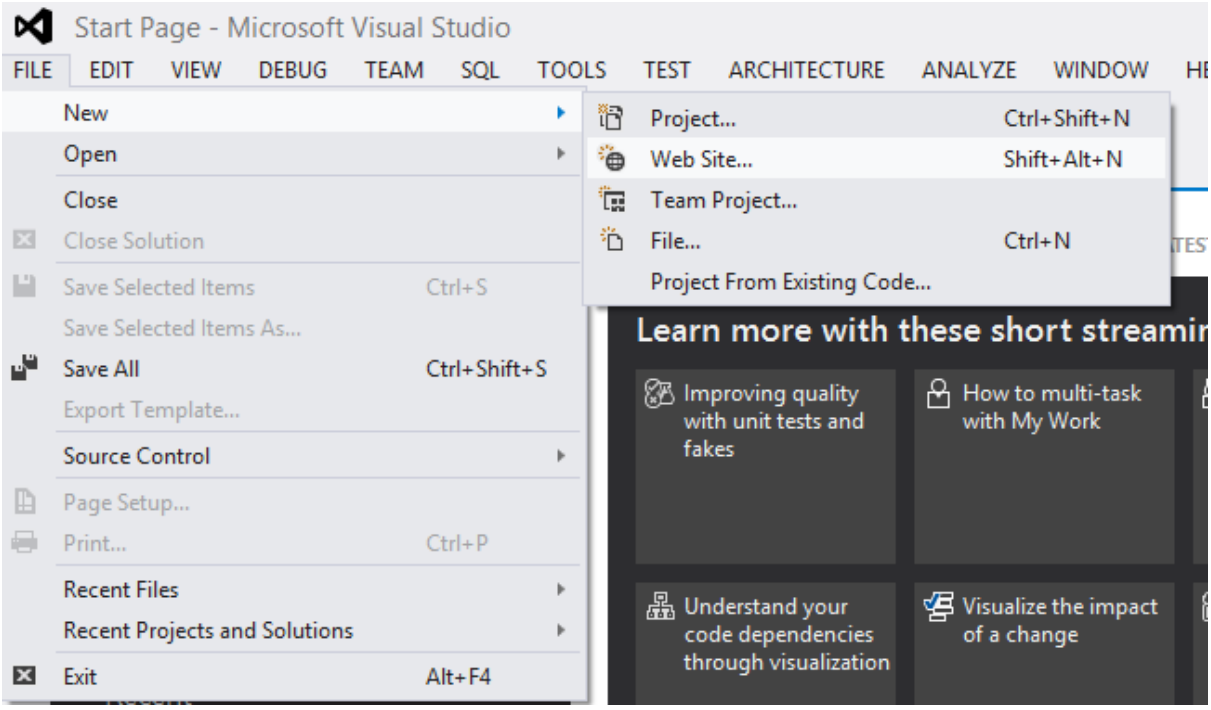


چنانچه گزینه start از بخش manage website در حالت اجرا باشد می توانیم وب سایت را browseکنیم

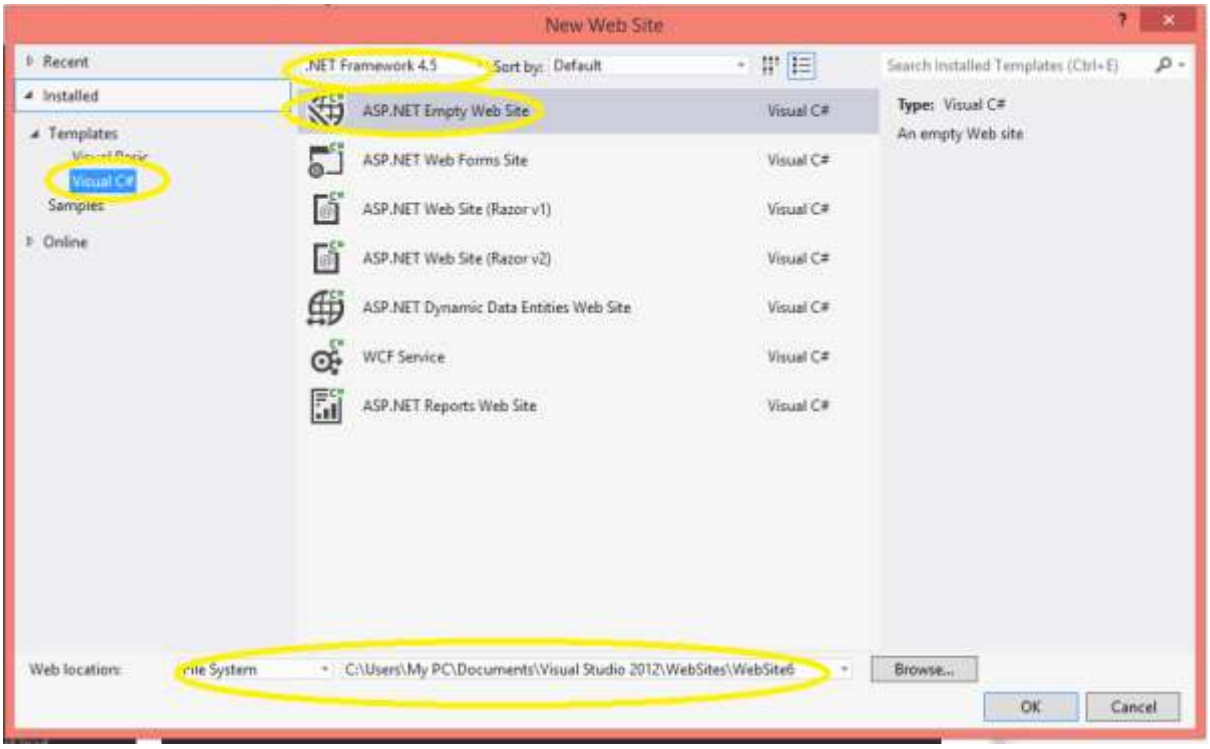


حال می توانیم از asp.netبرای طراحی صفحاب وب استفاده نماییم برای شروع مراحل زیر را انجام می دهیم

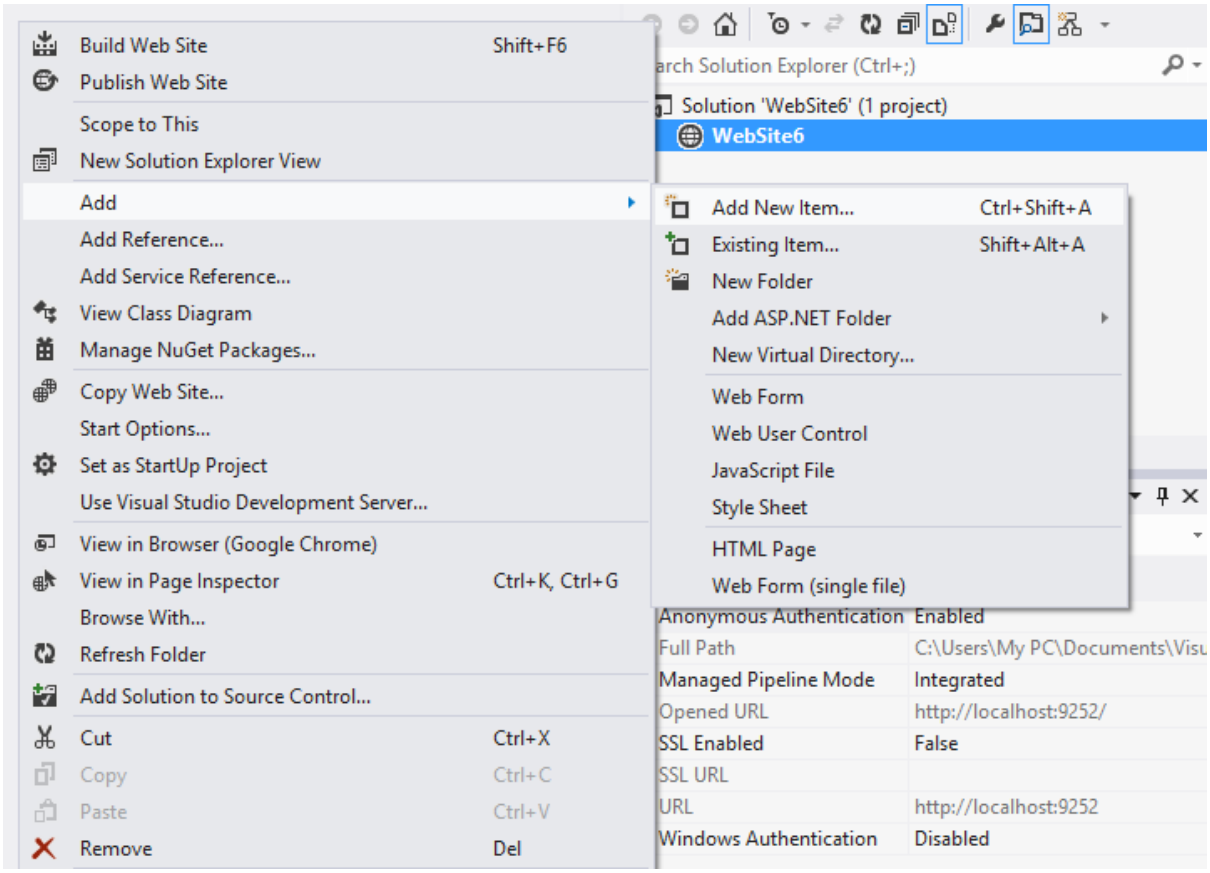
ا- ابتدا نرم افزار visual studio را اجرا می نماییم و سپس از منوهای مشخص شده استفاده می نماییم :



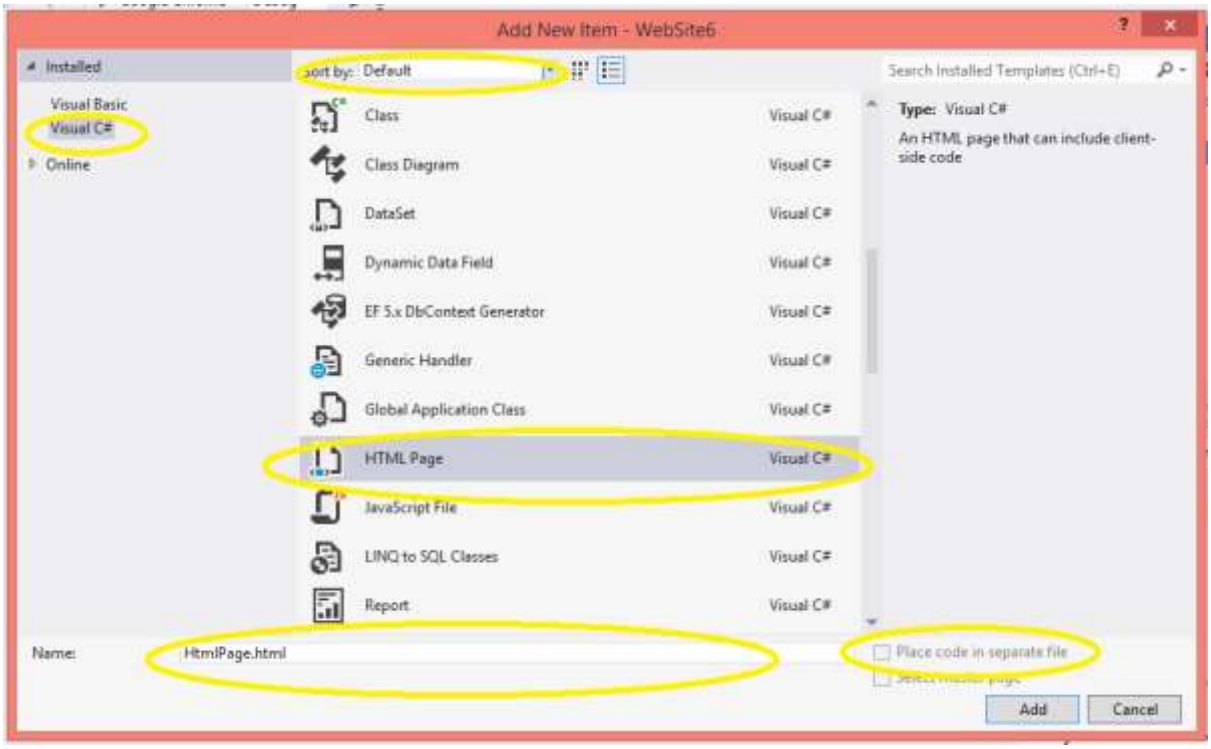
سپس به ترتیب گزینه های زیررا انتخاب می کنیم



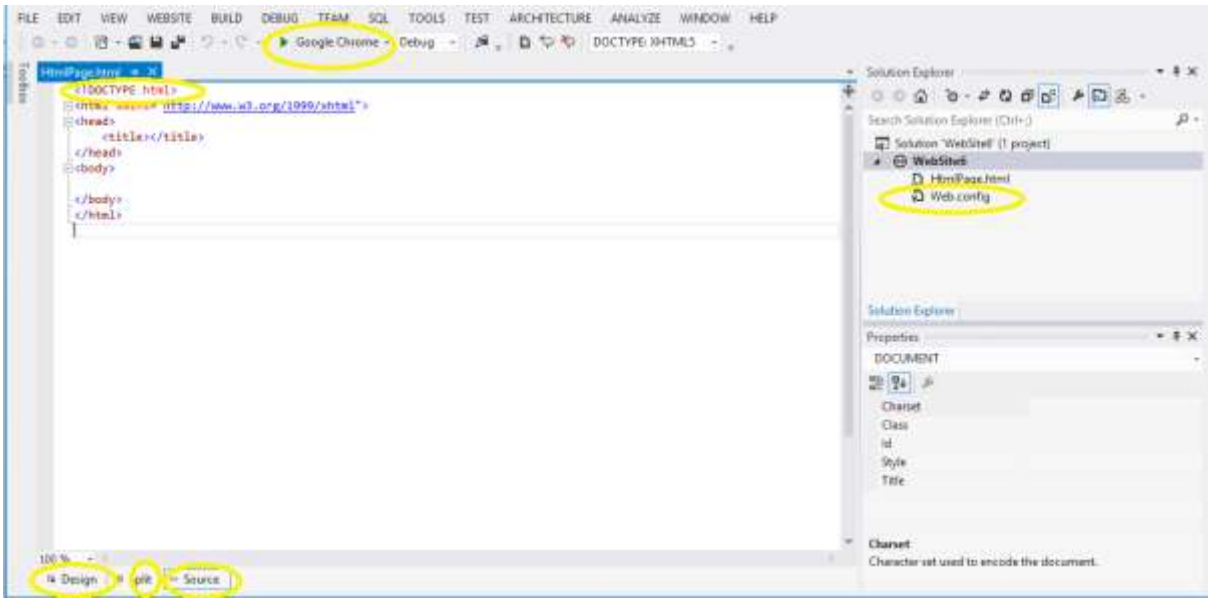
برروی فلدر وب سایت راست کلیک می نماییم :



منوی زیر ظاهر خواهد شد



به گزینه های زیر توجه کنید



یادآوری دستوراتی از html :

- HTMLمخففHyper Text Markup Languageاست.
- HTMLزبان **نشانه گذاری** است.
- یک زبان نشانه گذاری مجموعه ای از **تگ** های نشانه گذاری است.
- تگ ها محتوای سند را **توصیف** می کنند.
- اسناد HTML شامل تگ های HTML و متن های ساده می باشد.
- یک سند HTML **صفحه وب** نیز نامیده می شود.

تگ هایHTML

تگ های نشانه گذاری HTML معمولا تگ های HTML نامیده می شود .

- تگ هایHTML ، کلمات کلیدی (نام تگ) هستند که توسط براکت های زاویه ، مانند <html>احاطه شده اند.
- تگ های HTML معمولا به صورت جفت می آیند مانند و
- تگ اول تگ شروع و تگ دوم تگ پایان نامیده می شوند.
- تگ پایانی مانند تگ شروع، نوشته شده می شود ، اما با یک اسلش قبل از نام تگ.
- تگ های شروع و پایان برچسب های افتتاح و اختتام نیز نامیده می شود.

نمونه :

<tagname> محتوا</tagname>

عناصرHTML :

تگ های HTML و عناصر HTML اغلب برای توصیف چیزی مشابه استفاده می شوند .اما به بیان دقیق، یک عنصر HTML همه چیزی است که بین تگ شروع و تگ پایان قرار دارد، از جمله تگ ها :

این یک پاراگراف است.

مرورگرهای وب

هدف مرورگر وب (مانند گوگل کروم، اینترنت اکسپلورر، فایرفاکس، سافاری (خواندن اسناد HTML و نمایش آنها به عنوان صفحات وب است . مرورگر ها تگ های HTML را نمایش نمی دهد، اما با استفاده از تگ ها محتوای صفحه را تفسیر می کند:

نسخه های HTML

از روزهای اولیه وب، نسخه های بسیاری از HTML وجود داشته است :

نسخه	سال
HTML	۱۹۹۱
HTML +	۱۹۹۳
HTML ۲.۰	۱۹۹۵
HTML ۳.۲	۱۹۹۷
HTML ۴.۰۱	۱۹۹۹
XHTML ۱.۰	۲۰۰۰
HTML۵	۲۰۱۲
XHTML۵	۲۰۱۳

اعلامیه <!DOCTYPE>

اعلامیه <!DOCTYPE>به مرورگر کمک می کند تا یک صفحه وب را به درستی نمایش دهد. بسیاری از اسناد و مدارک مختلف بر روی وب وجود دارد، و یک مرورگر تنها زمانی می تواند صفحه HTML را ۱۰۰٪ به درستی نمایش دهد که نوع و نسخه مورد استفاده آن HTML را بداند .

اعلامیه های مشترک

HTML ۵

<!DOCTYPE html>

HTML ۴.۰۱

<!DOCTYPE HTML PUBLIC "-//W۳C//DTD HTML ۴.۰۱ Transitional//EN" "http://www.w۳.org/TR/html۴/loose.dtd">

XHTML ۱.۰

<!DOCTYPE html PUBLIC "-//W۳C//DTD XHTML ۱.۰ Transitional//EN" "http://www.w۳.org/TR/xhtml۱/DTD/xhtml۱-transitional.dtd">

در سال ۱۹۹۲ در دانشگاه مینه سوتا سیستمی به نام web به وجود آمد که دارای دو ویژگی خاص بود.

Graphic, Multi Media

Hyper Text (فقط کلیک کردن و در سایت حرکت کردن) صفحات این سیستم توسط برنامه ای به نام HTML ساخته شد.همچنین برای رد و بدل کردن اطلاعات از پروتکل http استفاده میشود:

HTTP(Hyper Text Transfer Protocol HTML Web page)

HTML یک text عادی و در حقیقت زبانی برای مارک کردن فایلهای text به یکدیگر می باشد که آن را با TAG مشخص کرده و به صورت <tag name> می نویسند.

فرمت کلی :

یک فایل HTML از دو بخش Body و Head ،تشکیل می شود. شکل ساده یک فایل HTML به صورت زیر است.

```
<HTML>
  <HEAD>
     <TITLE  This is the title> </TITLE>
  </HEAD>
  <BODY>
    This is the the body
  </BODY>
</HTML>
```

تگ head:

در برچسب HEAD از برچسب به نامهای TITLE و BASE و META استفاده می شود.

```
<HEAD>
<TITLE> </TITLE>
< META > </META>
< BASE>  </BASE>
</HEAD>
```

:TITLE

برای تعیین لقب صفحه (چیزی که در قسمت Status Bar دیده می شود).

:META

۱) برای تعیین نام و منبعی که برنامه توسط آن نوشته شده

۲) بهنگام کردن صفحات web توسط این برچسب انجام می شود.

۳) انتقال به یک صفحه دیگر web در زمان معیین.

مثال برای حالت اول : در این حالت برای وارد کردن آدرس web خودمان به موتورهای جستجو در web (مثل yahoo و google و ...) از META استفاده می کنیم:

```
<META name="keyword" content= "Hedayat,students,zahiri,yaghoubi,schoolnet"/>
<META name="description" content="This is Hedayat high school"/>
```

مثال برای حالت دوم و سوم:

```
<META name="vali" http_equiv="refresh" content ="زمان بر حسب ثانیه"/>
```

با این برچسب صفحه web بعد از ۱ دقیقه بهنگام (refresh) خواهد شد. در مثال بالا اگر در قسمت content به صورت زیر عمل کنیم صفحه web بعد از ۶۰ ثانیه به www.schoolnet.ir خواهد رفت:

```
Content="۶۰; URL= http://www.schoolnet.ir"
```

:BASE

برای مشخص کردن مبدا آدرس دهی از صفحات web می باشد.

```
<BASE href ="آدرس"/>
<BASE href="http://www.schoolnet.ir/~zahiri/index.htm">
```

نکته مهم: در برنامه نویسی HTML برچسب ها به دو صورت با پایان و بی پایان نوشته می شوند

جلسه دوم : یادآوری دستوراتی از html :

<TAG>.....</TAG> با پایان

<TAG> بی پایان

قسمت دوم یک فایل HTML را Body تشکیل می دهد که دارای Attributeهای زیر می باشد.

<BODY bgcolor="رنگ پس زمینه صفحه"

background=" آدرس عکسی که به عنوان پس زمینه در صفحه وب قرار می گیرد"

topmargin="یک فضای خالی بالای صفحه بر حسب پیکسل ایجاد ما کند"

leftmargin="یک فضای خالی سمت چپ صفحه بر حسب پیکسل ایجاد ما کند"

text="color" رنگ متن را مشخص می کند

link= "color"

alink ="color"

vlink ="color"/>

نکته مهم: در برنامه HTML و در نوشتن تگها بزرگ و یا کوچک نوشتن حروف هیچ تاثیری ندارد.

با این برچسب می توانیم مشخصات متن را به دلخواه خود درآوریم و فرمت کلی آن به صورت زیر است

<FONT> .....</FONT>

این تگ دارای Attributeهای زیر می باشد:

color: رنگ متن

size: اندازه متن

face: نوع متن

مثال: می خواهیم کلمه Schoolnet را با فونت نازنین و با اندازه normal و رنگ آبی بنویسیم.

```
<HTML>
 <HEAD>
</HEAD>
<BODY>
    <FONT size = "۳" color ="blue" face="NAZANIN" Schoolnet> </FONT>
</BODY>
</HTML>
```

نکته: اگر بخواهیم اندازه را نسبی مشخص کنیم یعنی نسبت به آنچه که قبلا بوده به صورت زیر عمل می کنیم

Size = +۲

:BOLD

اول و آخر متن مورد نظر قرار گرفته و آنرا Bold می کند.

<B> text … </B>

:ITALIC

اول و آخر متن مورد نظر قرار گرفته و آنرا Italic می کند.

<I> *… text* **</I>**

:UNDERLINE

اول و آخر متن مورد نظر قرار گرفته و آنرا Underline می کند.

<U> … text **</U>**

در HTML بوسیله تگ … <a> می توانیم یک متن یا عکس را به صفحه ای دیگر پیوند دهیم (Hyper link).مهمترین Attribute در این تگ، href می باشد.

فرمت کلی این تگ به صورت زیر است.

 text/image

مثال: در جمله click here to go to page zahiri home کلمه zahiri را به آدرس html.index لینک کنیم:

```
<BODY>
<p>
click here to go to
<a href ="http://www.schoolnet.ir"> schoolnet </a>
home page
</p>
</BODY>
```

<P> … </P>: در این تگ همه موضوعات آن در یک خط نوشته شده یا یک پاراگراف ایجاد می شود که در صورت بوجود آمدن پاراگراف در زیر خط اول، خط دوم را با فاصله زیاد می نویسد .برای حل این مشکل از تگ **
** استفاده می کنیم.

تگ <P> دارای یک Attribute است:

<p align="left/center/right">

**
**: این تگ از تگهایی است که پایان ندارد و آن را هر کجا که قرار دهیم کلمه بعدی را در یک خط پائین تر ولی با کمتر می نویسد.

<NOBR> … </NOBR>: اگر بخواهیم در آخر خط شکستگی ندا شته باشیم بین دو بخشی که شکسته می شود از این تگ استفاده می کنیم .

مثال: <NOBR>a۲</NOBR>

a ,۲ را هرگز از هم جدا نمی کند.

در HTML دارای شش نوع HEADING هستیم.

<H۱> … </H۱> بزرگترین

<H۲> … </H۲>

.

.

کوچکترین <H۶> … </H۶>

خود این تگها خاصیت راست چین و چپ چین و یا وسط چین شدن را هم دارند که برای فعال کردن آن از روش زیر استفاده می کنیم.

مثال: <H۲ align=”center”> vali </H۲>

تگ <HR> برای ما تک خط افقی سه بعدی ایجاد می کند و دارای Attributeهای زیر می باشد.

<HR align="left/center/right"

width=“طول خط بر حسب پیکسل یا درصد”

size=“ضخامت خط بر حسب پیکسل”

noshade: با نوشتن این کلمه خط سه بعدی نمی شود .

Color=“رنگ خط”>

در HTML هر چیزی را که بین تگ <PRE> … </PRE> به هر صورتی که بنویسیم با همان شکل در صفحه وب نشان می دهد.

مثال:

<PRE> Vali
Ali …
Reza a b cd
</PRE>

برای ما یک بلوک در متن ایجاد می کند.

مثال:

<DIV style="color: red"> … … </DIV>

استفاده دیگر برای تعیین **Direction** می باشد که سمت نوشتن را از راست به چپ یا چپ به راست می کند.

<DIV dir="rtl/ltr"> … … </DIV>

rtl= right to left

ltr= left to right

<big> … </big>: متن که بین آن باشد یک فونت درشتتر می نویسد.

<small> … </small>: متن که بین آن باشد یک فونت کوچکتر می نویسد.

برای درست کردن توان و اندیس دو تگ داریم.

مثال: برای توشتن و از دو تگ زیر استفاده می کنیم.

a_۱

a^{۲۰}

<IMG>: برای قرار دادن یک تصویر در web از این تگ استفاده می کنیم.

به چند دلیل نباید از تگ image زیاد استفاده کرد.

۱- به علت زیاد شدن تعداد عکسها صفحه دیر load می شود.

۲- ایجاد مشکل در Search Engine.

۳- عدم قابل استفاده بودن برای همه(عدم سرعت کافی، هزینه دار بودن برای user و ...)

طرز قرار دادن عکس:

طرز قرار دادن عکس:

<IMG src=”آدرس فایل تصویر را در این قسمت می نویسیم"
align="left/middle/right/top/bottom"
height="ارتفاع عکس بر حسب پیکسل"
width="طول عکس بر حسب پیکسل"
alt=" “
title=“”

هر چیزی که در این قسمت بنویسیم قبل از loadشدن تصویر و یا بعد از کامل شدن صفحه اگر موس را روی عکس ببریم این متن دیده خواهد شد.(در یک مستطیل زرد رنگ)"

Border=“ضخامت جدول دور عکس را بر حسب پیکسل نشان می دهد“
hspace=“فاصله عمودی دور تصویر بر حسب پیکسل“
vspace=“فاصله افقی دور تصویر بر حسب پیکسل“>

مثال:

<img border= "۳" src=‘‘zahiri.jpg " width= "۱۱۱"height= "۱۳۷" align="center "
hscape= "۱۰"
vscape= "۱۰" alt="This is Zahiri's picture”/>

سوال: برای لینک کردن یک عکس چه کاری را باید انجام دهیم؟

جواب:

```
<A href="URL" >
  <IMG src="URL"/>
</a>
```

در بسیاری از مواقع در یک صفحه وب مجبوریم قسمتی از یک عکس را به صفحه ای لینک کنیم برای این کار در html از Image Map استفاده می کنیم.

مثال: می خواهیم در t۱.gif در منطقه یکی دایره و دیگری چهارگوش را به صفحه ای دیگر لینک کنیم:

```
<MAP name="Zahiri" >
<AREA shape="circle" coord="۵۰,۵۰,۳۰" href="۱.htm">
<AREA shape="rect" cords="۱۰۰,۷۰,۱۳۰,۱۰۰"  href="۲.htm">
</MAP>
…
<IMG src="t۱.gif"  usemap="#Zahiri">
```

عنوان HTML

عنوان ها یا سر برگ های HTML با تگ های <h۱>تا <h۶>تعریف می شود.

```
<h۱>This is a heading</h۱>

<h۲>This is a heading</h۲>

<h۳>This is a heading</h۳>
```

پاراگرافHTML

پاراگراف های HTML با تگ <p>تعریف می شوند.

```
<p>This is a paragraph</p>
```

لینک هایHTML

لینک HTML با تگ <a>تعریف می شود.

This is a Link

عکس هایHTML

عکس های HTML با تگ img تعریف می شود.

<img src="image۱.jpg" width="۲۵۰" height="۱۸۰">

جداول HTML

جداول با تگ <table>تعریف شده است .
جدول به ردیف ها (با تگ (<tr>تقسیم می شود، و هر سطر به سلول های داده (با برچسب (<td>تقسیم می شود td .مخفف "table data" و دارای محتوای یک سلول داده می باشد. تگ <td>می تواند شامل متن، لینک ها، عکس ها، لیست ها، فرم ها، جداول و غیره باشد.

```
<table border="۱">
<tr>
<td>row ۱, cell ۱</td>
<td>row ۱, cell ۲</td>
</tr>
<tr>
<td>row ۲, cell ۱</td>
<td>row ۲, cell ۲</td>
</tr>
</table>
```

جداول HTML و عنصر حاشیه

اگر شما عنصر حاشیه را مشخص نکنید، جدول بدون حاشیه نمایش داده خواهد شد .گاهی اوقات این می تواند مفید باشد، اما بیشتر اوقات ، ما می خواهیم حاشیه ها را نشان دهیم .
برای نمایش جدول با حاشیه ، صفت border را مشخص کنید:

```
<table border="۱">
<tr>
```

```
<td>Row ۱, cell ۱</td>
<td>Row ۱, cell ۲</td>
</tr>
</table>
```

عناوین جدول HTML

اطلاعات عنوان در یک جدول با برچسب <th>تعریف می شود .

تمام مرورگرهای بزرگ ، متن را در عنصر <th>به صورت متمرکز و ضخیم نشان می دهند.

```
<table border="۱">
<tr>
<th>Header ۱</th>
<th>Header ۲</th>
</tr>
<tr>
<td>row ۱, cell ۱</td>
<td>row ۱, cell ۲</td>
</tr>
<tr>
<td>row ۲, cell ۱</td>
<td>row ۲, cell ۲</td>
</tr>
</table>
```

مثال جدول:

```
<table>
    <tr>
      <td width="۷۷۰px"height="۲۳۰px"colspan="۲"></td>
    </tr>
</table>
```

**طرح بندی - HTML استفاده از عنصر **

عنصر div یک سطح بلوک برای گروه بندی عناصر HTML است .در مثال زیر از پنج عنصر div به منظور ایجاد یک طرح بندی چند ستونی استفاده شده است، که همان نتیجه در مثال قبل را ایجاد می کند:

```
<! DOCTYPE html>
<html>
<body>
<div id="container" style="width:۴۰۰px ">
<div id="header" style="background-color:#FFA۵۰۰  ">;
</ body>
< /html>
```

شبه کلاس lang:

شبه کلاس lang: به شما اجازه می دهد تا قوانین ویژه ای را برای زبان های مختلف تعریف کنید .توجه IE۸ :از شبه کلاس lang: پشتیبانی می کند. به شرطی که <!DOCTYPE>مشخص شده باشد .در نمونه زیر lang: علامت کوتیشن را برای عناصر q با "lang="no تعریف می کند:

```
<html> lang="fa-IR"</html>
```

ویژگی ها (attributes):

```
<p dir="rtl"></p>
<p dir="ltr"></p>
<font></font>
<font size="۴"face="tahoma"color="green"></font>
```

- . ساختن منو

```
<OL>
<LH> آموزش وطراح </LH>
<LI> V </LI>
<LI> A </LI>
<LI> L </LI>
<LI> I </LI>
</OL>
```

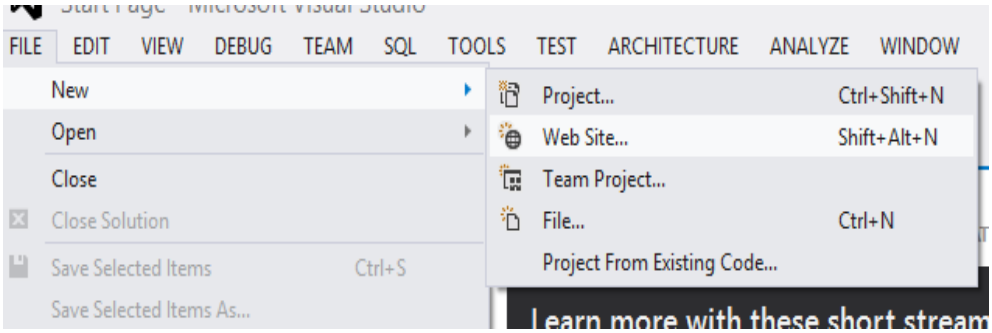
روش دوم

```
<UL>
<LH> … … </LH>
```

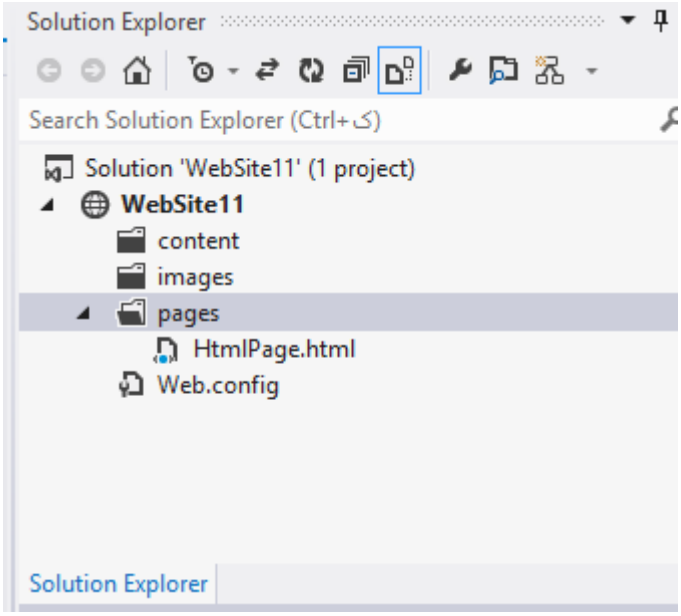
<LH> … … </LH>
<LH> … … </LH>
</UL>

جلسه سوم:طراحی لایه های صفحه با تکنولوژی div

۱ -ابتدا فایل جدیدی را باز می کنیم



۲-بروی پوشه کاری سه فلدر و یک صفحه html ایجاد می نماییم

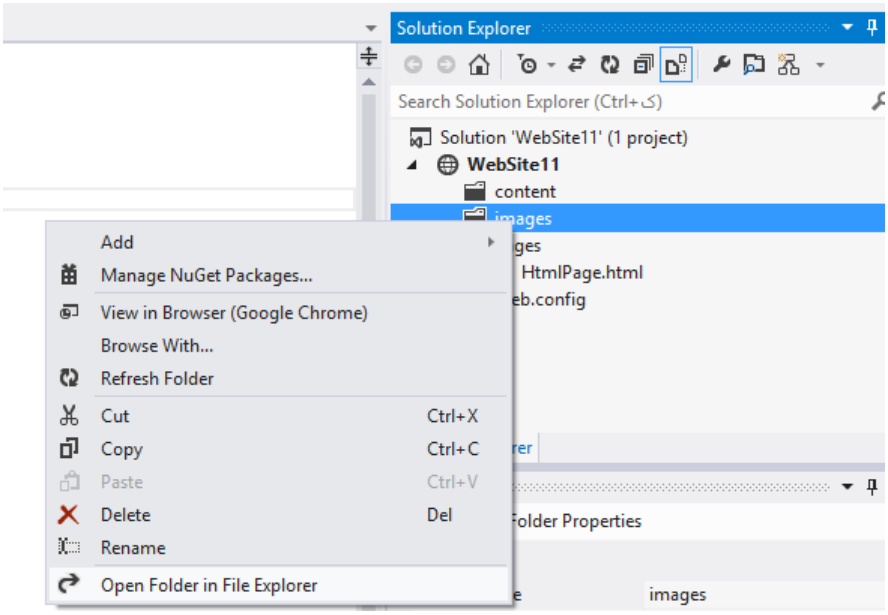


۳-در صفحه html کد های زیر در بخش body اضافه می نماییم

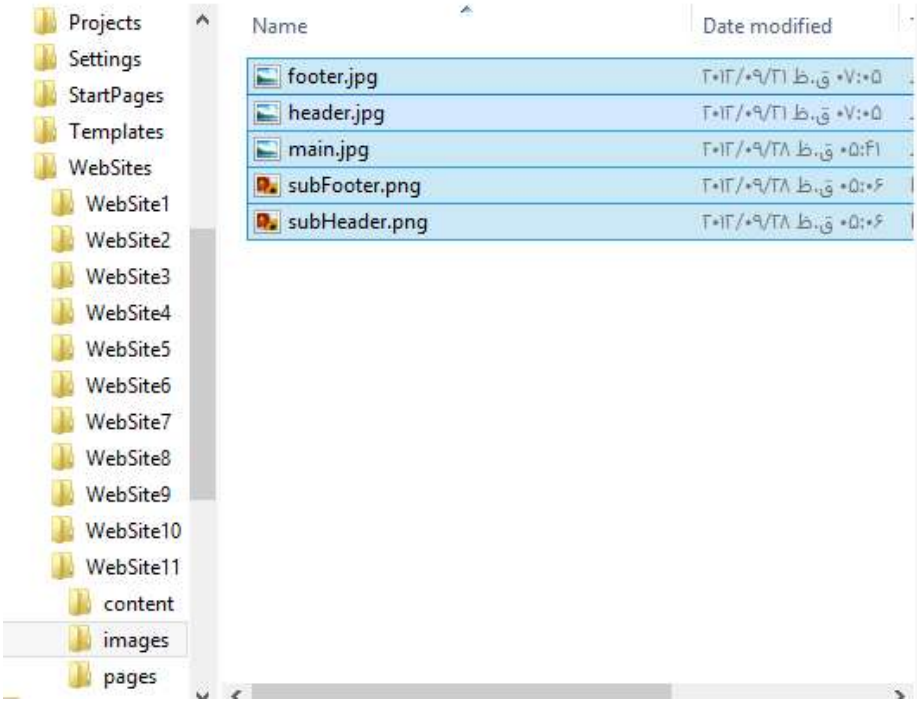
```
<!DOCTYPE html>
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
  <head>
     <title></title>
  </head>
  <body>
    <div></div>
    <div></div>
```

```
    <div></div>
    <div></div>
    <div></div>
  </body>
</html>
```

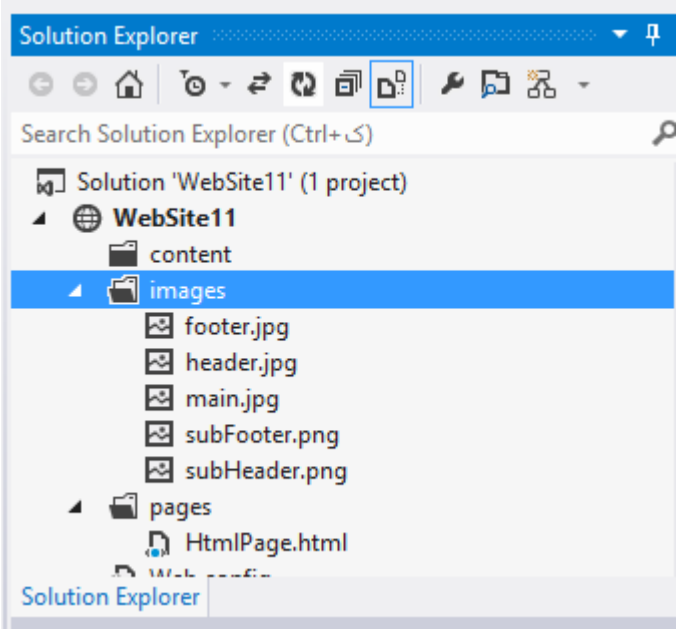
۴-برای لود کردن عکس های مورد نیاز در فلدر images به روش زیر عمل می نماییم :



فایل های مورد نیاز در پوشه مربوطه کپی می نماییم :



۵- بعد از کپی کردن فایلها در پوشه مربوطه پنجره را بسته و فلدر images را رفرش می کنیم .



۶- در صفحه html کد های زیر را اضافه می کنیم

```
<!DOCTYPE html>
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<head>
    <title></title>
</head>
<body>
    <div   style="width:   ۷۷۰px;   height:   ۲۳۰px;   background-image:
url(../Images/header.jpg);"></div>
    <div   style="width:   ۷۷۰px;   height:   ۵۲px;   background-image:
url(../Images/subHeader.png"></div>
    <div   style="width:   ۷۷۰px;   height:   ۲۳۰px;   background-image:
url(../Images/main.jpg)"></div>
    <div style="width: ۶۰۰px; height: ۲۳۰px; float: left"></div>
    <div   style="width:   ۷۷۰px;   height:   ۵۲px;   background-image:
url(../Images/subFooter.png)"></div>
    <div   style="width:   ۷۷۰px;   height:   ۱۰۲px;   background-image:
url(../Images/footer.jpg)"></div>
</body>
</html>
```

توجه : در این برنامه ویژگی div را با عبارت style مشخص کرده ایم که از css داخلی یا internal استفاده شده است در محیط css همانند محیط c# برای دادن مقادار به یک پارامتر از : و برای جدا سازی مقادیر از ; استفاده می نماییم

در DIV های تو در تو برای مشخص شدن ارنج و ترتیب و جای آنها از ویژگی float با دو مقدار right و left استفاده شده است .

CSS -مخفف کلمه **Cascading Style Sheets**میباشند.

-اولین ورژن استایلها در سال ۱۹۹۶ ساخته شد. که اولین ورژن آن CSS۱ بود.

-دیگر وژن آن CSS۲ می باشد که در سال ۱۹۹۸ ساخته شد که بیشتر برای ویرایش صفحات، نحوه نمایشها تگها و ... کاربرد دارد.

-فایل استایلها با فرمت CSS ذخیره میشوند.

-جدا کردن ظاهر صفحات اعم از طرح بندی و رنگ بندی و... از محتوای صفحات مانند کد های HTML و...

انواع استایلها:

همانطوری که قبلا نوشتم استایلها بر سه نوع میباشند:

۱:Inline Style. (internal)

این نوع استایل در داخل تگهای HTML به کار گرفته میشوند، و این استایل فقط بروی همان تگ به خصوص تاثیر خواهد گذاشت و به صورت زیر نوشته خواهد شد.

```
<p style="font-family: Tahoma;">This is a paragraph  </p>
```

در مثال بالا با استفاده از حالتهای مختلف در استایلها نوع فونت آنرا تغییر دادیم و همانطوری که در تعریف این نوع استایل نوشتم این استایل فقط و فقط تغییرات را بروی همین پاراگراف اعمال خواهد کرد.

۲:Embedded or Global Style. (external)

در این نوع، استایل نوشته شده در تمامی صفحه تاثیر خواهد گذاشت، اینرا بین دو تگ (<head></head>) به صورت زیر نوشته خواهد شد.

```
<style type="text/css">

pp { color: red; text-align: left; font-size: ۸pt }

</style>
```

این نوع حالت از تغییرات تعریف شده بروی تمامی تگها پاراگراف اعمال خواهد شد که رنگ متن : قرمز، جهت متن: چپ، و اندازه فونت ۸ پوینت خواهد بود. توجه داشته باشید در این مثال سه حالت مختلف یک استایل را برای تگ پاراگراف تعریف کردیم که با استفاده از (;) حالتهای مختلف آن از هم جدا شدند.

اگر بخواهید این حالت نوشته شده را برای چند تگ دیگر اعمال کنید می توانید تگها را بااستفاده از کاما (,) از هم جدا کنید برای مثال:

h۱, h۲, h۳, h۴ { color: gold; font-family: sans-serif }

۳: Linked or External style sheet .

نوع آخر هم معروف به استایلهای خارجی هستند که به صورت لینک فایل استایل را به صفحات خود لینک میدهید. فایلهای استایل با فرمت CSS ذخیره میشوند و به صورت زیر به صفحات لینک داده میشود.

<head><link rel="stylesheet" type="text/css" href="mystyle.css"> </head>

در این نوع از استایلها بعد از اینکه فایل استایل خود را آماده کردید آنرا با فرمت CSS ذخیره کرده و آنرا همانند بالا فایل استایل را به قالب لینک می کنیم .برای راحت بود در کار می توانید ابتدا استایلها را به صورت Internal استفاده کرده و بعد از اینکه کارتان تمام شد کدهای نوشته شده بین دو تگ (<style></style>)را در نت پد کپی کرده و آنرا با فر مت CSS ذخیره کرده و سپس با استفاده از فرمول بالا آن فایل را به قالب لینک دهید . یکی از مزیتهای این نوع استایل این میباشد که می توانید با داشتن یک فایل استایل برای چندین صفحه استفاده کنید و از دیگر مزیت های آن با این کار کدهای قالب را به چند فایل تقسیم کرده و قالب را منظم تر میشود . در مقالات بعدی کاربرد های بیشتری از سی اس اس را براتون شرح می دهم.

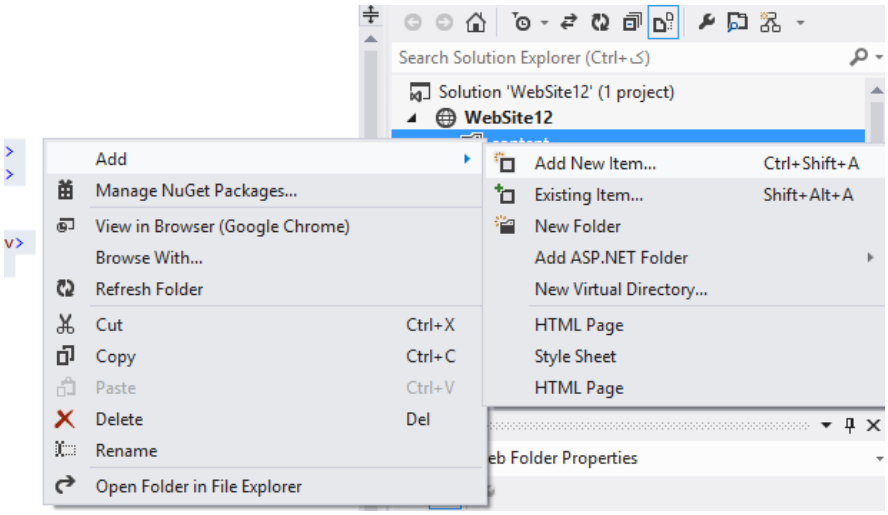
کار عملی :

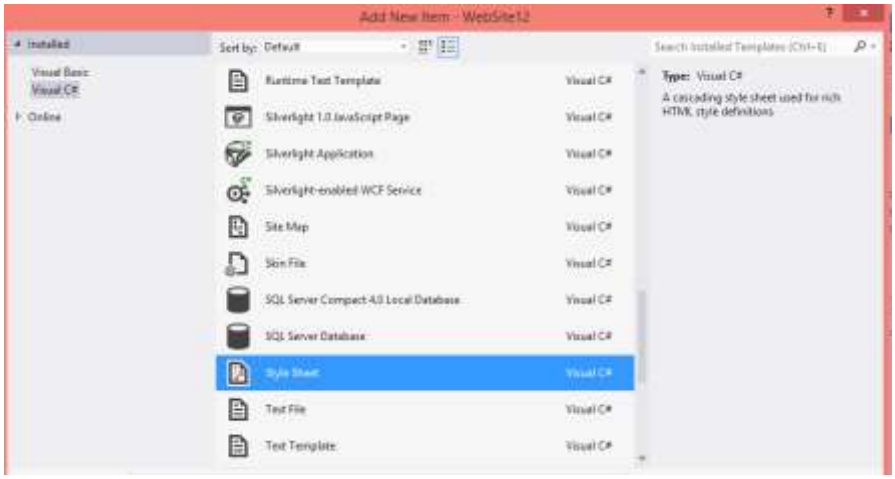
۱ -پروژه قبلی را مجدداً باز می نماییم

```
<!DOCTYPE html>
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<head>
    <title></title>
</head>
<body>
    <div style="width: ۷۷۰px; height: ۲۳۰px; background-image:
url(../Images/header.jpg);"></div>
    <div style="width: ۷۷۰px; height: ۵۲px; background-image:
url(../Images/subHeader.png"></div>
```

```
    <div    style="width:    ۷۷۰px;    height:    ۲۳۰px;    background-image:
url(../Images/main.jpg)"></div>
    <div style="width: ۶۰۰px; height: ۲۳۰px; float: left"></div>
    <div    style="width:    ۷۷۰px;    height:    ۵۲px;    background-image:
url(../Images/subFooter.png)"></div>
    <div    style="width:    ۷۷۰px;    height:    ۱۰۲px;    background-image:
url(../Images/footer.jpg)"></div>
</body>
</html>
```

همانطور که ملاحظه می کنید از CSS داخلی استفاده شده است . برای راحتی کار و تغییرات در صفحات طولانی و متعدد از CSS خارجی استفاده می کنیم که برای اینکار مراحل زیر انجام می دهیم .در داخل فلدر content دو فایلCSS با نام های site۱ و site۲باز می کنیم .





۲- در فایل site۱از روش .(دات) و فایل site۲ از روش #(شارپ) استفاده می نماییم و style های داخل فایل اصلی به داخل فایل های مربوطه می آورم .

محتوای site۱ :

```
.header
{
```

```
	width: ٧٧٠px; height: ٢٣٠px; background-image:
url(../Images/header.jpg);
}

.subHeader
{
	width: ٧٧٠px; height: ٥٢px; background-image:
url(../Images/subHeader.png);
}

.main
{
	width: ٧٧٠px; height: ٢٣٠px; background-image:
url(../Images/main.jpg);
}

.content
{
	width: ٦٠٠px; height: ٢٣٠px; float: left;
}

.content_left
{
	width: ٤٠px; height: ٤٠px; float: left;
}

.content_right
{
	width: ١٢١px; height: ٧٠px; float: right; direction: rtl;
}

.menu
{
	width: ١٧٠px; height: ٢٣٠px; float: right;
}

.subFooter
{
	width:٧٧٠px; height:٥٢px; background-
image:url(../Images/subFooter.png);
```

```
}

.footer
{
        width:۷۷۰px; height:۱۰۲px;background-
image:url(../Images/footer.jpg);
}
```

محتوای فایل site۲ :

```
#header
}
        width: ۷۷۰px; height: ۲۳۰px; background-image:
url(../Images/header.jpg;(
{

#subHeader
}
        width: ۷۷۰px; height: ۵۲px; background-image:
url(../Images/subHeader.png;(
{

#main
}
        width: ۷۷۰px; height: ۲۳۰px; background-image:
url(../Images/main.jpg;(
{

#content
}
        width: ۶۰۰px; height: ۲۳۰px; float: left;
{

#content_left
}
        width: ۴۰px; height: ۴۰px; float: left;
```

```
{

#content_right
}
        width: ۱۲۱px; height: ۷۰px; float: right; direction: rtl;
{

#menu
}
        width: ۱۷۰px; height: ۲۳۰px; float: right;
{

#subFooter
}
        width:۷۷۰px; height:۵۲px; background-
image:url(../Images/subFooter.png;(
{

#footer
}
        width:۷۷۰px; height:۱۰۲px;background-
image:url(../Images/footer.jpg;(
}
```

توجه:

برای استفاده از فایل css در برنامه دو مرحله زیر انجام می دهیم :

۱ -ابتدا با درگ کردن فایل css به انتهای تگ head به از title قرار می دهیم

۲ -در داخل هر دیو مشخص می کنیم که کدام بخش از css بخواند . برای اینکار اگر از نقطه استفاده کرده باشیم در داخل فایل html از عبارت class استفاده می کنیم و نام استال شیت را بنویسیم

** **

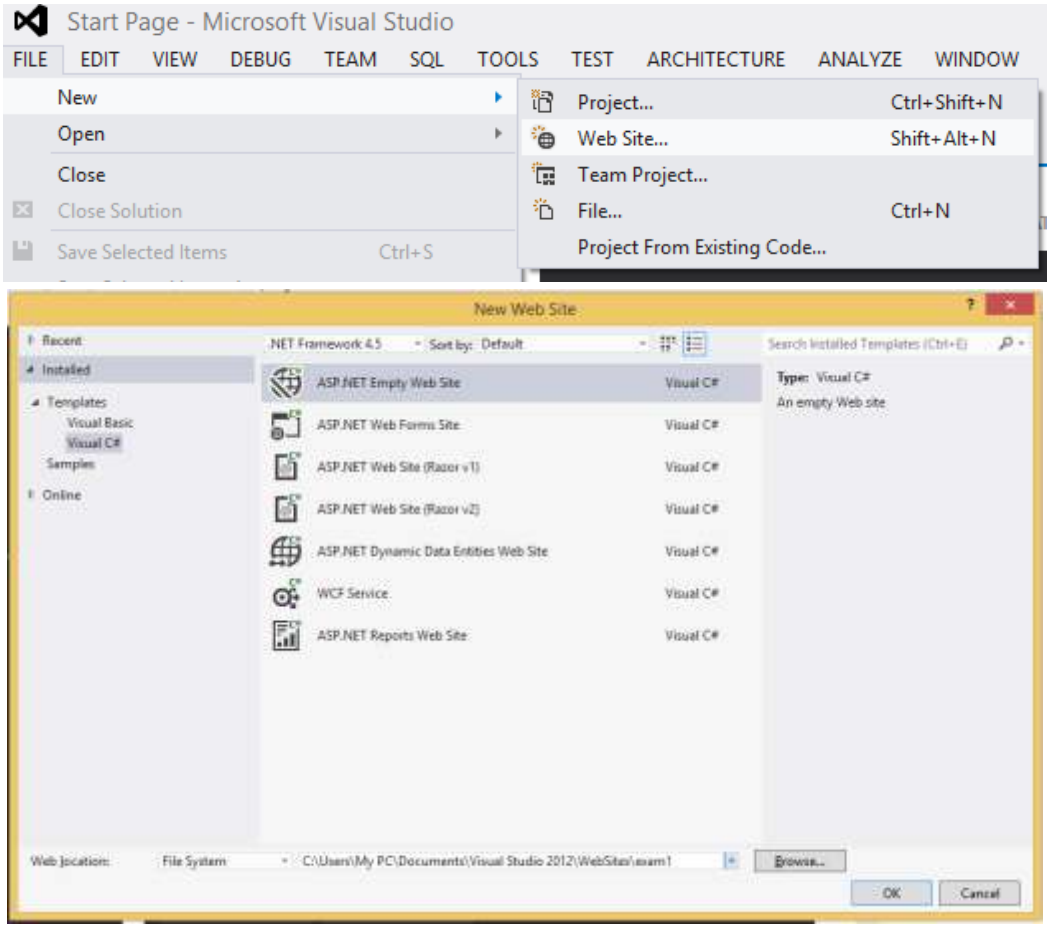
نام را بصورت intelligence شده می آورد و این از ویژگی class است .

اگر در فایل css از # استفاده کنیم باید در فایل html از عبارت id استفاده نماییم

** **

جلسه پنجم: روش form application

برای اجرای برنامه به روش form application مراحل زیر عمل می کنیم :

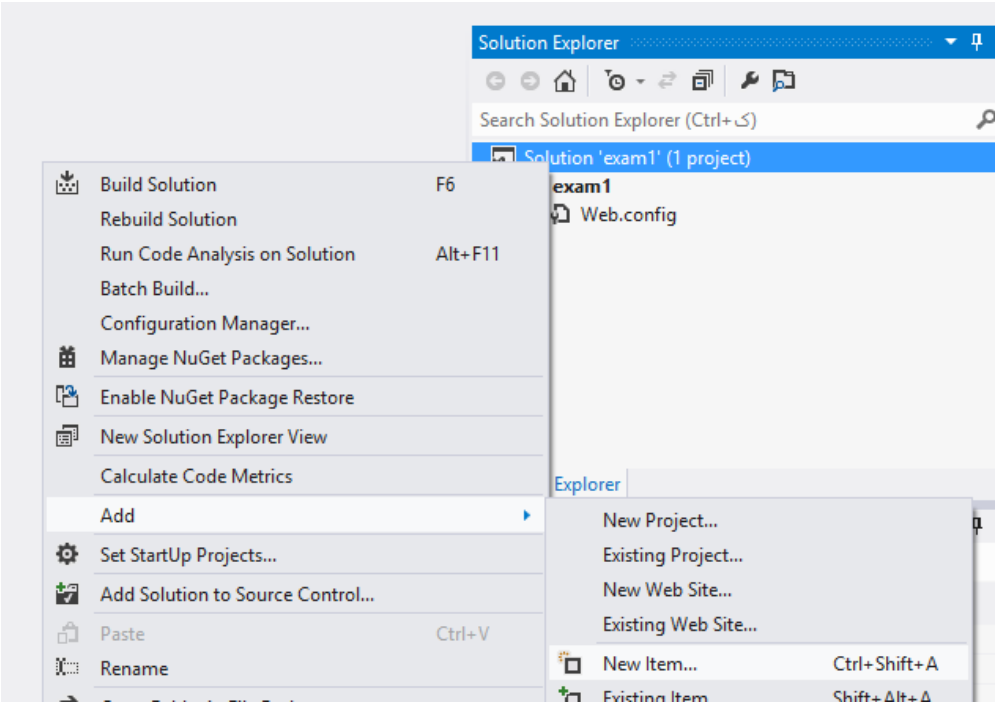


ونام exam۱ ایجاد می کنیم .

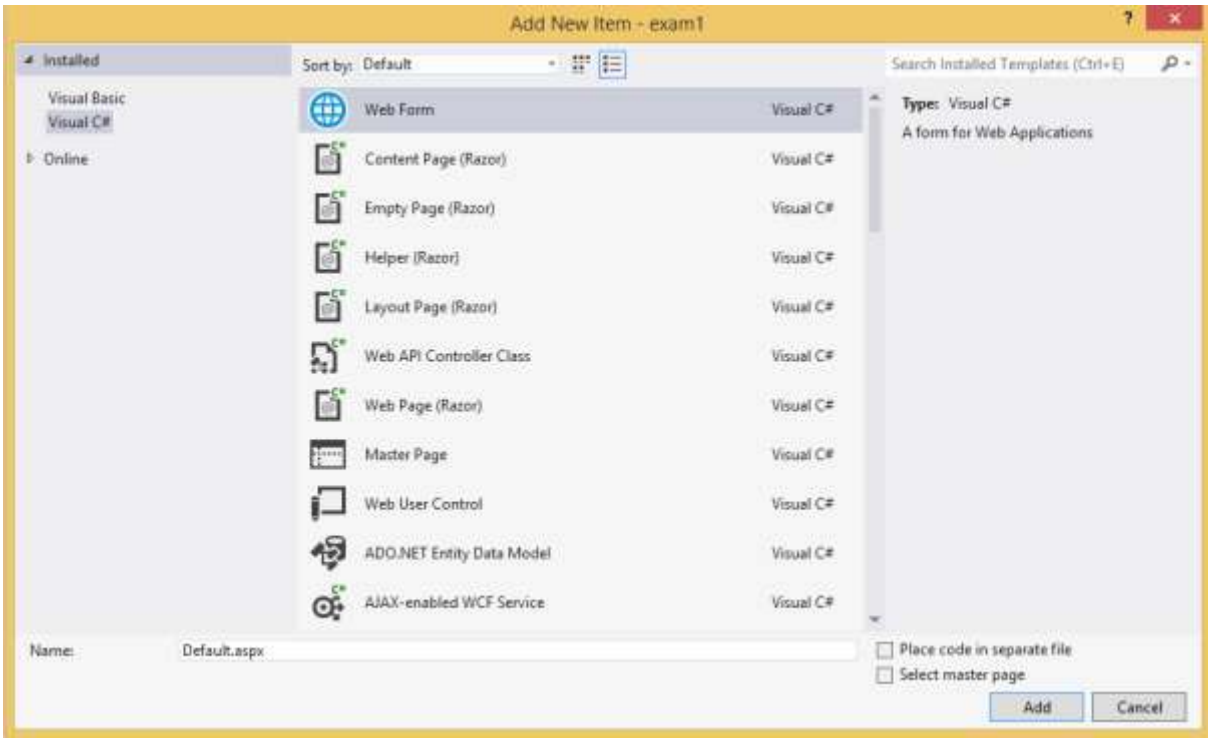
برای کارکردن با asp.netنیاز به صفحاتی داریم که فرمت asp.net دارند.برای ایجاد یک فایل به روش زیر عمل می کنیم .

بر روی پوشه exam۱ راست کلیک می کنیم ، سپس گزینه add و add new item را کلیک می کنیم

s.ali.taher@gmail.com



سپس به روش زیر عمل می کنیم



در صفحه مربوطه کدهای زیر ظاهر می گرددو

```
<%@ Page Language="C#" %>

<!DOCTYPE html>

<script runat="server">

</script>
```

```
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form۱" runat="server">
    <div>

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

در پروژه ویندوز اپلیکیشنی وقتی زبانی انتخاب می کردیم در کل پروژه همان زبان استفاده می شدولی در پروژه های ولی در وب سایت ها به صورت parpage کار می شود یعنی هر صفحه می تواند از زبانnet, را پشتیبانی کند.

در این فایل کدهای c# را در داخل تگ script می نویسیم

تگ runat="server" مشخص می کند که برنامه ما سرور ساید است یعنی کنترلهای ما سمت سرور اجرا خواهند شد و نتیجه به سمت client برمی گردانند.اگر این تگ در برنامه نداشته باشیم یعنی cilent side هستند.فرق بین این پروژه با پروژه html در این است که کدهای c# باید ذر تگ script نوشته شود و عبارت runat="server" و کدهای برنامه باید در تگ فرم نوشته شود.

در نوشتن لیبل ها وایدی هاو cssها از استاندارد کمل و بقیه موارد از استاندارد پاسکال استفاده می کنیم . استاندارد کمل frm Main

استاندارد پاسکال مثل :Hello World(اولین کلمه بزرگ بقیه کوچک).

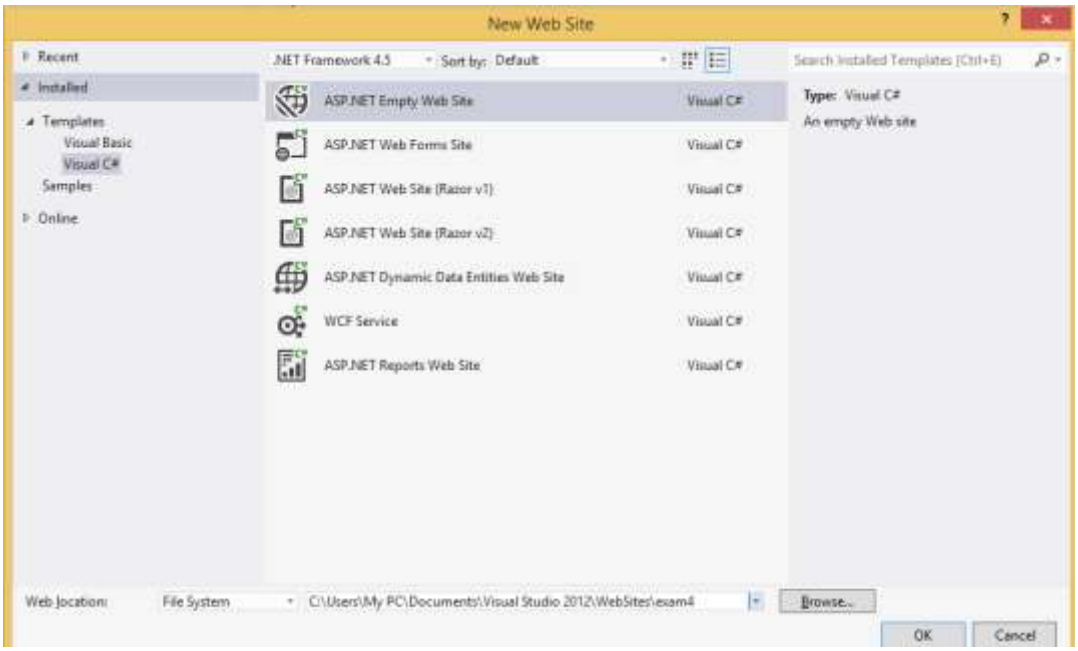
جلسه ششم :مسترپیج چیست ؟

مسترپیج به طراح این امکان را می دهد تا یک صفحه یا قالب (الگو) پیش ساخته را برای ساخت صفحات دیگر ، طراحی نماید . در واقع مسترپیج ظاهر ، قالب و نحوه عملکرد کلی گروهی از صفحات را در پروژه یا وب سایت شما تعیین می کند . سپس می توانید صفحاتی را به عنوان صفحات محتوا ، به صورت تکی ایجاد نموده که این صفحات شامل محتوا متغیر سایت شما بوده و آن ها را به مسترپیج متصل می کنید .

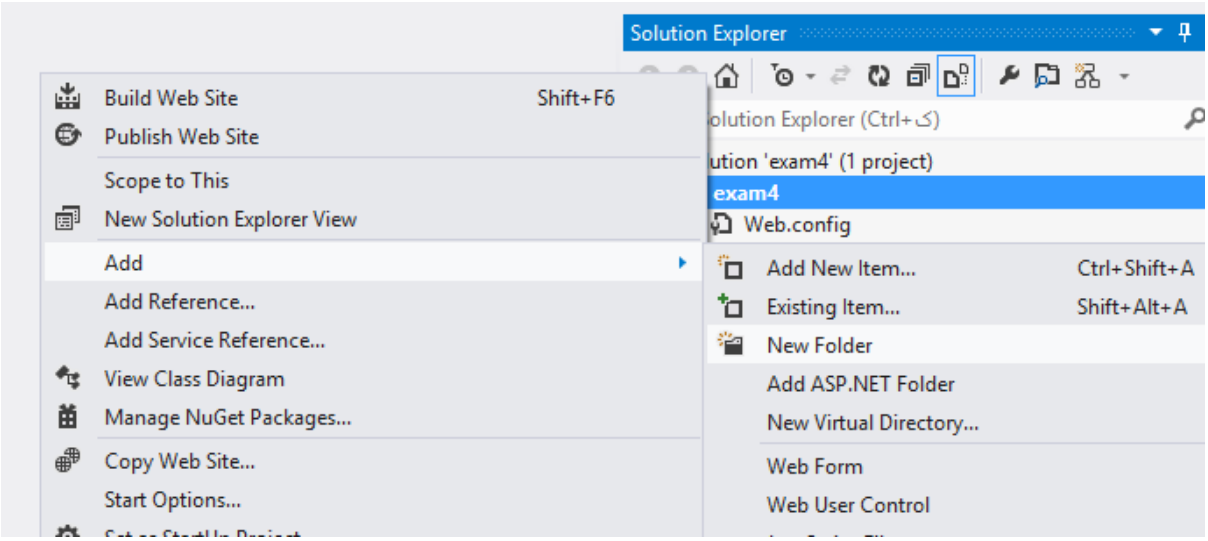
هنگامی که کاربر صفحه محتوا (Content Page) را درخواست می کند ، سرور ASP.Net صفحه محتوا را با صفحه مسترپیج ترکیب کرده و تم های موجود را به آن ها اعمال نموده و خروجی را در قالب یک فایل می سازد .

برای درک بهتر مسترپیج ها ، یک مثال می زنیم .

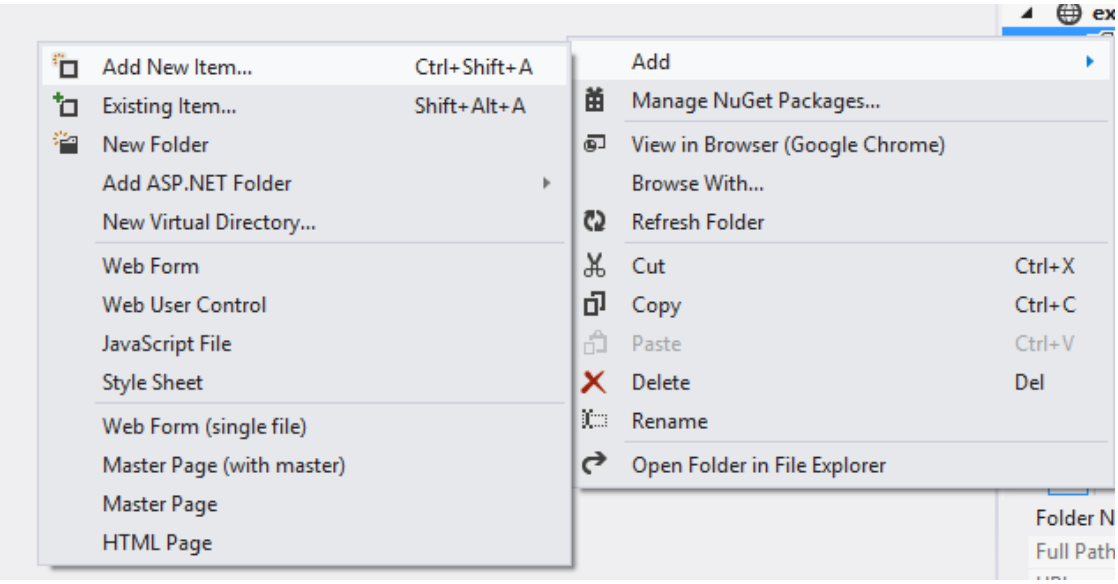
ابتدا مثل پروژه های قبلی عمل می کنیم :



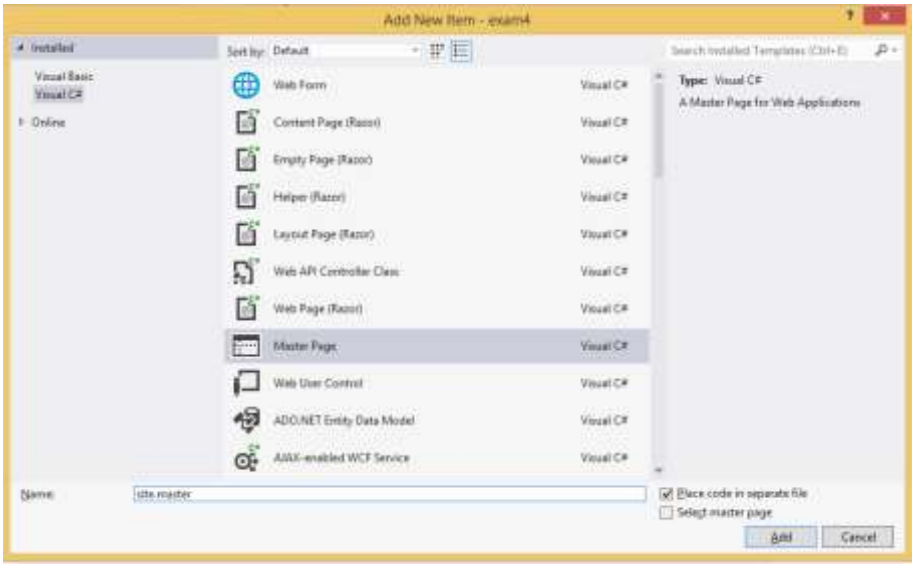
سپس پوشه ای به نام master pages ایجاد می کنیم .



روی فلدر master pages راست کلیک نموده و سپس add و add new item را کلیک می کنیم



سپس مراحل زیر را انجام می دهیم و نام siteبرای آن انتخاب می کنیم

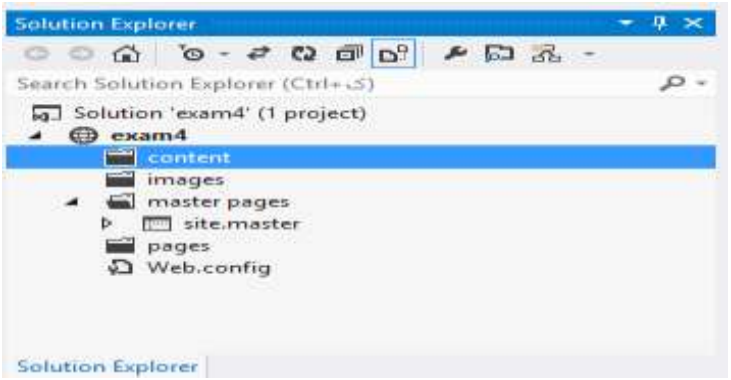


کد زیر ظاهر می گردد. و تگ های div اضافه می کنیم

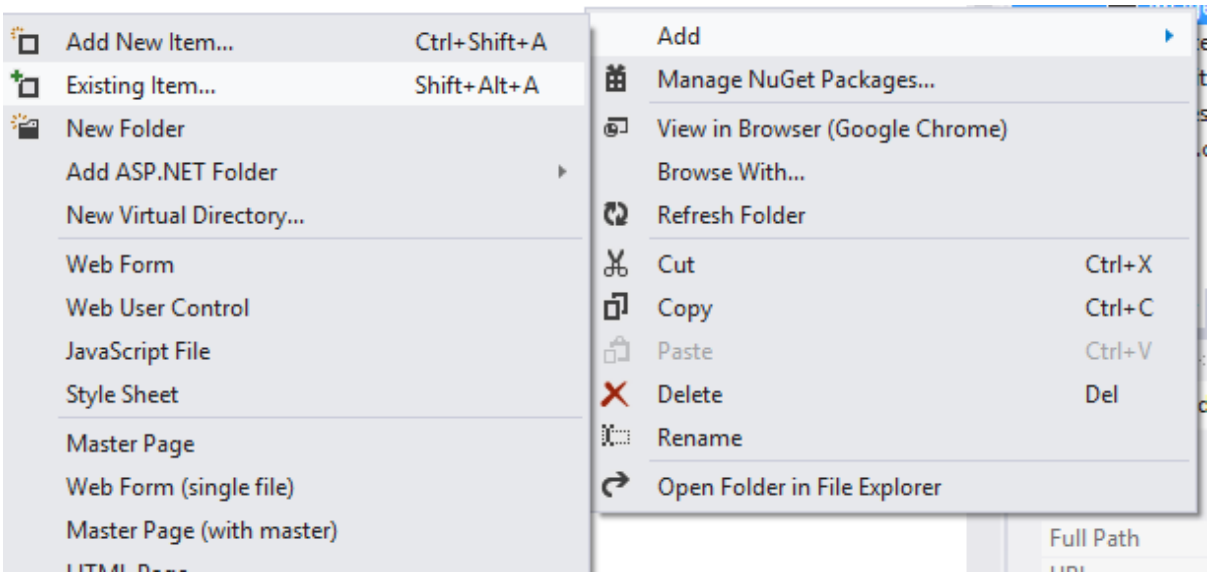
```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="site.master.cs" Inherits="master_pages_site" %>

<!DOCTYPE html>

<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
```

```
<body>
    <form id="form1" runat="server">
        <div>
            <div></div>
            <div></div>
            <div>
                <asp:ContentPlaceHolder
ID="ContentPlaceHolder1" runat="server">
                </asp:ContentPlaceHolder>
            </div>
            <div></div>
            <div></div>
        </div>
    </form>
</body>
</html>
```

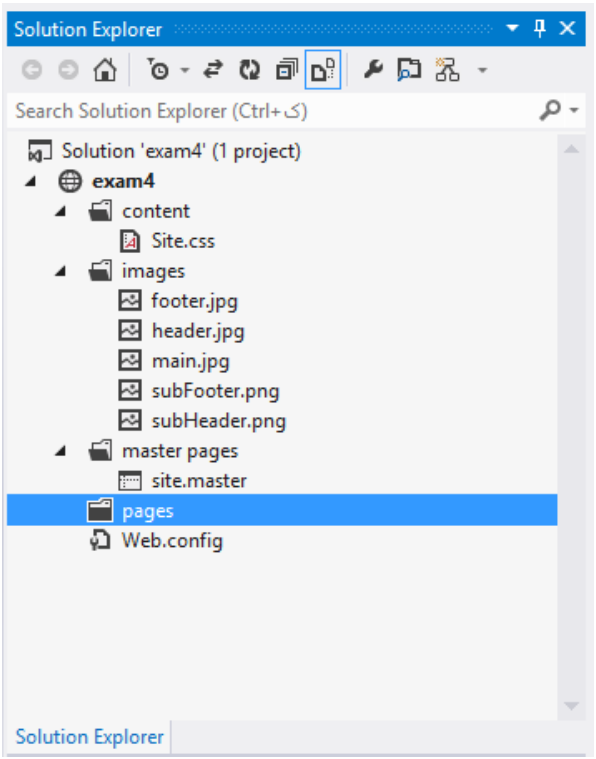
سپس فلدرهای pages و images وcontent به پروژه اضافه می کنیم .



در داخل فلدرهای محتوای پروژه های قبلی لود می کنیم



و سپس رفرش می کنیم

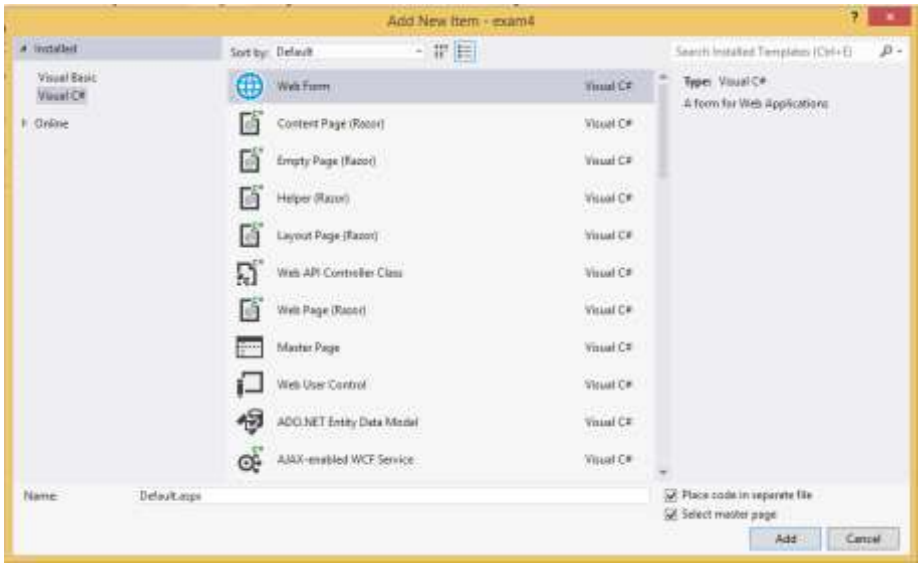


مثل تنظیمات css در فایل site.master تغییرات زیر لحاظ می کنیم

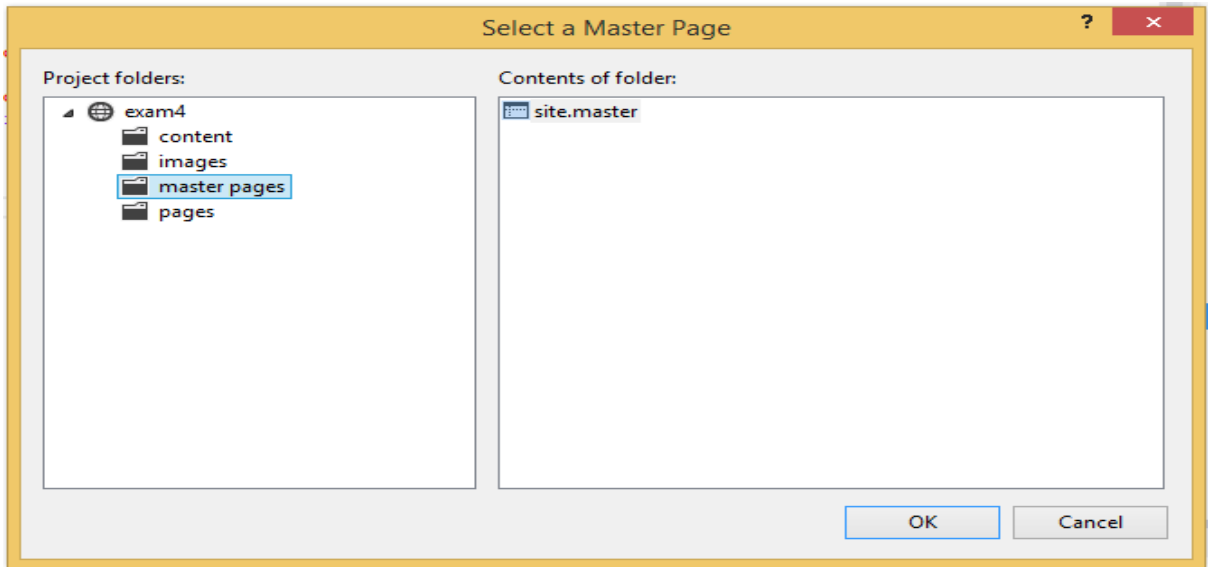
```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="site.master.cs" Inherits="master_pages_site" %>
<!DOCTYPE html>
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<head runat="server">
    <title></title>
```

```
    <link href="../content/Site.css" rel="stylesheet" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form\" runat="server">
        <div>
            <div class="header"></div>
            <div class="subHeader"></div>
            <div class="main">
                <asp:ContentPlaceHolder
ID="ContentPlaceHolder\" runat="server">
                </asp:ContentPlaceHolder>
            </div>
            <div class="footer"></div>
            <div class="subFooter"></div>
        </div>
    </form>
</body>
</html>
```

حال در design برنامه باید فرم ما کامل باشد .سپس روی فلدر pages راست کلیک می کنیم .گزینه add و add new item و صفحه زیر ظاهر می گردد.گزینه select master page را تیک می زنیم



سپس در صفحه زیر موارد مشخص شده انتخاب می کنیم و ok کلیک می کنیم .



تا کد default.aspx ظاهر گردد:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="pages_Default" %>

<asp:Content ID="Content۱" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content۲"
ContentPlaceHolderID="ContentPlaceHolder۱" Runat="Server">
</asp:Content>
```

استفاده از کنترهای html (کلاینت سایید):

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="pages_Default" %>
<asp:Content ID="Content۱" ContentPlaceHolderID="head"
runat="Server">
</asp:Content>
<asp:Content ID="Content۲"
ContentPlaceHolderID="ContentPlaceHolder۱" runat="Server">
    First Name:
    <input type="text" />
```

```
    <br />
    Last Name:
    <input type="text" />
    <br />
    Full Name:
    <input type="text" />
    <br />
    <input type="submit" value="submit" />
    <input type="button" value="cansel" />
</asp:Content>
```

جلسه هفتم : code behine

این کنترل های مربوط به client side هست و معمولا به زبان html نوشته می شود که می توانیم از toolbar و گزینه های html با درگ کردن استفاده کنیم . اما چنانچه از کنترل های asp.net استفاده کنیم و به روش زیر عمل می کنیم :

ابتدا یک web form جدید باز می کنیم :

کد زیر ظاهر می گردد:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="aspcontrols.aspx.cs" Inherits="pages_aspcontrols" %>
<asp:Content ID="Content۱" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content۲"
ContentPlaceHolderID="ContentPlaceHolder۱" Runat="Server">
</asp:Content>
```

برای اینکه از کنترلهای asp.netاستفاده کنیم به شکل زیر عمل می کنیم :

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="aspcontrols.aspx.cs" Inherits="pages_aspcontrols" %>

<asp:Content ID="Content۱" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content۲"
ContentPlaceHolderID="ContentPlaceHolder۱" Runat="Server">
   <div>
```

```
        <asp:Label ID ="lblfirstname" runat ="server"  Text
="firstname:"/>
        <asp:TextBox ID="txtfirstname" runat ="server" MaxLength
="۲۰" />
        <br />
        <asp:Label ID="lbllastname" runat ="server" Text
="lastname:" />
        <asp:TextBox ID="txtlastname" runat ="server" MaxLength
="۲۰" />
        <br />
        <asp:Label ID ="lblfullname" runat ="server"
text="fullname:" />
        <asp:TextBox ID="txtfullname" runat="server" MaxLength
="۴۰" />
        <br />
        <input type="submit" value="Submit" runat ="server" />
        <asp:Button ID="btncancel" runat ="server" Text
="cancel" />
    </div>

</asp:Content>
```

برای استفاده از تکنیک **Legend**:

```
asp:Content ID="Content1" ContentPlaceHolderID="cphHead" Runat="Server">
/asp:Content>
asp:Content ID="Content2" ContentPlaceHolderID="cphMain" Runat="Server">
   <fieldset>
       <legend>Legend</legend>
       <div class="field">
           <div class="lable">
               <asp:label id="lblFirtName" runat="server" text="FirtName" />
           </div>
           <div class="controls">
               <asp:textbox id="txtFirtName" runat="server" maxlength="20" />
           </div>
       </div>
   </fieldset>
/asp:Content>
```

```
.controls
{
    font-family:Georgia;

}
.field
{
    padding:1px;
    margin:1px;
}
.button
{
    font-family:Georgia;
}
```

:

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    txtFirstName.Text =
        txtFirstName.Text.Trim();
    txtLastName.Text =
        txtLastName.Text.Trim();

    txtFullName.Text =
        txtFirstName.Text + txtLastName.Text;
}
```

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    txtFirstName.Text =
        txtFirstName.Text.Trim();
    txtLastName.Text =
        txtLastName.Text.Trim();

    txtFullName.Text =
        string.Format("{0} {1}",txtFirstName.Text,txtLastName.Text);
}
protected void btnClare_Click(object sender, EventArgs e)
{
    this.txtLastName.Text = string.Empty;
    this.txtFirstName.Text = string.Empty;
}
```

جلسه هشتم: تگ گروه بندی Fieldset , Legend

s.ali.taher@gmail.com

تگ Fieldset برای گروه بندی در فرم ها بسیار کاربرد دارد . این تگ در Asp.netمعادل سروری کنترل Panel می باشد. برای تعیین عنوان یک گروه از تگ Legendدر ابتدای تگ Fieldset استفاده می گردد.

نکته ی بسیار مهم :در صورت استفاده ننمودن از کدنویسی سروری برای کنترل های Asp.net توصیه اکید می شود که از تگ های معادل آنها استفاده شود . بطور مثال بجای استفاده از کنترل Asp:Panel از تگ Fieldset و Legendاستفاده شود. زیرا در نهایت کنترل های Asp.net به‌تگ های Html خالص تبدیل شده و برای جلوگیری از این سربار اضافه از تگ های مستقیم آنها استفاده نمایید.

مزیت دیگر این روش کنترل بهتر آنها در Style آنها بوسیله CSS می باشد که کنترل های Asp.net بعضا در این مورد مشکل زا هستند.

نمونه ای از کاربرد تگ Fieldset و Legend را در مثال زیر می بینید:

```
<fieldset>
    <legend>عنوان گروه</legend>

    <p>این متن تست است.</p>

    <p>این متن تست شماره ۲ است.</p>
</fieldset>
```

خصوصیت اصلی

خصوصیت اصلی تگ Legend خصیصه ی Align یا تراز می باشد که محل قرار گیری عنوان را در این تگ تعیین میکند.

مقدار هایی که این خصوصیت می پذیرد عبارتند ازTop , Bottom , Left , Right :

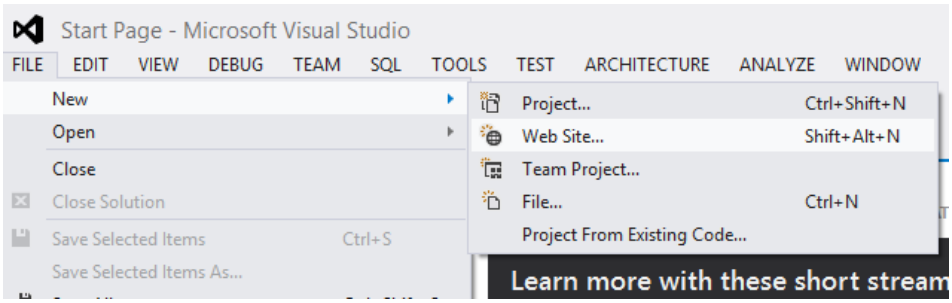
مثالی در این زمینه:

```
<fieldset>

    <legend align="left">عنوان گروه</legend>
```

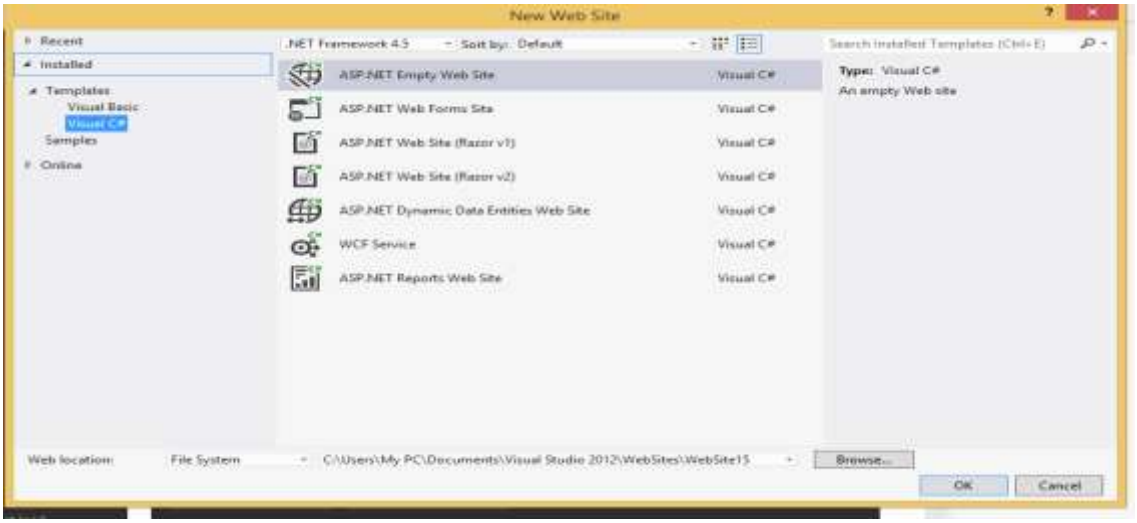
```
<p>این متن تست است.</p>

<p>این متن تست شماره ۲ است.</p>

</fieldset>
```

آشنایی با مفهوم user control:

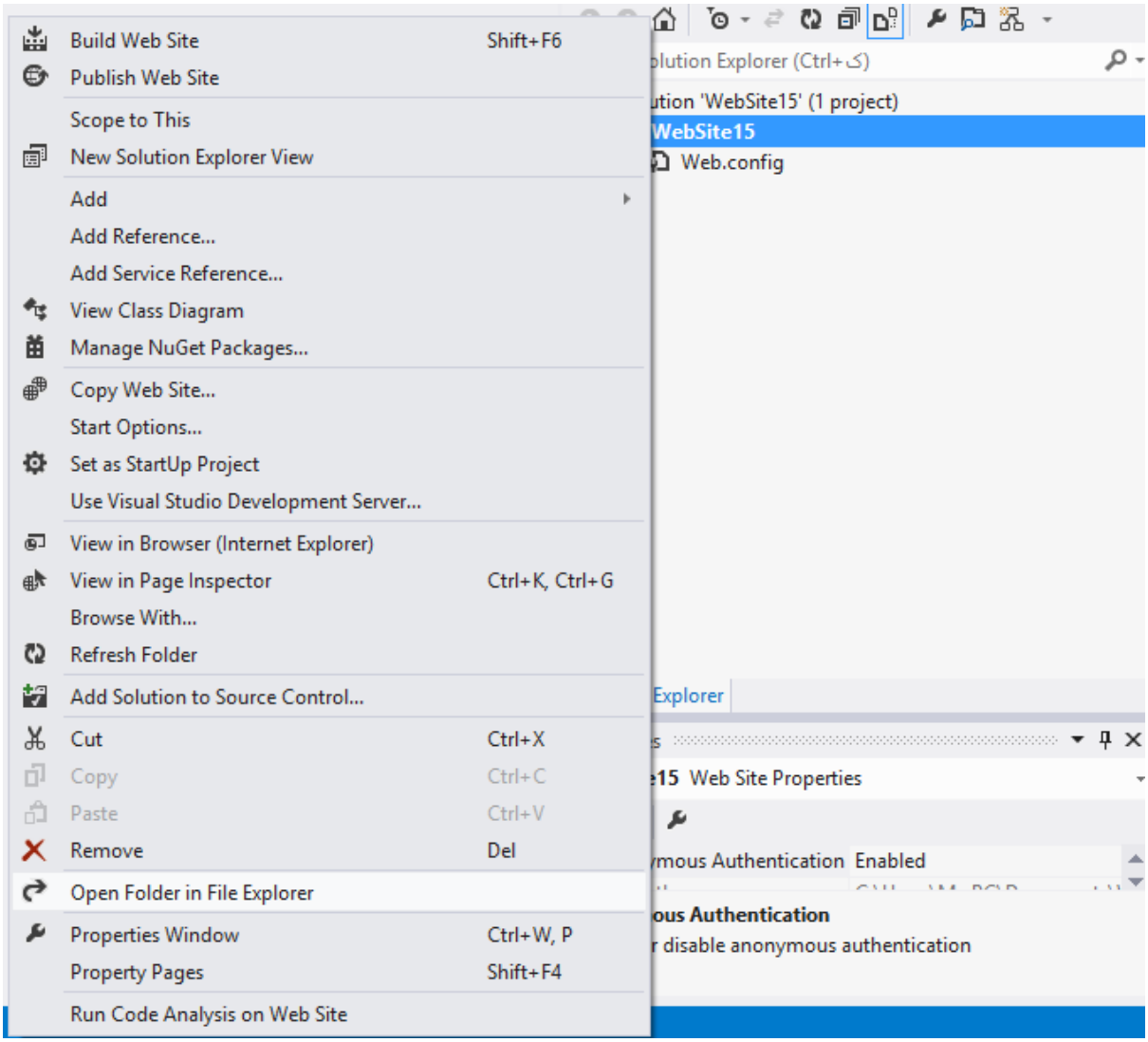
ابتدا پروژه جدیدی ایجاد می کنیم و فلدرهای قبلی را در آن قرار می دهیم :



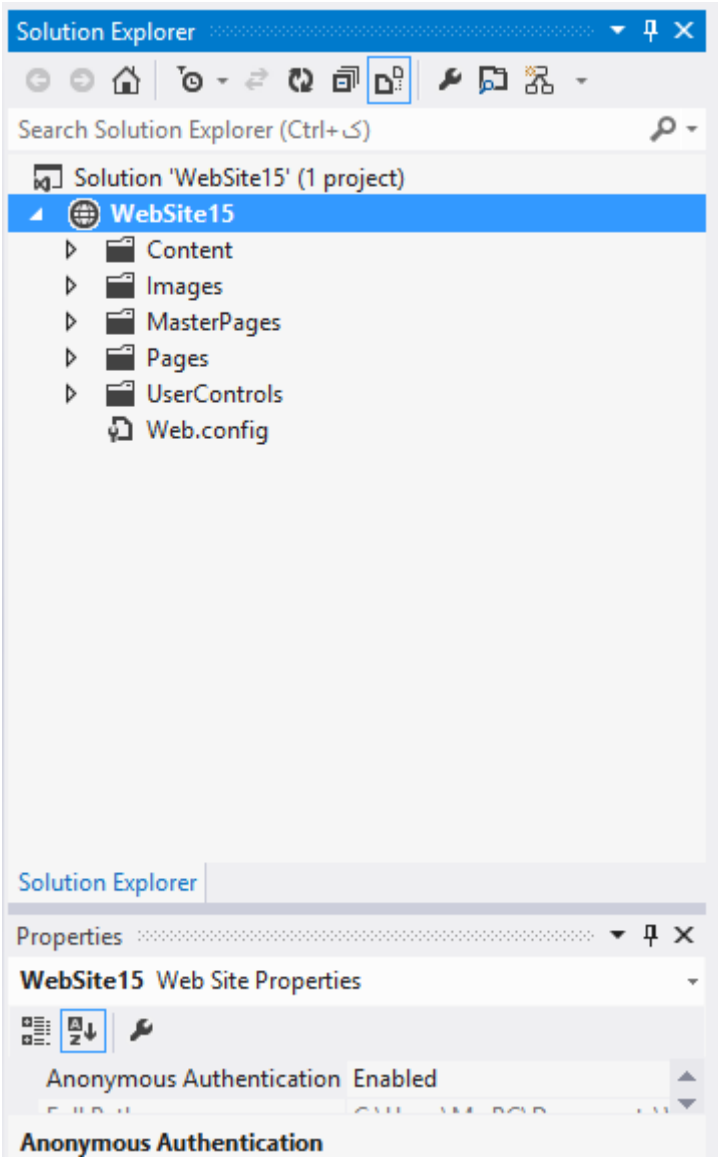
سپس:



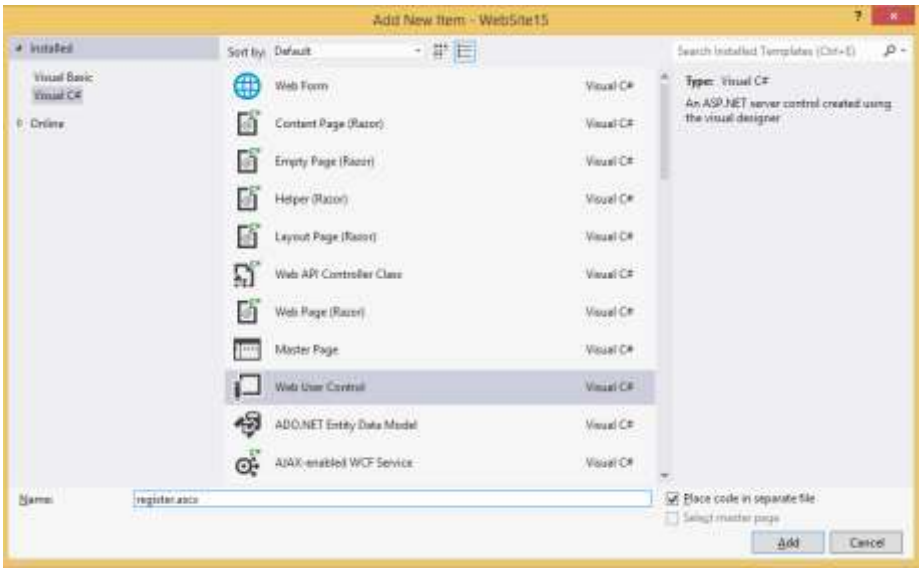
برای قرار دادن فلدر های قبلی روی نام پروژه راست کلیک می کنیم



در مکان باز شده فلدرها را کپی می نمابیم و سپس رفرش می کنیم :



همان طور که می بینید یک پوشه ی جدیدی به نام user control ایجاد نموده و در داخل این پوشه یک آیتم جدبد اضافه می کنیم :



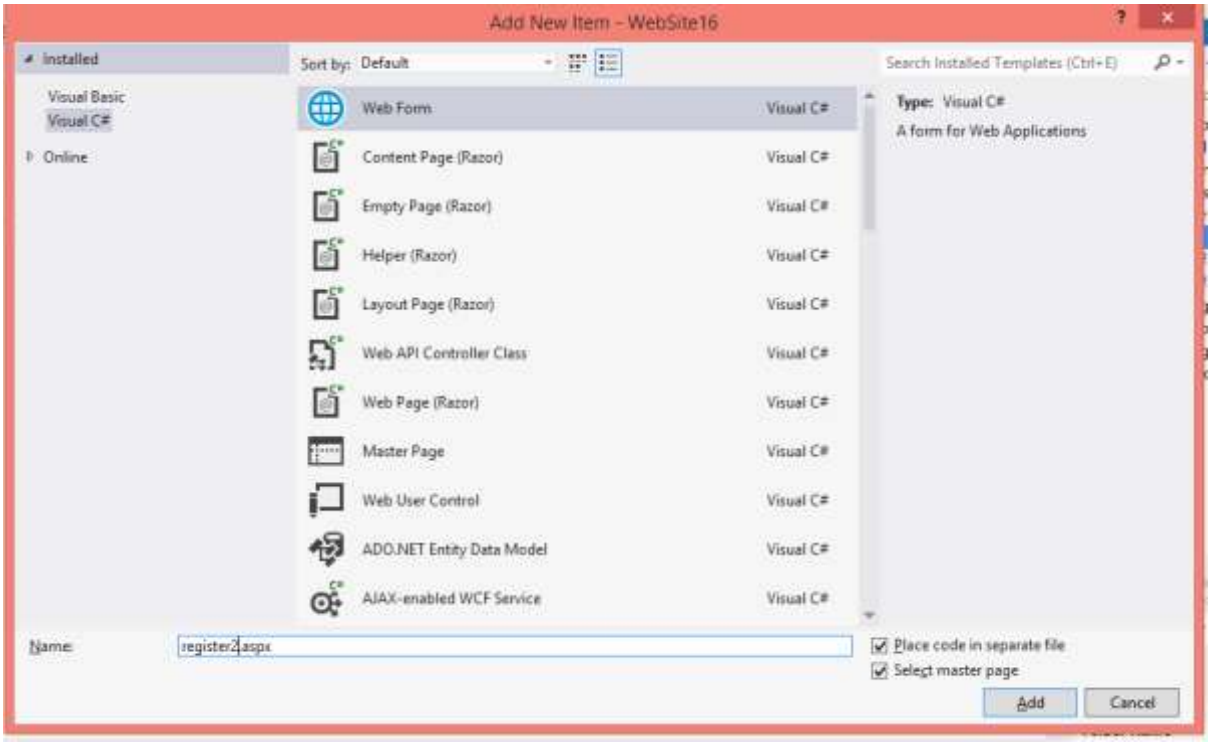
در صفحه باز شده می توانیم کد کنترلی را ایجاد کنیم

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Register.ascx.cs" Inherits="UserControls_Register" %>
<link href="../Content/Site.css" rel="stylesheet" />
<center>
<fieldset>
<legend>Register</legend>
<div class="field">
<div class="lable">
<asp:Label ID="lblFirstName" runat="server" Text="FirstName :"
/>
</div>
<div class="controls">
<asp:TextBox ID="txtFirstName" runat="server" MaxLength="٢٠" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblLastName" runat="server" Text="LastName :" />
</div>
<div class="controls">
<asp:TextBox ID="txtLastName" runat="server" MaxLength="٢٠" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblEmail" runat="server" Text="Email :" />
</div>
<div class="controls">
<asp:TextBox ID="txtEmail" runat="server" MaxLength="٢٠" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblPasswoed" runat="server" Text="Password :" />
</div>
<div class="controls">
<asp:TextBox ID="txtPassword" runat="server" MaxLength="٢٠" />
</div>
```

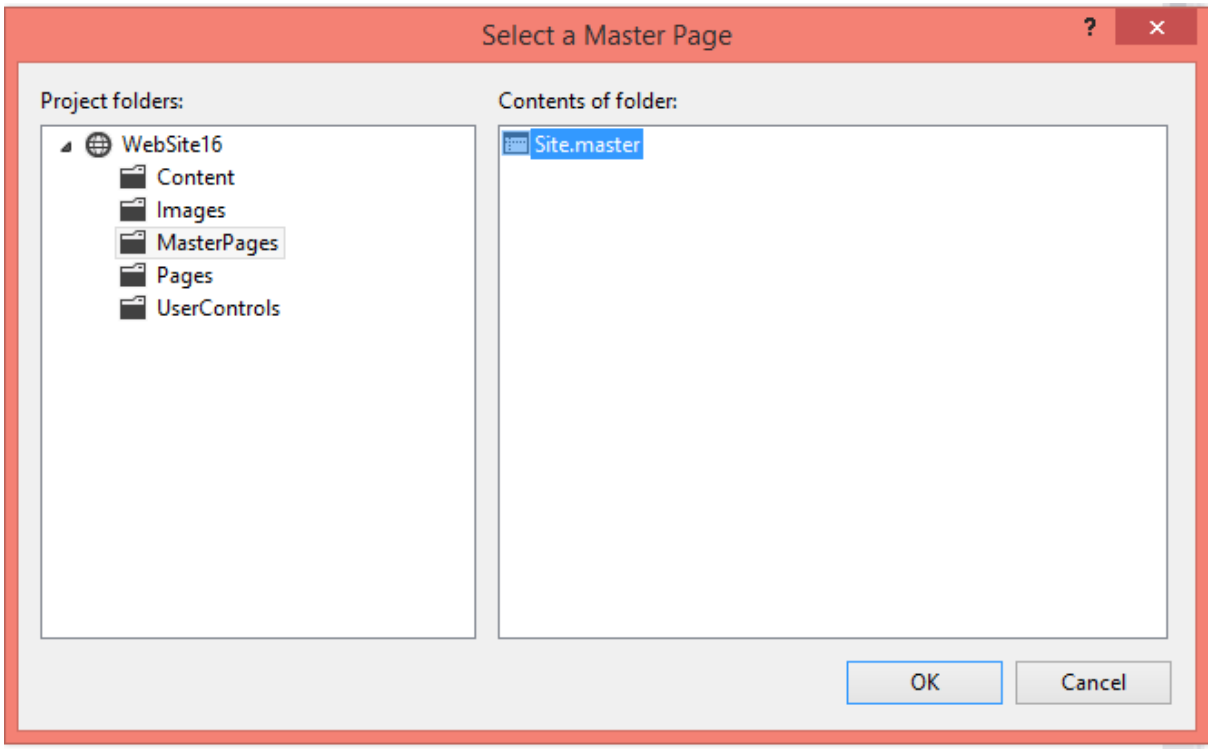
```
</div>

<div class="field">
<div class="button">
<asp:Button ID="btnRegister" runat="server" Text="Register" />
 
<asp:Button ID="btnCancel" runat="server" Text="Cancel" />

</div>
</div>
</fieldset>
</center>
```

حال می توانیم این یوزر کنترل را در مستر پیج خود لحاظ نماییم تا هر زمان که از مستر پیج ا ستفاده می نماییم اجزای یوزر کنترل نیز در صفحه اضافه گردد. برای اینکار به روش زیر اقدام می نماییم .

۱ -روی فولدر pages راست کلیک نموده و از add new item استفاده کرده و یک web page جدید ایجاد می کنیم



۲ -در حالی که تیک select master page خورده است گزینه add را کلیک می کنیم



۳ -در فولدر master pagesمستر پیج site.master را انتخاب ودکمه ok را کلیک می کنیم .

۴ -در صفحه باز شده register۲.aspx از فولدر usercontrols فایل register.ascx را مانند شکل زیر درگ می کنیم .

۵ -همزمان با درگ کردن دو خط زیر در فایل ایجاد می گردد

```
<%@ Register Src="~/UserControls/Register.ascx" TagPrefix="uc۱"
TagName="Register" %>

    <uc۱:Register runat="server" ID="Register" />
```


s.ali.taher@gmail.com

۶ - حالا می توانیم به دفعات از این یوزر کنترل در برنامه استفاده کنیم

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPages/Site.master"
AutoEventWireup="true" CodeFile="register۲.aspx.cs"
Inherits="Pages_register۲" %>

<%@ Register Src="~/UserControls/Register.ascx" TagPrefix="uc۱"
TagName="Register" %>

<asp:Content ID="Content۱" ContentPlaceHolderID="cphHead"
Runat="Server">
</asp:Content>
<asp:Content ID="Content۲" ContentPlaceHolderID="cphMain"
Runat="Server">
    <uc۱:Register runat="server" ID="Register" />
    <uc۱:Register  runat="server" ID ="r۲" />
</asp:Content>
```

جلسه نهم :طراحی cms (سیستم مدیریت محتوا)

سیستم های مدیریت محتوا و یا به اصطلاح CMS مخفف Content Managment Systems ابزار و برنامه هایی هستند که به شما کمک می کند تا مطالب خود را ایجاد و در یک منبع مشترک ذخیره سازی نمایید، و محتویات متنی و تصویری سایت خود را به صورت کاملاً پویا و آنلاین بروزرسانی کنید و از آن پس در مدیریت ارتباطات بین اجزا نیز به شما کمک خواهد نمود .

در ادامه به معرفی برترین CMS ها و Forum هایی که قابلیت پشتیبانی کامل از زبان پارسی را نیز دارند می پردازیم:

جوملا یک سیستم مدیریت محتوای کد باز و قدرتمند است که برای هر نوع سایت ساده و یا پیچیده قابل بکارگیری می باشد. زبان برنامه نویسی آن PHP بوده و از بانک اطلاعاتی MySQL استفاده می کند. از ویژگیهای آن می توان به ذخیره گاه صفحه اشاره نمود که به افزایش قدرت اجرای آن کمک می کند.

وردپرس نیز یک سیستم مدیریت محتوای متن باز است که به خوبی با سایت های جستجوگر هماهنگ می شود ، پلاگین های بسیار زیادی متناسب با نیاز کار ارائه کرده است ، در ضمینه های مختلف می توان از این سیستم استفاده کرد ، در ایران و سایر نقاط جهان طرفداران زیادی را دارست

مامبو را می توان مادر جوملا در نظر گرفت زیرا از بسیاری جهات ساختاری مشابه جوملا داشته و هر دو سیستم از یک سورس بهره جسته اند . مامبو با بهره گیری از زبان سطح بالای PHP و بکار گیری بانک های اطلاعاتی پهناور همانند جوملا امکان به جنبش در آوردن محتوای انتشار یافته در وبسایت را فراهم می کند.

پی اچ پی نیوک که از چند سال پیش طرفداران بسیاری در ایران پیدا کرده است ، یک سیستم مدیریت محتوای پیشرفته می باشد که بعنوان سیستم خودکار انتشار اخبار در اینترنت و اینترانت طراحی شده و از ویژگی های آن می توان به مدیریت آسان سایت و کاربران و همچنین تعامل صد در صد با بانک های اطلاعاتی اشاره نمود.

موویل تایپ یک سیستم انتشار وبلاگ مبتنی بر زبان پرل می باشد که در سال ۲۰۰۱ اولین نسخه جدی آن منتشر شد. این سیستم تمامی خصیصه های مورد نیاز یک وبلاگ را پشتیبانی می کند ، مواردی از قبیل سیستم اعضا ، نظر گذاری ، انتخاب قالب ، بخش مدیریت وبلاگ و غیره

e۱۰۷این سیستم مدیریت محتوای کد باز که از بانک اطلاعاتی معروف MySQL بهره می گیرد ، مدت زمان زیادی نیست که در دنیای وب مطرح شده ، با این حال از ویژگی های آن می توان به انعطاف پذیری و امکان انتشار سریع آن منطبق بر نیاز طراح و بکارگیری قالبهای مختلف اشاره نمود.

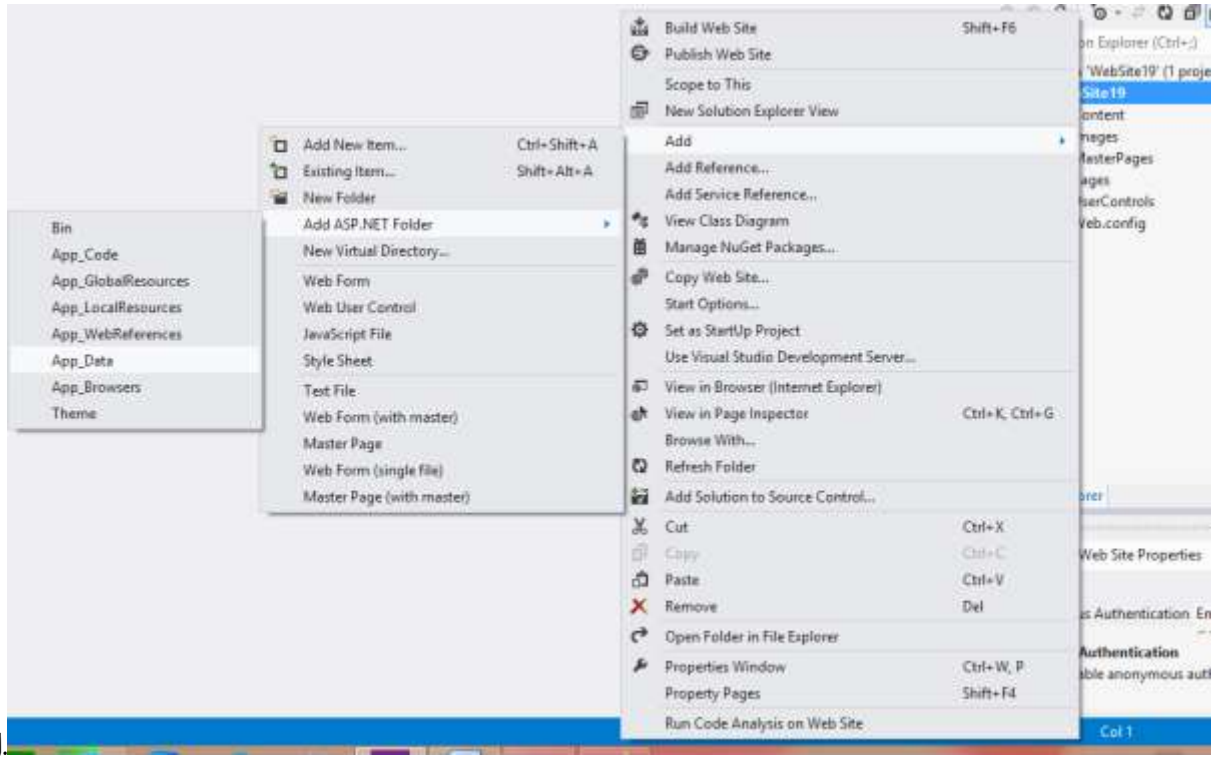
PHPBBمخفف PHP Bulletin Board یک سیستم انجمن اینترنتی کد باز است که با زبان سطح بالای PHPنوشته شده و قابلیت تنظیم قالب ، تنظیم زبان ، پشتیبانی از بانک های اطلاعاتی MySQL , PostgreSQL Microsoft SQL Server , Microsoft Accessو غیره از ویژگی های این انجمن می باشد.

SMFیک انجمن حرفه ایست که به شما در برقراری اجتماع آنلاینتان یاری می دهد . این انجمن از بسیاری جهات مشابه PHPBB بوده و دارای کامپوننت های جانبی مناسب بسیاریست که با نصب آنها می توانید تمامی نیازهای مدیریتی و کلوبری خود را برآورده سازید.

VBulletinاین انجمن بسایر قدرتمند و مبتنی بر زبان PHP، در آخرین نسخه خود از سیستم آجاکس جهت پست دهی کاربران استفاده می کند و از امنیت بالایی برخوردار است . نکته قابل توجه در مورد VBulletin رایگان نبودن آن است!

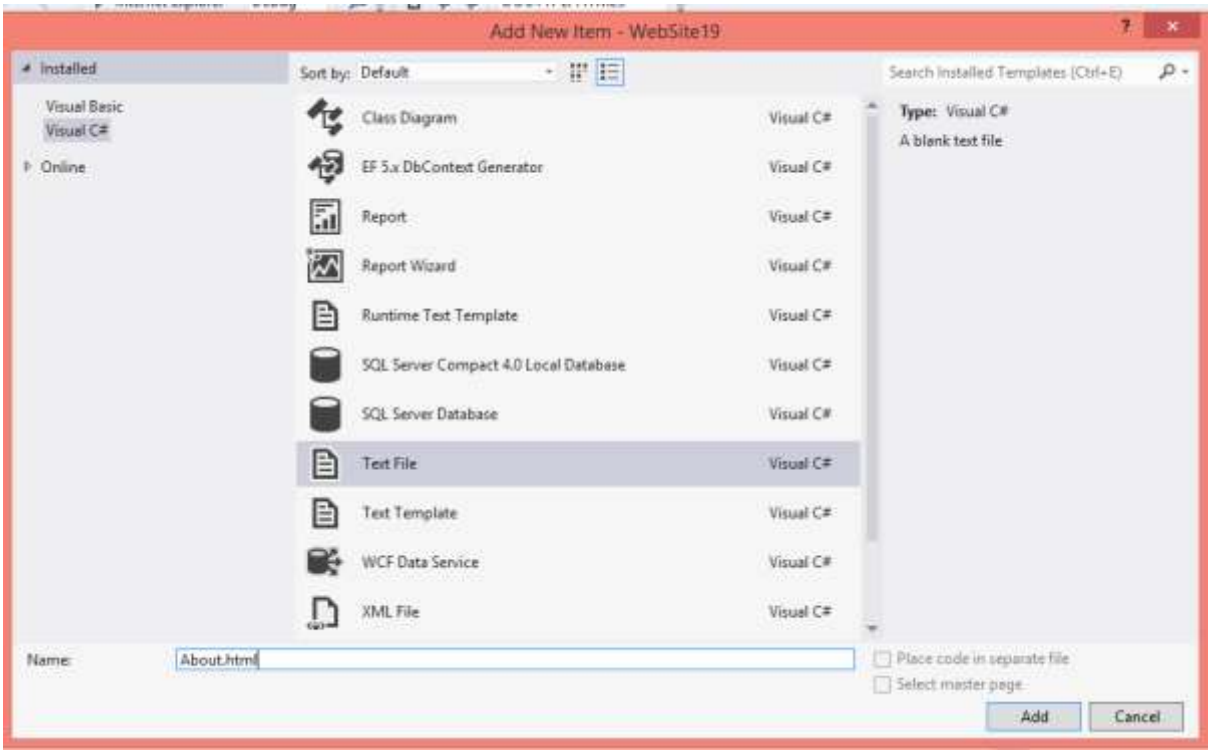
برای ایجاد یک cms مراحل زیر انجام می دهیم

۱ -ابتدا پروژه قبلی را ایجاد و یک فولدر جدید cms ایجاد می کنیم.



این فولدر امن asp.net است و امکان دسترسی آن برای دیگران ممکن نیست. Acces denide پیام مبنی بر عدم دسترسی به فایل اعلام می نماید.

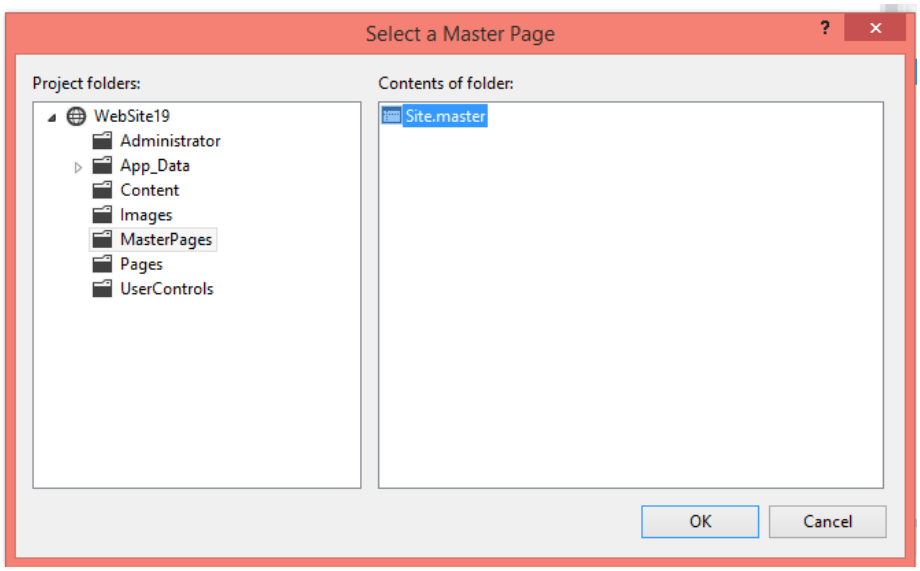
۲ -داخل آن فولدرpageContent ایجاد می کنیم و یک فایل html در آن ایجاد می کنیم



این صفحه می تواند مشخصات کاملی از admin یا شماره تماس یا اطلاعات دیگری که برای یک adminتعریف می گردد.مثلا داخل آن عبارتهای زیر اضافه می کنیم :

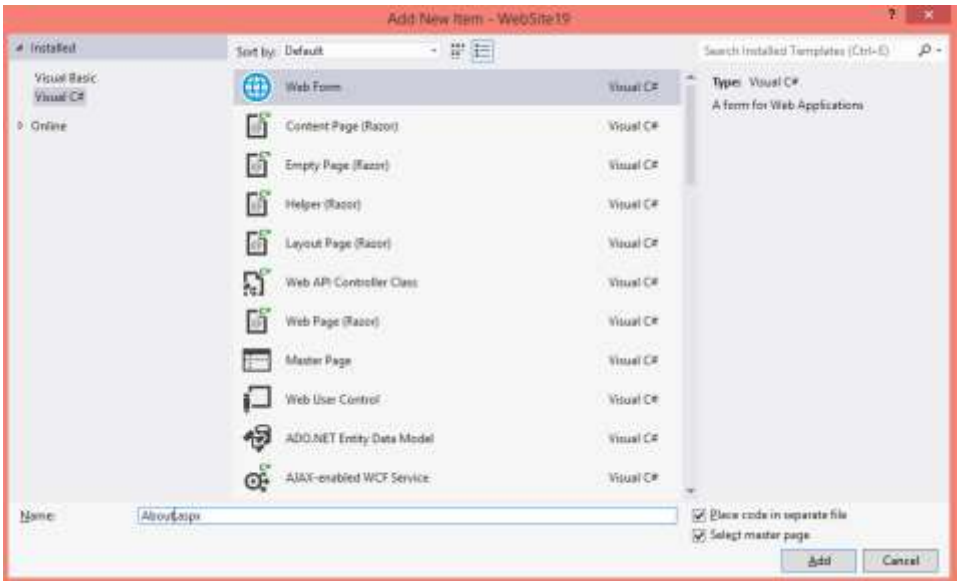
```
<p><strong>وبسایت صاحب تماس اطلاعات </strong><br /> <a
href="http://computer۹۴۰۱.blog.ir/"
target="_blank">http://computer۹۴۰۱.blog.ir/ </a></p>
<p>از ها آموزش شروع<br /> <strong>وب صفحات طراحی و نویسی برنامه آموزش:موضوع
۹۴</strong></p>
```

۳ -یک فولدر Administrator و در داخل آن یک وب فرم ایجاد می کنیم که از مستر پیج تغذیه بشود و نام آنرا EditPagecontent می نامیم



قرار است admin از طریق این صفحه دسترسی به صفحه About داشته باشد.

۴ -حال صفحه ای که کاربر قرار است کار بکند ایجاد می کنیم .بروی نام پروژه راست کلیک و گزینه add و add new item ویک وب فرم به نام About با انتخاب master page ایجاد می کنیم



بنابراین admin قرار است از طریق صفحه خودش EdiPageControl با استفاده از تغییراتی که در About.html که ایجاد می کند ، کاربر در صفحه About.aspx بتواند ببیند.

۵ -در صفحه About.aspx کد زیر اضافه می کنیم

```
<asp:Content ID="Content۲" ContentPlaceHolderID="cphMain"
Runat="Server">
    <div>
        <asp:Literal id="litpagecontent" runat ="server" />
    </div>
</asp:Content>
```

۶ -در هدر دو لینک اضافه می کنیم

وارد مستر پیج می شویم و در قسمت ساب هدر کد های زیر اضافه می کنیم

```
<div class="subHeader">
                <asp:HyperLink ID="lnkAbout" runat="server"
Text="About" NavigateUrl="~/About.aspx" CssClass="link" />
                 
                <asp:HyperLink ID="lnkEditPage" runat="server"
Text="Edit Page"
NavigateUrl="~/Administrator/EditPageContent.aspx"
CssClass="link" />
            </div>
```

۷-وارد صفحه کدهای سی شارپ about می شویم و کد های زیر اضافه می کنیم

```
protected void Page_Load(object sender, EventArgs e)
```

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

    }

    private void Initialize()
    {
        string strFileName =
            "About.html";
        string strRootRelativePath =
            "~/App_Data/PageContent";
        string strRootRelativePathName =
            string.Format("{0}/{1}",        strRootRelativePath,
strFileName);

        string strPathName =
            Server.MapPath(strRootRelativePathName);

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

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

            }
            catch (Exception ex)
            {

                litPageContent.Text = ex.Message;
            }
```

```
            finally
            {
                if (oStreamReader != null)
                {
                    oStreamReader.Dispose();
                    oStreamReader = null;
                }
            }
        }
```

متد : tryیعنی تلاش

یعنی تلاش کن تا قطعه کدی که داخل try هست رو اجرا کنی

و اگر با خطامواجه شدی catch را اجرا کن

```
try
{
}
catch
{
}
```

حتما شما هم متوجه شدید که وقتی رخداد یک استثناء را با استفاده از try و catch کنترل می‌کنیم، هر چیزی که بعد از بسته شدن تگ catch بنویسیم، در هر صورت اجرا می شود .

```
    try {
          int i=۰;
          string s = "hello";
          i = Convert.ToInt۳۲(s);
    } catch (Exception ex)
    {
          Console.WriteLine("Error");
```

```
}
Console.WriteLine("I am here!");
```

پس فلسفه استفاده از بخش finally چیست؟

در قسمت finally منابع تخصیص داده شده در try را آزاد می‌کنیم. کد موجود در این قسمت به هر روی اجرا می‌شود چه استثناء رخ دهد چه ندهد. البته اگر استثناء رخ داده شده در لیست استثناء هایی که برای آنها catch انجام دادیم نباشد، قسمت finally هم عمل نخواهد کرد مگر اینکه از catch به صورت سراسری استفاده کنیم. اما مهمترین مزیتی که finally ایجاد می‌کند در این است که حتی اگر در قسمت try با استفاده از دستوراتی مثل returnیا break یا continue از ادامه کد منصرف شویم و مثلا مقداری برگردانیم، چه خطا رخ دهد یا ندهد کد موجود در finally اجرا می‌شود در حالی که کد نوشته شده بعد از try catch finally فقط در صورتی اجرا می‌شود که به طور منطقی اجرای برنامه به آن نقطه برسد اجازه بدهید با یک مثال توضیح دهم. اگر کد زیر را اجرا کنیم:

```
public static int GetMyInt()
{
    try {
         for (int i=۱۰;i>=۰;i--)
            Console.WriteLine(۱۰/i);
        return ۱;
    } catch
    {
        Console.WriteLine("Error!");
    }
    finally {
        Console.WriteLine("ok");
    }
    Console.WriteLine("can you reach here?");
    return -۱;
}
```

```
public static int GetMyInt()
 {
```

```
public int GetUserId(string nickname)
{
     SqlConnection connection = new SqlConnection(...);
     SqlCommand command = connection.CreateCommand();
     command.CommandText = "select id from users where
nickname like @nickname";
     command.Parameters.Add(new SqlParameter("@nickname",
nickname));
     try {
        connection.Open();
        return Convert.ToInt۳۲(command.ExecuteScalar());
     }
     catch(SqlException exception)
     {
     // some exception handling
        return -۱;
     } finally {
        if (connection.State == ConnectionState.Open)
        connection.Close();
     }
     // if all things works, you can not reach here
}
```

برنامه خطای تقسیم بر صفر می دهد اما با توجه به کدی که نوشتیم، عدد - ۱ به خروجی خواهد رفت. در عین حال عبارت ok و can you reach here در خروجی چاپ شده است . اما حال اگر مشکل تقسیم بر صفر را حل کنیم، آ یا باز هم عبارت can you reach here در خروجی چاپ خواهد شد؟

مشاهده می‌کنید که مقدار ۱ برگردانده می‌شود و عبارت can you reach here در خروجی چاپ نمی‌شود ولی همچنان عبارت ok که در finally ذکر شده در خروجی چاپ می‌شود. یک مثال خوب استفاده از چنین وضعیتی، زمانی است که شما یک ارتباط با بانک اطلاعاتی باز می کنید، و نتیجه یک عملیات را با دستور return به کاربر بر می‌گردانید. مسئله این است که در این وضعیت چگونه ارتباط با دیتابیس بسته شده و منابع آزاد می گردند؟ اگر در حین عملیات بانک اطلاعاتی، خطایی رخ دهد یا ندهد، و شما دستور آزاد سازی منابع و بستن ارتباط را در داخل قسمت finally نوشته باشید، وقتی دستور return فراخوانی می‌شود، ابتدا منابع آزاد و سپس مقدار به خروجی بر می‌گردد.

جلسه یازدهم: ادامه سیسنم مدیریت محتوا cms :

همانطور که قبلا گفته شد در فایل About.aspx.cs موارد زیر اضافه شد :

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

}

private void Initialize()
{
   string strFileName =
      "About.html";
   string strRootRelativePath =
      "~/App_Data/PageContent";
   string strRootRelativePathName =
      string.Format("{۰}/{۱}",strRootRelativePath,strFileName);


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

      try
      {
         oStreamReader = new System.IO.StreamReader
            (strPathName, System.Text.Encoding.UTF۸);
         litPageContent.Text = oStreamReader.ReadToEnd();

      }
      catch (Exception ex)
      {

         litPageContent.Text = ex.Message;
```

در این تابع موارد زیر اجرا می گردد:
1 - مسیر فایل را در رشته قرار می دهیم
2 - با استفاده از دستور Server.MapPath . آن را به آدرس فیزیکی تبدیل می کنیم

در این تابع موارد زیر اجرا می گردد:
1 - اگر فایل مورد نظر ایجاد شده است
2 - یک ابجکت به نام oStreamReader بساز و مقدار null در آن قرار ده

در این تابع موارد زیر اجرا می گردد:
1-oStreamReader را قالب دهی کن
2-از تابع oStreamReader.ReadToEnd() استفاده کن و محتوای فایل اختصاصی در تکس مربوطه قرار بده

```
        }

        finally
        {
            if (oStreamReader!=null)
            {
                oStreamReader.Dispose();
                oStreamReader = null;
            }
        }
```

حال در داخل فایل EditPageContent.aspx مقادیر زیر اضافه می کنیم :

```
<div id="divPageMessages" runat="server" class="pageMessages"
visible="false" >
        <asp:Literal ID="litPageMessages" runat="server" />
    </div>
    <fieldset>
        <legend>Edit Pages</legend>
        <div class="field" >
            <div class="lable">
                <asp:Label ID="lblFileName" runat="server" Text="File Name
:" />
            </div>
            <div class="controls">
                <asp:TextBox ID="txtFileName" runat="server"
MaxLength="۴۰" />
            </div>
        </div>

        <div class="field" >
            <div class="lable">
                <asp:Label ID="lblEdit" runat="server" Text="Edit :" />
            </div>
            <div class="controls">
                <asp:TextBox ID="txtEditPage" runat="server"
MaxLength="۱۰۰۰" TextMode="MultiLine" Rows="۱۰" Columns="۷۰" />
            </div>
        </div>
        <div class="field">
            <div class="button">
                <asp:Button ID="btnOpen" runat="server" Text="Open"
AccessKey="O" OnClick="btnOpen_Click" />
                 
```

```
                <asp:Button ID="btnSave" runat="server" Text="Save"
AccessKey="S" OnClick="btnSave_Click" />
            </div>
        </div>
    </fieldset>
```

در بروی دکمه open دابل کلیک نموده و کد زیر در آن اضافه می کنیم :

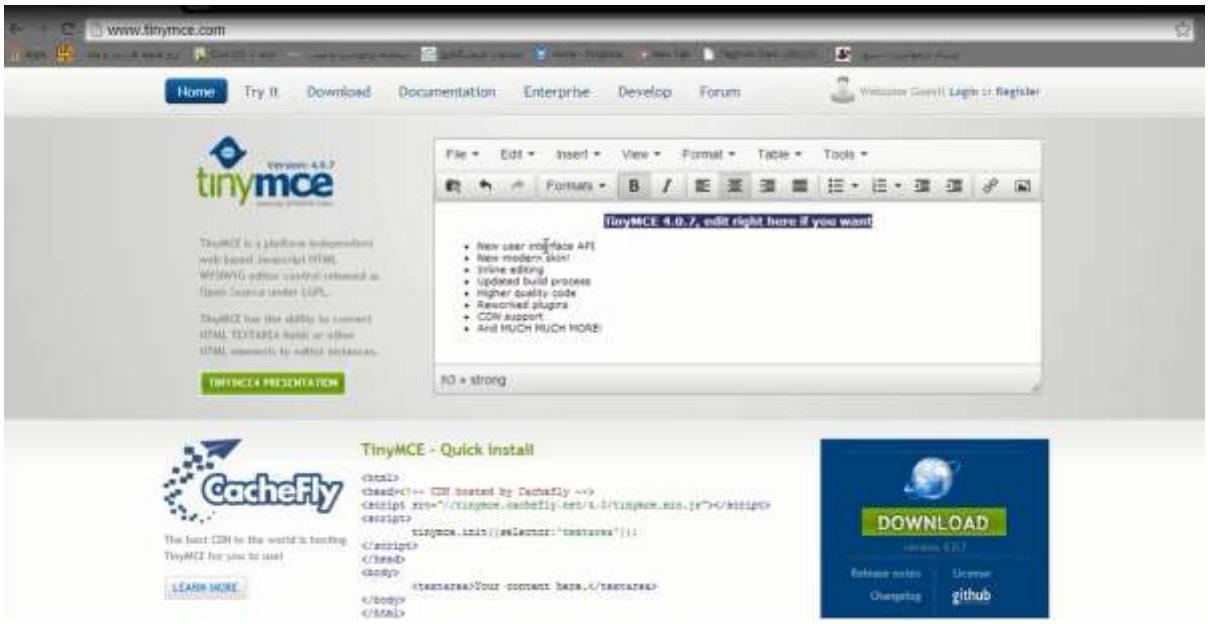
```
protected void btnOpen_Click(object sender, EventArgs e)
    {
        txtFileName.Text =
            txtFileName.Text.Trim();

        if (txtFileName.Text==string.Empty)
        {
            string strErrorMessage =
                "You Did Not Specify File Name For Opening !";
            DisplayErrorMessage(strErrorMessage);
            return;
        }

        string strFileName = txtFileName.Text;
        string strRootRelativePath = "~/App_Data/PageContent";
        string strRootRelativePathName =
            string.Format("{۰}/{۱}",strRootRelativePath,strFileName);

        string strPathName = Server.MapPath(strRootRelativePathName);

        if (System.IO.File.Exists(strPathName)==false)
        {
            string strErrorMessage =
                string.Format("The File [{۰}] Dose Not Exists For Opening
!",txtFileName.Text);
            DisplayErrorMessage(strErrorMessage);
            return;
        }

        System.IO.StreamReader oStreamReader = null;

        try
        {
            oStreamReader = new System.IO.StreamReader(strPathName,
System.Text.Encoding.UTF۸);
            txtEditPage.Text = oStreamReader.ReadToEnd();

            string strInformationMessage =
```

```
                string.Format("The File [{·}] Opened Successfully . .
.",txtFileName.Text);
            DisplayInformationMessage(strInformationMessage);
        }
        catch (Exception ex)
        {

            DisplayErrorMessage(ex.Message);
        }

        finally
        {
            if (oStreamReader!=null)
            {
                oStreamReader.Dispose();
                oStreamReader=null;
            }
        }
```

سپس برروی دکمه save دابل کلیک می کنیم :

```
    }
protected void btnSave_Click(object sender, EventArgs e)
    {
        txtFileName.Text =
            txtFileName.Text.Trim();

        if (txtFileName.Text==string.Empty)
        {
            string strErrorMessage =
                "You Did Not Specify File For Opening!";
            DisplayErrorMessage(strErrorMessage);

        }

        string strFileName = txtFileName.Text;
        string strRootRelativePath = "~/App_Data/PageContent";
        string strRootRelativePathName =
            string.Format("{·}/{١}",strRootRelativePath,strFileName);

        string strPathName=Server.MapPath(strRootRelativePathName);

        System.IO.StreamWriter oStreamWriter = null;

        try
        {
            oStreamWriter = new System.IO.StreamWriter
```

```
                (strPathName, false, System.Text.Encoding.UTF٨);
            oStreamWriter.Write(txtEditPage.Text);

            string strInformationMessage =
                string.Format("The File [{٠}] Saved Saccessfully . .
.",txtFileName.Text);
            DisplayInformationMessage(strInformationMessage);
        }
        catch (Exception ex)
        {

            DisplayErrorMessage(ex.Message);
        }

        finally
        {
            if (oStreamWriter!=null)
            {
                oStreamWriter.Dispose();
                oStreamWriter = null;
            }
        }
    }
}
```



جلسه دوازدهم : نحوه کار با ولیدیتورها:

کنترل های Validation در ASP.NET

بررسی و ارزیابی صحت اطلاعات ورودی توسط کاربران ، یکی از عملیات بسیار مهم در برنامه های کامپیوتری

خصوصا" برنامه های وب با توجه به ماهیت معماری آنها می باشد . پیاده کنندگان برنامه های وب می بایست پس از درج داده ورودی توسط کاربر و قبل از ارسال آن به لایه های دیگر ، آنها را ارزیابی و صرفا" پس از تائید ، پردازش های مورد نیاز را بر روی آنها انجام دهند . اعتقاد عملی به سیاست فوق باعث پیشگیری تعداد زیادی از حملات در برنامه های وب می گردد . در صورت عدم رعایت موارد اشاره شده ، شرایط لازم به منظور سوء استفاده از پتانسیل فوق فراهم و مهاجمان می توانند با بهره برداری از نقاط ضعف موجود ، حملات خود را برنامه ریزی نمایند .
در این مقاله به بررسی امکانات ارائه شده در ASP.NET به منظور ارزیابی صحت داده ورودی خواهیم پرداخت .
آشنائی و بکارگیری روش های مناسب به منظور بررسی صحت داده ورودی یکی از عملیات مهم در جهت ایمن سازی برنامه های وب نیز می باشد . پس با انگیزه و دقت مضاعف کار را دنبال می نمائیم .

ضرورت و ماهیت کنترل های Validation

زمانی که از یک TextBox برای درج داده بر روی یک فرم استفاده می گردد ، همواره این احتمال وجود خواهد داشت که کاربران اطلاعات را متناسب با انتظار یک برنامه وارد نمایند . به عنوان مثال ، فرض کنید که در یک فرم وب از دو TextBox برای درج داده عددی توسط کاربران استفاده شده است و در ادامه می بایست اطلاعات ورودی در اختیار یک روتین جهت انجام پردازش های مورد نیاز (به عنوان نمونه ، انجام عملیات محاسباتی بر روی داده های ورودی) قرار داده شود . در صورتی که کاربران داده را در محدوده مجاز وارد ننمایند و یا ورودی از نوع عددی نباشد ، روتین مورد نظر در انجام پردازش های خود با مشکل مواجه خواهد شد . بنابراین ، می بایست همواره قبل از این که داده ورودی در اختیار روتین های مورد نظر جهت انجام پردازش های مشخص شده قرار داده شود ، آنها را بررسی و در صورت رعایت مجموعه سیاست های تعریف شده ، امکان استفاده از آنها را برای اسکریپت ها فراهم نمود .
به منظور ارزیابی داده ورودی توسط کاربران با توجه به داده مورد نیاز اسکریپت ها ، از روش های متعددی می توان استفاده نمود . متداولترین نیاز ، وجود یک مقدار ورودی است . در چنین مواردی ، انتظار داریم قبل از این که یک اسکریپت بتواند پردازش های مورد نیاز را بر روی داده ورودی انجام دهد، در TextBox مقداری درج شده باشد.
ماهیت داده ورودی در این مقطع مهم نمی باشد و مهم این است که داده ئی توسط کاربر در مکان مورد نظر وارد شده باشد . در برخی موارد لازم است که داده ورودی از یک نوع خاص باشد (به عنوان نمونه یک مقدار عددی) تا بتوان آن را در اختیار روتین های مورد نظر به منظور انجام پردازش های محاسباتی قرار داد . در برخی موارد دیگر ، لازم است که داده ورودی در یک محدوده خاص باشد و داده درج شده خارج از محدوده مورد نظر ، غیرمعتبر گردد.
برای بررسی و ارزیابی صحت داده در ASP.NET مجموعه ای از کنترل های validation ارائه شده است . از کنترل های فوق می توان به منظور بررسی و تست عدم درج داده ، مقایسه داده ورودی با یک مقدار خاص ، بررسی وجود مقادیر در یک محدوده خاص و سایر فرم های داده استفاده نمود تا این اطمینان ایجاد گردد که داده مناسب و معتبر در اختیار اسکریپت ها و به منظور انجام پردازش ها قرار داده می شود .

کنترل های validation در ارتباط مستقیم با کنترل TextBox بوده و تست و ارزیابی خود را بطور اتوماتیک و پس از کلیک بر روی کنترل هائی نظیر Button ، LinkButton و ImageButton انجام خواهند داد . در صورتی که ماحصل بررسی یک کنترل validation توام با موفقیت نباشد ، validator یک پیام خطاء را نمایش و به کاربران اعلام می نماید که داده ورودی فاقد شرایط لازم برای استفاده در برنامه می باشد.
در شکل زیر کنترل های Validation ارائه شده در ASP.NET ۲.۰ ، نشان داده شده است .

کنترل های Validation

در ادامه به بررسی کنترل ه‌ای فوق خواهیم پرداخت .

کنترل <asp:RequiredFieldValidator>

متداولترین نوع ارزیابی داده ورودی ، بررسی آن به منظور اطمینان از درج داده توسط کاربر است . کنترل <asp:RequiredFieldValidator> ، مسئولیت بررسی داده ورودی در یک Textbox را برعهده گرفته و در صورت عدم درج داده توسط کاربر ، یک پیام خطاء را نمایش خواهد داد . در چنین مواردی ، روتین هائی که می بایست پس از کلیک بر روی دکمه موجود از داده درج شده در TextBox استفاده نمایند ، فرصت انجام چنین کاری را به دلیل عدم درج داده توسط کاربر پیدا نخواهند کرد.

شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است :

```
<asp:RequiredFieldValidator  id="id"
Runat="Server"
```

```
  ControlToValidate="controlID"
  Display="Dynamic|None|Static"
  ErrorMessage="string"
  SetFocusOnError="False|True"
  ValidationGroup="name"
/>
```

توضیحات:

- از یک id صرفا" زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپپ مراجعه گردد.
- به خصلت ControlToValidate ، مقدار id کنترل textBox که مسئولیت بررسی آن به کنترل RequiredFieldValidator واگذار شده است ،نسبت داده می شود.
- خصلت Text، مشخصه Text به عنوان متنی استفاده می شود که کنترل تایید اعتبار روی صفحه نمایش می دهد .این می تواند یک آستریکس (*) برای نشان دادن یک خطا باشد .یا متنی مانند Enter Your Name .
- CssClass، این مشخصه به ما اجازه می دهد تا خاصیت class در css را تعیین کنیم که روی متن پیغام خطا اعمال می شود.
- خصلت ErrorMessage ،این مشخصه پیغام خطای استفاده شده در کنترل validationSummary را نگه می دارد.وقتی مشخصه Text خالی باشد، مقدار ErrorMessage نیز به عنوان متنی استفاده می شود که روی صفحه ظاهر می شود.
- خصلت SetFocusOnErrorMessage ، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد.

  این مشخصه تعیین می کند که آیا اسکریپت روی کلاینت ،روی اولین کنترلی که تولید خطا کرده است فوکوس را قرار دهد.

- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند مثلاً برای کنترلهای صفحه login از یک گروه و برای search از گروه دیگر) .
- خصلت Display : کنترل RequiredFieldValidator ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار TextBox مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد (ولو این که خطائی اتفاق نیافتد) . در

صورتی که مقدار خصلت Display معادل Dynamic در نظر گرفته شود ، مکانی برای نمایش پیام خطاء رزو نخواهد شد و بطور پویا و همزمان با بروز خطاء ایجاد می گردد .با مقدار None ،کنترل اصلاً قابل دیدن نخواهد بود .این برای زمانی مفید است که از یک ValidationSummaryاستفاده کنید.

- خصلت IsValid ، معمولا این مشخصه را در زمان طراحی اعمال نمی کنیم بلکه این کار را در زمان اجرا انجام می دهیم زیرا اطلاعاتی در باره این که آیا آزمایش تایید اعتبار با موفقیت انجام شده است فراهم می آورد . کلاس page نیز دارای یک مشخصه IsValidاست که نتیجه ترکیب شده همه کنترلهای درون صفحه یا گروه تایید اعتبار را برمی گرداند.

تفاوت مشخصات Textو ErrorMessage

در validationSummary از عبارت Errormessage استفاده می شود.

مثال

در این مثال زمانی که کاربر بر روی button کلیک نمود ، کنترل RequiredFieldValidator بطور اتوماتیک عملیات بررسی و ارزیابی داده ورودی در TextBox را آغاز می نماید (قبل از این که روتین Get_Data فراخوانده شود) . در صورتی که TextBox خالی باشد ، یک پیام خطاء نمایش داده شده و cursor مجددا" در TextBox قرار خواهد گرفت . در صورتی که در TextBox داده ئی توسط کاربر درج گردد ، تست ارزیابی با موفقیت انجام و روتین مربوط فراخوانده شده و مقدار ورودی را در خروجی نمایش خواهد داد . در این مثال ، مقدار خصلت Display کنترل RequiredFieldValidator ، معادل Dynamic در نظر گرفته شده است ، بنابراین در صورت عدم تولید یک پیام خطاء ، مکانی بر روی صفح ه برای نمایش آن در نظر گرفته نخواهد شد . بدین ترتیب به کنترل label اجازه داده می شود که در کنار TextBox نمایش داده شده و از فضای مشابه پیام خطاء استفاده نماید (استفاده مشترک از یک مکان بر روی فرم وب به منظور نمایش پیام خطاء و یا داده ورودی توسط کاربر) .

```
<SCRIPT Runat="Server">

   Sub Get_Data (Src As Object, Args As EventArgs)
     Output.Text = "You entered '" & MyTextBox.Text & "'"
   End Sub

</SCRIPT>
<form Runat="Server">
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:RequiredFieldValidator Runat="Server"
  ControlToValidate="MyTextBox"
  ErrorMessage="Please enter a data value"
  Display="Dynamic"
  SetFocusOnError="True"/>
```

```
<asp:Label id="Output" Runat="Server"/>

</form>


<asp:RequiredFieldValidator ID="rfvUserName" runat="server"
ControlToValidate="txtUserName" CssClass="validator"
Display="Dynamic" EnableClientScript="true"

SetFocusOnError="true" Text="UserName Is Required . . ."
 />
```

کنترل <asp:RangeValidator>

کنترل فوق مسئولیت بررسی و ارزیابی داده ورودی در یک TextBox را برعهده گرفته و ماموریت آن حصول اطمینان از این موضوع است که داده درج شده در محدوده مورد نظر است. این محدوده توسط دو خصلت MinimumValue و MaximumValue مشخص می گردد. type در فرآیند فوق می تواند در ارتباط با نوع های متفاوت داده نظیر Date ، اعداد صحیح ، اعداد اعشاری و یا رشته (مقدار پیش فرض) اعمال گردد .

شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است :

```
<asp:RangeValidator id="id" Runat="Server"
  ControlToValidate="controlID"
  Display="Dynamic|None|Static"
  ErrorMessage="string"
  MaximumValue="value"
  MinimumValue="value"
  SetFocusOnError="False|True"
  Type="Currency|Date|Double|Integer|String"
  ValidationGroup="name"
/>
```

توضیحات

- از یک id صرفا" زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپپ مراجعه گردد .
- خصلت ErrorMessage ، پیام مورد نظر در صورت بروز خطاء را مشخص می نماید .
- خصلت SetFocusOnErrorMessage ، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .

- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند) .
- خصلت Display : کنترل RangeValidator ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار TextBox مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد (ولو این که خطائی اتفاق نیافتد) . در صورتی که مقدار خصلت Display معادل Dynamic در نظر گرفته شود ، مکانی برای نمایش پیام خطاء رزو نخواهد شد و بطور پویا و همزمان با بروز خطاء ایجاد می گردد .
- خصلت Type : در صورتی که نوع داده ورودی مشخص نگردد ، نوع آن به صورت پیش فرض string در نظر گرفته خواهد شد . یک TextBox خالی ، به عنوان یک نوع داده معتبر ارزیابی خواهد شد . بنابراین ، لازم است که به همراه کنترل RangeValidator از یک کنترل RequiredFieldValidator نیز استفاده گردد تا این اطمینان حاصل شود که با عدم درج داده در TextBox مربوطه با آن به عنوان یک داده معتبر برخورد نخواهد شد .
- خصلت های MinimumValue و MaximumValue ، حداقل و حداکثر محدوده مجاز برای داده ورودی را مشخص می نمایند .

مثال

در این مثال به منظور ارزیابی و تست داده ورودی در یک TextBox از دو کنترل RangeValidator و RequiredFieldValidator استفاده شده است . بنابراین می بایست حتما" در TextBox مقداری درج گردد . با توجه به این که مقدار خصلت Type معادل integer و مقادیر خصلت های MinimumValue و MaximumValue به ترتیب صفر و نه در نظر گرفته شده اند ،داده ورودی می بایست عددی بین صفر تا نه باشد .

```
<SCRIPT Runat="Server">

  Sub Get_Data (Src As Object, Args As EventArgs)
    Output.Text = "You entered '" & MyTextBox.Text & "'"
  End Sub

</SCRIPT>

<form Runat="Server">
```

```
Enter a value between ۰ and ۹:<br/>
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:RangeValidator Runat="Server"
  ControlToValidate="MyTextBox"
  Type="Integer"
  MinimumValue="۰"
  MaximumValue="۹"
  ErrorMessage="Please enter an integer in the range ۰ to ۹"
  Display="Dynamic"
  SetFocusOnError="True"/>
<asp:RequiredFieldValidator Runat="Server"
   ControlToValidate="MyTextBox"
   ErrorMessage="Please enter a data value"
   Display="Dynamic"
   SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>
```

کنترل <asp:CompareValidator>

کنترل فوق مسؤلیت بررسی مقدار درج شده در یک TextBox را برعهده دارد . بدین منظور داده ورودی با یک مقدار خاص و یا مقدار یک کنترل موجود بر روی فرم مقایسه می گردد . نوع داده درج شده در TextBox می تواند از نوع Currency ، Date ، اعشاری ، صحیح و یا رشته (مقدار پیش فرض) باشد .

شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است :

```
<asp:CompareValidator id="id" Runat="Server"
ControlToCompare="controlID"
ControlToValidate="controlID"
Display="Dynamic|None|Static"
ErrorMessage="string"
Operator="Equal|NotEqual|GreaterThan|GreaterThanEqual|LessThan
          |LessThanEqual|DataTypeCheck"
SetFocusOnError="False|True"
```


s.ali.taher@gmail.com

```
Type="Currency|Date|Double|Integer|String"
ValidationGroup="name"
ValueToCompare="value"
/>
```

توضیحات

- از یک id صرفا" زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپپ مراجعه گردد .
- ControlToCompare، این مشخصه حاوی ID کنترلی است که مقایسه کننده با آن مقایسه می شود.وقتی این مشخصه تعیین شود ، ValueToCompare بی اثر می شود.
- Operator، این مشخصه تعیین کننده نوع عملیات مقایسه است .
- به خصلت ControlToValidate ، مقدار id کنترل textBox که مسئولیت بررسی آن به کنترل <asp:CompareValidator> واگذار شده است ،نسبت داده می شود.
- خصلت ErrorMessage ، پیام مورد نظر در صورت بروز خطاء را مشخص می نماید .
- خصلت SetFocusOnErrorMessage ، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .
- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند) .
- خصلت Display : کنترل <asp:CompareValidator> ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار TextBox مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد (ولو این که خطائی اتفاق نیافتد) . در صورتی که مقدار خصلت Display معادل Dynamic در نظر گرفته شود ، مکانی برای نمایش پیام خطاء رزو نخواهد شد و بطور پویا و همزمان با بروز خطاء ایجاد می گردد .
- مقدار ورودی می تواند با یک مقدار مشخص شده توسط خصلت ValueToCompare و یا مقدار یک کنترل دیگر موجود در صفحه (مشخص شده توسط خصلت ControlToCompare) ، مقایسه گردد . به صورت پیش فرض ، عملیات مقایسه برای "برابری " انجام خواهد شد . در صورت نیاز می توان از سایر عملگرهای مقایسه ای که توسط خصلت Operator مشخص می گردند ، استفاده نمود . عملیات مقایسه بر اساس نوع داده درج شده در Textbox که توسط خصلت Type مشخص می گردد ، انجام خواهد شد .
- خصلت Type : در صورتی که نوع داده ورودی مشخص نگردد ، نوع آن به صورت پیش فرض string در نظر گرفته خواهد شد . یک TextBox خالی ، به عنوان یک نوع داده معتبر ارزیابی خواهد شد . بنابراین ، لازم است که به همراه کنترل <asp:CompareValidator> از یک کنترل

RequiredFieldValidator نیز استفاده گردد تا این اطمینان حاصل شود که با عدم درج داده در TextBox مربوطه با آن به عنوان یک داده معتبر برخورد نخواهد شد.

مثال

در این مثال ، کاربر می بایست یک عدد مثبت را وارد نماید و در صورتی که مقدار ورودی منفی باشد یک پیام خطاء نمایش داده می شود . به همراه کنترل CompareValidator از یک کنترل RequiredFieldValidator نیز استفاده شده است تا عدم درج داده (خالی‌بودن) به عنوان یک داده معتبر ارزیابی ن گردد . مقدار ورودی می بایست از نوع اعشاری (Type=Double) و بزرگتر از (Operator=GreaterThan) صفر (ValueToCompare=۰) باشد تا به عنوان یک داده معتبر ارزیابی گردد .

```
<SCRIPT Runat="Server">

Sub Get_Data (Src As Object, Args As EventArgs)
   Output.Text = "You entered '" & MyTextBox.Text & "'"
End Sub

</SCRIPT>

<form Runat="Server">

 Enter a positive number:<br/>
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:CompareValidator Runat="Server"
  ControlToValidate="MyTextBox"
  ValueToCompare="۰"
  Type="Double"
  Operator="GreaterThan"
  ErrorMessage="Please enter a number greater than ۰"
  Display="Dynamic"
  SetFocusOnError="True"/>
<asp:RequiredFieldValidator Runat="Server"
  ControlToValidate="MyTextBox"
  ErrorMessage="Please enter a data value"
  Display="Dynamic"
```

```
  SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>
```

کنترل <asp:CustomValidator>

در زمان بررسی و ارزیابی داده ورودی ممکن است به مواردی برخورد نمائیم که با ترکیب یک RequiredFieldValidator ، RangeValidator و یا CompareValidator خواسته ما تامین نگردد. در چنین مواردی می توان از کنترل <asp:CustomValidator> به منظور انجام تست های اضافه استفاده نمود .

شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است :

```
<asp:CustomValidator          id="id"          Runat="Server"
ControlToValidate="controlID"
Display="Dynamic|None|Static"
ErrorMessage="string"
SetFocusOnError="False|True"
ValidationGroup="name"
OnServerValidate="subprogram"
/>
```

توضیحات

- خصلت های کنترل <asp:CustomValidator> مشابه سایر کنترل های validation می باشند با این تفاوت که از خصلت OnServerValidate به منظور فراخوانی یک برنامه فرعی برای بررسی و ارزیابی داده ورودی نیز استفاده می گردد .
- یک TextBox خالی به عنوان یک داده معتبر در نظر گرفته خواهد شد . بنابراین می با یست به همراه کنترل CustomValidator از یک کنترل RequiredFieldValidator نیز استفاده گردد تا عدم درج داده به عنوان یک داده معتبر ارزیابی نگردد .
- برنامه فرعی صدا زده شده دارای آرگومان های خاص ServerValidateArgs می باشد . در صورتی که بررسی و ارزیابی صحت داده ورودی توام با موفقیت نباشد ، مقدار خصلت IsValid معادل false خواهد

شد . از آرگومان Value ، به عنوان مرجعی به منظور مراجعه به مقدار TextBox (کنترل مشخص شده توسط خصلت ControlToValidate) استفاده می گردد .

- خصلت ErrorMessage ، پیام مورد نظر در صورت بروز خطاء را مشخص می نماید .
- خصلت SetFocusOnErrorMessage ، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .
- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند) .
- خصلت Display : کنترل <asp:CustomValidator> ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار TextBox مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد (ولو این که خطائی اتفاق نیافتد) . در صورتی که مقدار خصلت Display معادل Dynamic در نظر گرفته شود ، مکانی برای نمایش پیام خطاء رزو نخواهد شد و بطور پویا و همزمان با بروز خطاء ایجاد می گردد .

مثال

در این مثال به منظور ارزیابی و تست داده ورودی در یک TextBox از دو کنترل CustomValidator و RequiredFieldValidator استفاده شده است . داده ورودی می بایست یک عدد صحیح بین صفر تا نود و نه باشد .

پس از کلیک بر روی دکمه Submit ، روتین Get_Data فعال و قبل از هر چیز عملیات بررسی و ارزیابی صحت داده ورودی انجام خواهد شد . کنترل RequiredFieldValidator در ابتدا بررسی لازم در خصوص درج داده در TextBox را انجام و در ادامه روتین Validate_TextBox توسط خصلت OnServerValidate کنترل CustomValidator فراخوانده می شود . روتین فوق ، تست های لازم را بر روی داده ورودی انجام خواهد داد (برای مراجعه به داده ورودی از Args.Value استفاده شده است) . در صورتی که هر یک از تست های انجام شده توام با موفقیت نباشد ، خصلت Args.Isvalid مقدار false را خواهد گرفت (داده ورودی می بایست یک عدد مثبت بین صفر تا نود و نه باشد) .

پس از انجام فرآیند بررسی داده ورودی ، روتین Get_Data ، اجرا خواهد شد . روتین فوق در ابتدا و پس از کلیک بر روی دکمه button فراخوانده می گردد و اجرای آن تا زمانی که عملیات بررسی و ارزیابی داده ورودی به اتمام نرسیده باشد ، به تاخیر خواهد افتاد . پردازش های انجام شده در روتین Get_Data مشروط به انجام موفقیت آمیز تست ارزیابی و صحت داده است . بنابراین ، همه چیز وابسته به شرط Page.IsValid شده است و در صورتی که Args.IsValid معادل false شده باشد ، مقدار Page.IsValid نیز false خواهد شد.

```
<SCRIPT Runat="Server">

 Sub Validate_TextBox (Src As Object, Args As ServerValidateEventArgs)

  If Not IsNumeric(Args.Value) Then
      MyValidator.ErrorMessage = "Please enter a number"
      Args.IsValid = False
    Else
    If Not Args.Value Mod ١ = ٠ Then
       MyValidator.ErrorMessage = "Please enter an integer"
       Args.IsValid = False
   End If
   If Args.Value < ٠ Then
      MyValidator.ErrorMessage = "Please enter a positive integer"
      Args.IsValid = False
   End If
   If Args.Value > ٩٩ Then
     MyValidator.ErrorMessage = "Please enter a positive integer between ٠ and
٩٩"
     Args.IsValid = False
   End If
 End If

End Sub
Sub Get_Data (Src As Object, Args As EventArgs)

If Page.IsValid Then
   Output.Text = "You entered '" & MyTextBox.Text & "'"
End If

End Sub
</SCRIPT>

<form Runat="Server">

Enter a positive integer:<br/>
```

```
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:CustomValidator id="MyValidator" Runat="Server"
  ControlToValidate="MyTextBox"
  Display="Dynamic"
  SetFocusOnError="True"
  OnServerValidate="Validate_TextBox"/>
<asp:RequiredFieldValidator Runat="Server"
  ControlToValidate="MyTextBox"
  ErrorMessage="Please enter a data value"
  Display="Dynamic"
  SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>
```

مثال :

```
<div class="controls">
                <asp:TextBox ID="txtPassword" runat="server"
MaxLength="۴۰"
                    CssClass="password" />
                <asp:RequiredFieldValidator ID="rfvPassword"
runat="server"
                    ControlToValidate="txtPassword" CssClass="validator"
                    Display="Dynamic" EnableClientScript="true"
                    SetFocusOnError="true" Text="Password Is Required . .
."
                     />
                 <asp:RegularExpressionValidator ID="revPassword"
runat="server"
                    ControlToValidate="txtPassword" CssClass="validator"
                    Display="Dynamic" EnableClientScript="true"
SetFocusOnError="true"
                    Text="Password Is Not Valid . . ."
                    ValidationExpression="[a-zA-Z۰-۹]{۶,۲۰}" />
            </div>
        </div>
```

جلسه سیزدهم :طراحی منو:

```
<asp:Menu ID="mnuMain" runat="server" RenderingMode="Table"
MaximumDynamicDisplayLevels="٣"
Orientation="Vertical" DisappearAfter="۴۰۰"
DynamicHorizontalOffset="٢"
BackColor="#FFFBD۶" ForeColor="□ ۹۹۰۰۰"
StaticSubMenuIndent="٢٠px"
StaticPopOutImageUrl="~/Images/RightToLeftArrow.gif"
DynamicPopOutImageUrl="~/Images/RightToLeftArrow.gif">

<Items>
<asp:MenuItem Text="Home . . ." NavigateUrl="~/Default.aspx"
ToolTip="Home" />

<asp:MenuItem Text="Learn . . ." NavigateUrl="~/Default.aspx"
ToolTip="Learn">
<asp:MenuItem Text="٨٧-٩١" NavigateUrl="~/Default.aspx"
ToolTip="٨٧-٩١" />
<asp:MenuItem Text="٩٢" NavigateUrl="~/Default.aspx"
ToolTip="٩٢" />
</asp:MenuItem>

<asp:MenuItem Text="Help . . ." NavigateUrl="~/Default.aspx"
ToolTip="Help"/>
<asp:MenuItem Text="ContactUs . . ."
NavigateUrl="~/Default.aspx" ToolTip="ContactUs" />

<asp:MenuItem Text="AboutUs . . ." NavigateUrl="~/Default.aspx"
ToolTip="AboutUs">
<asp:MenuItem Text="System" NavigateUrl="~/Default.aspx"
ToolTip="System" />
<asp:MenuItem Text="Team" NavigateUrl="~/Default.aspx"
ToolTip="Team">

<asp:MenuItem Text="Support" NavigateUrl="~/Default.aspx"
ToolTip="Support" />
```

```
<asp:MenuItem Text="Sales" NavigateUrl="~/Default.aspx"
ToolTip="Sales" />

</asp:MenuItem>
</asp:MenuItem>

<asp:MenuItem Text="News . . ." NavigateUrl="~/Default.aspx" />

</Items>
<StaticSelectedStyle BackColor="#FFCC۶۶" />
<StaticHoverStyle BackColor="□ ۹۹۰۰۰" ForeColor="White" />
<StaticMenuItemStyle HorizontalPadding="۵px"
VerticalPadding="۲px" CssClass="fixMenuItem" />

<DynamicSelectedStyle BackColor="#FFCC۶۶" />
<DynamicMenuStyle BackColor="#FFFBD۶" CssClass="fixMenu" />
<DynamicHoverStyle BackColor="□ ۹۹۰۰۰" ForeColor="White" />
<DynamicMenuItemStyle HorizontalPadding="۵px"
VerticalPadding="۲px" CssClass="fixMenuItem" />
</asp:Menu>
```

مقدمه:

طراحی منوهای ساده و کارآمد که دسترسی به تمامی صفحات و موضوعات سایت شما را فراهم نماید ، یک موضوع حیاتی است . کاربران بایستی در هر صفحه بتوانند به صفحات اصلی دیگر دسترسی داشته و بدانند در کجای سایت قرار دارند .

ASP.Net دارای کنترل های درون ساخته ای است که به وسیله آنها می توان ید ، انواع منو ها را ایجاد نمایید . همچنین ASP.Net دارای یک امکان به نام نقشه وب سایت می باشد ، که به وسیله آن ، کلیه مسیرهای مورد نظر خود را درون یک فایل با پسوند sitemap تعیین می کنید . فایل sitemap ، حکم یک فایل داده ای از نوع XML را داراست که می توان سایر کنترل های منوی ASP.Net را به آن متصل نمود . در این حالت کنترل های منو اطلاعات فایل و مسیرها را از فایل sitemap دریافت می کنند . این فایل ، نقشه مکان صفحات و فایل

های وب سایت شماست .

ASP.Net دارای ۳ نوع کنترل برای ایجاد منوها و سیستم های مسیریاب به شرح زیر می باشد :

- منوهای دینامیک (کنترل Menu).
- منوهای درختی (کنترل TreeView).
- منوهای مسیریاب (کنترل SiteMapPath).

در ادامه به توضیح و آموزش کار با این کنترل ها خواهیم پرداخت .

مرحله اول - ایجاد فایل نقشه وب سایت: (sitemap)

گفتیم که در ASP.Net ، می توانید مسیرها و فایل های اصلی سایت را در یک فایل تقشه سایت با پسوند sitemap. تعیین نمایید . این فایل به زبان XML نوشته شده و می تواند به عنوان منبع داده ای برای تمامی کنترل های منوی ASP.Net به کار رود .

کد زیر ، کد یک فایل ساده نقشه سایت را نشان می دهد . به آن دقت نموده ، سپس توضیحات لازم را ارائه خواهیم کرد :

کد

```
<?xml version="۱.۰" encoding="ISO-۸۸۵۹-۱" ?>
 <siteMap>
   <siteMapNode title="Home" url="~/Default.aspx">
     <siteMapNode title="Web Design" url="~/Web/Index.aspx">
         <siteMapNode title="HTML" url="~/HTML/Index.aspx"/>
        <siteMapNode title="CSS" url="~/CSS/Index.aspx"/>
     </siteMapNode>
   </siteMapNode>
 </siteMap>
```

قوانین ایجاد و تعیین مسیرها در یک فایل sitmap همانند کد فوق ، به شرح زیر است :

- تمامی مسیرها و تگ ها بایستی درون تگ اصلی <SiteMap>تعریف شوند.
- این فایل ها به زبان XML نوشته می شوند.
- تگ <SiteMap>، فقط می تواند یک عنصر <SiteMapNode>را به عنوان فرزند داشته باشد که معمولا به صفحه اصلی یا Home اشاره می کند.

- هر تگ <SiteMapNode>می تواند به تعداد دلخواه عنصر فرزند (Child Nodes) به صورت تگ درونی <SiteMapNode>داشته باشد.
- درون هر تگ <SiteMapNode>، عنوان لینک توسط خاصیت title و آدرس آن نیز توسط خاصیت urlتعیین می شود.

نکته : فایل sitemap با نقشه فایل ، بایستی حتما در پوشه اصلی سایت (root) قرار بگیرد . همچنین هر سایت یک فایل نقشه می تواند داشته باشد .

آموزش طراحی منوهای دینامیک به وسیله کنترل: Menu

از کنترل <asp:Menu> ، می توان برای ایجاد منوهای استاندارد در ASP.Net استفاده نمود . این منوها معمولا به صورت Pop-Up بوده و آیتم های اصلی آن ثابت بوده و سپس با قرار گرفتن موس بر روی آیتم های اصلی ، زیر منوها باز می شوند .

کد	<asp:SiteMapDataSource **id="nav۱"** runat="server" /> <form id="Form۱" runat="server"> <asp:Menu ID="Menu۱" runat="server" **DataSourceId="nav۱"** /> </form>
خروجی	Home •

شکل کلی ایجاد این منو با استفاده از کنترل Menu و یک فایل SiteMap به صورت زیر است :

توضیح کد :

در ابتدا یک کنترل <asp:SiteMapDataSource> را ایجاد کرده و id آن را تعیین می کنیم . این کنترل یک کنترل سرور داده است که می تواند به فایل تقشه سایت (SiteMap) متصل شده ، اطلاعات آن را خوانده و سپس به کنترل منوی مورد نظر انتقال دهد .
سپس کنترل Menu را تعریف کرده و با قرار دادن مقدار خاصیت DataSourceId آن برابر با nav۱ که id کنترل asp:SiteMapDataSource است ، آنها را به هم متصل کرده ایم . در این صورت این کنترل منو ، اطلاعات دریافتی از کنترل سرور داده خود را نمایش خواهد دارد .

نکته مهم : کنترل asp:SiteMapDataSource به صورت اتوماتیک به فایل نقشه سایت (web.sitemap) که در پوشه اصلی سایت (root) قرار دارد ، متصل می شود .

ایجاد منو با استفاده از کنترل: TreeView

کنترل <asp:TreeView> ، برای ایجاد یک منو با ساختار درختی استفاده می شود ، که آیتم های خود را با رعایت سلسله مراتب نمایش می دهد .

لینک هایی که در سطح پایین تر باشند ، به صورت پیش فرض نمایان بوده و با زدن آیکون های + یا - در کنترل ، می توان آنها را باز یا بسته نمود .

شکل کلی تعریف و ایجاد یک منو با کنترل TreeView به صورت زیر است :

کد	<asp:SiteMapDataSource **id="nav۱"** runat="server" /> <form id="Form۱" runat="server"> <asp:TreeView ID="Menu۲" runat="server" **DataSourceId="nav۱"** /> </form>
خروجی	⊟ Home ⊟ Web Design HTML CSS

توضیح کد مثال :

همانند مثال قبل ، کنترل asp:SiteMapDataSource را به عنوان فایل داده ای استفاده کرده تا اطلاعات فایل نقشه سایت (sitemap) را خوانده و به کنترل asp:TreeView انتقال دهد .

همچنین توسط خاصیت DataSourceId ، کنترل asp:TreeView را به کنترل asp:SiteMapDataSource متصل کرده ایم .

آموزش ایجاد منو با استفاده از کنترل: SiteMapPath

کنترل SiteMapPath منوی خود را به صورت یک سری آیتم پشت سر هم که مسیر خاصی را نمایش می دهند ، نشان می دهد . به وسیله این نوع منوها ، می توانید در هر لحظه بدانید در کجای سایت قرار داشته و همچنین با کلیک بر روی آیتم های قبل از آدرس صفحه جاری ، به مسیر پیموده شده قبلی خود باز گردید.

برخلاف کنترل های Menu و TreeView ، کنترل SiteMapPath از کنترل

asp:SiteMapDataSource به عنوان منبع داده ای استفاده نکرده و خود به صورت پیش فرض و اتوماتیک به فایل نقشه سایت متصل می شود .

کد تعریف این کنترل به صورت زیر است :

کد	<form id="Form۱" runat="server"> <asp:SiteMapPath ID="Menu۳" runat="server" /> </form>

جلسه چهاردهم :آموزش آپلود کردن وب سایت روی هاست در دات نت پنل(WebSite Panel)

همیشه یکی از مشکلاتی که افراد پس از خرید هاست دارند این است که می پرسند:

- چطور دامنه را به هاست متصل کنیم؟(تنظیمات DNS)
- چطور سایتمان را در سرور هاست آپلود کنیم؟
- چگونه سایتمان را پیکربندی کنیم؟(بحث در فایل web.config)
- چطور یک دیتا بیس در هاست ایجاد کنیم؟
- چگونه محتویات دیتا بیس را از سیستم خود به هاست منتقل کنیم؟
- چطور دیتا بیس را به سایت متصل کنیم؟
- چطور ftp را فعال کنیم و چطور از آن استفاده کنیم؟

نحوه تنظیم DNS روی هاست (تنظیم DOMAIN NAME SERVER)

برای تنظیم دامنه خود روی سرور هاست ابتدا به سایتی که دامنه را خریداری کرده اید بروید، وارد حساب کاربری خود شده و دامنه خود را انتخاب کنید .سپس گزینه تغییر DNS را انتخاب کنید. ۲ تا DNS برای دامنه خود setکنید.

اولیcns۵.my-hosting-panel.com

دومیcns۶.my-hosting-panel.com

اگر از قبل DNS دامنه شما روی سایت دیگری قرار داده شده است قبلی ها را پاک کنید و این ۲ را وارد کنید. حال باید تاحداکثر ۴۸ ساعت منتظر بمانید تا تغییرات اعمال شود. معمولا کمتر از ۱۲ ساعت انجام میشود.

نحوه آپلود کردن سایت روی سرور هاست

از طریق username و پسوردی که به وسیله این شرکت به ایمیلتان ارسال شده به کنترل پنل خود وارد شوید روی Domains کلیک کنید، سپس Add Domain ، سپس روی Domain کلیک کنید. نام دقیق دامنه یتان را در کادر وارد کنید، فقط پشت Create Web Site تیک بزنید و دکمه Add Domain را فشار دهید. وبسایت شما ایجاد شده است . در این مرحله اگر تغییرات DNS به خوبی اع مال شده باشد می توانید سایتخود را در اینترنت ببینید . منتها یک صفحه قراردادی را می بینید با عنوان UNDER CONSTRUCTIONکه خالی می باشد . در هر مرحله از کار اگر مشکلی وجود داشت می توانید با ایمیل support@iranhost۲۴.comبا پشتیبانی سایت در ارتباط باشید . حالا وقت آن است که فایل های خود راآپلود کنید. از کنترل پنل روی File Manager کلیک کنید. سپس روی نام وب سایتتان کلیک کنید و بعد به پوشه ی wwwroot کلیک کنید. اینجا همان جایی است که می توانید فایل هایتان را upload کنید. برای راحتی کار توصیه می کنیم فایل هایتان را به صو رت zip آپلود کنید و سپس unzip کنید .سایتتان را امتحان کنید. اگر درست کار نمی کند، نگران نباشید احتمالا مشکل از تنظیمات web.config می باشد.

تنظیمات فایل WEB.CONFIG روی هاست

این فایل از مهمترین فایل های سایت شما می باشد که اگر بخوبی تنظیم نشده باشد سایت شما با error مواجه می شود .اگر سایت شما به صورت local یا محلی در سیستم کامپیوترتان کار می کند ولی در سرور مشکل دارد، ابتدا فایل config را از داخل سایت edit کنید و این سطر را داخل <system.web>اضافه کنید تا errorسایت با توضیحات آن مشخص شود تا سر از کارتان در بیاورید:

```
<customErrors mode="Off"/>
```

حالا وقتی سایت را اجرا می کنید دقیقا ارور ها داخل مرورگر به نمایش در می آید. در این مرحله شاید عیب کارتان را پیدا کنید و آن را بر طرف کنید. همچنین باید اطمینان حاصل کنید که تگ های زیر را به web.config اضافه کرده اید:

```
<system.serviceModel>
<serviceHostingEnvironment aspNetCompatibilityEnabled="true" />
</system.serviceModel>
<system.webServer>
<directoryBrowse enabled="false"/>
<defaultDocument>
<files>
<clear />
<add value="default.aspx"/>
</files>
</defaultDocument>
</system.webServer>
```

اگر اضافه نکرده اید حتما همین الان اضافه کنید. دقت کنید در کد بالا باید به جای default.aspx نام صفحه ی اصلی سایتتان را اضافه کنید که می تواند اسامی مختلفی باشد مانند :

index.html, index.php, index.aspx, default.htm, default.asp & ...

این صفحه با ورود به سایتتان نمایش داده می شود . تا اینجای کار کلی به یک سایت ایده آل نزدیک شده اید. حالا وب سایتتان را تست کنید، اگر بازهم error گرفتید اصلا نگران نباشید. این طبیعیه، مگر یادتان رفته هنوز پایگاه داده یا دیتابیس را ایجاد و متصل نکرده اید؟

ایجاد دیتا بیس و مدیریت آن

ایجاد و مدیریت دیتا بیس در پنل WebSite Panel بسیار آسان می باشد. شما ابتدا از داخل پنل وارد قسمت دیتا بیس می شوید، یک دیتا بیس و یک کاربر دیتا بیس (Create User)ایجاد می کنید. وارد دیتا بیسی که ساخته اید می شوید، برای مدیریت آن روی Browse Database کلیک کنید. از اینجا میتوانید دیتا بیس را به ایجاد کنید یا از دیتا بیسی که قبلا ساخته اید Script بگیرید و به صورت Query وارد دیتا بیس جدید کنید.

طریقه ATTACH کردن یا SCRIPT گرفتن از پایگاه داده (دیتا بیس)

شاید بارها در داخل ویندوز یک فایل دیتا بیس Attach کرده اید و از آن در برنامه هایتان استفاده کرده اید . اما در هاست این روش روش مناسبی نیست و اکثر مواقع با مشکلاتی روبه رو می شود. راه بهتر و آسانتر این است که از دیتابیس Script بگیرید . آن را در دیتا بیس سرور اجرا کنید . مراحل ایجاد و اجرای Script به اینصورت می باشد:

ابتدا از داخل پنل هاست به دیتا بیس مورد نظر وارد می شوید و Browse Database را کلیک می کنید. حالا درون قسمت مدیریت دیتا بیس هستید. گزینه ی tools و سپس Query را انتخاب می کنید، Scriptمورد نظر را در اینجا کپی میکنید و Submit را کلیک می کنید. اگر نمی دانید Script را باید از کجا بیاورید که در اینجا کپی کنید به این طریق عمل کنید:

sqlServerرا از داخل ویندوز باز کنید، روی دیتا بیس مورد نظر راست کلیک کنید، Tasksو سپس Generate scriptsرا انتخاب کنید، بقیه مراحل روتین است طی کنید تا Script ساخته شود و در یک پنجره به نمایش در آید. کار هنوز تمام شده نیست، در خط اول script جلوی use نام دیتا بیسی که در سرور ساخته اید را جایگزین کنید]USE نام دیتا بیس[

سپس از طریق find and replace تمام نوشته های [dbo] را با نام کاربری دیتا بیس در سرور جایگزین کنید[Database_username] <--- [dbo] .

این script آماده ی کپی و اجرا شدن در سرور می باشد.

با فرض این که دیتا بیس و سایت شما به صورت کامل در سرور مستقر می باشند باید یک ارتباط بین این دو برقرار شود که این کار بسیار آسان و با افزودن چند خط به web.config میسر می شود:

```
<connectionStrings>
<add name="connectionStringName" connectionString="Data Source=۲۰۴.۹۳.۱۷۸.۱۵۷;
Initial Catalog=DatabaseName; User ID=DatabaseUserName;
Password=DatabaseUserPassword"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

این تگ ها باید داخل تگ <configuration>باشند.

طریقه ایجاد اکانت FTP و آپلود از طریق این پروتکل

ابتدا از داخل پنل مدیریت هاست یک اکانت ftp می سازیم. سپس با یکی از نرم افزار های ftp مانند FileZilla به راحتی فایل ها را upload می کنیم IP host .برای این اتصال ۲۰۴.۹۳.۱۵۶.۱۴۰و port=۲۱ می باشد.

```
<?xml version="۱.۰"?>
<configuration>
<system.web>
<compilation debug="true" targetFramework="۴.۰"/>
<httpRuntime targetFramework="۴.۰"/>
<customErrors mode="Off"/>
</system.web>
</configuration>
```

در این راهکار قصد داریم تا نحوه قرار دادن یک کنترل Menu و تنظیم آن برای نمایش فهرست محتویات سایت Asp.Net را آموزش دهیم . یکی از امکانات مناسبی که در هر سایت باید قرار بگیرد ، منو یا فهرست موضوعات سایت است . به وسیله کنترل Menu در Asp.Net و بدون نوشتن کد خاصی به راحتی می توانید منوی مورد نظر خود را بسازید . کنترل Menu انواع حالت های نمایش منو را چه بصورت استاتیک و چه بصورت داینامیک در اختیار شما قرار می دهد .

در حالت منوی استاتیک ، تمام شاخه های اصلی منو و زیر شاخه های آن باز بوده و کاربر قادر به مشاهده و کلیک بر روی آنهاست .

اما در حالت داینامیک معمولا سرعنوان ها فقط نمایش داده شده و با حرکت موس بر روی هر سرعنوان ، زیر منوی

آن بصورت Pop-Up نمایش داده می شود .
کنترل Menu این امکان را می دهد که ترکیبی از هر دو حالت استاتیک و داینامیک را نیز بکار ببرید.

اما نحوه تعیین کردن موضوعات برای نمایش درفهرست Menu ؟! . شما می توانید منو ها و زیرمنوهای کنترل را بصورت از پیش تعیین شده در هنگام طراحی تعریف نمایید و یا اینکه کنترل را به یک منبع داده ای که دارای ساختار درختی و سلسله مراتبی است ، مثل XmlDataSource پیوند دهید .
در این راهکار موارد زیر آموزش داده خواهند شد :

- ساخت یک منوی ساده و استاتیک و سپس اتصال هر عنوان به صفحه مربوطه.
- ساخت یک منوی پیشرفته تر که برای دریافت فهرست موضوعات خود به یک فایل XML به نام Web.sitemapمتصل می شود.
- تنظیم جهت چیدمان و نمایش منو.
- تلفیق منو های استاتیک با داینامیک و نمایش در هنگام اجرا.

پیش نیازها :

برای اجرای این راهکار نیاز دارید تا برنامه های زیر بر روی سایت شما نصب شده باشد :

۱. Microsoft Visual Studio .
۲. Microsoft .NET Framework version ۲.۰یا بالاتر (همراه با ویژوال استودیو نصب می شود) .

مرحله اول – طراحی یک سایت ASP.Net :

در مرحله اول از راهکار بایستی یک سایت ASP.Net را برای اجرای پروژه خود ایجاد نمایید . فرض بر این است که سایت را قبلا ایجاد کرده و یا با نحوه اجرای آن آشنایی دارید ، در غیر اینصورت به راهکار شماره ۱-۱ : نحوه ایجاد یک وب سایت ASP.Net در ویژوال استودیو بروید .

مرحله دوم – ساخت یک منوی ساده و ایستاتیک :

برای ساخت یک منوی ساده در صفحه مورد نظرمان مراحل زیر را انجام دهید :

۱. صفحه مورد نظر را باز کرده و به حالت Design بروید.
۲. از منوی ToolBox و بخش Navigation یک کنترل Menu را کشیده و بر روی صفحه قرار دهید.

۳. دراین مثال ، قصد داریم تا یک منوی افقی طراحی نماییم . بنابراین کنترل Menu را انتخاب کرده و از قسمت **Properties**، مقدار خاصیت **Orientation**را بر روی **Horizontal**قرار دهید.

تنظیم کردن منوی قرار داده شده بر روی صفحه:

در این بخش ، بوسیله ویرایشگر Menu Item Editor، آیتم های کنترل منو را تنظیم می کنیم :

۱. بر روی کنترل Menu راست کلیک کرده و سپس گزینه Edit Menu Items را انتخاب نمایید . پنجره Menu Item Editor، باز می شود.
۲. در زیر قسمت Items ، آیکون Add a root item را انتخاب کنید.
۳. در زیر قسمت Properties آیتم جدید ، مقدار خاصیت Text را بر روی Home و Navigate URLرا به Default.aspx ، تغییر دهید.
۴. مجددا با کلیک بر روی گزینه Add a root item ، یک آیتم جدید دیگر را به منو اضافه کنید.
۵. مقدار خاصیت Text آیتم جدید را بر روی Books و خاصیت Navigate آن را بر روی Books.aspxتعیین کنید.
۶. کار فوق را برای گزینه دیگر به نام Apps انجام داده و آدرس ضفحه مقصد را نیز Apps.aspx تعیین نمایید .

حال اگر به بخش Design صفحه دقت نمایید ، آیتم هایی که برای کنترل Menu را تعیین کرده اید ، مشاهده خواهید کرد.

۷. صفحات Home.aspx ، Books.aspxو Apps.aspx را به پروژه خود اضافه نمایید . حال زمان تست کردن منوی ساخته شده است ، مراحل زیر را انجام دهید:
   ۱. به صفحه اصلی یا Defalt.aspx رفته و کنترل های Ctrl + F۵ را برای اجرای پروژه فشار دهید.
   ۲. صفحه اجرا شده و کنترل Menu را با آیتم های تعیین شده بر روی صفحه مشاهده خواهید کرد.
   ۳. موس را بر روی هر آیتم ببرید ، آدرس مقصد آیتم در منوی Status Bar مرورگر نمایش داده می شود . با کلیک بر روی هر گزینه به صفحه مقصد آن منتقل خواهید شد.

مرحله سوم – ساخت یک کنترل Menu متصل به یک فایل نقشه سایت (Site Map) :

در بخش قبل ، یک کنترل Menu ساده را ایجاد و آیتم های آن را به صورت دستی در هنگام Design صفحه تعیین کردیم . در این بخش قصد داریم تا به جای تعیین آیتم های کنترل Menu از قبل ، آن را به یک فایل نقشه سایت به عنوان یک XmlDataSource متصل کنیم . این کار به کنترل این امکان را می دهد که ساختار و آیتم

کد	<siteMap> <siteMapNode title="Home" description="Home" url="default.aspx" > <siteMapNode title="Products" description="Our products" url="Products.aspx"> <siteMapNode title="Hardware" description="Hardware choices" url="Hardware.aspx" /> <siteMapNode title="Software" description="Software choices" url="Software.aspx" /> </siteMapNode> <siteMapNode title="Services" description="Services we offer" url="Services.aspx"> <siteMapNode title="Training" description="Training classes" url="Training.aspx" /> <siteMapNode title="Consulting" description="Consulting services" url="Consulting.aspx" /> <siteMapNode title="Support" description="Support plans" url="Support.aspx" /> </siteMapNode> </siteMapNode> </siteMap>

های خود را از یک فایل مجزا XML در هنگام اجرای صفحه دریافت نماید . شما می توانید هر زمان که بخواهید فایل XML مورد استفاده را بدون نیاز به کار با کنترل Menu ، آپدیت و به روز رسانی نمایید . سپس این تغییرات به صورت اتوماتیک در کنترل Menu اعمال می شود .

برای انجام این کار مراحل زیر را به ترتیب انجام دهید . در این مثال از یک کنترل Menu دوم استفاده می کنیم :

۱. مجدد از منوی ToolBox و قسمت Navigation ، یک کنترل دیگر Menu را انتخاب کرده و بر روی صفحه قرار دهید.
۲. اکنون ما به یک فایل نقشه سایت (site map) نیاز داریم . برای ایجاد این فایل مراحل زیر را انجام دهید:
   - در منوی Solution Explorer ، بر روی نام وب سایت کلیک راست کرده و گزینه Add New Itemرا انتخاب کنید.
   - از پنجره باز شده ، یک فایل Site Map را انتخاب کرده و بر روی گزینه Add کلیک نمایید.
۳. (۳کد XML زیر را در فایل Site Map ایجاد شده قرار دهید . مطابق تصویر زیر:

۴. کد XML فوق ساختار و آیتم های کنترل Menu را تعین می کند . تگ های SiteMap ode درون تگ <siteMapNode>اول ، بصورت عنصر فرزند یا زیر منو ، منوی اصلی نمایش داده می شوند.
۵. (۴فایل Site Map را ذخیره نمایید.

اتصال کنترل Menu به فایل: Site Map

اکنون نوبت آن است که یک کنترل داده وب مسیریاب مثل SiteMapPath را بر روی صفحه قرار داده و آن را به فایل Site Map متصل کنید . سپس کنترل Menu خود را نیز برای دریافت اطلاعات به کنترل سرور داده مسیریاب متصل نمایید .

برای این منظور مراحل زیر را انجام دهید :

۱. صفحه اصلی مورد نظر خود را باز کرده و به حالت Design بروید.
۲. بر روی گزینه Smart tag کنترل Menu کلیک نمایید تا پنجره Menu Tasks باز شود.
۳. از منوی باز شده ، کنترل کرکره ای Choose Data Source را انتخاب کرده و گزینه New Data Sourceرا انتخاب نمایید.
۴. پنجره Data Source Configuration Wizard ، باز می شود . گزینه Site Map را انتخاب کرده و یک ID در قسمت Specify an ID for the data source ، برای آن تعیین کرده و Ok نمایید.

پس از انجام مراحل فوق ، کنترل Menu به فایل (site map) متصل شده است . برای تست صفحه با زدن کلیدهای Ctrl + F۵ آن را اجرا نمایید . صفحه باز شده و منو بر روی آن قابل مشاهده است . با حرکت موس بر روی منو آیتم های آن را مشاهده کرده و کلیک نمایید .

نظیم درجات نمایش منو های استاتیک و داینامیک :

همانطور که گفتیم ، کنترل Menu دارای دو حالت برای نمایش منوها و زیرمنوهای خود است (استاتیک و داینامیک) .

در نمایش استاتیک ، تمام منو ها و زیرمنوهای کنترل بصورت باز هستند و کل کنترل قابل مشاهده است . کاربر می تواند بر روی هر کدام از منو ها نیز کلیک نماید .

اما در حالت داینامیک ، فقط منوها یا زیرمنوهایی که طراح تعیین کرده در ابتدا نمایش داده می شوند و بقیه مخفی هستند . زیرمنوهای مخفی در هنگامی که کاربر اشاره گر موس را بر روی منوی Parent آن ببرد ، نمایش داده می شود .

کنترل Menu ای که در مثال قبل ساختیم ، بجز سر منوهای اصلی آن ، بقیه منوهایش داینامیک و مخفی هستند . بوسیله کنترل Menu شما این توانایی را دارید که تعیین نمایید چه درجه ای از منوها و زیرمنوها ، استاتیک بوده و یا داینامیک باشند . بوسیله مراحل زیر این قابلیت را برای کنترل منوی خود تنظیم می کنیم:

۱. بر روی کنترل Menu در صفحه کلیک کرده و به منوی Properties آن بروید.

۲. مقدار خاصیت **StaticDisplayLevels**، آن را به ۲ تغییر دهید .

در این حالت ۲ درجه از منوهای کنترل (سرمنوها و یک مرحله زیرمنوهای آنها) بصورت استاتیک و بقیه درجات منوها ، مخفی بوده و به صورت دینامیک باز می شوند.

۳. با اجرای صفحه ، تفاوت این حالت را در عمل مشاهده کنید.

آموزش ASP.NET > کنترل های پیمایش > کنترل SITEMAPPATH

کنترل: SiteMapPath

کنترل SiteMapPath یک مسیر نمایش از صفحه ابتدا تا صفحه جاری را بصورت یک لینک چند بخشی به کاربر نمایش می دهد .

شما می توانید با کلیک بر روی هر یک از بخش های این کنترل به یک مرحله قبلتر از صفحه جاری بروید.

این کنترل امکانات زیادی برای تغییر شکل و تنظیم لینک ها در اختیار طراح قرار می دهد .

کنترل SiteMapPath اطلاعات خود را از یک فایل SiteMapPath دریافت میکند . فایل SiteMapPath فایلی است که شامل اطلاعات کلیه صفحات و مسیرها در سایت شما همراه با توضیحی مثل متن لینک, عنوان و آدرس مقصد (URL) می باشد .

ایجاد یک فایل SiteMapPath برای سایت خودتان بسیار مفید است زیرا می توانید از این فایل برای دادن اطلاعات به سایر کنترل های پیمایش Asp .Net استفاده کرد و همچنین آن را به عنوان راهنما برای موتورهای جستجو قرار دهید .

همچنین چنانچه تغییر در مسیر ها و فایل های سایتتان بوجود آید کافی است اطلاعات فایل SiteMapPath را به روز کنید .

در ورژن های قدیم تر Asp و یا زبان های دیگر چنانچه لینک یا آدرسی در سایت تغییر می کرد ، مجبور بودیم تمام لینک ها و صفحات که آن لینک در آن وجود دارد را اصلاح نمایید. اما با امکاناتی که کنترل های پیمایش Asp.Net دارند با تغییر لینک در فایل داده ای آدرس های فایل SiteMapPath این تغییر بصورت اتوماتیک در تمام سایت اعمال می شود .

کد زیر یک فایل نمونه SiteMapPath را نمایش می دهد :

```
<siteMap>
    <siteMapNode title="Home" description="Home" url="~/default.aspx" >
        <siteMapNode title="Services" description="Services we offer"
url="~/Services.aspx" />
        <siteMapNode title="Training" description="Training classes"
url="~/Training.aspx" />
  <siteMapNode title="Consulting" description="Consulting services"
url="~/Consulting.aspx" />
      </siteMapNode>
    </siteMapNode>
</siteMap>
```

شکل کلی قرار دادن یک کنترل Site Map بر روی صفحه بصورت زیر است :

```
< asp:SiteMapPath ID="SiteMapPath۱" Runat="server">  </asp:SiteMapPath >
```

شما به راحتی و بدون هیچ کد نویسی خاصی می توانید به وسیله کنترل SiteMapPath مسیر پیمایش برای سایت خود بسازید .

توجه داشته باشید که به وسیله کنترل SiteMapPath کاربر می تواند مسیر عقب (صفحاتی که قبلا مرور کرده و یا در هرم لینک های سایت در رده بالاتری هستند) برود . اما به کاربر امکان حرکت به سمت جلو را نمی دهد .

همواره آخرین لینک در کنترل SiteMapPath صفحه جاری است .

کنترل: Menu

از کنترل Menu ، می توانید برای ایجاد منوهای ایستاتیک و یا دینامیک در صفحات ASP.Net استفاده نمایید .

منوهایی همانند لیست موضوعات در همین سایت و یا منویی از محصولات و ... .

شما می توانید آیتم های موجود در کنترل منو را به صورت دستی برای آن تعیین کرده و یا با اتصالش به یک پایگاه یا فایل داده ای ، آیتم های آن منبع داده ای را توسط کنترل نمایش د هید.

بدون نوشتن کد یا دستوری ، می توانید عملکرد این کنترل را به راحتی در محیط ویژوال استودیو تنظیم نمایید.

شکل کلی تعریف و نمایش یک کنترل Menu در صفحه های ASP.Net به صورت زیر است :

```
<asp:Menu ID="Menu۱" runat="server">
  <Items>
    <asp:MenuItem Text="طراحی وب">
```

```
        <asp:MenuItem Text="HTML" NavigateUrl="~/HTML/Index.aspx" />
        <asp:MenuItem Text="CSS" NavigateUrl="~/CSS/Index.aspx" />
    </asp:MenuItem>
    <asp:MenuItem Text="برنامه نویسی وب">
        <asp:MenuItem Text="ASP.Net"
NavigateUrl="~/ASP_NET/Standard/Index.aspx" />
        <asp:MenuItem Text="PHP" NavigateUrl="~/PHP/Index.aspx" />
      </asp:MenuItem>
  </Items>
</asp:Menu>
```

طراحی وب

برنامه نویسی وب

نکات مهم درباره کنترل Menu :

۱. آیتم های کنترل Menu ، در مجوعه < Items >کنترل ، تعیین می شوند.
۲. برای ساخت هر آیتم ، درون مجموعه < Items >کنترل ، یک تگ < asp:MenuItem >تعیین می کنیم.
۳. عنوان هر آیتم ، توسط خاصیت Text ، درون تگ آن تعیین می شود.
۴. آدرسی که در صورت کلیک بر روی هر آیتم ، کاربر به ان ارسال می شود را توسط خاصیت NavigateUrlتعیین می کنیم.

نحوه قرار دادن یک کنترل Menu بر روی صفحه ASP.Net :

۱. ابتدا یک صفحه ASP.Net در محیط Visual Studio ایجاد کرده و یا صفحه از قبل طراحی شده خود را باز کنید.
۲. از منوی Toolbox و از قسمت کنترل های Navogation یک کنترل Menu را کشیده و روی صفحه قرار دهید.
۳. همجنین می توانید به صورت مستقیم در فسمت کدنویسی Source صفحه کد مربوط به کنترل را تایپ نمایید.

کنترل: TreeView

کنترل TreeView برای نمایش اطلاعات سلسله مراتبی مثل محتویات یک کتاب یا فهرست مطالب یک سایت به شکل درخت وار استفاده می شود .

امکانات و قابلیت های کنترل TreeView :

کنترل TreeView دارای ویژگی ها و قابلیت های زیر می باشد :

- اتصال اتوماتیک به یک منبع داده ای که باعث می شود تا گره ها (Nodes) کنترل به یک منبع سلسله مراتبی مثل فایل XML متصل شوند.
- امکان ساخت یک منو و یا فهرست برای مطالب سایت با کمک از کنترل SiteMapDataSource
- متن گره ها می توانند نوشته عادی یا لینک به صفحات دیگر باشند.
- می توانید ظاهر کنترل را با استفاده از Theme و استایل های مختلف به شکل دلخواه درآورید.
- دسترسی حین اجرا و قابل برنامه ریزی به کنترل TreeView که این امکان را به شما می دهد تا گره ها و لینک های کنترل را بصورت دینامیک تعیین نمایید.
- امکان Callback کردن صفحه به سرور در هنگام لینک بر روی گره ها.
- قابلیت نمایش یک کادر انتخابی (CheckBox) در کنار هر گره.

شکل کلی استفاده از یک کنترل TreeView به صورت زیر است :

```
<asp:TreeView ID="TreeView۱" runat="server">
  <Nodes>
    <asp:TreeNode Text="طراحی وب">
        <asp:TreeNode Text="HTML" NavigateUrl="~/HTML/Index.aspx" />
        <asp:TreeNode Text="CSS" NavigateUrl="~/CSS/Index.aspx" />
    </asp:TreeNode>
    <asp:TreeNode Text="برنامه نویسی وب">
        <asp:TreeNode Text="ASP.Net" NavigateUrl="~/ASP_NET/Standard/Index.aspx" />
        <asp:TreeNode Text="PHP" NavigateUrl="~/PHP/Index.aspx" />
     </asp:TreeNode>
  </Nodes>
</asp:TreeView>
```

طراحی وب
HTML

نحوه قرار دادن یک کنترل TreeView بر روی صفحه ASP.Net :

۱. ابتدا یک صفحه ASP.Net در محیط Visual Studio ایجاد کرده و یا صفحه از قبل طراحی شده خود را باز کنید.
۲. از منوی Toolbox و از قسمت کنترل های Navogation یک کنترل TreeView را کشیده و روی صفحه قرار دهید.
۳. همجنین می توانید به صورت مستقیم در فسمت کدنویسی Source صفحه کد مربوط به کنترل را تایپ نمایید.

نحوه کار با کنترل : TreeView

کنترل TreeView نمایش چندین نوع اطلاعات را داراست :

۱. اطلاعات استاتیک یا ثابت که در هنگام طراحی توسط طراح صفحه تعیین می شود.
۲. اتصال به یک منبع داده ای و دریافت اطلاعات به آن.
۳. اضافه شدن اطلاعات به صورت برنامه ریزی شده در حین اجرا بل‌بر عملکرد کاربر.

۱) نمایش اطلاعات ثابت یا استاتیک :

شما می توانید گره ها و لینک های کنترل TreeView از قبل و در هنگام طراحی تعیین نمایید . برای این منظور بایستی گره های مورد نظر خود را در مجموعه < nodes > کنترل تعیین کنید . هر گره توسط یک تگ < asp:TreeNode > ایجاد شده و یک فرزند کنترل TreeView محسوب می شود .

```
<asp:TreeView ID="TreeView۱" runat="server">
  <Nodes>
    <asp:TreeNode Text="طراحی وب">
      <asp:TreeNode Text="HTML"
NavigateUrl="~/HTML/Index.aspx" />
      <asp:TreeNode Text="CSS" NavigateUrl="~/CSS/Index.aspx"
/>
    </asp:TreeNode>
  </Nodes>
</asp:TreeView>
```

- طراحی وب
  - HTML
  - CSS

۲) اتصال کنترل TreeView به یک منبع داده ای :

شما می توانید کنترل TreeViwe را به یک منبع داده ای که از خاصیت IHierarchicalDataSource پشتیبانی می کند مثل XmlDataSource متصل نمایید . در هنگام اتصال کنترل به یک منبع داه ای می توانید تعیین نمایید تا گره های کنترل به کدام فیلد بانک اطلاعاتی متصل شود . برای دریافت اطلاعات بیشتر به بخش اتصال کنترل TreeViwe به یک منبع داده ای بروید .

۳) نمایش اطلاعات به صورت برنامه ریزی شده با استفاده از TreeNodesCollection :

شما می توانید اطلاعات کنترل TreeView را بصورت برنامه ریزی شده و در حین اجرا به کنترل ارسال نمایید . برای این منظور از خاصیت Nodes کنترل استفاده می شود . این خاصیت با استفاده از کلاس TreeNodesCollection کار می کند . برای دریافت اطلاعات بیشتر به بخش توضیح Nodes بروید .

جلسه پانزدهم : آموزش دسترسی به پایگاه داده – ASP.NET

ASP.NET امکان دسترسی و استفاده از منابع داده ی زیر را به برنامه نویس می دهد:

MySQL)، Oracle، SQL Server، Access،Databases (e.g.
XML documents
Business Objects
Flat files

ASP.NET فرایندهای پیچیده و سنگین دسترسی به داده را پنهان می کند و همچنین سطح بسیار بالاتری از کلاس ها و اشیاء که دسترسی به داده از طریق آن ها بسیار سهل است را ارائه می دهد. این کلاس ها تمامی کدنویسی های پیچیده که برای اتصال به داده، بازیابی داده، پرسمان از داده (data querying) و دستکاری داده ها انجام می شود را پنهان می کند.

ADO.NET یک تکنولوژی است که به مثابه ی یک پل ارتباطی بین اشیاء کنترلی (control object) ASP.NET و منبع داده (data source)backend عمل می کند. در این فصل به نحوه ی دسترسی به داده و کار با آن به طور مختصر خواهیم پرداخت.

بازیابی و نمایش داده

به منظور بازیابی و نمایش داده در ASP.NET به دو نوع data control احتیاج داریم:

data source control (کنترل منبع داده) – اتصال به داده، انتخاب و گزینش داده و دیگر کارها و عملیاتی همچون صفحه بندی (paging) و ذخیره ی داده در حافظه ی پنهان (caching) را مدیریت می کند.

data view control (کنترل مقید سازی و نمایش داده) – داده را متصل کرده سپس نمایش می دهد، همچنین امکان دستکاری و مدیریت داده ها را برای برنامه نویس فراهم می آورد.

به مفاهیم مقید سازی و اتصال داده (data binding) و کنترل های منبع داده ای (data source control) در بخش های بعدی با جزئیات بیشتر خواهیم پرداخت. در این قسمت، برای دسترسی به داده از یک کنترل SqlDataSource و از کنترل GridView جهت نمایش و مدیریت داده بهره می گیریم.

همچنین از پایگاه داده ی Access که دربردارنده ی اطلاعاتی درباره ی کتاب های .NET. موجود در بازار است استفاده خواهیم کرد. اسم پایگاه داده ASPDotNetStepByStep.mdb بوده و جدول داده ای (data table) که بکار می بریم را DotNetReferences نام گذاری کرده ایم.

جدول مورد نظر دارای ستون های زیر خواهد بود:

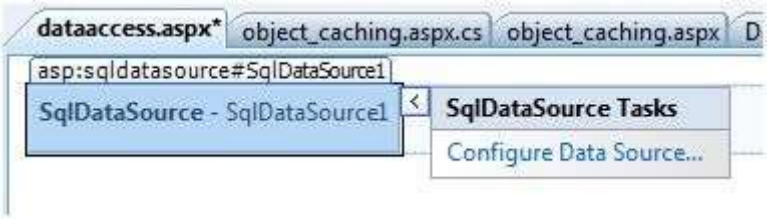
ID، Title، AuthorFirstName، AuthorLastName، Topic و Publishe.

ذیل تصویری از جدول داده ای مذکور مشاهده می کنید:

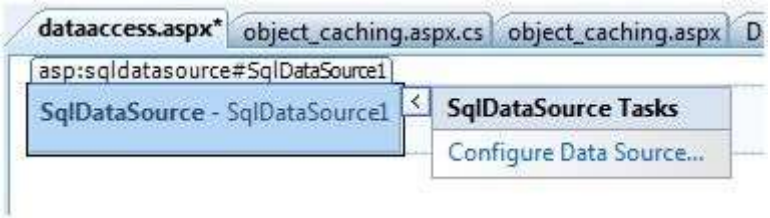
ID	Title	AuthorLastNam	AuthorFirstNam	Topic
1	Essentials .NET	Box	Don	Gestalt of .NET
2	Programming Microsoft Visual C	Shepherd	George	C++ in the .NET World
3	ASP.NET Step by Step	Shepherd	George	ASP.NET from square one
4	Programming Microsoft ASP.NET	Esposito	Dino	ASP.NET comprehensive reference
5	Windows Forms Programming in	Sells	Chris	Windows UIs usin .NET
6	Applied Microsoft .NET Framewo	Richter	Jeffrey	Comprehensive .NET refrence
7	.NET Compact Framework Progra	Yao	Paul	How to do .NET on small devices
8	.NET Framework Essentials	Thai	Thuan	How to do .NET development
9	Microsoft Visual Basic .NET Prog	MacDonald	Matthew	Digestible Visual Basic examples
10	Designing Microsoft ASP.NET Ap	Reilly	Douglas	ASP.NET Design topics
11	The C	Hejlsberg	Anders	Definitive C# Reference
12	Programming Windows with C	Petzold	Charles	The original Windows programming autho
13	The CL Infrastructure Annotated	Miller	Jim	Info from someone really close to the CLR

اکنون مراحل زیر را طی می کنیم:

یک وب فرم ایجاد کرده، سپس کنترل SqlDataSourceControl را به آن اضافه کنید.

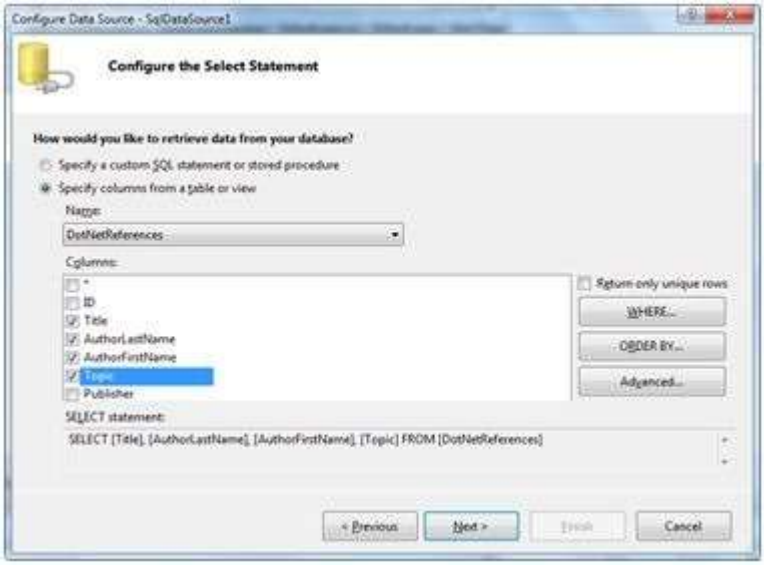


روی گزینه ی Configure Data Source کلیک کنید.



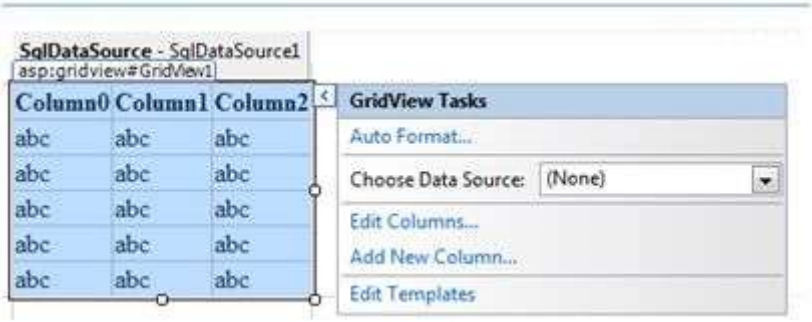
حال روی دکمه ی New Connection کلیک کرده تا اتصال با یک پایگاه داده برقرار گردد.

پس از اینکه اتصال با پایگاه داده برقرار شد، آن را برای استفاده در آینده ذخیره کنید. در مرحله ی بعدی از شما خواسته می شود دستور Select را پیکربندی (config) کنید:



اکنون ستون ها را انتخاب کرده و دکمه ی next را فشار دهید تا مراحل به پایان برسد. به دکمه های WHERE، ORDER BY، Advanced دقت کنید. دکمه های نام برده به شما امکان ارائه ی عبارت های (clause) Where، Orderby و همچنین تعریف دستورات (command) insert، update و delete اس کیو ال را به ترتیب فراهم می کند. از این طریق شما قادر خواهید بود داده ها را مدیریت یا دستکاری کنید.

کنترل GridView را به فرم اضافه کنید. منبع داده ای مورد نظر را انتخاب کرده و با استفاده از گزینه ی AutoFormat کنترل را قالب بندی (فرمت) کنید.



پس از انجام این کار، کنترل GridView قالب بندی شده عنوان های ستون ها را نمایش می دهد. اکنون برنامه آماده ی اجرا است.

SqlDataSource - SqlDataSource1

Title	AuthorLastName	AuthorFirstName	Topic
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc

برنامه را اجرا کنید.

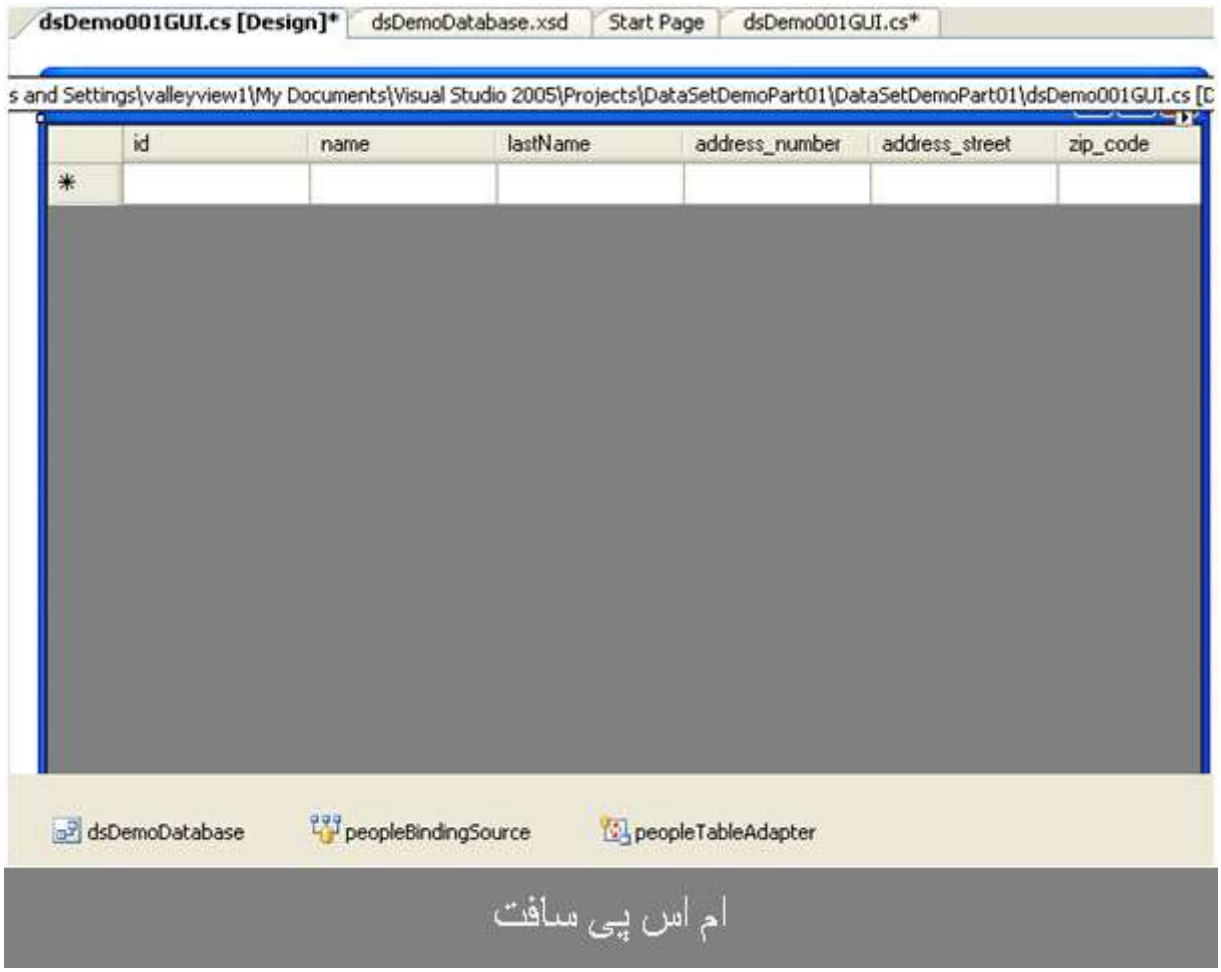
کد Content file:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="dataaccess.aspx.cs" Inherits="DataAccess.dataaccess" %>

<!DOCTYPE html PUBLIC "-//W۳C//DTD XHTML ۱.۰ Transitional//EN"
"http://www.w۳.org/TR/xhtml۱/DTD/xhtml۱-transitional.dtd">
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml">
<html xmlns="http://www.w۳.org/۱۹۹۹/xhtml" >

<head runat="server">
<title>
Untitled Page
</title>
</head>
<body>
<form id="form۱" runat="server">
<div>
```

```
<asp:SqlDataSource ID="SqlDataSource١" runat="server" ConnectionString
="<%$ ConnectionStrings:ConnectionString٥ %>"ProviderName="<%$
ConnectionStrings:ConnectionString٥.ProviderName %>" SelectCommand=
"SELECT * FROM [EEE]"></asp:SqlDataSource>
<asp:GridView ID="GridView١" runat="server"
AutoGenerateColumns="False" CellPadding="٣"
DataSourceID="SqlDataSource١" BackColor="#DEBA٨٤" BorderCol
or="#DEBA٨٤" BorderStyle="None" BorderWidth="١px"CellSpacing
="٢">
<RowStyle BackColor="#FFF٧E٧" ForeColor="□ ₡٤٥١٠" />
<Columns>
<asp:BoundField DataField="Title" HeaderText="Title"
SortExpression="Title" />
<asp:BoundField DataField="AuthorLastName"
HeaderText="AuthorLastName" SortExpression="AuthorLastN
ame" />
<asp:BoundField DataField="AuthorFirstName"
HeaderText="AuthorFirstName" SortExpression="AuthorFirstN
ame" />
<asp:BoundField DataField="Topic"
HeaderText="Topic" SortExpression="Topic" />
</Columns>
<FooterStyle BackColor="#F٧DFB٥" ForeColor="□ ₡٤٥١٠" />
<PagerStyle
ForeColor="□ ₡٤٥١٠" HorizontalAlign="Center" />
<SelectedRowStyle BackColor="□ ٧٣₳٩C"
Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#A٥٥١٢٩" Font-Bold="True"
ForeColor="White" />
<SortedAscendingCellStyle BackColor="#FFF١D٤" />
<SortedAscendingHeaderStyle BackColor="#B٩٥C٣٠" />
<SortedDescendingCellStyle BackColor="#F١E٥CE" />
<SortedDescendingHeaderStyle BackColor="□ ٩٣٤٥₣" />
</asp:GridView>
```

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

جلسه شانزدهم:آموزش ساخت Dataset در سی شارپ



در این مقاله قصد آموزش چگونگی ایجاد یک **table adapter** و **dataset** در سی شارپ را دارم .در تجربه هایی که داشته ام ، dataset همیشه آزار دهنده بوده است ، زیرا برای تغییر یک خاصیت در دیتابیس باید جدول ها و table adapter ها را در فایل xds آپدیت کنید و نسخه ی جدید نرم افزار را انتشار دهید.

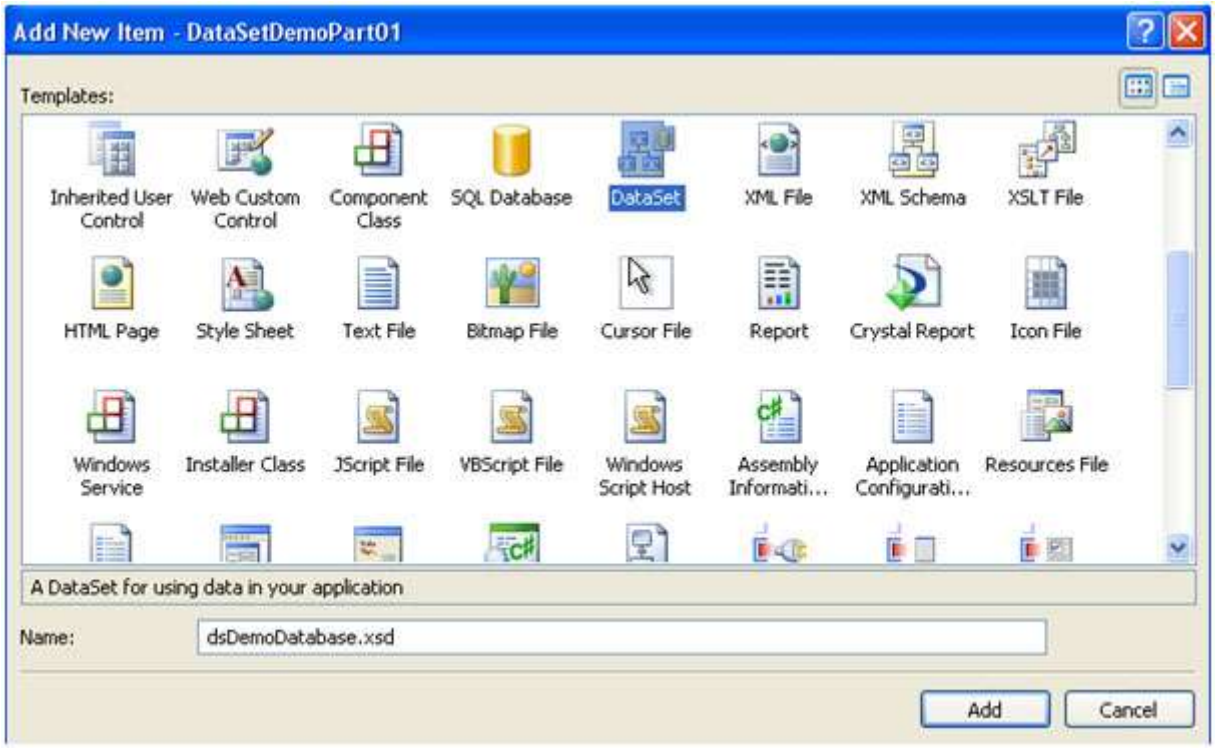
اما ایجاد connection و query ها با dataset آسان تر و سریع تر از ایجاد یک کلاس Connection Manager است ، (اما همیشه این بهترین کار نیست). اگر قصد ساخت یک برنامه ای دارید که شامل Data Grid View میشود و نیاز به افزودن ، آپدیت و حذف در آن است ؛ dataset بهترین گزینه است.

ایجاد dataset و table adapter

۱ چارچوب خود را ایجاد کنید.

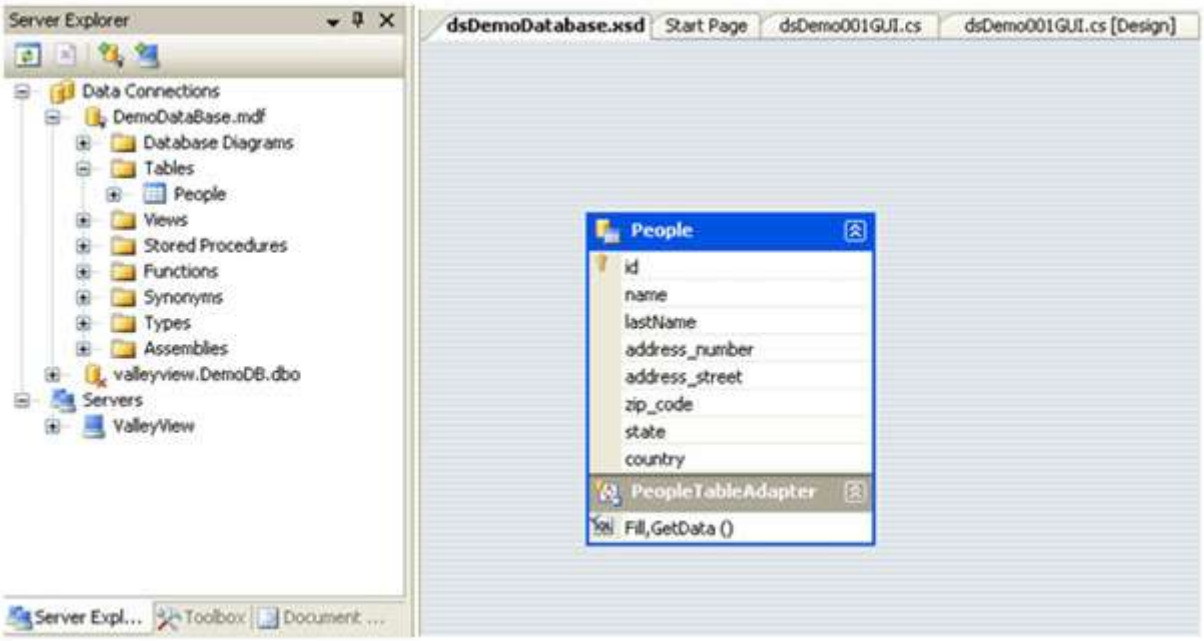
۲ یک کنترل Data Grid View به برنامه اضافه کنید.

۳ یک فایل Dataset (xds) ایجاد کنید.



۴ فایل xds خود را باز کنید.

۵ جدولی که قصد دارید با آن کار کنید را از قسمت Server explorer بکشید. جدول و table adapter شما ایجاد خواهد شد.



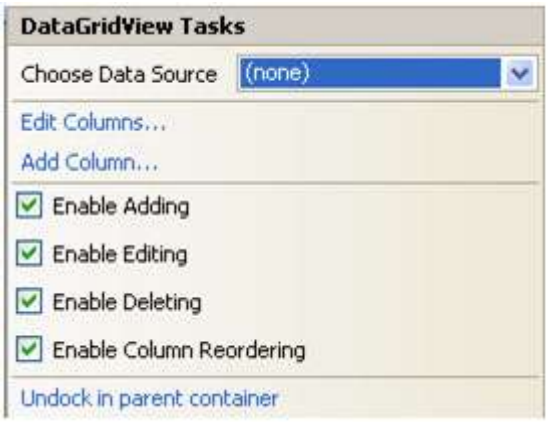
اگر بر روی table adapter کلیک کنید و مشخصات آن را چک کنید ، میتوانید ببینید که دستورات Select، insert، update و delete به صورت اتوماتیک تولید شده اند.

به این بستگی دارد که جدول تان چگونه طراحی شده است. زمانی که من با یک جدول بدون کلید اصلی کار میکردم ، دستور update تولید نشده بود . میتوانید دستور را اضافه کنید ، اما زمانی که موقع update یک مورد در data grid view پیش بیاید ، ممکن است مشکلاتی پیش بیاید و update صورت نگیرد.

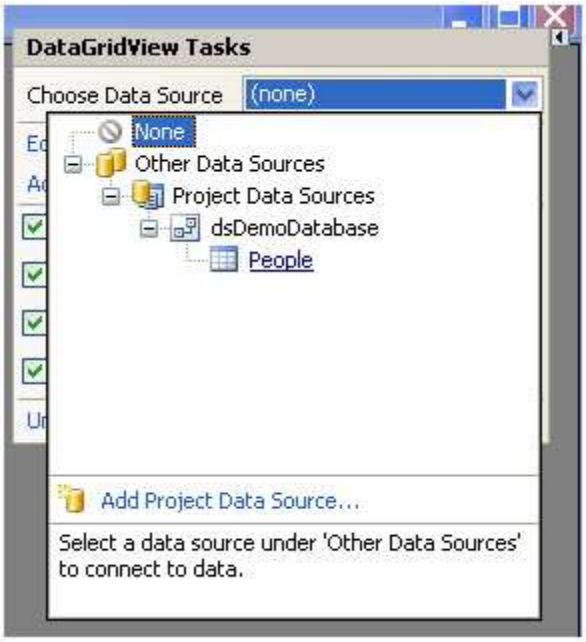
اگر میخواهید با این مشکلات رو به رو نشوید ، یک کلید اصلی در هنگام طراحی به جدولتان در دیتابیس اختصاص دهید.

حال قطعه کدی که به صورت اتوماتیک در رویداد load form ایجاد شده است را چک کنید. این قطعه کد data grid view را با داده های موجود در جدولتان پر میکند.

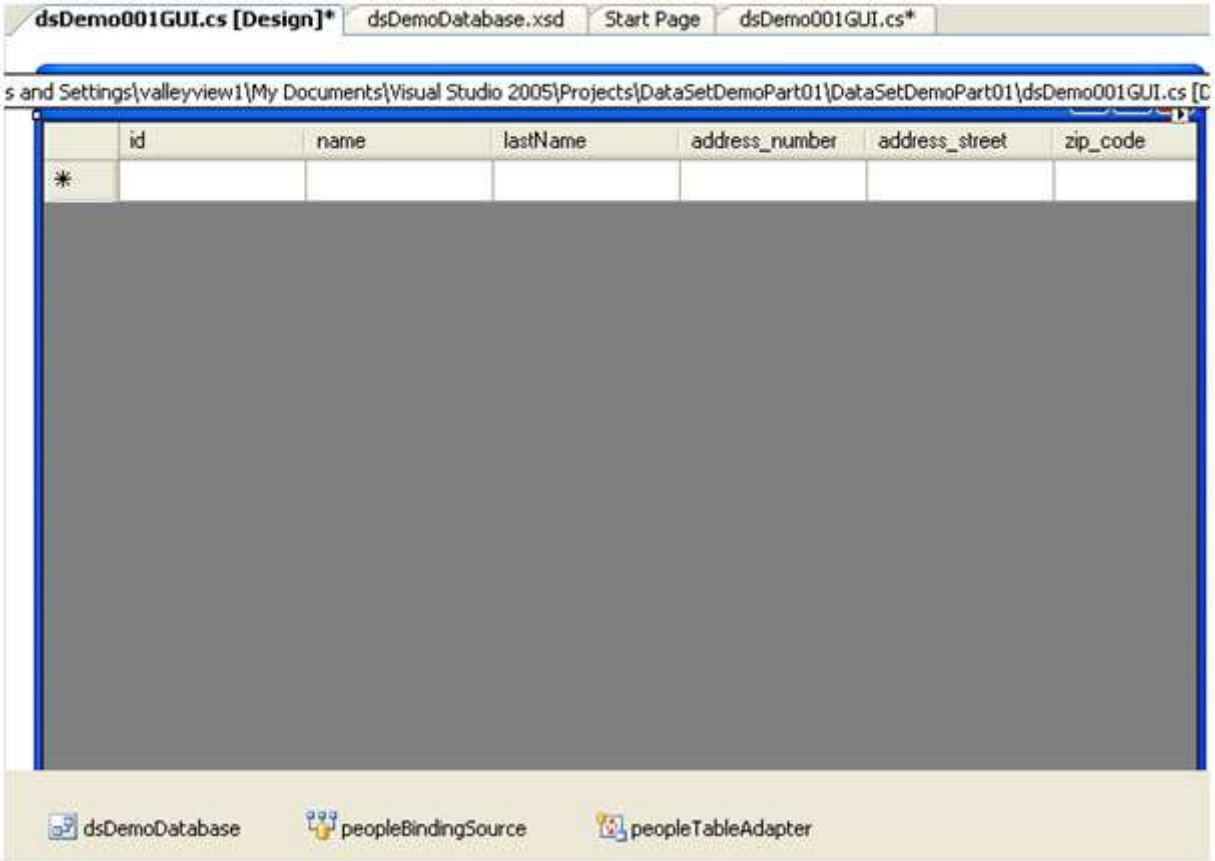
۶ برای دسترسی به خصوصیات insert، update و delete در کنترل data grid view خود بر روی دکمه ی کنترل option کلیک کنید.



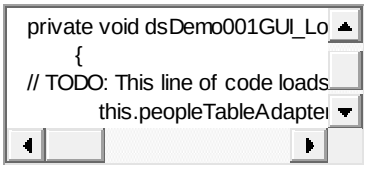
۷ در همان قسمت ، Data Table و Data Source خود را انتخاب کنید.



حال ستون های data grid view به صورت اتوماتیک ساخته میشود . همچنین اشیای table adapter ، اشیای Binding Source و dataset .



همچنین در رویداد Load Form این کد برای پر شدن با اطلا عات جدول در هر بار لود شدن فرم window استفاده میشود.



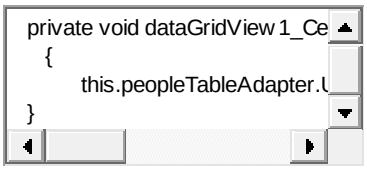

```
private void dsDemo۰۰۱GUI_Load(object sender, EventArgs e)
      {
// TODO: This line of code loads data into the 'dsDemoDatabase.People'
table. You can move, or remove it, as needed.
          this.peopleTableAdapter.Fill(this.dsDemoDatabase.People);
      }
```

۸ حال باید یک button که برای حذف مورد انتخاب شده است ، اضافه کنید.

۹ حال وقت آن رسیده است که از رویداد برای آپدیت موارد از data grid view استفاده کنید. Grid view را انتخاب کنید و به بخش مشخصات رویداد بروید.

۱۰ رویداد CellEndEdit را ایجاد کنید.

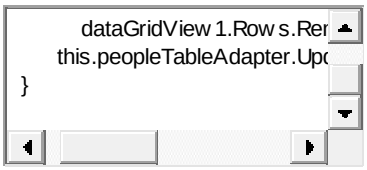
۱۱ در قسمت کد به متد رویداد بروید و کد زیر را تایپ کنید




```
private void dataGridView۱_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
  {
      this.peopleTableAdapter.Update(dsDemoDatabase);
}
```

حال برای تنظیم grid view خود احتیاج دارید که Add Row مربوط به Allow User را True کنید.

حال یک دکمه ی رویداد کلیک ایجاد کنید و کد زیر را در آن وارد کنید:




```
private void button۱_Click(object sender, EventArgs e)
    {
        foreach (DataGridViewRow row in dataGridView۱.SelectedRows)
            dataGridView۱.Rows.Remove(row);
        this.peopleTableAdapter.Update(dsDemoDatabase);
    }
```

دکمه سطر انتخاب شده از data grid view را حذف میکند و سپس متد Update را از شئ Table Adapter فراخوانی میکند. همچنین خاصیت انتخاب grid view به انتخاب همه ی سطر ها تغییر میکند تا از بروز استثنا در زمان حذف سطر جلوگیری کند.

بسته به کاری که در حال انجامش هستید ، Table Adapter متد Update را فراخوانی میکند و جدول در Dataset و base را آپدیت میکند. اگر میخواهید سطر ها را حذف کنید ، delete query را فراخوانی کنید ؛ اگر میخواهید سطر اضافه کنید ، insert query را فراخوانی کنید و اگر میخواهید مقادیر سلول های موجود را تغییر دهید ، query Update را فراخوانی کنید.

جمع بندی

برای استفاده از Dataset در یک جدول باید یک جدول با طراحی در دیتابیس خود داشته باشید (شامل یک کلید اصلی و افزودن خودکار ستون).

Dataset میتواند مقدار زیادی از زمان برای ایجاد Connection string و queries را برایتان ذخیره کند.

در این مطلب آموزشی شما متوانید تمامی پروژه های برنامه نویسی به زبان C# را خودتان انجام بدین

در برنامه نویسی تنها مطلب مهم برقراری با پایگاه داده که SQL است و همچنین که بتوانید دستوراتی مانند Insert، update و Delete را توسط فرم برنامه خود به دیتابیس اعمال نمایید

تعریف : به مجموعه کامپوننت هایی که برای دسترسی به داده های بانک اطلاعاتی در .NET استفاده می شود Ado.Net گفته می شود .

می توانیم بگوییم که دو روش برای اتصال به بانک وجود دارد اتصال متصل(online)و غیر متصل .

کار با بانک اطلاعاتی بصورت متصل :

هنگام استفاده از اشیا و متد های مربوط به این نوع اتصال ارتباط بین فرم و بانک در تمام مدت باید برقرار باشد برای همین موضوع این نوع کار با بانک متصل می نامند. معمولا این نوع کار با بانک از سرعت بسیار بالاتری نسبت به روش دوم که غیر متصل نام دارد برخوردار است:

کلاسهای ارتباط با بانکهای اطلاعاتی :

توجه - در اینجا در مثالها از بانک اطلاعاتی SqlServer استفاده میشود.

برای استفاده از این کلاس ها باید فضای نام زیر را با استفاده از using به برنامه اضافه کرد .

```
System.Data.SqlClient ;
```

۱- کلاس SqlConnection :

این کلاس وظیفه برقراری ارتباط بین برنامه و بانک اطلاعاتی را بر عهده دارد.

- هنگامی که می خواهید یک نمونه از ان کلاس را ایجاد کنید باید پارامتری را به نام Connection String به آن ارسال کنید .

Connection String رشته ای است که شامل تمام داده های مورد نیاز برای برقراری اتصال به یک بانک اطلاعاتی می شود .

ویژوال استودیو با استفاده از ویزارد AddConnection و اطلاعاتی که کاربر وارد می کند چنین رشته ای را ایجاد کرده و در اختیار SqlConnection قرار می دهد .

اغلب بهتر است که متن لازم برای ConnectionString را خودمان بنویسیم که به صورت زیر است :

"Data Source = local;Initial Catalog = university;Integrated Security = True"

در کد بالا: local نام سروری است که بانک بر روی آن قرار دارد که در اینجا چون سرور خود کامپیوتر ماست مقدار آن را local قرار داده ایم که می توانیم به جای آن از "." (نقطه) هم استفاده کنیم .
University نام بانکی است که قرار است ما با اطلاعات آن کار کنیم .
-متن ConnectionString به صورت پارامتر به شیئ جدید SqlConnection فرستاده می شود به صورت زیر:

```
SqlConnection Con = New SqlConnection( "Data Source = local;Initial Catalog = university;Integrated Security = True");
```

در کد بالا : Con یک نمونه جدید از نوع SqlConnection است که برای استفاده از آن آن را ساخته ایم .

متصل و قطع کردن اتصال به یک بانک اطلاعاتی :
با استفاده از متدهای Open و Close در کلاس SqlConnection به بانک متصل شده و یا اتصال خود را قطع کنیم .

```
Con.Open();و Con.Close();
```

۲- کلاس SqlCommand :
این کلاس حاوی یک دستور Sql برای اجرا بر روی داده های دریافت شده از بانک اطلاعاتی است این دستور می تواند یک دستور SELECT برای انتخاب داده هایی خاص ، یک دستور INSERT برای درج داده های جدید در بانک اطلاعاتی ، یک دستور DELETE برای حذف داده ها از بانک اطلاعات و یا حتی فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی می باشد .
ایجاد آن به صورت زیر می باشد :

```
SqlCommand Cmd = New SqlCommand();
```

نکته – در برنامه های بانک اطلاعاتی معمولا از اشیای ایجاد شده از کلاس SqlCommand به تنهایی استفاده نمی کنند بلکه آنها را همراه با DataSet ها و DataAdapter ها به کار می برند .

همچنین اشیای SqlCommand می توانند به همراه اشیای ایجاد شده از کلاس DataReader مورد استفاده قرار گیرند .

خاصیت connection کلاس SqlCommand :

قبل از اینکه بتوانیم از یک شئ از کلاس SqlCommand استفاده کنیم باید بعضی از خاصیت های آن را تنظیم کنیم اولین خاصیتی که باید تنظیم شود خاصیت Conection است .

این خاصیت می تواند یک مقدار از نوع SqlConnection را دریافت کند :

```
Cmd .Connection = Con ;
```

توضیح کد بالا :

Cmd همان شیئی است که از کلاس SqlCommandقبلا ساخته ایم .

Con شیئی است که از نوع SqlConnection قبلا ساخته ایم .

خاصیت CommandText کلاس SqlCommand :

خاصیتبعدی که باید از کلاس SqlCommand تنظیم شود خاصیت CommandText است .

این خاصیت متنی را دریافت می کند که می تواند حاوی یک دستور Sql و یا فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی باشد که باید روی داده ها اجرا شود .

متد ExecuteNonQuery :

این متد دستورات را بر روی بانک اطلاعات اجرا می کند .

برای استفاده از این متد باید ابتدا اتصال خود را به بانک اطلاعاتی برقرار کنید سپس با فراخوانی این متد دستور موجود در شئ Command را اجرا کنید .

کلاس SqlDataReader :

از طریق این کلاس می توانیم اطلاعات را از بانک دریافت کنیم . که به صورت زیر تعریف می شود :

```
SqlDataReader  dr;
```

متد ExecuteReader :

برای دریافت اطلاعات از بانک از متد ExecuteReader شی تقاضا استفاده می کنیم ، یک شی از نوع SqlDataReader تعریف کرده و مقدار ExecuteReader را برابر آن قرار می دهیم

```
SqlDataReader dr = Cmd.ExecuteReader();
```

سپس از طریق متد Read شی DataReader اطلاعات را در یک حلقه واکشی می کنیم .

ارائه مراحل کلی یک ارتباط و مثال

برای تقاضا از یک جدول در بانک اطلاعاتی باید مراحل زیر طی شود :

۱- اضافه نمودن فضا نام های مورد نیاز : برای ارتباط با بانک های اطلاعاتی به فضا نام System.Data و همچنین فضا نام System.Data.SqlClient برای کار با دیتابس SQLServer و فضا نام System.Data.Oledb برای کار با بانک Access نیازمندیم .

۲- تعیین رشته اتصال (ConnectionString) : رشته اتصال رشته شامل تنظیماتی جهت اتصال به بانک اطلاعاتی می باشد ، این رشته برای هر بانک متفاوت خواهد بود

۳-تعیین شی اتصال (Connection Object) : کلاسی است برای برقراری ارتباط با بانک اطلاعاتی است ، این شی که از کلاس DbConnection ارث بری می کند اعمالی مانند باز و بسته کردن اتصال با بانک را از طریق رشته اتصال انجام می دهد.

۴- تعیین رشته تقاضا (Text Command) : همان دستورات SQL است که جهت یک تقاضا ارائه می گردد ، این تقضا جهت دریافت اطلاعات (Select) درج اطلاعات (Insert) ، ویرایش اطلاعات (Update) و یا حذف اطلاعات(Delete) یا... صورت می گیرد .

۵- تعیین شی تقاضا (Command Object) : کلاسی است جهت ارسال و دریافت تقاضا از طریق شی اتصال به بانک اطلاعاتی

۶- باز کردن اتصال

۷- اجرای درخواست

۸- دریافت اطلاعات (در صورتی که تقاضا Select باشد)

۹- بستن اتصال

- «درج ، حذف و بروز رسانی » برای درج ، حذف و بروز رسانی به صورت متصل و مستقیم از متد ExecuteNoneQuery مربوط به شی Command استفاده می شود . این شی درخواست مربوط به insert,delete,update یا هر درخواست دیگری را بدون هیچ درخواستی انجام میدهد و خروجی آن تعداد سطر های تحت تاثیر درخواست می باشد.

```
string connectionString = "Data Source=(local);Initial
Catalog=univercity;Integrated Security=true;”;
SqlConnection Con = new SqlConnection(connectionString);
string commandText = "insert into student(name,family) values('ali','arefi’)”;
SqlCommand Cmd = new SqlCommand(CommandText, Con);
Con.Open();
Cmd.ExecuteNonQuery();
Con.Close();
```

معرفی خاصیت ها و دو کد نمونه Placeholder ها :متغیرهایی هستند که در یک دستور Sql قرار می گیرندو می توانند در زمان اجرای برنامه جای خود را با عبارتی خاص عوض کنند این متغیرها با علامت @ در یک دستور مشخص می شوند . و هنگامی که از آنها در یک دستور Sql استفاده کنیم قبل از اجرای دستور باید تمامی آنها را با مقادیر مناسب تعویض کنیم . که این کار به صورت اتوماتیک توسط برنامه در زمان اجرای دستور انجام می شود .اما باید پارامترهایی را ایجاد کرده و آن را در لیست parameters در شئ ایجاد شده از کلاس SqlCommand قرار دهیم تا برنامه بداند هنگام اجرای دستور هر placeholder را باید با مقدار چه متغیری در برنامه عوض کند . نکته – هیچ ضرورتی ندارد که نام یک placeholder همنام فیلدی باشد که قرار است مقدار placeholder در آن قرار بگیرد .خاصیت parameters کلاس SqlCommand :برای دسترسی به لیست پارامترهایی که در یک شئ از کلاس SqlCommand وجود دارد می توانیم از خاصیت parameters در این کلاس استفاده کنیم .

این خاصیت حاوی لیستی از placeholder ها به همراه متغیرهای وابسته به آنها است بنابراین در کد قبل از اجرای دستور ، باید به وسیله ی این لیست مشخص کنیم که هر placeholder با مقدار چه متغیری باید تعویض شود .مثالی از درج رکورد در بانک اطلاعات : در این کد می خواهیم در جدول student از بانک university ، یک رکورد اضافه کنیم این جدول شامل سه فیلد می باشد که اطلاعات آن توسط کاربر د ر TextBox ها وارد می شود و برنامه با گرفتن این اطلاعات آنها را در جدول بانک ذخیره می کند.

```
Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security = True"; SqlConnection Con = New SqlConnection(Strcon);

SqlCommand Cmd = New SqlCommand();

Cmd.Connection = Con;

Cmd.CommandText= " insert into student
id=@id,lname=@lname,fname=@fname";
Cmd.Parameters.AddWithValue("@id",TextBox١.Text);

Cmd.Parameters.AddWithValue("@lname",TextBox٢.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox٣.Text);

Con.Open();

Cmd.ExecuteNonQuery();

Con.Close();
```

توضیح کد بالا : در خط ۱ یک متغیر از نوع String تعریف کردیم و راطلاعات اتصال به بانک را در آن قرار دادیم (connectinString) .خط ۲ : شیئی از نوع SqlConnection ساختیم و رشته StrCon را به عنوان پارامتر به آن فرستادیم .خط ۳ : شیئی از نوع SqlCommand ساختیم .خط ۴ : خاصیت Connection کلاس SqlCommand را برابر شیئ ساخته شده از کلاس SqlConnection قرار دادیم .خط ۵ : دستور اجرایی Sql را به شیئ SqlCommand نسبت دادیم (با استفاده از خاصیت CommandText).خط ۶ ، ۷ ، ۸ : با استفاده

از خاصیت Parameters کلاس SqlCommand به placeholder ها مقدار دادیم .خط ۹ : اتصال به بانک را برقرار می کنیم .خط ۱۰ : متد اجرایی ExecuetNonQuery را اجرا می کنیم .خط ۱۱ : اتصال برنامه با بانک را قطع می کنیم .مثالی از اصلاح (update) اطلاعات یک رکورد :حالا می خواهیم اطلاعات یک رکورد از جدول student را اصلاح کنیم و تغییرات را ثبت نماییم . برای این منظور جدول مورد نظر دارای یک کلید است که می توان با استفاده از آن به تمامی اطلاعات رکورد موردنظر دسترسی پیدا کرد . در این جدول فیلد id (شماره دانشجویی) کلید جدول است .

```
Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security = True";

SqlConnection Con = New SqlConnection(Strcon);
SqlCommand Cmd = New SqlCommand();
Cmd.Connection = Con;

Cmd.CommandText= "update student  lname=@lname,fname=@fname where id=@id";

Cmd.Parameters.AddWithValue("@id",TextBox۱.Text);

Cmd.Parameters.AddWithValue("@lname",TextBox۲.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox۳.Text);

Con.Open();

Cmd.ExecuteNonQuery();
Con.Close();
```

در توضیح کد بالا باید این رو بگم که تمام مراحل آن مانند کد insert می باشد به جز دستور Sql که در خط ۵ آمده است و در اینجا update می باشد .

کلاس SqlDataAdapter :این کلاس در برنامه های بانک اطلاعاتی ، همانند پلی بین جداول اطلاعاتی و نیز داده های موجود در حافظه که به وسیله ی DataSet نگهداری می شوند ، عمل می کنند . و برای استفاده از آن در برنامه باید یک شیئ از نوع آن ساخته شود .

```
SqlDataAdapter da = New SqlDataAdapter ();
```

این کلاس برای دسترسی به بانک اطلاعاتی از شیئ ایجاد شده از کلاس SqlCommand ای که به آن نسبت داده می شود استفاده می کند .و برای دسترسی به بانک اطلاعات از کلاس SqlCommand و SqlConnection استفاده می کند .

```
da . SelectCommand = New SqlCommand();
```

خاصیت SelectCommand :

کلاس SqlDataAdapter دارای خاصیتی این خاصیت است .خاصیت SelectCommand حاوی شیئی از نوع SqlCommand است که از دستور موجود در آن شیئ برای دریافت داده های موردنیاز در برنامه از بانک اطلاعاتی به کار می رود یعنی SqlDataAdapter ، دستوری را که در خاصیت SqlCommand نگه‌داری می شود را روی بانک اطلاعاتی اجرا کرده و نتایج آن را در کلاس هایی مانند DataSet و یا DataTable قرار می دهد تا در برنامه مورد استفاده قرار گیرند .علاوه بر این ، کلاس SqlDataAdapter دارای خاصیت هایی به نام InsertCommand،DeleteCommand و UpdatCommand است که به هر یک شیئی از نوع SqlCommand را قبول می کنند و DataAdapter از دستور ذخیره شده در هر یک از آنها به ترتیب بای حذف ، درج و ویرایش داده ها در بانک اطلاعاتی استفاده می کند.

* هنگامی که بخواهید با استفاده از کلاس DataAdapter اطلاعات مورد نیاز خود را از یک بانک اطلاعاتی دریافت کنید ابتدا باید خاصیت SelectCommand را در DataAdapter تنظیم کنید

*خاصیت SelectCommand شیئی از نوع SqlCommand دریافت کرده که این شیئ مشخص می کند داده ها چگونه باید از بانک اطلاعات انتخاب شده و نیز چه داده هایی باید انتخاب شوند .

*اشیاء از نوع SqlCommand نیز دارای خاصیتهایی هستند که قبل از استفاده باید آنها را تنظیم کرد این خاصیت ها عبارتند از :

- Connection : یک شیئ از کلاس SqlConnection در این قسمت قرار گرفته و نحوه ی اتصال به بانک اطلاعاتی را مشخص می کند .

```
da.SelectCommand.Connection = Con;
```

- CommandText : دستور Sql و یا پروسیجر ذخیره شده در بانک اطلاعاتی که باید توسط این شیئ اجرا شود ، در این قسمت ذخیره می شود .

```
da.SelectCommand.CommandText = "select fields from table ";
```

توضیح کد بالا :در اینجا نوع دستور select می باشد و منظور از fields ، نام فیلدهایی است که می خواهیم اطلاعات آنها را استخراج کنیم اگر بخواهیم همه ی آنها را استخراج کنیم از * استفاده می کنیم و منظور از table نیز نام جدولی است که می خواهیم اطلاعات را از آن استخراج کنیم . نمونه کد :برای مثال می خواهیم اطلاعات فیلدهای نام ، نام خانوادگی و شماره دانشجویی از جدول دانشجو را استخراج کنیم و برای کارمورد نظر استفاده کنیم (پس ما در اینجا فقط اطلاعات را استخراج می کنیم).

```
SqlConnection Con = New SqlConnection(connectionstring) ;
SqlDataAdapter da = New SqlDataAdapter ();
da.SelectCommand.Connection = Con ;
da.SelectCommand.CommandText" = select fname,lname,id  from student ;"
```


s.ali.taher@gmail.com

دسترسی اطلاعات و اتصال داده ها

دسترسی به اطلاعات :

در ویژوال C# برای دسترسی به اطلاعات و نمایش آنها سه کامپوننت مهم و اصلی وجود دارند که عبارتند از :
Binding Source، BindingNavigator، Table Adapter، DataSet .

* کامپوننتهای BindingNavigator، Binding Source و DataSetرا می توانید در قسمت Data جعبه ابزار ببینید .

* کامپوننت TableAdapter نیز بر اساس مسیری که برای دسترسی به اطلاعات درون بانک اطلاعاتی و نمایش آنها طی می کنیم به صورت اتوماتیک ایجاد می شود . در ادامه توضیح مختصری در مورد کامپوننتهای مطرح شده با هم مرور می کنیم .

کامپوننت DataSet :

مانند یک موتور اطلاعاتی کوچک عمل می کند با استفاده از DataSet ابتدا به بانک وصل می شویم اطلاعات مورد نیاز را در حافظه DataSet قرار می دهیم سپس ارتباط با بانک را قطع می کنیم از این پس هر تغییراتی که خواستیم می توانیم بر روی اطلاعات درون DataSet اعمال کنیم سپس در آخر تمام تغییرات را بر روی بانک اطلاعاتی اعمال کنیم .

- با استفاده از این کامپوننت اطلاعات درون جداولی نگهداری می شوند و با استفاده از کامپوننت DataView به چندین روش پرس و جوهایی را روی داده ها انجام داد .

کامپوننت DataGridView :

این کنترل برای نمایش داده های موجود در یک بانک اطلاعاتی در فرم برنامه به کار می رود.برای کار با آن کافی است آن را به منبع داده های خود ، مثلا یکی از جدولهای موجود در بانک اطلاعاتی متصل کرده و سپس این کنترل را تنظیم کنیم تا دادهای جدول مورد نظر همانند یک جدول نمایش دهد (ستونهای این جدول نام فیلدها و ردیفهای آن اطلاعات مربوط به فیلدها که هر کدام در یک رکورد نگهداری می شوند) .

- علاوه بر این به وسیله این کنترل می توانید عنوان ستونهای داده ها و یا نوع نمایش آنها را نیز بدلخواه تعیین کنیم

کامپوننت BindingSource :

این کنترل همانند پلی برای ایجاد ارتباط بین داده های موجود در منبع داده ای شما (مانند DataSet) و کنترل هایی که برای نمایش داده ها مورد استفاده قرار می گیرند(مانند TextBox) به کار می رود .
بنابراین هنگامی که بخواهید به وسیله ی کنترل هایی و یا به هر دلیل دیگری بخواهید به آنها د منبع اطلاعاتی دسترسی داشته باشید ، این ارتباط باید از طریق این کامپوننت صورت بگیرد .

کامپوننت BindingNavigator :

این کنترل یک رابط گرافیکی استاندارد برای حرکت بین رکوردهای موجود در یک بانک اطلاعاتی ایجاد می کند.
همچنین مانند کنترل DataGridView می تواند به کنترل BindingSource متصل شده و از طریق آن به داده های موجود در برنامه دسترسی داشته باشد .

کامپوننت TableAdapter :

این کامپوننت در جعبه ابزار وجود ندارد بلکه با توجه به روشی که کامپوننت های داده ای دیگر را در برنامه قرار داده و آنها را تنظیم می کنید و به صورت اتوماتیک ایجاد می شود

- این کامپوننت حاوی پرس و جوهایی برای انتخاب داده های موجود در بانک اطلاعاتی و نیز اطلاعاتی در مورد نحوه اتصال برنامه به بانک است .

- همچنین حاوی متدهایی است که به وسیله آنها می توان داده ها را از جداول بانک اطلاعاتی بدست آورد و در کامپوننت هایی مانند DataSet قرار داد و سپس در برنامه از آن داده ها استفاده کرد .

- این کامپوننت این قابلیت را دارد که بر اساس دستور Select ای که برای انتخاب داده ها از بانک اطلاعاتی برای آن وارد می کنید دستورات Insert، Update و نیز Delete مناسب برای تغییر داده های انتخاب شده در بانک اطلاعاتی ایجاد کند .

* اتصال داده ها :

اتصال داده یعنی اینکه داده های را که به وسیله ی کامپوننت BindingSource به آنها دسترسی دارید را به یک کنترل خاص نسبت دهید (مثلا به یک DataGridView یا TextBox ، ...) .

به عبارت دیگر یک کنترل را بتوانید به نحوی تنظیم کنید که داده های مورد نیاز خود را بوسیله کامپوننت های دسترسی داده ها در برنامه دریافت کند و سپس آنها را به صورت اتوماتیک به کاربر نمایش دهد.

- در C# بعضی از کنترل ها وجود دارند که مخصوص این کار طراحی شده اند مانند کنترل DataGridView و یا TextBox .


Microsoft
ASP.net

ASP چیست ؟**ASP▪ یک تکنولوژی مبتنی بر سرویس دهنده بوده که امکان اجرای اسکریپت ها ی موجود در یک صفحه وب را از طریق یک سرویس دهنده اینترنت فراهم می نماید .**

ASP▪ تکنولوژی متعلق به شرکت ما کروسافت است .

ASP▪ از کلمات Page Server Active مشتق شده است .

ASP▪ برنامه ای است که با مدیریت IIS اجراء می گردد.

▪یک فایل ASP مشابه فایل Html است .

▪محتویات یک فایل ASP شامل : متن ، Html، XML و اسکریپت است .

▪اسکریپت های موجود در یک فایل ASP بر روی سرویس دهنده اجراء می گردنند.

▪فایل های ASP دارای انشعاب asp می باشند.

▪ پس از درخواست فایل های ASP توسط کاربران ، در ابتدا محتویات (اسکریپت ها) مربوطه بر روی سرویس دهنده اجراء و در ادامه نتایج بصورت تگ های Html برای کاربر ارسال خواهد شد.

آشنائی اولیه با ASP کلاسیک می تواند دارای جنبه های مثبتی از بعد فراگیری ASP.NET با شد .

ASP+ همان ASP.NET است . زمانیکه ما کروسافت ASP.NET را طراحی نمود ،در ابتدا از نام فوق استفاده گردید .

ASP.NET چیست ؟**ASP نسخه شماره سه ، آخرین نسخه در این زمینه بوده و ما هرگز شاهد عرضه نسخه شماره چهار محصول فوق نخواهیم بود. ASP.NET نسل جدید ASP است و نمی توان ادعا نمود که ASP.NET نسخه ارتقاء یافته ASP کلاسیک است .**

ASP.NET یک نمونه و نگرش جدید به برنامه ها و اسکریپت های مبتنی بر سرویس دهنده می باشد.

ASP.NET عضوی از فریمورک جدید دات نت شرکت ما کروسافت بوده و سه سال زمان صرف نوشتن آن شده است . ASP.NET با نسخه ASPکلاسیک کاملا" سازگار نیست .

دات نت فریمورک دات نت فریمورک، زیر ساخت پلات فورم جدید دات نت است . فریمورک فوق ، یک محیط عمومی برای ایجاد ، بکارگیری و اجرای برنامه های تحت وب و سرویس های وب را فراهم می نماید . دات نت فریمورک شامل دو بخش اساسی است :

▪Common language runtime

▪Common class libraries نظیر : ADO.NET,ASP.NET و Windows Forms

دات نت فریمورک ، با الهام از امکانات دو بخش فوق ، قادر به ارائه سرویس ها و خدمات متفاوت به مجموعه وسیعی از سیستم های کامپیوتری است . دات نت فریمورک از زبانهای متعددی حمایت می نماید .
C++,C#,VB,Jscript نمونه هائی در این زمینه می باشند.

تفاوت های ASP کلاسیک و ASP.NET **ASP.NET▪ زبان های متعددی را حمایت می نماید .**
ASP.NET▪ مجموعه گسترده ای از کنترل های جدید و عناصر مبتنی بر XML را ارائه می دهد.
ASP.NET▪ قابلیت اعتبارسنجی کاربران با توانائی بالا را دارا است .
▪افزایش کارائی سیستم از طریق اجرای کدهای کمپایل شده (نه تفسیر شده !)
▪کدهای ASP.NET بطور کامل با نسخه ASP کلاسیک سازگار نمی باشند.
برخی از ویژگی های جدید در ASP.NET :

- **تعداد بالای زبانهای برنامه رویسی حمایت شده**
- **کنترل های قابل برنامه نویسی**
- **برنامه نویسی مبتنی بر Event**
- **استفاده از عناصر مبتنی بر Xml**
- **اعتبار سنجی کاربران با استفاده از Account و قوانین**
- **افزایش کارائی با توجه به کمپایل نمودن کدها**
- **پیکربندی و بکارگیری آسان**

ASP.NET دارای مجموعه ای وسیع از کنترل های HTML است . اکثر عناصر موجود بر روی یک صفحه وب را می توان بعنوان یک شی قابل کنترل ASP.NET در نظر گرفت .رفتار اشیاء فوق را می توان با استفاده از اسکریپت ها ، کنترل و هدایت نمود. ASP.NET همچنین دارای مجموعه ای از کنترل های ورودی شی گراء نظیر : ListBox های قابل برنامه نویسی و کنترل های بررسی صحت د اده های ورودی است . با استفاده از یک کنترل جدید (Data Grid) عملیات ذخیره سازی ،مرتب سازی وسایر عملیات مورد نیاز در رابطه با بانک های اطلاعاتی را پشتیبانی و حمایت می نماید. تمامی اشیاء ASP.NET بر روی یک صفحه وب می توانند باعث بروز یک "رویداد" گردند. در چنین مواردی کدهای نوشته شده در ASP.NET ، مجری سیاست برخورد با رویداد بوجود آمده خواهند بود . عناصر استفاده شده در ASP.NET عموما" متکی بر XML می باشند . نظیر کنترل Adrotator که از Xml برای ذخیره ساری اطلاعات و تنظیمات مربوط به آگهی ها استفاده می نماید. ASP.NET ، اعتبارسنجی کاربران متکی بر فرم را با استفاده از تکنولوژی هائی نظیر : مدیریت کوکی، تغییر مسیر کاربرانی که هویت آنها تایید نشده و ... انجام می دهد. به محض درخواست

یک صفحه ASP.NET ،صفحه مورد نظر ترجمه (کمپایل) و یک نسخه از آن در حافظه باقی خواهد ماند (Cached) . بدیهی است که در این حالت افزایش چشمگیری را از بعد کارآئی خواهیم داشت .ASP.NET با نسخه قبلی خود (ASP کلاسیک) کاملا" سازگار نیست .بنابراین در برخی از کدهای نوشته شده با ASP کلاسیک می بایست تغییراتی را اعمال نمود. فایل های ASP.NET دارای انشعاب aspx می باشند. بدین ترتیب می توان بر روی یک سرویس دهنده فایل های ASP.NET (فایل های با انشعاب aspx) و فایل های ASP کلاسیک (فایل های با انشعاب asp) بطور همزمان استفاده کرد.

نصب ASP.NET برای نصب ASP.NET به امکانات زیر نیاز خواهد بود :

- یکدستگاه کامپیوتر با قابلیت اجرای ویندوز بر روی آن .
- نصب یکی از نسخه های ویندوز ۲۰۰۰ و یا XP
- نصب برنامه IIS
- قبل از نصب ASP.NET می بایست تمامی Service Pack های مربوط به محصولات نرم افزاری که قصد نصب آنها را داشته باشیم ،آماده نمود.
- در صورتیکه قبلا" نسخه Beta مربوط به ASP.NET بر روی سیستم نصب شده باشد ،لازم است که در ابتدا نسخه فوق از روی سیستم برداشته گردد.
- نصب .NET ، قبل از نصب می بایست کیت کامل محصول فوق را تهیه نمود. کیت فوق حدودا" ۱۳۰ مگابایت بوده و می توان آن را از سایت ماکروسافت و یا سایر سایت ها نظیر : www.asp.net تهیه کرد .

صفحات وب ASP.NET

یک صفحه ASP.NET ، در اولین نگاه ،مشابه یک صفحه Html است . برای آشنائی با صفحات ASP.NET ، یک صفحه ساده Html را ایجاد که مسئولیت آن نمایش یک پیام خاص در خروجی باشد. (Hello.html)

مثال : یک صفحه ساده Html

```
<html>
<body bgcolor="Blue">
<center>
<h۲>Hello HTML Tags </h۲>
</center>
</body>
</html>
```

در صورتیکه بخواهیم فایل فوق را به یک صفحه ASP.NET تبدیل نمائیم ،کافی است محتویات فایل فوق را در فایل جداگانه ای قرار داده و فایل فوق را با نام دلخواه و انشعاب aspx. ذخیره نمائیم .در این حالت محتویات صفحه ASP.NET بصورت زیر خواهد بود :

مثال : یک صفحه ساده ASP.NET

```
<html>
<body bgcolor="Blue">
<center>
<h۲>Hello ASP.NET </h۲>
</center>
</body>
</html>
```

یک صفحه ASP.NET از برخی جهات دارای عملکردی مشابه صفحات HTML است . صفحات HTML دارای انشعاب Html بوده و زمانیکه مرورگر درخواست یک صفحه Html را از سرویس دهنده داشته باشد ،سرویس دهنده بدون انجام هیچگونه اصلاحات و یا عملیات خاصی، صرفا" صفحه مورد نظر را برای مرورگر ارسال خواهد نمود. یک صفحه ASP.NET دارای انشعاب .aspx بوده و زمانیکه مرورگر درخواست یک صفحه ASP.NET را از سرویس دهنده داشته باشد ،سرویس دهنده دستورالعمل های اجرائی موجود در صفحه را پردازش و در ادمه نتایج بدست آمده ، برای سرویس گیرنده (مرورگر) ارسال خواهند شد.

در مثالی که ارائه گردید ما شاهد حضور و استفاده از دستورالعمل های اجرائی در فایل ASP.NET نبودیم . در ادامه با افزودن دستورالعمل های اجرائی به بررسی تفاوت های موجود بین صفحات ایستای Html و صفحات پویای ASP خواهیم پرداخت . **ASP کلاسیک** برنامه نویسان وب چندین سال است که از ASP کلاسیک استفاده می نمایند .ASP.NET با ASP کلاسیک کاملا" سازگار نبوده و اغلب صفحات ASP کلاسیک ، با اعمال تغییراتی اندک قادر به استفاده بر روی بستر دات نت و مشابه صفحات ASP.NET خواهند بود.

صفحات پویا در ASP کلاسیک بمنظور بررسی توانائی صفحات ASP در جهت نمایش اطلاعات پویا ،مثال زیر را در نظر بگیرید.مثال : یک صفحه ساده ASP کلاسیک بمنظور نمایش اطلاعات پویا

```
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello ASP Classic!</h۲>
<p><%Response.Write(now())%></p>
</center>
</body>
</html>
```

کدهای محصور بین <٪ ٪> توسط سرویس دهنده اجراء می گردند. Response.Write ، امکان نوشتن اطلاعات در خروجی را فراهم می نماید . ( )NOW تابعی است که زمان و تاریخ سرویس دهنده را برمی گرداند.

صفحات پویا در ASP.NET **کدهای زیر همان مثال قبلی با رگرش ASP.NET است**

مثال : یک صفحه ساده ASP.NET بمنظور نمایش اطلاعات پویا

```
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello ASP.NET!</h۲>
<p> <%Response.Write(now())%></p>
</center>
</body>
</html>
```

در دو مثال ارائه شده ،اختلاف بین صفحات ASP.NET و ASP کلاسیک مشهود نیست . بمنظور بررسی تفاوت های موجود، بین ASP کلاسیک و ASP.NET مثال زیر را در نظر بگیرید.

مثال : یک صفحه ساده ASP کلاسیک بمنظور نمایش اطلاعات پویا

```
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello ASP.NET!</h۲>
<p> <%Response.Write(now())%></p>
</center>
</body>
</html>
```

کدهای نوشته شده (به رنگ قرمز که اصطلاحا" Render Block) نامیده می شوند،در ابتدا بر روی سرویس دهنده اجراء و در ادامه نتایج برای سرویس گیرنده (مرورگر) ارسال می گردد. مسئله فوق می تواند نشاندهنده یکی از محدودیت ASP کلاسیک باشد. در این نوع صفحات Block Render ها می بایست در محلی قرار بگیرند که می خواهیم در خروجی نمایش داده شوند. با استفاده از ASP کلاسیک نمی توان کدهای اجرائی را از تگ های Html جدا نمود. بدیهی است در چنین مواردی خوانائی و پشتیبانی صفحات به شدت زیر سوال خواهد رفت .

کنترل های سرویس دهنده ASP.NET

ASP.NET بنوعی مسئله " کدهای نوشته شده به سبک اسپاگتی ! " را با بکارگیری کنترل های سرویس دهنده برطرف نموده است . کنترل های سرویس دهنده ، تگ هائی هستند که توسط سرویس دهنده قابل درک و فهم می باشند. سه نوع کنترل های سرویس دهنده وجود دارد :

- کنترل های Html سرویس دهنده (تگ های سنتی Html)
- کنترل های وب سرویس دهنده (تگ های جدید ASP.NET)
- کنترل های اعتبارستجی سرویس دهنده (تگ های مسئول اعتبار سنجی داده های ورودی)

کنترل های Html سرویس دهنده این نوع کنترل ها ، همان تگ های استاندارد Html بوده با تفاوت که می بایست از خصلت : Runat=Server بهمراه تگ مربوط به آنان ، نیز استفاده گردد . مثال زیر نحوه جداسازی کد های اجرائی از تگ های Html ،با استفاده از کنترل های Html سرویس دهنده را نمایش می دهد .

مثال : یک صفحه ساده ASP.NET و استفاده از HTML Controls Server

```
<%TimeStamp.InnerText=now()%>
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello HTML Server Controls </h۲>
<p id="TimeStamp" runat="server"></p>
</center>
</body>
</html>
```

خصلت Runat=server بهمراه تگ <P> باعث شده که تگ فوق ، بعنوان یک کنترل سرویس دهنده ایفای وظیفه نماید. در این حالت مجموعه کدهای اجرائی به محلی خارج از محدوده تگ های Html منتقل شده اند.

کنترل های وب سرویس دهنده این نوع کنترل ها مشابه کنترل های Html با پیچیدگی بیشتری می باشند. این نوع کنترل ها هرگز بعنوان یک بخش وابسته و بصورت یک خصلت در کنار تگ های Html استفاده نخواهند گردید ، بلکه برای حضور خود دارای ماهیتی کاملا" مستقل هستند. از این نوع کنترل ها در اغلب برنامه هائی که سیاست ارتباط دوسویه با کاربران دنبال می گردد ، استفاده می شود. فرم های ورودی ، نمونه ئی مناسب از جایگاه استفاده از این نوع کنترل ها می باشند. کنترل های فوق همواره با تگی شروع خواهند شد که ابتدای آن مزین به واژه : < asp:> است . برنامه زیر نحوه استفاده از کنترل های وب سرویس دهنده را نشان می دهد.

مثال : یک صفحه ساده ASP.NET و استفاده از Web Server Controls

```
<%TimeStamp.Text=now() %>
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello Web Server Controls </h۲>
<p><asp:label id="TimeStamp" runat="server"
/></p>
</center>
</body>
</html>
```

در مثال فوق از یکی از کنترل های وب سرویس دهنده ، با نام label استفاده شده است . کنترل کننده فوق یکی از ده ها کنترل از قبل تعریف شده در این زمینه بوده که برای ASP.NET قابل فهم است . **کنترل های اعتبارستجی سرویس دهنده**با استفاده و بکارگیری کنترل های تعیین صحت داده ها ، می توان عملیات مربوط به بررسی صحت داده های ورودی توسط کنترل های ورودی سرویس دهنده نظیر : TextBox ، انجام داد. در چنین مواردی زمانی که داده وارد شده متناسب با سیاست و قانون تعریف شده از قبل نباشد ، می توان پیام مناسبی را نمایش داد. بصورت پیش فرض عملیات موسوم به بررسی صحت داده ها در یک صفحه (Page Validation) همزمان با فشردن یک Button نظیر : کنترل های ImageButton و یا LinkButton صورت می پذیرد. **رویدادها در دات نت** در صفحات ASP.NET می توان مجموعه کدهائی را نوشت که در صورت بروز یک شرط خاص و یا تحقق یک شرایط ویژه (وقفه) فعال و خدمات تعریف شده خود را ارائه نمایند. در مثال های قبل ، با کد های زیر آشنا شدیم .

مثال : یک صفحه ساده ASP.NET و استفاده از Controls Web Server

```
<% TimeStamp.Text=now()%>
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello W۳Schools!</h۲>
<asp:label id="TimeStamp" runat="server" />
</center>
</body>
</html>
```

چگونه می توان اطمینان پیدا نمود که Render Block (کدهای قرمز رنگ) اجراء خواهد شد. بمنظور اطمینان از اجرای کدهای مورد نظر در زمان مورد نظر ، می بایست یک Event Handler را اضافه نمائیم.

مثال : یک صفحه ساده ASP.NET و استفاده از Controls Web Server و رویداد Page_Load

```
<script runat="server">
Sub Page_Load(Sender As Object,E As EventArgs)
TimeStamp.Text=now()
End Sub
</script>
<html>
<body bgcolor="yellow">
<center>
<h۲>Hello W۳Schools!</h۲>
<asp:label id="TimeStamp" runat="server" />
</center>
</body>
</html>
```

Event Handler ، روتینی است که مسئول اجرای کدهائی به ازای تحقق یک رویداد است.رویداد Page_Load یکی از ده ها Event قابل فهم توسط ASP.NET است .

مرجع :

- **کنترل های Html Server** : تگ های مبتنی بر HTML که توسط سرویس دهنده قابل فهم می باشند . تمام کنترل های فوق می بایست بهمراه تگ مربوطه که دارای خصلت runat=server است ، استفاده گردند .

Description	**HTML Server Control**
Controls an <a> HTML element	HtmlAnchor
Controls a <button> HTML element	HtmlButton
Controls a <form> HTML element	HtmlForm
Controls other HTML element not specified by a specific HTML server control, like <body>, , , etc.	HtmlGeneric

Controls an <image> HTML element	HtmlImage
Controls <input type="button">, <input type="submit">, and <input type="reset"> HTML elements	HtmlInputButton
Controls an <input type="checkbox"> HTML element	HtmlInputCheckBox
Controls an <input type="file"> HTML element	HtmlInputFile
Controls an <input type="hidden"> HTML element	HtmlInputHidden
Controls an <input type="image"> HTML element	HtmlInputImage
Controls an <input type="radio"> HTML element	HtmlInputRadioButton
Controls <input type="text"> and <input type="password"> HTML elements	HtmlInputText
Controls a <select> HTML element	HtmlSelect
Controls a <table> HTML element	HtmlTable
Controls <td>and <th> HTML elements	HtmlTableCell
Controls a <tr> HTML element	HtmlTableRow
Controls a <textarea> HTML element	HtmlTextArea

کنترل های Web Server : تگ های خاصی از ASP.NET بوده که توسط سرویس دهنده قابل فهم می باشند . گرامر بکارگیری کنترل های فوق بصورت زیر است :

```
<asp:control_name id="some_id"  runat="server" />
```

Description	**Web Server Control**
Displays a sequence of images	AdRotator
Displays a push button	Button
Displays a calendar	Calendar
Displays a check box	CheckBox
Creates a multi-selection check box group	CheckBoxList
Displays fields of a data source in a grid	DataGrid
Displays items from a data source by using	DataList

templates	
Creates a drop-down list	DropDownList
Creates a hyperlink	HyperLink
Displays an image	Image
Displays a clickable image	ImageButton
Displays static content which is programmable (lets you apply styles to its content)	Label
Creates a hyperlink button	LinkButton
Creates a single- or multi-selection drop-down list	ListBox
Displays static content which is programmable (does not let you apply styles to its content)	Literal
Provides a container for other controls	Panel
Reserves space for controls added by code	PlaceHolder
Creates a radio button	RadioButton
Creates a group of radio buttons	RadioButtonList
	Repeater
Creates a table	Table
Creates a table cell	TableCell
Creates a table row	TableRow
Creates a text box	TextBox
Displays an XML file or the results of an XSL transform	Xml

- **کنترل های Validation Server** : از کنترل های فوق بمنظور بررسی صحت داده های ورودی توسط کاربر استفاده می گردد . گرامر بکارگیری کنترل های فوق بصورت زیر است :

```
<asp:control_name id="some_id" runat="server" />
```

Validation Server Control	Description
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value
CustomValidator	Allows you to write a method to handle the validation of the value entered
RangeValidator	Checks that the user enters a value that falls between two values

Ensures that the value of an input control matches a specified pattern	RegularExpressionValidator
Makes an input control a required field	RequiredFieldValidator
Displays a report of all validation errors occurred in a Web page	ValidationSummary

اجزای تشکیل دهنده یک صفحه وب دارای خصوصیتی مشترک بوده و آن کد تولید کننده آن صفحات می باشد که البته به زبان خاصی نوشته می شوند این زبان Hypertext Markup Language یا همانHTML می باشد. که مختصراً در خصوص آنها توضیحاتی ارائه می شود.

Hypertext : در لغت به معنای ابرمتن یا فوق متن تلقی می گردد ، و منظور از آن مجموعه متنی است که وظم با متون خاص شده باشدکه این متون خاص می توانند لینکها یا ارتباطی باشند که با دیگر صفحات در ارتباط است

Markup : این کلمه نیز در لغت به معنای نشانه گذاری می باشد و شاید علت استفاده از آن قابلیت نشانه گذاری و ایجاد ارتباط بین صفحات توسط همان ابرمتن ها باشد .

Language : به معنای زبان بوده و در این عبارت معرف وجود زبانی با قابلیت نشانه گذاری توسط متن ها و فوق متن ها می باشد .

عناصر تشکیل دهندهHTML :

HTML دارای عناصر مختلفی می باشد که از آن جمله می توان به موارد زیر اشاره کرد :

۱ -فرم : **فرم ها همان عناصر اولیه ی ا صفحاتی هستند که کاربر با آنها در ارتباط می باشد .**

۲ -متن ها : **عناصر پر مصرف و ساده که عمده صفحات وب را شامل می شود .**

۳ -لینک ها (ابرمتن ها) : **این عناصر متونی هستند که قابلیت اتصال به یک فرم یا آدرس دیگر را دارند .**

۴ -تصاویر : **تصاویر در صفحات وب از اعضای اصلی بوده که می تواند جلوه ای ویژه در صفحات ایجاد نماید .**

۵ -جدول ها : **جداول نیز به منظور دسته بندی و ایجاد ساختار های متناسب نیز کاربرد ویژه ای در صفحات وب دارد .**

۶ -چند رسانه ای (مولتی مدیا) : **هر چند این عناصر در HTML اولیه موجود نبوده است ولی در نسخه های بعدی اضافه شده و می توان امکاناتی نظیر صوت و یا تصویر را در صفحات وب ایجاد کرد .**

۷ -عناصری نظیر کدهای فلش یا جاوا و یا عناصری نظیر استیل شیت ها نیز سایر عناصر **موجود در صفحات وب می باشند که از ذکر موردی آنها در اینجا خودداری می شود .**

شروع کار :

با توجه به ذکر مختصری از HTML حال برای آغاز چه باید کرد ؟ در پاسخ به این سوال به ابزارهایی جهت ایجاد صفحات وب نیاز داریم که عبارتند از :

یک محیط متن جهت ایجاد فایل HTML نظیر Notepad (این ساده ترین محیطی است که می توان بوسیله آن کد HTML را ایجاد کرد اما محیط هایی نظیر Microsoft FrontPage و یا سایر زبان های پشتیانی کننده از html نظیر Microsoft Visual Studio می تواندابن امکانات را فراهم کند) .

یک Web Browser نظیر Internet Explorer و یا Opera که بتوان صفحه ایجاد شده را در آن مشاهده نمود .

به این ترتیب می توان یک صفحه وب ساده را ایجاد نمود . در زیر یک کد ساده جهت ایجاد یک صفحه وب ذکر شده است که جزئیات آن را در ادامه توضیح می دهیم .

```
<html>

        <head>
<title>Page ۱</title>
        </head>


        <body>
<p>Welcome to HTML</p>
        </body>

</html>
```

نکته : نرم افزار Notepad را باز کرده و متن فوق را درون آن بنویسید سپس آن را با یک نام دلخواه و پسوند htm یا html ذخیره کنید . در ادامه بر روی فایل ذخیره شده را اجرا کنید تا نتیجه کار را ببینید .

ایجاد یک صفحه وب بوسیله html به سادگی فوق می باشد قابل ذکر است که به دستورات فوق در html تگ می گویند نظیر تگ <head> که البته تگها می توانند بصورت جفت و یا مستقل باشند نظیر تگ head که جفت است و به <head> تگ head باز و به تگ </head> تگ head بسته گفته می شود .

توجه : همه تگ ها در html در میان علامت <> قرار می گیرند .

اما جهت آنکه بتوان قابلیتهای بیشتری را به صفحات وب اعمال کنیم لازم است با سایر تگ ها آشنا شویم در ادامه توضیحاتی مختصر جهت معرفی این تگ ها ارائه می دهیم .

تگهای HTML

<html> </html>

این جفت تگ به عنوان آغاز و پایان کدنویسی html می باشد ، یعنی با تگ باز آن کدنویسی آغاز و با تگ پایان آن انتهای کدنویسی خواهد بود سایر کدها بایستی حتماً در میان این دو تگ قرار بگیرد .

<head> </head>

جفت تگ head به منظور مشخص کردن ناحیه ای به عنوان راس در صفحه وب بکار می روند در این ناحیه تگ های ویژه ای را می توان به کار برد که عمدتا درباره خصوصیاتی است که با صفحه مذکور در ارتباط می باشد. از آن جمله می توان به عنوان صفحه اشاره کرد .

<title> </title>

در میان این دو تگ هر عنوانی را که ذکر نمایید به عنوان نام صفحه وب شما تلقی می گردد . توجه شود این جفت تگ حتماً بایستی در ناحیه head و در میان آن جفت تگ قرار بگیرد .

<body> </body>

این جفت تگ بدنه اصلی صفحه وب را مشخص می کند یعنی هر آنچه در یک صفحه وب مشاهده می گردد حتماً باید در این قسمت معرفی شده باشند در ادامه سایر تگ ها مربوط به جزئیات درون این بخش می باشد .

عبارت درون این جفت تگ بصورت ذخیم (bold) نشان داده می شود

عبارت درون این جفت تگ به عنوان یک پاراگراف مشخص می شوند . این تگ دارای خصوصیتی است (align)که نوع چیدمان را در صفحه مشخص می کند که عبارتست از :

راست چین یا right

چپ چین یا left

وسط چین یا center

در مثال زیر از این خاصیت استفاده شده است .

 Welcome to HTML

به این ترتیب عبات Welcome to HTML در وسط صفحه وب نشان داده می شود .

<h۱> </h۱>

این جفت تگ وظیفه مشخص کردن سایز متنی را که درون آن است را به عهده دارند.

برای آنکه تغییرات این تگ را متوجه شوید مثال قبل را درون این جفت تگ قرار داده و سپس ص فحه مربوطه را ذخیره کنید .

```
<body>
        <h۱>
        <p>Welcome to HTML</p>
        </h۱>
</body>
```

ساختار بدنه (body) به شرح فوق تغییر می کند .

سایر سایزبندی در صفحات وب از شماره ۱ تا ۷ می باشد به شرح زیر :

<h۱> Welcome</h۱>

<h۲> Welcome </h۲>

<h۳> Welcome </h۳>

<h۴> Welcome </h۴>

<h۵> Welcome </h۵>

<h۶> Welcome </h۶>

<h۷> Welcome </h۷>

که خروجی آن به صورت مقابل است .

Welcome
Welcome
Welcome
Welcome
Welcome
Welcome
Welcome

<hr>

این تگ از جمله تگ های مستقل می باشد که وظیفه آن ایجاد یک خط افقی به منظور جداکردن مطالب فوق با مطالب بعدی می باشد . در مثال بعدی از این تگ استفاده شده است .

<marquee> </marquee>

عبارت درون این جفت تگ با توجه به تنظیمات مربوطه بصورت افقی درون صفحه حرکت می کند مثال زیر را ببینید : (تغییرات فقط درون بدنه است)

```
<body>
        <h۱>
        <p align="justify">Welcome to HTML</p>
        </h۱>
        <hr>
        <marquee> this is example </marquee>
        <hr>
</body>
```

این تگ دارای خصوصیات زیر می باشد :

" "=align محل قرار گرفتن متن را تعیین می کند که با کلمات top, middle, bottom مقدار دهی می شود.

" "=behavior این خصوصیت نحوه حرکت متن را کنترل می کند که آنرا برابر با scroll اگر قرار دهیم، متن بصورت متناوب از یکطرف صفحه وارد و از طرف دیگر خارج می شود و اگر برابر با alternate قرار دهیم ، متن از صفحه خارج نمی شود و در عرض مرورگر حرکت می کند، همچنین اگر برابر با slide باشد ، متن از یکطرف وارد صفحه شده و در طرف دیگرمتوقف می شود.

" "=bgcolor رنگ زمی نه آن تگ را مشخص می کند که یا نام رنگ یا کد هگز آنرا می نویسید.

" "=direction جهت ورود متن به صفحه را کنترل می کند و با کلمات left, right, top, down که از چپ ، راست، بالا و پایین می تواند وارد شود.

" "=hspace حاشیه چپ و راست را کم وزیاد می کند.

" "=loop تعداد چرخش متن را کنترل می کند.

" "=scrolldelay سرعت حرکت متن را تعیین می کند.

" "=vspace حاشیه بالا و پایین متن را مشخص می کند.

```
<body>
        <h٧>
        <p align="justify">Welcome to HTML</p>
        </h٧>
        <hr>
        <marquee behavior="alternate" direction="right" scrolldelay="۶۰">
                this is example
        </marquee>
        <hr>
</body>
```

<!-- -- >

این جفت تگ به منظور نوشتن یک عبارت توضیحی یا comment گذاری به کار می رود هیچگونه دخالتی در خروجی ندارد . در مثال بعدی از این تگ استفاده شده است .

---

 this is link of yahoo web site

جفت تگ فوق یک لینک را ایجاد می کنند ، بگونه ای که در مقابل herf=" " آدرس اینترنتی و درمیان دو تگ باز و بسته مشابه فوق متنی را که تمایل دارید نمایش داده شود . بدین ترتیب شما متنی را درون صفحه خود ایجاد کرده اید که به سایت یاهو لینک داده شده است .

```
<body>
        <hr>
        <a href="www.yahoo.com"> this is link of yahoo web site </a>
</body>
```

<img src="pic۰۱.jpg">

این تگ به منظور استفاده از عکس در صفحه وب می باشد که بایستی حتما قسمت src که مربوط به آدرس عکس می باشد مقداردهی شود در زیر یک نمونه مثال ساده ذکر شده است .

```
                                                        <body>
                </h٧><p align="justify">Welcome to HTML</p>    <h٧>
                                <marquee> this is example </marquee>
                                                        <hr>
                                                        <!--
                                        this is comment
                                                        -->
```

سایر خصوصیات این تگ به شرح زیر است :

Height, Width

این تگ خصوصیات دیگری هم دارد، یکی از آنها که کاربرد زیادی هم دارد خ صوصیات طول height و عرض width می باشد که توسط آنها اندازه یک عکس را داخل صفحه می توانید کنترل کنید. مقیاس اندازه گیری طول و عرض بر حسب پیکسل Pixel می باشد، اگر شما این خصوصیات را کنترل نکنید عکس در اندازه اصلی خود ظاهر می شود. بطور مثال شما می خواهید عکسی را وارد کنید که در صفحه باید به اندازه ۱۰۰×۱۰۰ فضا اشغال کند:

<img src="images/pic۰۱.jpg" height="۱۰۰" width="۱۰۰">

این نکته را در نظر داشته باشید که سرعت کامل شدن عکس یعنی download شدن آن در صفحه بستگی به حجم آن دارد نه اندازه عکس، پس می توانید اندازه های یک عکس را اضافه کنید ولی دقت کنید که کیفیت عکس خراب نشود.

Alignment

شما می توانید محل قرارگرفتن عکس را نسبت به عناصر اطراف خودش با خصوصیت " "=align تعیین کنید و می توانید کلمات left , right , top , middle , bottom را برای این خصوصیت بکار برید . مثلا بصورت زیر

<img src="images/pic۰۱.jpg" align="middle">

در ضمن اگر می خواهید خود عکس در وسط صفحه قرارگیرد باید از تگ قبل از تگ <img> استفاده کنید و بعد از آن را بنویسید:

<img src="images/pic۰۱.jpg">

Border

اگر مایل هستید برای عکس کادر بگذارید می توانید از خصوصیت border=" " استفاده کنید و آنرا برابر با یک عدد قرار دهید که هر چه این عدد بزرگتر باشد ، کادر دور عکس هم ضخیم تر است به صورت پیش فرض border="۰" است یعنی هیچ کادری مشاهده نمی شود.

Alt

تا حالا متوجه شدید که در یک وب سایت هنگامی که نشانگر موس بر روی یک عکس که قرار می گیرد یک کادر متنی کوچک باز شده بنام tool tip و اطلاعاتی را راجع به آن عکس می دهد؟ پس شما هم اینکار را انجام دهید، خصوصیت alt=" " را داخل تگ گذاشته و هر متنی را که جلویش بنویسید در صفحه هنگامی که موس بر روی عکس قرار گیرد ، زیرنشانگر نمایان می شود . بهتر است که شما همی شه این کار را انجام دهید چون مرورگرهایی وجود دارد که عکس را نمایش نمی دهد پس با اینکار بیننده از حضور آن عکس خبر دار می شود

<img src="images/pic۰۱.jpg" alt="this is a picture">

hspace, vspace

دو خصوصیت دیگر هست که حاشیه و فضای خالی دور عکس را کنترل می کند،" hspace=" "برای حاشیه چپ و راست عکس و" vspace=" "برای بالا و پایین. عددی را که وارد می کنید در مقیاس پیکسل است.

Image link

از این تگ برای آنکه بتوان یک عکس را دارای لینک کرد استفاده می شود یعنی هرگاه بیننده روی عکس کلیک کرد به صفحه ای دیگر هدایت شود. برای این منظور بایستی تگ عکس را در میان تگ لینک قرار دهید.

 <img src="images/pic۰۱.jpg">

<bdo> </bdo>

این تگ جهت رعایت نوشتاری از راست به چپ و یا از چپ به راست می باشد که برای ما فارسی زبانان بسیار با ارزش است ایت تگ دارای خصوصیت dir="" می باشد و با ltr یعنی از چپ به راست و rtl یعنی از راست به چپ مقدار دهی می شود.

برای مشخص کردن نوع یا شکل حروف ، رنگ و یا اندازه آنها از این تگ استفاده می کنیم که دارای خصوصیات زیر می باشد.

face=" " این خصوصیت نوع فونت را تعیین می کند که باید براپبا یکی از فونتهای استاندارد سیستم باشد .

size=" " اندازه هر حرف را تعیین می کند که از اعداد یک تا هفت با علامت مثبت به نشانه افزاینده و با علامت منفی به نشانه کاهش سایز استفاده می شود .

color=" " برای تعیین رنگ حروف از خاصیت استفاده می شود .

<big> </big>

<small> </small>

برای تعیین اندازه حروف در یک حد خاص از این تگها استفاده می شود

برای نوشتن حروف بصورت کج italic این تگ را بکار می بریم.

<u> </u>

این تگ خطی زیر کلمات خواهد کشیدو مخفف کلمه Underline است .

این تگ هم مانند <b> عمل می کند و متن را ذخیم تر نشان می دهد.

<acronym> </acronym>

کاربرد این تگ برای مختصرنویسی است یعنی اگر یک کلمه مانند HTML دارید با استفاده از آن می توانید کلمات کامل را در یک کادر کوچک به اسم tooltip قرار دهید بطوریکه وقتی نشانه گرموس روی آن کلمه باشد این کادر باز شده و آنها را نمایش می دهد.کلمات کامل را باید در خصوصیت " "=title قرار دهید.

<acronym title="Hypertext Markup Language"> HTML </acronym>

<del> </del>

هنگامی که بخواهید یک مطلب را حذف کنید از این تگ ا ستفاده می کنید و یک خطی روی متن بین آنها کشیده می شود که بطور معمول با تگ <ins> بکار می رود. این تگ دارای دو خصوصیت منحصر می باشد، " "=cite که می تواند آدرس یک فایل باشد تا توضیحی راجع به علت حذف آن مطلب بدهد و " "=date که تاریخ و زمان حذف را معین می کند.

لیست ها:

سه نوع لیست وجود دارد ، اول لیستهای با ترکیب منظم (Ordered list) دوم لیست ها با ترکیب نا منظم (Unordered list) و سومی لیستهای توصیفی (Definition list) . شاید این نوع نامگذاری بخاطر وجود اعداد یا حروف بترتیب در لیست منظم است که در دیگری فقط نقطه های توپر هست که ترتیبی را نمی توان برای آنها در نظر گرفت .

 list ۱

 list ۲

برای درست کردن لیست در یک صفحه از این تگها باید استفاده کنیم، بدین صورت که در آغاز قسمتی که می خواه یم لیست درست کنیم تگ را که مخفف ordered list می باشد را می نویسیم تا مرورگر بفهمد لیست از آنجا آغاز می شود و سپس در ابتدای هر گزینه از لیستمان تگ را اضافه می کنیم و در انتهای گزینه ها تگ پایان دهنده را می نویسیم.

Type : نوع شماره گذاری را مشخص می کند مثلا اعداد از ۱ به بعد یا حروف رومی و غیره

<ol type="i">

این تگ را Unordered list می نامند و مانند تگهای بالا برای درست کردن لیست در یک صفحه می باشد با این تفاوت که بجای حروف و اعداد از دایره یا مربع های توپر استفاده می کند که مانند بالا می توانید از خصوصیت type که با کلمات "disc", "circle", "square" مقدار دهی می شود استفاده کنید.

<ul type="disc">

<dl> <dt> <dd> </dl>

آخرین مدل، لیستهای توصیفی هستند که برای تعریف و توصیف یک کلمه بکار می روند که کلمه مشخص را با تگ <dt> و توصیف آنرا با <dd> بکار می برند.

```
<dl>
        <dt> Html
  <dd> Hypertext Markup Language.
</dl>
```

Anchor

یک نوع لینک هم هست که دو نقطه را در داخل یک صفحه بهم متصل می کند که به آن anchor می گویند. کاربردهای گوناگونی دارد ولی بیشترین کاربرد این نوع لینک برای مواقعی است که طول یک صفحه از سایت زیاد شده و در انتهای همان صفحه شما یک لینک می گذارید تا با کلیک کردن بر روی آن بیننده سایت شما به بالای همان صفحه هدایت شود . برای اینکار دو سری تگ لازم است که باید بنویسید، یکی جایی است که می خواهید لینک به آن متصل شود که باید آن نقطه از صفحه را نامگذاری کنید و این اسم باید داخل همان صفحه تک باشد یعنی در جای دیگر بکار نبرده باشید و تگ آن به این صورت است:

>"<a name="top

در بین دو علامت " " و به جای کلمه top هر اسمی می توانید بکار برید این نکته را در ذهن داشته باشید که این تگ درمرورگر ظاهر نمی شود .اما تگ دیگری که نیاز است خود کد لینک است و بجای نوش تن آدرس فایل در خصوصیت href شما نامی را که انتخاب کردید به اضافه علامت # را می نویسید:

top of the page

---

mailto

علاوه بر این لینکها شما می توانید برای آدرسهای ایمی ل هم لینک درست کنید که تگ آن به صورت زیر می باشد:

 my email

کلمه :mailto به مرورگر می فهماند که باید یک ایمی ل به آدرس بعد از آن فرستاده شود. البته این لینکها برای بیننده هایی که ایمی ل هایشان را با برنامه هایی مانند Outlook express چک می کنند ، مفید است چون وقتی که روی این نوع لینک کلیک شود برنامه پیش فرض مدیریت ایمی ل در سیستم عامل کاربر باز می شود پس برای بیننده هایی که آدرس ایمی ل یاهو دارند این کد مفید نیست بهتر است که آدرس کامل ایمی ل را نوشته و به صورت یک لینک درست کنید تا برای کلیه بینندگان سایت مفید باشد، مانند مثال زیر:

 my_email@domain.com

البته شما می توانید حتی موضوع و متن ایمی ل را تعیین کنید . اگر قصد همچین کاری را دارید پس باید بلافاصله بعد ازآدرس ایمی ل داخل تگ یک علامت سوال اضا فه کنید تا مرورگر بفهمد که این آدرس ادامه دارد و بعد از علامت سوال کلمه subject= را می نویسید که این کلمه نشانگر موضوع ایمی ل است و هرچیزی که جلوی آن نوشته شود به عنوان موضوع ایمی ل در برنامه مشخصه نمایان خواهد شد و اگر متن ایمی ل هم بخواهید نوشته شود بایی بعد از موضوعی که نوشتید علامت & را بگذارید وسپس کلمه body= را که نشانگر متن ایمی ل است و در جلوی علامت مساوی هر متنی را می توانید وارد کنید . به فرض می خواهید که ایمیل با موضوع Test و متن Hello به آدرس ایمیل فرستاده شود:

 my_email@domain.com

تگ جدول :

جدولها یکی از بهترین و مفیدترین عنصرها در صفحات وب می باشند، با استفاده از آنها ما می توانیم اطلاعات و عناصررا در یک صفحه سازمان دهی و مرتب کنیم . کلیه اطلاعات یا عناصر دیگر وب را می توانیم داخل ردیف ها یا ستون های یک جدول قرار دهیم بدون آنکه خطوط جدول مشخص باشد و یا در صورت نیاز خطوط آنها نمایان می شود. کمتر کسی را پیدا می کنید که از این عنصر استفاده نکند و زمانیکه تجربه کا فی برای طراحی یک سایت بدست آوردید به اهمی ت این عنصر پی خواهید برد .تگ مشخصه جدول <table></table> می باشد. اما برای اضافه کردن ردیف به یک جدول از <tr></tr>و برای ستون از تگ <td></td> استفاده می کنیم. پس ساختار کلی یک جدول بدین صورت می باشد:

```
<table>
   <tr>
  <td>this is a table۱.</td>
  <td>this is a table۲.</td>
   </tr>
</table>
```

هر چه تگهای ردیف و ستون بیشتر باشد به همان اندازه ما در جدولمان خانه خواهیم داشت . مجموعه کد بالا نشان دهنده یک جدول با یک ردیف و یک ستون است یعنی این جدول دارای یک خا نه می باشد . یکی از ویژگیهای جدول که محبوبیت آنرا زیاد می کند اینست که هر خانه از آن می تواند به طور مجزا زمی نه رنگی و یا عکسی در زمی نه آن داشته باشد البته اندازه های هر خانه هم می تواند متفاوت باشد

border

خب برای اینکه بتوانیم یک جدول را در مرورگر خود ببینیم باید با خصوصیت border=" " آشنا شویم. این خصوصیت دور جدول یک کادردرست می کند البته اگر این خصوصیت را برابربا صفر قرار دهیم یا آنرا ننویسیم کادر دور جدول در مرورگر نمایان نخواهد شد و هر چه عدد بزرگتری بگذاریم کادر ضخیم تری خواهیم داشت، در ضمن رنگ کادرهم با bordercolor=" " که برابر با عدد هگز رنگ است تعیین می شود.border ویژگیهای دیگری هم دارد، به طور مثال شما می توانید خصوصیت frame=" " را به تگ <table>اضافه کرده و آنرا برابر با یکی از کلمات void, above, below, hsides, vsides,lhs, rhs, box قرار دهید تا کادر دور جدول را کنترل کنید. این کلمات هر کدام یک قسمت از کادر را نمایان می کنند.

rule

ویژگی دیگری هم هست ولی بهتره اول یک جدول درست کنید تا مطلب جا بیافتد، برنامه Notepad را باز کنید و کد زیر را وارد کرده و به اسم Table.htm ذخیره کنید:

```
<table border="۴" bordercolor="#ff۰۰۰۰">
  <tr>
     <td>cell ۰۱</td>

     <td>cell ۰۲</td>
  </tr>
  <tr>
     <td>cell ۰۳</td>

    <td>cell ۰۴</td>
  </tr>
</table>
```

حالا خصوصیت" "=rules را به تگ<table> اضافه کنید و هر بار یکی از کلمات all, none, groups, rows, cols را جلوی آن قرار دهید و فایل را دوباره ذخیره کنید تا تاثیر این خصوصیت را ببینید .

Alignment

می توانید با استفاده از خصوصیت" "=align که در تگ <td>می نویسید محل قرارگرفتن یک عنصر مانند متن را تعیین کنید که می توانید این خصوصیت را با کلمات left, right, center, justify مقداردهی کنید .

height, width

با خصوصیات " "=width , " "=height هم اندازه طول و عرض یک جدول را می توان کنترل کرد که باید آنها را برابر با عدد در مقیاس پیکسل قرار دهید در ضمن شما می توانید از درصد ٪ هم استفاده کنید مانند "٪۹۰=width" که این جدول نود درصد عرض یک صفحه را اشغال می کند.

bgcolor

برای تعیین رنگ زمی نه یک جدول از " "=bgcolor که با عدد هگز رنگها مقدار دهی می شود استفاده می کنیم. این خصوصیت را در تگ های <tr> و <td> هم می توانید بکار برید .

cellspacing, cellpadding

تگ <table> دو خصوصیت دیگر هم دارد که شما با بکار بردن آنها می توانید فاصله بین خانه های جدول را از هم کنترل کنید یعنی فضای خالی بین خانه ها اضافه کنید، این خصوصیت این است cellspacing=" " و دیگری فاصله بین متن داخل یک خانه در جدول با لبه های چارچوب آن خانه است که خصوصیت cellpadding=" " این وظیفه را به عهده دارد. هرچه مقداردهی عددی آنها بزرگتر باشد فاصله ها بیشتر خواهد بود

header: <th> </th>

حالا که با درست کردن ردیف و ستون در جدول آشنا شدید، می توانید برای هر ستون و ردیف عنوان گذاری کنید و یک تیتر به بالای ستون یا به ابتدای یک ردیف اضافه کنید . در هر قسمت که شما به یک تیتر و عنوان نیاز داشتید بجای تگ <td> </td> از تگ <th> </th> استفاده کنید که در این حالت متن نوشته شده در آن خانه بصورت ضخیم ظاهر شده و از بقیه متمایز می شود.

colspan, rowspan

گاهی اوقات شما نیاز دارید تا چند خانه در یک ردیف یا یک ستون از جدول را ترکیب کنید تا یک خانه درست شود و بطور مثال یک تیتر و عنوان در آن قرار دهید که برای ترکیب خانه های یک ستون از خصوصیت rowspan=" " و برای یک ردیف از colspan=" " در تگ <td> استفاده می کنید و مقدار آنرا برابر با تعداد خانه هایی که باید یکی شوند، قرار می دهید.

caption

بطور معمول هر جدول دارای یک اسم و یا یک توضیح می باشد که آنرا با تگ <caption></caption> در می ان تگهای <table></table> مشخص می کنند. این تگ دارای خصوصیت align=" " می باشد با کلمات top, bottom, left, center, right مقداردهی می شود.

thead, tbody, tfoot

اگر جدول شما دارای ستونها و ردیفهای زیادی است پس تکرار خصوصیات در هر تگ مربوطه بسیار دشوار است . شما می توانید بصورت گروهی این خصوصیات را کنترل کنید . برای کنترل ردیفها باید آنها را به سه قسمت تقسیم کرد، قسمت سرشامل فقط خانه های ردیف اول، بدنه شامل همه ردیفهای وسط و قسمت انتهایی شامل فقط خانه های آخرین ردیف، که قسمت سر را با <thead></thead> ، قسمت بدنه را با <tbody></tbody> و قسمت انتهایی را با <tfoot></tfoot> کنترل می کنیم. این تگ دارای خصوصیت align نیز می باشد.

colgroup

همچنین با تگ <colgroup></colgroup> می توانید خصوصیات کلیه ستونهای یک جدول را کنترل کنید که این تگ علاوه بر خصوصیات گفته شده در این بخش دارای خصوصیت span=" " نیز می باشد که توسط آن تعداد ستونهایی که باید کنترل شوند را تعیین می کنید.

summery

یک خصوصیت دیگر نیز برای تگ <table> هست بنام summery=" " که توصیفی از جدول می باشد ولی در جایی نمایش داده نمی شود . این خصوصیت برای مرورگرهایی که فقط متن را نمایش می دهند و یا مرورگرهای مخصوص افراد ناتوان ، مفید می باشد که بهتر است از آن استفاده کنید

به این موضوع هم توجه داشته باشید که شما می توانید داخل یک جدول، جدول دیگری هم درست کنید یعنی داخل هرخانه جداگانه یک جدول بسازید که به این روش Nested table گفته می شود.

افزودن فایل صوتی به زمی نه صفحه :

با اضافه کردن تگ <bgsound> به سورس کد صفحه وب، می توانید آهنگ یا صدایی را در زمی نه صفحه وارد کنید تا هنگامی که بیننده صفحه را باز می کند آن فایل بدون نیاز به برنامه کاربردی دیگری پخش شود. این تگ در حال حاضر کاربردی ندارد چون فقط مرورگراینترنت اکسپلورر نسخه های ۵.۵ به پایین آنرا پشتیبانی می کنند. این تگ دارای دو خصوصیت ، src برای آدرس دهی فایل صوتی و loop برای کنترل تکرار دفعات پخش می باشد.

<bgsound src=”mymusic.mid” loop=”۲”>

<meta>

متا تگها اطلاعات مربوط به موتورهای جستجوگر را کنترل می کنند . در این تگها هر نوع اطلاعات که برای جستجو نیاز باشد را وارد می کنید ، مانند کلمات کلیدی یا نام سازنده سایت و یا تاریخ راه اندازی و از این قبیل که در بخش سایتهای جستجوگر بیشتر توضیح خواهم داد. خصوصیاتی را که در این تگ می توانید بکار برید،

http-equiv=” “, name=” “ , content=” “

برای معرفی کلمات کلیدی سایت باید خصوصیت http-equiv=”keyword” قرار دهیم و کلمات مورد نظر را در خصوصیت “ ”=content وارد کنیم،

<meta http-equiv=”keyword” content=”html, web, amoozesh, amouzesh“>

همچنین می توانید نوع رمزگذاری encoding مرورگر را در این قسمت مع ین کنید که برای بکار بردن حروف فارسی باید از utf-۸ استفاده کنید که در قسمت فارسی نویسی توضیح خواهم داد.

<meta http-equiv=”content-type” content=”text/html;charset=utf-۸“>

این تگ به مرورگر می فهماند که باید از حروف و علامتهای یونیکد برای نمایش صفحه استفاده کند. برای وارد کردن اطلاعات دیگر هم مانند زیر عمل می کنید:

<meta name=”author” content=”your-name“>

اگر شما نمی خواهید که صفحه ای در هارد دیسک کاربر سایت شما ذخیره شود تا بتوان آنرا بصورت offline هم ببینند، از این تگ باید استفاده کنید:

<meta name=”pragma” content=”no-cache”>

یا اینکه آن صفحه هر لحظه اطلاعاتش عوض می شود و می خواهید تا بیننده سریع آن اطلاعات را ببیند، از خاصیت refresh مرورگر استفاده کرده و این تگ را اضافه می کنید:

<meta http-equiv=”refresh” content=”۵”>

که آن عدد زمان بر حسب ثانیه است . اگر می خواهید که بعد زمان معین همان صفحه به یک صفحه دیگر یا یک وب سایت دیگر هدایت شود خصوصیت زیر را اضافه کنید

<meta http-equiv=”refresh” content=”۵;url=http://www.sitename.com/page۰۱.htm”>

تعیین الگو و style :

هر style باید برای یک تگ خاص تعریف شود . به محتوای داخل تگی که می خواهیم یک style نسبت دهیم selector یا انتخاب کننده می گویند که االگوی تعریف شده در تگ آغاز کننده یک عنصر تا تگ پایان دهنده آن تاثیر خواهد داشت.

برای تعیین و تعریف یک الگو ابتدا باید selector را مشخص و بنویسیم مانند h۱ سپس خصوصیات و مقادیر آنها را بین دو علامت { } بگذاریم که طرز نوشتن خصوصیت و مقدار آن بین { } به این ترتیب می باشد:

ابتدا خصوصیت را نوشته سپس علامت : می گذاریم و بعد مقدار مشخصه آنرا می نویسیم ،

h۱ { color:green }

اگر بخواهیم چند خصوصیت را به یک selector نسبت دهیم ، بعد از هر کدام یک ; می گذاریم،

h۱ { color:green; text-align:center }

روشهای تعریف ااگو و style :

Style ها را به سه روش می توان تعریف کرد:

۱- به عنوان یک الگوی خارجی که اطلاعات در یک فایل نوشته می شود و با پسوند .css ذخیره می شود که در تگ <link> داخل قسمت head آدرس دهی و فراخوانده می شود.

۲- style را می توانیم در قسمت head نوشته و جاسازی کنیم که با تگهای <style></style> مشخص می شوند.

۳- با قرار دادن style به عنوان خصوصیت در یک تگ و مقدار دهی آن. مانند تگ زیر:

نکته: اگر برای یک selector به چند روش style تعیین شود، آن الگویی که به تگ نزدیکتر است اجرا خواهد شد، پس طبق این اصل روش سوم نسبت به بقیه موثرتر است . در ضمن هیچ محدود یتی در تعیین selector ها ویا خصوصیات آنها نیست. به زیر توجه کنید:

فراخوانی یک استیل شیت:

<link rel=”stylesheet” type=”text/css” href=”آدرس فایل “>

s.ali.taher@gmail.com

s.ali.taher@gmail.com

اسکریپت

زبان های اسکریپتی برای ارائه تحولات و ایجاد پویایی در صفحات وب ایجاد شدند . این زبان ها از روی زبان های برنامه نویسی ساخته شدند و بهمین دلیل دارای تشابه بسیاری با هم هستند . این زبان ها در اصل نمونه کوچک شده زبان های مادر خود هستند . تعدادی از فرمان ها و امکانات زبان های بزرگ در این زبان ها حذف شده اند . مثلا امکان نوشتن فایل یا پاک کردن فایل ها برروی سیستم کاربر مانند زبان های برنامه نویسی وجود ندارد . البته این زبان ها برای استفاده در زمینه کاری شبکه طراحی شده اند و حذف این دستورات علل خاصی (از جمله بالا رفتن امنیت و ...) داشته است .

Script VB

این زبان نمونه کوچک شده زبان VB است و کسانی که با ویژوال بیسیک کار کرده اند در یاد گیری این زبان بیشتر راه را پیموده اند و کافیست تفاوت های آن را با ویژوال بیسیک یاد گیرند . این زبان برای افزودن افکت به صفحات وب نیز کاربرد دارد . ولی کاربرد برتر آن در نوشتن برنامه های سمت سرور است که بر اساس تکنولوژی ASP صورت میگیرد . این زبان به عنوان زبان پیش فرض برای نوشتن صفحات ASP است . یاد گیری این زبان برای کسانی که تا کنون برنامه نویسی نکرده اند بسیا ر آسان است و شروع بسیار خوبی برای ورود به دنیای برنامه نویسی است .

Script JAVA

این زبان ساختاری شبیه زبان C دارد و بیشتر برای ایجاد افکت بر روی کامپیوتر کاربر استفاده میشود (Client-side) .احتمالا تا کنون سایت هایی را دیده اید که در آن کلمه خاصی دنبال موس میدود . یا هنگام وارد شدن به آن سایت مرورگر شما در صفحه ویندوزتان میلرزد . این قبیل کد ها که فقط روی سیستم کاربر اجرا میشوند و نیازی به پردازش توسط سرور ندارند را کد های سمت کاربر (کلاینت ساید) میگویند . البته این زبان نیز قابلیت های استفاده به صورت server-side را داراست . اما چون استفاده از زبان VBS آسانتر است معمولا از VBS برای نوشتن برنامه های سرور-ساید استفاده میشود . یکی از تفاوت های این دو زبان در طرز نوشتن حروف است . در VBS تفاوتی ندارد که دستورات را با حروف کوچک یا بزرگ بنویسید ، اما در JavaScript اگر دستوری که با حروف کوچک است با حروف بزرگ بنویسید با Error در صفحات خود مواجه میشوید.برای دیدن قدرت زبان JavaScript میتوانید از سایت AnfyTeam دیدن کنید . این سایت همچنین امکان دانلود برنامه ای برای ساخت افکت های جاوا اسکریپت خود را در اختیارتان میگذارد .

HTML Scripts

پیت ها را به صفحاتتان اضافه کنید تا آنها را پویاتر و فعال تر بسازید.

وارد کردن یک اسکریپت در یک صفحهHTML

یک اسکریپت در HTML با برچسب SCRIPT معرفی می شود. توجه کنید که برای تعیین زبان اسکریپت نویسی از TYPE استفاده خواهید کرد.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

اسکریپت بالا این خروجی را خواهد داشت :

Hello world!

چگونه مرورگرهای قدیمی را راه بیندازیم

یک مرورگر که نمی تواند برچسب <script> را تشخیص دهد ، آن را بعنوان متن در صفحه نمایش خواهد داد . برای جلوگیری از مرورگر از انجام این کار شما باید script را در برچسب های توضیح مخفی کنید. یک مرورگر قدیمی که نمی تواند script را تشخیص دهد توضیحات را نادیده خواهد گرفت و محتویات آ ن برچسب را نشان نخواهد داد.درحالیکه یک مرورگر جدید می فهمد که اسکریپت ها باید اجرا شوند حتی اگر با برچسب توضیحات محدود شده باشند .

JavaScript:

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
```

VBScript:

```
<script type="text/vbscript">
<!--
document.write("Hello World!")
'-->
</script>
```

برچسب<noscript>

در مجموع برای مخفی کردن اسکریپت درون توضیحات شما می توانید یک برچسب < noscript> اضافه کنید.این برچسب برای نشان دادن یک متن درصورتیکه اسکریپت اجرا نشود بکار می رود .این برچسب برای مرورگرهایی بکار می رود که برچسب script را تشخیص نمی دهند و اسکریپت های درون را پشتیبانی نمی کنند . بنابراین این مرورگرها بجای آن ، متن داخل برچسب < noscript> را نشان می دهد .

JavaScript:

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

VBScript:

```
<script type="text/vbscript">
<!--
document.write("Hello World!")
'-->
</script>
<noscript>Your browser does not support VBScript!</noscript>
```

برچسب هایscript

برچسب	توضیحات
<script>	یک اسکریپت تعریف می کند
<noscript>	یک متن جایگزین برای زمانی که اسکریپت کار نکند تعریف می کند
<object>	یک شی تعریف می کند
<param>	تنظیمات زمان اجرا برای اسکریپت تعریف می کند
<applet>	به جای آن از object استفاده کنید

برچسب های HTML می توانند موجودیت داشته باشند .موجودیت های خاص هر برچسب در زیر لیست شده . موجودیت های فوق هسته هستند و موجودیت زبان برای همه برچسب ها استاندارد است.

موجودیت های هسته

موجودیت	مقدار	توضیحات
class	class_rule or style_rule	کلاس یا عنصر
id	id_name	یک id یکتا برای عنصر
style	style_definition	تعریف یک style درون خطی
title	tooltip_text	یک متن برای نمایش در یکtool tip

موجودیت های زبان

موجودیت	مقدار	توضیحات

dir	ltr | rtl	چیدمان متن را مشخص می کند
lang	language_code	کد زبان را مشخص می کند

موجودیت های کلمات کلیدی

موجودیت	مقدار	توضیحات
accesskey	character	یک میانبر صفحه کلید برای عناصر معرفی می کند.
tabindex	number	درخواست tab برای یک عنصر را تعریف می کند.

چیزی که در HTML۴.۰ جدید است توانایی انجام رویدادهای HTML در یک مرورگر است، مثل شروع یک جاوا اسکریپت وقتی که یک کاربر روی عنصر HTML کلیک می کند .در زیر یک لیست از موجودیتهاست که برای تعریف عملیات یک رویداد می توانند در یک برچسب وارد شوند .

رویدادهای پنجره

فقط در body,frameset معتبر است .

موجودیت	مقدار	توضیحات
onload	script	وقتی که پرونده بارگیری شود اسکریپت اجرا خواهد شد.

onunload	script	وقتی که پرونده بارگیری نشود اسکریپت اجرا خواهد شد

رویدادهای عناصر فرم

فقط در فرم معتبر است .

موجودیت	مقدار	توضیحات
onchange	script	هنگلمی که عنصر تغییر کند اسکریپت اجرا می شود
onsubmit	script	هنگامی که فرم ارسال می شود اسکریپت اجرا می شود
onreset	script	هنگامی که فرم reset می شود اسکریپت اجرا می شود.
onselect	script	هنگامی که عنصر انتخاب می شود اسکریپت اجرا می شود.
onblur	script	هنگامی که تمرکز از عنصر برداشته می شود اسکریپت اجرا می شود.
onfocus	script	هنگامی که روی عنصر متمرکز می شویم اسکریپت اجرا می شود.

رویدادهای صفحه کلید

موجودیت	مقدار	توضیحات
onkeydown	script	هنگامیکه کلید فشرده می شود چه کاری انجام شود
onkeypress	script	هنگامی کلید زده می شود چه کاری انجام شود.
onkeyup	script	هنگامی که کلید آزاد می شود چه کاری انجام شود

رویدادهای ماوس

موجودیت	مقدار	توضیحات
onclick	script	با کلیک ماوس چه کاری انجام شوی
ondblclick	script	با دوبار کلیک کردن ماوس چه کاری انجام شود
onmousedown	script	هنگامی که دکمه ماوس فشرده می شود چه کاری انجام شود
onmousemove	script	هنگامی که اشاره گر ماوس جابجا می شود چه کاری انجام شود
onmouseout	script	هنگامی که اشاره گر ماوس از یک عنصر خارج می شود چه کاری انجام شود

onmouseover	script	هنگامی که اشاره گر ماوس روی یک عنصر قرار می گیرد چه کاری انجام شود
onmouseup	script	هنگامی که کلید ماوس رها می شود چه کاری انجام شود.

s.ali.taher@gmail.com

s.ali.taher@gmail.com

شاید تا حالا این سه حرف رو خیلی جاها دیده باشید ، ولی ندونید چیه !

CSS مخفف Style Sheets Cascading هست . با استفاده از زبان ساده ی CSS شما می تونید تنظیمات خاصی رو روی تگ های html (یا زبان های Murkup دیگه مثل xml) اعمال کنید !

هدف CSS جداسازی طراحی بدنه ی یک فایل html از شیوه ی نمایش اون فایل html هست . با CSS شما می تونید تعیین کنید که هر تگ html به چه شکلی نمایش داده بشه . مثلا رنگ ، اندازه و نوع فونت متن درون تگ رو مشخص کنید . یا اینکه پس زمینه و ویژگی های دیگه ی تگ رو تعیین کنید .

! سوال : شاید با خودتون بگید که هر تگ html خودش یک سری ویژگی هایی داره که با دادن مقدار به اونها میشه نوع نمایش تگ رو مشخص کرد . پس دیگه CSS واسه چیه ؟

مثلا تگ <font>در زیر رو ببینید که ویژگی های رنگ و اندازه و نوع فونت رو در خودش داره:

<font/>Learn.com۷<“face=”’Tahoma “size=”’۱۲px “color=”’red font>

خروجی این کد متن Learn.com۷ هست که با رنگ قرمز ، اندازه ی ۱۲ پیکسل و با فونت Tahoma نمایش داده میشه.

(اصطلاح : در html اصطلاحا به ویژگی هایی که روی هر تگ می تونید اعمال کنید attribute های اون تگ میگن . مثلا size و color از attribute های تگ font هستند)

خب پس با توجه به اینکه هر تگی خودش می تونه نوع نمایشش رو تعیین کنه ، دیگه چه لزومی به استفاده از CSS هست ؟

! جواب : شما فرض کنید می خواید تو صفحه ی وبتون ۱۰۰ بار (به تعداد زیاد) در جاهای مختلف صفحه ، متنی رو با شیوه ی نمایش خاصی بیارید . اگر بخواهید از attribute های هر تگ برای نوع نمایش اون استفاده کنید ، مثلا در مثال بالا باید ۱۰۰ بار تگ font رو همراه با هر سه ویژگی color، size و face در کد صفحه ی وبتونن بیارید .

ولی حالا توی CSS در نظر بگیرید : شما تنظیمات color، size و face رو فقط در یک جای کد صفحه ی وبتون میارید و به این دسته تنظیمات یک اسم (توی css اصطلاحا Selector) اختصاص میدید و به جای اینکه در هر ۱۰۰ مرتبه تنظیمات رو توی خود تگ بیارید ، فقط میگید که آقای تگ font برو و تنظیمات فلان Selector رو روی خودت اعمال کن . به صورت خاص در این مثال پیاده سازی CSS اون به شکل زیره :

اول باید Selector مد نظرتون رو با تنظیمات نمایشش در تگ<style> تعریف کنید . به شکل زیر :

```
<style>
{;font-family:Tahoma ;fontStyle۱{color:red; font-size:۱۲px.
<style/>
```

و بعد هر جایی که خواستید می تونید از این Selector که در اینجا به نام fontStyle۱ تعریف شده به شکل زیر استفاده کنید :

<font/>Learn.com۷<”**fontStyle۱**“=class font>

Learn.com۷از ویژگی class در همه ی تگهای html می تونید برای مشخص کردن تنظیمات اعمالی روی تگ استفاده کنید . که البته تنظیماتی با چنین اسمی باید قبلا توی تگ Style نوشته باشید . مثلا در کد بالا تمام تنظیمات سلکتور fontStyle۱ روی تگ font اعمال میشه !

(در این پست ، هدف آموزش CSS نیست . این مثال صرفا برای درک بهتر از اینکه چرا باید از CSS استفاده کنیم آورده شده . در پست های آینده به صورت مفصل وارد بحث آموزش CSS خواهیم شد)

:: خوب حالا این همه قصه گفتیم ، مزیت استفاده از CSS در مقابل ویژگی های خود تگهای html چیه ؟

۱) کاملا مشخصه که اگه بخواید از attribute های خود html استفاده کنید ، بدلیل اینکه باید اونها رو در هر تگ دوباره و چندباره تکرار کنید حجم فایل html شما خیلی زیاد میشه ، در صورتی که ما با استفاده از CSS میتونیم یک دسته از attribute ها رو فقط یکبار و در یکجا تعریف کنیم و تگهای مختلف به اون دسته تنظیمات ارجاع بدن و نیازی به آوردن تنظیمات در خود تگ و اتلاف فضای بیشتر نباشه. با CSS کلی در اندازه ی فایل صفحه ی وبتون صرفه جویی می کنید !

۲) خوب هر چی اندازه ی فایل وبتون کوچیکتر باشه سرعت لود شدن صفحه ی وب هم مسلما بیشتر میشه . که اینکار با CSS میسره !

۳) تغییر دادن صفحه ی وبتون به مراتب راحتتر میشه. تو مثال بالا فرض کنید بعد از یه مدتی استقلالی میشید و عاشق رنگ آبی . و تصمیم می گیرید همه ی اون ۱۰۰ تا جایی که Learn.com۷ رو با رنگ قرمز تو صفحه وبتون آورده بودید آبی کنید . حالا اگه از attribute های html استفاده کرده باشید باید تمام کد صفحه تون رو جستجو کنید و هر جا متن learn.com۷ رو آوردید برید و مقدار ویژگی color تگ font رو از red به blue تغیر بدید . ولی اگه از CSS استفاده میکردید فقط همون یکجا (توی Selector مورد نظر) red رو تبدیل به blue می کردید و همه ی تگهایی که به این Selector ارجاع میدادند تغیرات رو متوجه میشن !

۴) تنظیمات CSS رو می تونید توی فایل جداگونه بزارید و به این ترتیب استفاده و نگهداریش راحتتره!

s.ali.taher@gmail.com

پایگاه های داده از اجزاء مهم نرم افزار های امروزی هستند. آن ها با انجام اعمال ذخیره و بازیابی **اطلاعات ، امکان ذخیره سازی اطلاعات را به خوبی فراهم می کنند. دیگر لازم نیست برای هر سیستم نرم افزاری یک مدیریت پایگاه داده ایجاد کرد. بلکه این نرم افزار ها کار را برای بسیار آسان کرده و ما اکنون باید به فکر ساختار اطلاعات و نحوه ساماندهی آن ها در ساختار های این برنامه ها باشیم.**

انواع مختلفی از پایگاه داده وجود دارد .

پایگاه هایی که بر مبنای یک فایل هستند. مانند MS Access و فاکس پرو.

پایگاه هایی که بر مبنای تئوری خادم مخدوم - Server Client - هستند. این پایگاه ها در اندازه های مختلف قرار دارند و اصطلاحا به آن ها RDBMS - Relational Data Base Management Systems گفته می شود. از این قبیل سیستم ها می توان به MS SQL SERVER، MY SQL و یا Oracle اشاره کرد.

آشنایی با نسخه‌های SQL Server ۲۰۰۵

SQL Server۲۰۰۵ در نسخه‌های مختلفی عرضه شده است که از نظر کارایی و قیمت بسیار متفاوت هستند و البته برای کاربران مختلفی نیز طراحی شده‌اند.

SQL Server۲۰۰۵ Enterprise Edition (۳۲-bit and ۶۴ bit)

این نسخه در حقیقت نسخه پیشرفته این نرم‌افزار است و برای محیطی طراحی شده است که یا دارای بانک اطلاعاتی بسیار بزرگ (از نظر تعداد رکورد یا حجم اطلاعات) یا دارای تعداد پردازش آنلاین بسیار زیاد یا نیازمند تحلیل پیچیده اطلاعات است. این نسخه همه قابلیت‌های این نرم‌افزار را دارد.

SQL Server۲۰۰۵ Standard Edition (۳۲-bit and ۶۴-bit)

این نسخه در حقیقت برای شرکت‌های متوسط مناسب است. نکته مهم این است که این نسخه نیز به راحتی بانک‌های اطلاعاتی بزرگ را پشتیبانی می‌کند. اما بعضی از ویژگی‌های مورد نیاز برای تحلیل پیچیده اطلاعات را ندارد و البته قیمت بسیار مناسب‌تری نسبت به نسخه Enterprise دارد.

SQL Server۲۰۰۵ Workgroup Edition (۳۲-bit only)

این نسخه برای شرکت‌های کوچک و سرویس‌دهنده‌های وب و البته بانک‌های اطلاعاتی که به عنوان نسخه پشتیبان استفاده می‌شوند، مناسب است. این نسخه در بین نسخه‌هایی که به کاربران نهایی عرضه می‌شود، کمترین قیمت را دارد.

SQL Server ۲۰۰۵ Developer edition (۳۲-bit and ۶۴-bit)

این نسخه از نظر خصوصیات و ویژگی‌ها هیچ تفاوتی با نسخه Entrprise ندارد. اما مختص برنامه‌نویسان است. لذا قیمت بسیار پایینی دارد و شرکت‌های برنامه‌نویسی در زمینه بانک‌های اطلاعاتی از این نسخه استفاده می‌کنند. بدین ترتیب این شرکت‌ها دیگر مجبور نیستند برای تست برنامه خود نسخه Entrprise را تهیه کنند.

SQL Server۲۰۰۵ Express Edition (۳۲-bit only)

این نسخه کاملاً مجانی است و می‌تواند به عنوان یک سرویس‌دهنده یا یک سرویس گیرنده با حجم کوچکی از اطلاعات استفاده شود. بدین‌ترتیب برنامه‌نویسانی نیز که می‌خواهند برنامه‌ای بنویسند که دارای یک بانک اطلاعات کوچک است دیگر نیازی نیست که نگران بانک اطلاعاتی خود باشند. چرا که می‌توانند برنامه خود را همراه با یک نسخه مجانی Expees Edition ارائه کنند.

مواردی در خصوص نصب SQL Server ۲۰۰۵

اگر بخواهید SQL Server ۲۰۰۵ را روی کامپیوتر شخصی خودتون نصب کنید، توجه داشته باشید که اگر ویندوزتون XP هست نمی‌تونید نسخه Enterprise رو روی سیستمتون نصب کنید و باید از نسخه‌های Standard و یا Developer استفاده کنید. ضمناً ویندوز XP شما حتماً باید SP۲ به بالا باشد.

ولی اگر بخواهید SQL Server ۲۰۰۵ رو برای شرکت و یا سازمانی نصب کنید، باید نسخه Enterprise رو روی Windows Server ۲۰۰۳ با SP۱ به بالا نصب شود.

سیتم عامل

در مورد این‌که SQL Server۲۰۰۵ روی کدام سیستم عامل را نصب کنیم، بسته به این‌که شما کدام نسخه را بخواهید نصب کنید، انتخاب‌های زیادی وجود دارد. اما پیشنهاد می‌کنم که روی سیستم‌عامل windows server۲۰۰۳ نصب کنید.

چراکه با این کار می‌توانید از امکانات امنیتی این سیستم‌عامل در پایگاه داده خود بهره ببرید. توجه داشته باشید که اگر شما windows server۲۰۰۳ ندارید، بدین معنی نیست که نمی‌توانید از SQL Server۲۰۰۵ استفاده نمایید؛ می‌توانید SQL Server۲۰۰۵ را روی windows XP SP۲ نصب کنید.

اگر windows Server۲۰۰۳ را نصب کردید، حتماً از قسمت نرم‌افزارهایی که با خود سی‌دی ویندوز عرضه می‌شوند، نرم‌افزار IIS را نیز نصب کنید. برای این کار کافی است به Control Panel>add or remove Programs>add/remove windows component مراجعه کنید و روی Aplication server دو بار کلیک کنید و سپس با انتخاب گزینه Internet Information services نرم‌افزار IIS را نصب کنید. به خاطر داشته باشید که پیش از نصب حتماً سی‌دی ویندوز را در دستگاه گذاشته باشید (شکل ۱).

عرضه شدن نسخه ۲۰۰۵ پایگاه داده‌ای مشهور مایکروسافت یعنی SQLServer، بازار نرم‌افزارهای بانک اطلاعاتی را به شدت تحت تأثیر خود قرار داد. امکانات گسترده‌ای که در این نسخه جدید تعبیه شده، طراحی و پیاده‌سازی بانک‌های اطلاعاتی را سرعت بخشیده و تلفیق و انطباق آن با انواع فناوری‌های نو مانند XML وADO.NET، باعث افزایش قدرت و کارایی آن شده است. در این مقاله به برخی از ویژگی‌های جدید این برنامه نگلهی خواهیم داشت.

Snapshot Isolation Level

یکی از روش‌هایی که به انواع متدهای قفل کردن ردیف‌های یک جدول بانک‌اطلاعاتی در نسخه جدید اضافه شده است، شیوه تصویربرداری از رکورد است. در روش‌های قبلی، اگر یک یا چند رکورد بانک اطلاعاتی توسط دستور BeginTrans که شروع یک فرآیند را مشخص می‌کند در شرف تغییر یا حذف قرار می‌گرفتند، تا مادامی که فرآیند مذکور توسط دستور Commit Trans تأیید یا توسط RollBack منتفی نشود، از هیچ جا و برنامه‌ای نمی‌توان رکوردهای مذکور را حتی با دستور ساده SELECT خواند. اما در روش جدید قفل‌گذاری، در صورت بروز چنین رویدادی سایر کاربران می‌توانند همواره آخرین ارزش رکوردهای مذکور را با این فرض که هنوز هیچ تغییری در آن‌ها ایجاد نشده است بخوانند و مورد استفاده قرار دهند.

با نسخه جدید SQL Server، برنامه‌نویسان بانک‌های اطلاعاتی قادرند از امکانات و قابلیت‌های موجود در پلتفرم دات‌نت و کلیه توابع و کلاس‌های ساخته شده در آن بهره‌مند شوند. یکی از ابتدایی‌ترین و در عین حال اساسی‌ترین این قابلیت‌ها، امکان استفاده از دو زبان مهم و کاربرپسند دات‌نت یعنی ویژوال بیسیک و سی‌شارپ در پیاده‌سازی اجزای مختلف یک بانک‌اطلاعاتی است. این عامل نه‌تنها باعث می‌شود که برنامه‌نویسان برای نوشتن ماژول‌هایی مثل تریگرها، روال‌ها (Stored Procedures) در توابع به جای استفاده از زبان استاندارد و در عین حال پیچیده T-SQL ، بتوانند از زبان‌های محیط دات‌نت با تمام ساختارها، دستورات، کلاس‌ها، آرایه‌ها، و خلاصه تمام ویژگی‌های یک زبان شی گرا استفاده کنند، بلکه این همکاری نزدیک بین موتور برنامه‌نویسی دات‌نت یعنی CLR (که مسؤول تبدیل کدهای نوشته شده دات‌نت به زبان سیستم‌عامل است) و موتور بانک اطلاعاتی SQLServer باعث شده تا به غیر از تنوع زبان‌های برنامه‌نویسی قابل استفاده درSQLServer، تغییر قابل توجهی نیز در کارایی ماژول‌های مذکور پیش آید. در واقع موضوع از این قرار است که اصولاً کدهای نوشته شده به زبان‌های دات‌نت، ابتدا توسط کامپایلر به زبان (IL) ترجمه می‌شوند. سپس CLR این کد میانی را به کد قابل فهم سیستم‌عامل تبدیل و آماده اجرا می‌نماید. این کار سبب می‌شود تا کدهای نهایی به دلیل این‌که بسیار به سیستم‌عامل نزدیک می‌باشد سریع‌تر از کدهای TSQL (که فقط توسط موتور بانک اطلاعاتی قابل اجرا هستند) اجرا شوند و در زمان اجرا از کارایی بیشتری برخوردار باشند. البته این مسأله بدین معنی نیست که استفاده از زبان‌های دات‌نت همیشه بر زبان‌های SQL ارجحیت دارد، بلکه منظور آن است که در برخی موارد ممکن است آن قدر منطق و الگوریتم یک ماژول پیچیده باشد که برنامه‌نویس استفاده از زبان‌های دات‌نت را به دلیل آسان‌تر بودن ساختار و دستورات آن به زبان SQL ترجیح دهد. بنابراین زمانی که بیشترعملیات یک ماژول مربوط به خواندن و نوشتن اطلاعات باشد بهتر است از همان دستورات استاندارد SQL یعنی SELECT، UPDATE،DELETE وINSERT استفاده کرده و بی‌جهت منابع سیستم را صرف تعریف متغیرها و کلاس‌های دات‌نت ننماید. اما در ماژول‌هایی که بیشتر عملیاتشان شامل پردازش اطلاعات مثل انجام عملیات‌های ریاضی یا مقایسه اطلاعات با یکدیگر است بهتر است تا هم از امکانات برنامه‌نویسی و هم از سرعت و کارایی بالای دات‌نت در این زمینه بهره برد و ماژول‌های مذکور را با زبان‌های دات‌نت پیاده‌سازی کرد.

ADO .NET

طبق یک سنت نه‌چندان قدیمی برنامه‌نویسی در محیط ویندوز، برنامه‌نویسان SQLServer، بانک اطلاعاتی موردنظرشان را برروی سرور و برنامه کاربردی نوشته شده با زبانی مثل ویژوال بیسیک را بر

روی کلاینت‌ها قرار می‌دهند. سپس از طریق این برنامه کاربردی و با استفاده از اشیایی از جنس ADO داده‌های موردنیاز خود را از سمت سرور دریافت کرده و یا به آن ارسال می‌کنند. اکنون این ارتباط به لطف نسخه جدید SQLServer و همچنین محیط دات‌نت، با امکانات جدیدADO.NET بسیار کامل‌تر از قبل شده است. این ارتباط جدید با استفاده از مکانیسمی به نام اعلان (Notification) به یک ارتباط دوطرفه فعال تبدیل شده به طوری که ADO.NET قادر است پیغام‌هایی را از سمت پایگاه داده به سمت کلاینت ارسال کند. به عنوان مثال فرض کنید که شما با استفاده ازADO تعدادی از رکوردهای یک جدول بانک اطلاعاتی را انتخاب کرده و مشغول کار برروی آن‌ها هستید. در همین هنگام کاربر دیگری از طریق کلاینت و ADO خود، رکوردی در محدوده رکوردهای مورد انتخاب شما را تغییر می‌دهد یا حذف می‌کند. در این وقت موتور پایگاه داده با ارسال پیغامی به ADO شما، این مسأله را با استفاده از فراخوانی یک رخداد (Event) شی ADO به اطلاعتان می‌رساند.

علاوه براین قابلیت جدید، فناوری جدید دیگری هم با استفاده ازADO.NET به نسخه جدید SQLServer اضافه شده و آن امکان چند پرس‌وجوی همزمان توسط یک شی ADO است. در این شیوه اگر یک شی ADO با استفاده از دستور SELECT مشغول خواندن تعدادی از رکوردهای یک جدول بانک اطلاعاتی باشد، می‌تواند بدون این‌که منتظر به پایان رسیدن این عملیات شود، تعداد دیگری از رکوردهای یک جدول دیگر بانک اطلاعاتی را بخواند. این قابلیت جدید با نام (Multiple Active Result Set (MARS که قبلاً فقط در کرسرهای سمت سرور (server side) و آن هم نه با کارایی بالا وجود داشت اکنون در کرسرهای سمت راست کلاینت هم وجود دارد و تفاوت عمده آن با شکل قدیمی هم علاوه بر مورد مذکور، امکان ایجاد چند کرسر در یک شی ADO به صورت همزمان است. SQLServer نسخه ۲۰۰۵ به خوبی از تمام این ویژگی‌ها، پشتیبانی می‌کند.

تکنولوژی XML

اکنون که XML به یک استاندارد ارتباطی بین سکوهای مختلف تبدیل شده است، نسخه جدید SQLServer هم از توجه کافی به آن و ایجاد یک انقلاب در ساده‌تر استفاده کردن از آن طفره نرفته است. در نسخه ۲۰۰۰ کاربران قادر بودند تا با استفاده از دستور FOR XML نتیجه یک پرس‌وجوی SELECT از یک بانک اطلاعاتی را به درون یک فایل XML بریزند یا مثلاً با دستور OPEN XML می‌توانستند یک فایل XML را باز کرده و شروع به خواندن دستورات درون آن نمایند.

از آن‌جا که در نسخه جدید SQLServer توجه خاصی به این استاندارد و زبان ارتباطی شده است، یک نوع داده جدید (Data type) به انواع داده‌های قبلی و استاندارد SQL مثل int، char و امثال آن اضافه شده است. این نوع داده جدید که XML نام دارد و دارای خصوصیات یک نوع داده موجود در یک

محیط شی‌گرا است، دارای متدهای پیشرفته‌ای چون nodes()، Value()، exist()، query و modify() بوده و قادر است انواع پردازش‌های قابل انجام برروی اسناد XML را به راحتی انجام دهد. عملیات جستجو، تغییر، حذف و درج مقادیر موردنظر در داخل یک فایل XML را می‌توان با استفاده از متدهای مذکور و صرفاً با چند خط برنامه‌نویسی انجام داد. همچنین در این نسخه برخلاف نسخه ۲۰۰۰، با استفاده از دستور FOR XML می‌توان یک شیء از جنس XML را بدون ارسال آن به کلاینت، برروی سرور ساخته و از آن نگهداری کرد. با این کار می‌توان جداولی را که مرتباً مورد رجوع کاربران قرار می‌گیرند هراز گاهی در قالب XML به داخل حافظه آورد و کاربران مذکور به جای رجوع به جداول اصلی در هارددیسک، با استفاده از دستورات ویژه جستجو درXML ، متغیر مذکور را در حافظه سرور مورد جستجو قرار دهند و بدین‌وسیله یک نوع عمل Cache کردن را جهت افزایش سرعت دسترسی به اطلاعات تکراری شبیه‌سازی کنند. در این حالت، کاربران به جای استفاده از دستورSELECT استاندارد می‌توانند از OPEN XML که در نسخه ۲۰۰۵ قادر است متغیرهای جدید از نوع XML را بخواند استفاده کرده و به سرعت به اطلاعات موردنیاز خود دسترسی پیدا کنند. این قابلیت جدید آن‌قدر در سریع‌تر کردن جستجو در برنامه‌های تحت وب مهم و مؤثر است که جای هیچ مشکلی را در استفاده از آن باقی نمی‌گذارد.

سرویس اعلان (Notification)

همان‌طور که گفتیم سیستم اعلان درSQLServer قادر است پیغام‌هایی را طی زمان‌های مشخص به سمت کاربران بفرستد. مثلاً تصور کنید که تعدادی کاربر در حال اتصال به یک بانک حاوی اطلاعات مربوط به ارزش سهام در بورس هستند. از آن‌جایی که ممکن است قیمت سهام هر شرکت یا مؤسسه برای تعدادی از کاربران از اهمیت زیادی برخوردار باشد، می‌توان این سیستم را طوری تنظیم کرد تا هرگاه ارزش سهام خاصی که موردنظر هر کاربر است تغییر کرد، به صورت اتوماتیک به وی اعلام شود. کاربر هم می‌تواند این تغییرات را برروی برنامه کاربردی خود، تلفن همراه (در قالب Windows Messenger،SMS) و یا ایمیل به‌صورت مرتب دریافت و مشاهده کند.

سرویس گزارش‌گیری (Reporting Service)

اگر در زمان نصب SQL Server ۲۰۰۵ این گزینه فعال شود، بخش گزارشگیری خودکار روی سیستم شما نصب خواهد شد . این سرویس امکاناتی را برای ایجاد گزارش از بانک های اطلاعاتی مختلف در اختیارتان قرار می دهد. گزارشهای ایجاد شده توسط ا ین سرویس Web-enabled هستند و قابلیت پخش روی انواع دستگاهها را دارند . سرویس جدید تولید گزارش‌های متنوع در نسخه ۲۰۰۵ به یکی از

جالب‌ترین و پرکاربردترین قابلیت‌های این نسخه تبدیل شده است، وجود یک موتور گزارشگر قوی در سمت سرور و یک ابزار مناسب ساخت گزارش با واسط کاربر عالی، باعث شده تا برنامه‌نویسان بتوانند گزارش‌های موردنظر خود را با کارایی و سرعت مناسب در سمت سرور بسازند به طوری که این گزارش‌های سمت سرور توسط هر برنامه کاربردی سمت کلاینت در هر پلتفرمی با همان امکانات اتصال به SQLServer قابل مشاهده است. . شما می توانید این گزارشات را با فرمتهای مختلف Excel و Word و PDF و Html و ... ایجاد کنید.

سرویس تحلیل (Analysis Service)

قابلیت پردازش های تجزیه و تحلیلی آنلاین (OLAP) با سرعت بالا، تجزیه وتحلیل پیشرفته برای مجموعه دیتابیس های پیچیده و بزرگ با استفاده از راههای متعدد ذخیره سازی اطلاعات .در حقیقت اگر بخواهیم روی دیتابیس‌های خیلی بزرگ گزارشگیری کنیم، پروسه ساخت گزارش خیلی کند است. برای این منظور یک مخزن اطلاعات جنبی در نظر گرفته می‌شود و گزارشات از روی آن تهیه می‌شود. نتیجه گزارش بر روی یک فضای جداگانه قرار می‌گیرد و چنانچه اطلاعات تغییر کند، تغییرات ایجاد شده بصورت اتوماتیک بر روی گزارشات اعمال می‌شود.

Integration Services

این سرویسها یک پلت فرم هستند که راه حل هایی برای ایجاد یکپارچگی اطلاعات با سرعت بالا ارائه می کنند و شامل بسته های نرم افزاری پردازش extract, transform, and load برای data warehousingاست.

SQL Server Agent

به بیان ساده، این سرویس مسئول دفتر SQL SERVER است. مثلاً در صورت تعریف توسط ادمین، هر شب ساعت ۱۰ از اطلاعات کپی پشتیبان می‌گیرد.

Authentication Mode

چنانچه گزینه‌ی Windows Authentication Mode انتخاب گردد، دسترسی کاربران و کلمه عبور آنان به SQL Serverبر اساس وجود و درستی آن‌ها در ویندوز انجام می‌گیرد. ولی اگر Mixed Mode انتخاب شود باید برای دسترسی به SQL Server مجدداً نام کاربری و سطح دسترسی تعریف کرد.

Collation

شمایی در SQL Server که نحوه Sort حروف الفبای یک زبان خاص و همچنین نحوه Compare

کردن حروف الفبای آن زبان را در بر می‌گیرد.

تنظیمات collation به صورت پیش فرض از تنظیمات سیستم عامل گرفته می شود. ودر صورتی که سیستم عامل شما در بخش Regional settings دارای تنظیمات لازم فارسی باشد، نیاز به تغییر در این بخش ندارید.

Business Intelligence موجود در SQL Server ۲۰۰۵

سیستم های هوشمند و تحلیلی که بیشتر با عنوان سیستم های (OLAP) شناخته می شوند درون SQL Server ۲۰۰۵ قرار گرفته شده است . یکپارچگی هوش تجاری با موتور پایگاه داده قابلیت های فراوانی را به طراحان برنامه های کاربردی و تحلیل گران داده های سازمانی ارائه می دهد.

بهره گیری از مکانیسم امنیتی نوین و مطمئن با مدیریت آسان تر

توسط این نسخه می توانید دسترسی های خاص تر به افراد خاصی بدهید، طراحی جدید Schema به شما امکان می دهد به کاربران خود فقط دسترسی هایی را بدهید که به آن نیاز دارند.

قابلیت بسط پذیری در سازمان های بسیار بزرگ

بدون شک یکی از اصلی ترین مشکلات SQL Server ۲۰۰۰ ، عدم قابلیت یا بهتر است بگوئیم عدم کارایی این نسخه با حجم وسیعی از داده ها در سطح Enterprise بوده است. در واقع یکی از مهم ترین نقاط مورد توجه در طراحی این نسخه از SQL Server ، قابلیت رقابت این سیستم با رقبای تجاری مانند Oracle و DB/۲ بوده است.

بهبودهای ایجاد شده در زبان

در SQLServer ۲۰۰۵ تغییرات بسیار مثبتی در زبان SQL T ایجاد شده است. این تغییرات در زمینه‌های مختلف مثل مدیریت خطاها، جستجوهای بازگشتی (Recursive Query) و حتی در بدنه موتور پایگاه داده‌ها انجام شده و کارایی کلی ذخیره و یا خواندن اطلاعات را به نحو مطلوبی افزایش داده است. به عنوان مثال دردستورات

TSQL، دو اپراتور جدید دیده می‌شود، که PIVOT وUNPIVOT نام دارند. این دو اپراتور که در قسمت FROM یک پرس‌وجو مورد استفاده قرار می‌گیرند می‌توانند نتیجه یک جستجوی انجام شده توسط دستور SELECT را به جای برگرداندن در قالب ردیف‌ها یا رکوردهای پشت‌سرهم، به صورت ستون‌های مختلف یک یا چند رکورد برگردانند. در این روش یکی از ستون‌های (فیلدهای) یک جستجو

به عنوان محور معرفی شده و بقیه ستون‌ها براساس آن به صورت افقی طبقه‌بندی می‌شوند. به یک مثال توجه کنید:

```
SELECT CUSTOMER ID, order No
FROM orders PIVOT CustomerID
```

همان‌طور که مشاهده می‌کنید با استفاده از اپراتور مذکور، نتیجه پرس‌وجوی انجام شده به این صورت که هر ردیف به یک شماره مشتری و چندین شماره سفارش مربوطه به آن مشتری در قالب ستون‌های مختلف است، در می‌آید. این همان چیزی است که سال‌ها در**SQLServer** وجود نداشت و ابزارهای مختلف گزارش‌سازی مثل **CrystalReport** آن را با نام **Cross Tab** به کاربران خود ارایه می‌دادند. در همین رابطه اپراتور**UNPIVOT** هم عمل عکس اپراتور مذکور را انجام می‌دهد.

اپراتور دیگری که می‌تواند نقش مهمی را در دستورات **SQL** بازی کند **APPLY** نام دارد که در قسمت **FROM** یک دستور **SQL** به کار می‌رود. با استفاده از این دستور می‌توان خروجی یک تابع (**Function**) را با یک یا چند جدول دیگر ترکیب (**Join**) کرد همان‌طور که می‌دانید در ۲۰۰۵ **SQLServer** توابع می‌توانند یک یا چند ردیف یک جدول اطلاعاتی را برگردانند که این خروجی می‌تواند با یک جدول دیگر با استفاده از اپراتور مذکور ترکیب شود.

مدیریت خطا

در نسخه‌ای قدیمی **SQLServer** برای کشف و مدیریت خطا از سیستم **Error Handling** استفاده می‌شد. این شیوه کشف خطا که در زبانی مثل ویژوال بیسیک ۶ هم مورد استفاده قرار می‌گرفت با استفاده از دستور **GOTO** می‌توانست کنترل و خط اجرای روال را از یک محل به محل دیگر و در واقع از محل بروز خطا به محل مدیریت و آشکار کردن (**Raise**) آن ببرد و بدین‌وسیله پیغام خطایی را به کار نشان دهد. نسخه جدید **SQLServer** با تأثیر از پلتفرم دات‌نت، از دستورات ویژه کشف و مدیریت خطا با عنوان **Exception Handling** استفاده می‌کند. این روش با استفاده از دستورات جدید **TRY/CATCH** شیوه بهتری از مدیریت خطا را به اجرا می‌گذارد. در این روش برخلاف روش قبل، تمام خطاهای اتفاق‌افتادنی مثل خطاهای مربوط به تبدیل داده‌ها به یکدیگر (**DataConversion**) به خوبی مدیریت شده و از بروز خطاهایی که منجر به اتمام ناقص عملیات یک روال یا تریگر می‌شود جلوگیری به عمل می‌آید.

فرض کنیم که پلیس ۱۱۰ یک نرم افزار در اختیار دارد که توسط آن آمار جرائم را نگهداری می کند. در این آمار، پلیس اطلاعات مربوط به مکان و زمان وقوع جرم و نوع جرم مثلا دزدی را نگهداری می کند.

پس از مدتی پلیس می تواند اطلاعات مربوط به دزدی در یک مکان خاص را بررسی کند . مثلا پلیس در بررسی و تحلیل اطلاعات خود پی می برد که در فلان محله خاص بین ساعت ۸ الی ۱۰ صبح هیچ مورد دزدی گزارش نشده است بنابراین می تواند نیروهای خود را در آن زمان خاص در آن محله خاص کاهش داده و در جای دیگری که امکان وقوع دزدی بیشتر است استفاده کند.