

ایجاد قالب برای صفحات وب

کپی برداری بدون ذکر نام منبع مجاز نیست

منظور از قالب سازی برای صفحات چیست؟ (مقایسه با روش

parsi e-book

های دیگر...)

قالب سازی بدین معنی است که با ایجاد یکسری نقاط مشترک

برای تمامی صفحات بصورت جداگانه و قرار دادن نوعی ارجاع

از آن نقاط در تمامی صفحات وب، یک نوع مدیریت جامع را به

ساختار وب سایت تحمیل کنیم. تعریف ذکر شده کمی گنگ و

پیچیده به نظر می آید. بنابراین با ذکر مثالی سعی در باز کردن

مساله فوق می نمایم.

فرض کنید که قرار است سایتی را با بیش از ده صفحه ایجاد

parsi e-book

WWW.PARSI-BOOK.4T.COM

نمایید. مسلما برای ایجاد این سایت طرحی را در نظر می گیرید.

بطور مثال اینچنین فرض می کنید که آرم و نام سایت را در

قسمت بالا قرار می دهید، منوهای اصلی سایت را در سمت

راست، در پایین موارد مربوط به کپی رایت، در چپ تبلیغات

سایت و نهایتا فضای باقیمانده را به قسمت اصلی سایت (که از

این پس آنرا Main Part می نامیم) یعنی مکانی برای نمایش

اطلاعات و عملکرد سایت خود اختصاص می دهید. مسلما تمامی

قسمت های ذکر شده در تمامی صفحات ثابت هستند (یا

تقریبا ثابت هستند) و فقط فضای باقیمانده (Main Part) که

معمولا بیشترین فضای صفحه را به خود اختصاص می دهد پویا

می باشد، که در این مکان با توجه به درخواست و عملکرد کاربر

اطلاعاتی نمایش داده می شود. حالت معمول طراحی بدین گونه

هست که یکبار شمای ذکر شده طراحی و سپس در تمامی

صفحات قرار می گیرد. پس از آن در هر صفحه با توجه به نیاز

آن، مواردی به Part Main آن اضافه می شود (مواردی چون

forms, buttons, datagrid، انواع کنترل‌های وب، متن های

مورد نیاز صفحه و...).

حال فرض کنید که قرار شود یک منوی جدید به لیست منوها

اضافه گردد. بنابراین مجبور هستیم که این منو جدید را در

تمامی ده صفحه فوق اضافه کنیم. شاید در نگاه اول این مساله

مهمی نباشد ولی اگر صفحات فوق به صد صفحه و یا بیشتر

افزایش یابد، دیگر این مساله کوچک نیست.

parsi e-book
WWW.PARSIBOOK.4T.COM

استفاده از فریم(Frame)ها به عنوان اولین راه حل

شاید خیلی ها به فریم های موجود در HTML متوسل شوند، که

این به نوبه خود اگر چه مفید هست ولی سختی ها و کاستی های

خود را دارد. در استفاده از فریم ها نیاز به دانش HTML،

آشنایی با مدل شی سند (DOM) و همچنین تا حدودی نیاز به

JavaScript می باشد. در ضمن در حالت های پیشرفته تر نیاز

بیشتری به موارد فوق است و این در حالی است که یک ساختار

اجباری را به سایت نیز تحمیل می کند. بطور مثال اسکرولهای هر

فریم جداگانه می باشد و اغلب در بسیاری از حالتها نحوه

نمایش صفحه را از یک حالت پیوسته به صورتی گسسته درمی

آورد. به عنوان نمونه در نظر بگیرید منوهای سمت راست

صفحه در یک فریم راست و صفحه اصلی (Main Part) در فریم

سمت چپ تعریف شود. حال در صورتی که منوها و متن صفحه

اصلی از اندازه صفحه خارج شوند، نیاز به اسکرول پیدا می کنند

و در نتیجه در صفحه دو اسکرول یافت می شود که این ساختار

معمولا کاربر پسند نیست. همچنین در Address bar مربوط

به اکسپلورر فقط آدرس فریم دربرگیرنده ظاهر می شود، که

در صورت فشردن دکمه refresh اکسپلورر، دوباره به

صفحه اول باز می گردد که این به نوبه خود یک فاجعه در

navigating می باشد!

parsi e-book
WWW.PARSIBOOK.4T.COM

پس راه حل چیست؟

خوب مسلماً اولین راه حلی که از دید آشنایان با ASP.NET

بنظر می رسد، استفاده از کنترل‌های کاربر یا همان User

controls می باشد. مسلماً این یک راه حل بسیار خوب و جامع

است ولی معایبی دارد که آنرا نیز می توان با اعمال روشهایی

خاص برطرف ساخت. از آنجایی که راه حل اصلی مورد نظر در

این مقاله بر پایه همین مدل است، ابتدا به بررسی این روش

پرداخته و سپس آنرا کامل می کنیم. بنابراین برای درک مرحله

بعد نیاز است که این مرحله بخوبی درک شده باشد. در ضمن

در کدهای زیر از روش code behind استفاده شده است تا

مطلب واضح تر باشد.

قالب سازی به کمک User Control ها

نحوه عمل

در این روش بدین صورت عمل می شود که قسمتهای یکسان

موجود در تمام صفحات در قالب User control ها تعریف می

شوند و سپس این کنترلها در تمامی صفحات قرار داده می

شوند. برای بررسی این روش به یک مثال نیاز داریم. در این

مثال بدین گونه فرض می شود که صفحه وب ما شامل سه

قسمت می باشد:

(۱) مکان لوگو		
(۲)		(۳) قسمت اصلی (Main)

منوها	(Part
-------	-------

و با انتخاب هر منو متن قسمت ۳ یا همان Main Part تغییر می

کند.

طراحی بدین صورت شکل می گیرد که به جز قسمت ۳ که

همیشه غیر ثابت و متغیر هست، قسمت های ۱ و ۲ بصورت

User Control (که از اینجا به بعد به منظور خلاصه سازی با نام

اختصاری UC نامبرده می شود) تعریف و استفاده می شوند. این

قسمت ها را پنل (Panel) نیز می گویند. پنل ۱ را به صورت زیر

تعریف می کنیم: (فایل با نام Panel_logo.ascx) این تعریف

کنترل panel_logo می باشد. در ضمن در این قسمت از

هیچگونه برنامه نویسی استفاده نشده است.

```

---file                                     Panel_logo.ascx---

Control                                     Language="vb"                                @%.>

"AutoEventWireup="false

" =Inherits="site.Panel_logo" TargetSchema

</. "http://schemas.microsoft.com/intellisense/ie5

src="images/logo.gif" "IMG height="74>

<"border="0 "width="370" vspace="2

```

نکته مهم: باید توجه کنید که به هیچ وجه در UCها از فرمهای با

ویژگی runat=server استفاده نشود. اگرچه این کار مجاز

است ولی عملاً کاربرد کنترل شما را بسیار محدود می کند.

توجه کنید، کنترلهایی که بدین صورت ایجاد می گردند در

هنگام ترجمه کدهای HTML آن بصورت مستقیم در مکان

فراخوانی کنترل قرار خواهند گرفت. حال کنترل دوم را ایجاد می

کنیم. این کنترل را بصورتی ایجاد می کنیم که با انتخاب هر منو،

پس از رفتن به صفحه مربوطه، منوی مربوط به آن صفحه غیر

فعال شود:

```
---Panel_right.ascx file---

"Control Language="vb @/>

AutoEventWireup="false"

"Codebehind="Panel_right.ascx.vb

" =Inherits="site.Panel_right" TargetSchema

</. "http://schemas.microsoft.com/intellisense/ie5
```

```

<table>
<tr>
A><"td      id="td1"      runat="server">
<td/><A/>  Home  Page  <"href="index.aspx
A      ><"runat="server      "td      id="td2">
<td/><A/>      Searching      <"href="search.aspx
A><"td      id="td3"      runat="server">
<td/><A/>  About  Us  <"href="AboutUs.aspx
<tr/>
<table/>

```

همانطور که دیده می شود در این پنل سه لینک به سه صفحه

متفاوت موجود است. همچنین این سه لینک هر کدام در یک

تگ <td> با ویژگی runat=server قرار داده شده اند، تا بتوان

از آنها در Codebehind مربوطه استفاده نمود. توجه کنید که

قرار است با رفتن به هر صفحه، به این کنترل بفرماییم که در

کدام صفحه است و کدام لینک را باید غیرفعال کند. پس نیاز

است که یک property برای این کنترل معرفی شود که این کار

را برای ما انجام دهد. بنابراین در Codebehind مربوط به پنل

راست داریم:

```
---Panel_right.ascx.vb                                     file---  
  
td1      As      Protected      WithEvents  
      System.Web.UI.HtmlControls.HtmlTableCell  
Protected      WithEvents      td2      As  
      System.Web.UI.HtmlControls.HtmlTableCell  
Protected      WithEvents      td3      As
```

System.Web.UI.HtmlControls.HtmlTableCell

که باعث معرفی td1,td2,td3 در کد می شود. سپس

property مربوطه به صورت زیر تعریف می شود:

```
Public WriteOnly Property current_page() As Integer
    Get
        Return 1
    End Get
    Set(ByVal Value as Integer)
        Select Case Value
            Case 1
                "td1.InnerHtml" = "Home Page"
            Case 2
                "td2.InnerHtml" = "Searching"
            Case 3
                "td3.InnerHtml" = "About Us"
        End Select
    End Set
End Property
```

Set	End	End Property
-----	-----	--------------

بدین ترتیب مثلاً اگر خصیصه `current_page` به مقدار ۲

تنظیم شود، به جای لینک `searching` فقط متن آن قرار

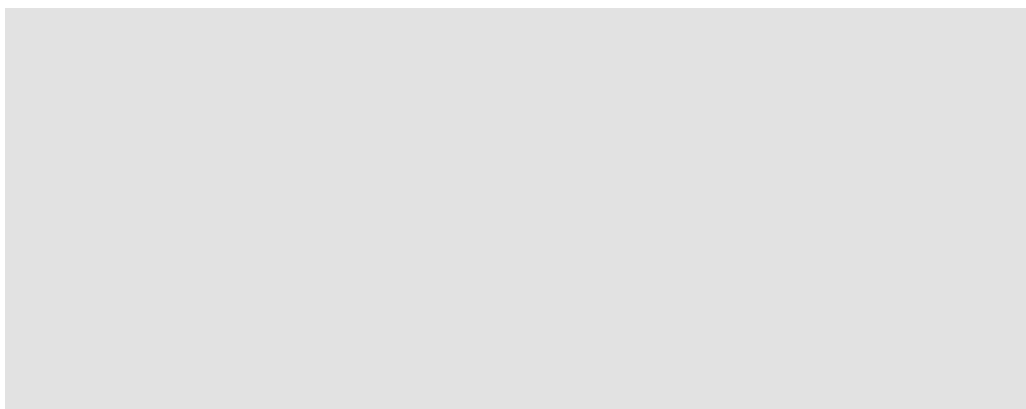
خواهد گرفت که نشانگر غیرفعال بودن آن لینک و یا به عبارتی

نشانه قرار داشتن در صفحه `searching` می باشد.هم اکنون

جزئیات قالب ما معین است و فقط کافی است که شکل صفحه

قالب را معین کنیم. صفحه قالب را با نام `template.aspx`

ایجاد و از آن استفاده می نمایم:



```

</
Register      TagPrefix="Mysite"      @%>

</ "Src="Panel_logo.ascx "TagName="Panel_logo

Page      Language="vb"      @%>

      "AutoEventWireup="false
      "Codebehind="template.aspx.vb
      </"Inherits="site.template

      <HTML>

<head/>      My      site      <head>

      <body>

      <table>

      <tr>

      <"colspan="2      td>

      "mysite:panel_logo      id="panel_logo1>

```

```
<mysite:panel_logo/><"runat="server
```

```
<td/>
```

```
<tr/>
```

<td>متن و کنترل‌هایی که مخصوص هر صفحه هست و جداگانه

تعریف می شوند در اینجا قرار [می گیرند(Main Part)

```
<td/>
```

```
<td>
```

```
"id="panel_right1
```

```
mysite:panel_right>
```

```
<mysite:panel_right/><"runat="server
```

```
<td/>
```

```
<tr>
```

```
<tr/>
```

WWW.PARSI

```
<table/>
```

```
<body/>
```

```
<HTML/>
```

بدین ترتیب قالب ما ساخته می شود و برای ایجاد هر صفحه

جدید کافی است کدهای فوق (البته به غیر از directive

مربوط به page که باید مخصوص همان صفحه باشد) را در

آنجا قرار داد. نمونه ساده ای از استفاده این قالب به صورت

زیر است: (بطور مثال برای صفحه aboutus.aspx)

```

</
TagPrefix="Mysite" Register @%.>

</. "TagName="Panel_logo" Src="Panel_logo.ascx

Page Language="vb" @%.>

AutoEventWireup="false"

"Codebehind="aboutus.aspx.vb

</. "Inherits="site.aboutus

<HTML>

<head/> My site <head>

<body>

<table>

<tr>

<"colspan="2 td>

"mysite:panel_logo id="panel_logo1>

```

<mysite:panel_logo/><"runat="server

 ||
 <td> درباره ما... سابقه کاری شرکت ما به صورت زیر می |

باشد: $\dots-2\dots-1$

 | |

"id="panel_right1

mysite:panel_right>

"current_page="3

<mysite:panel_right/><"runat="server

 ||
| | |

<table/>

<body/>

<HTML/>

همانطور که در این کد ملاحظه می شود، directive مربوط به

Page تغییر کرده است، متن قسمت "صفحه اصلی" (Main

Part) نیز با متن مورد نیاز در این صفحه تنظیم شده و نهایتاً

ویژگی current_page، از کنترل panel_right، با مقدار ۳

تنظیم شده که باعث می شود لینک سوم در هنگام بازدید از

این صفحه غیرفعال گردد. توجه کنید که تغییرات فوق فقط

مختص این صفحه است و به هیچکدام از صفحات دیگر در

ارتباط نیست. بنابراین اصل قالب بندی به هیچ وجه زیر سوال

نمی رود. حال برای درک بیشتر این قضیه فرض کنید که ۲۰۰

صفحه با قالب فوق ایجاد کردید و حال تصمیم می گیرید که در

پنل مربوط به لوگو مکانی را برای قرار دادن یک عکس تبلیغاتی

اختصاص دهید. فقط کافی است که بدون کنترل

panel_logo یک خط بصورت زیر اضافه شود:

```
IMG height="74" src="images/adv.gif" width="570  
<"vspace="2" border="0 "
```

همین!... با تغییر فوق این تغییر را تمامی صفحات مشاهده

خواهید کرد و دیگر لازم نیست که این خط را به تمامی صفحات

اضافه و از درست عمل کردن آن مطمئن شوید. و یا اینکه

تصمیم می گیرید یک منوی جدید به لیست منوها اضافه کنید.

همانند مورد فوق خط مشابهی را در صفحه

panel_right.ascx اضافه کرده و یک case دیگر نیز به

دستور select در behind code مربوطه اضافه می کنید و

دیگر تمام! حتی لازم نیست current_page مربوط به صفحات

دیگر را تغییر دهید. همچنین شما می توانید به کنترل های خود

ویژگی visible را هم اضافه نمایید تا در صفحات مورد نیاز از

نمایش بعضی پنل ها (مانند پنل تبلیغات) جلوگیری به عمل

آورید. چیزی که اجرای آن با فریم ها بسیار مشکل و دردسر

ساز می باشد. همانطور که دیده می شود، اصول شئ گرایی

تماما در این روش رعایت شده است.

در این روش شما می توانید با استفاده از caching کنترلها،

سرعت و کارایی برنامه را تا حد قابل توجهی بالا ببرید.

همچنین در موارد پیشرفته تر و تعداد صفحات بالا مشاهده

خواهید کرد که:

- از حجم برنامه شما بسیار کاسته می شود.
- خوانایی کدهای HTML شما افزایش می یابد و در نتیجه ایراد

زدایی از کدهای HTML نیز بسیار آسانتر خواهد شد.

- سرعت کار و تمرکز شما نیز بطور قابل توجهی افزایش می

یابد.

روش ذکر شده بسیار خوب و کامل است. ولی این روش

ایراداتی دارد که پس از اشاره به آنها، به نحوه رفع آن با روش

دوم خواهیم پرداخت:

فرض کنید پس از مدتی تصمیم می گیریم که ظاهر سایت را

تغییر دهیم. مثلاً یک پنل تبلیغات به سمت چپ اضافه کنیم. اگر

دقت کنید، با مشکلی همانند اولین مشکل ذکر شده در مقاله

برخورد خواهیم کرد. در اینجا دیگر پنلی به این نام در صفحات

ما وجود ندارد و ما مجبور خواهیم بود که این پنل را ایجاد و در

تمامی صفحات قرار دهیم. و این در حالی است که ساختار

HTML موجود در صفحات بهم نخورد. اغلب ساختار HTML

صفحات بسیار پیچیده تر از مثال فوق است و بنابراین نیاز است

که در کدهای HTML تمامی صفحات نیز بازنگری شود و این

نیز بر مشکل اول افزوده می شود. مشکل اساسی آنجا آغاز می

شود که قصد کنیم ظاهر سایت را کاملاً تغییر داده و کل قالب

را عوض کنیم. اگر چه شاید این مشکل مانند مشکلات مدل

ایجاد صفحات بدون قالب نباشد، ولی حداقل نیاز است که این

تغییرات برای تمامی صفحات موجود اجرا شود و قالب جدیدی

نیز ساخته شود. پس از آن است که دیگر نیاز به تغییرات

نیست، ولی اگر دوباره نیاز به اینچنین تغییری پیدا شد چه؟...

مسئله این راه حل کاملاً پاسخگو نیست.

در راه حل زیر مشکل فوق بررسی و برطرف می شود:

قالب سازی پویا بوسیله سگمنت کردن قالب اولیه اگر در روش

قبلی دقت کرده باشید، مشکل از وجود کدهای HTML موجود

در قالب بود و اگرچه در تمامی صفحات ثابت بودند ولی با

اندکی تغییر در ساختار صفحه نیاز به تغییر آن و در نتیجه تغییر

قالب و نهایتاً قالب گیری مجدد صفحات ساخته شده است. اگر

بتوان بصورتی عمل کرد که قالب اصلی ما همیشه شکل ثابتی

داشته باشد و بهیچ وجه تغییر نکند، مشکل برطرف می شود. به

عبارتی با قالب سازی برای قالب اولیه و یا به عبارتی پویا سازی

قالب، دیگر و در هیچ شرایطی نیاز به تغییر قالب مورد استفاده

صفحات را نداشتیم و با تغییر شکل قالب، خودبخود در تمامی

صفحات نیز تغییرات صورت می گرفت. قبل از بررسی روش

مورد نظر توجه شما را به نکته ای جلب می کنم:

همانطور که احتمالا همه شما تا کنون متوجه شده اید، هر سایت

مجموعه ای از پنل های فرعی و تقریبا ثابت، و یک قسمت اصلی

(Main Part) که حاوی مطالب صفحه مورد نظر است می

باشد. برای درک بیشتر این موضوع همین الان به چند سایت

مختلف نگاه کنید... بطور مثال همین سایت IranASP.NET

شامل دو پنل چپ و راست و یک پنل بالا و پایین است. قسمت

وسط آن نیز، قسمت اصلی صفحه (Main Part) را تشکیل می

دهد. در سایت یاهو نیز در صفحه مربوط به میل آن یک پنل

در چپ، یک پنل در بالا و یک پنل تبلیغات در بالاترین نقطه

وجود دارد. در قسمت راست صفحه هم اصلی ترین قسمت)

(Part Main) وجود دارد. می توان گفت که این قسمت یا همان

Main Part بیشترین فضای صفحه را در تمامی سایتها اشغال می

کند. پس هم اکنون به این امر واقف هستید که "قسمت اصلی"

جزو لاینفک هر سایت هست و کلیه تغییرات در سایت حول این

قسمت می باشد.

برای توضیح این روش از مثال ذکر شده در روش قبل استفاده

می کنیم و آنرا تکمیل می نماییم. اکنون همانند مثال ذکر شده

در روش قبل یک قالب ایجاد کنید. مکانی که باید کدهای

قسمت اصلی در آنجا قرار بگیرند را در نظر بگیرید. از همین

مکان کدها را به دو قسمت بالایی و پایینی تفکیک کنید. یعنی

کدهای HTML ای که بالای قسمت Main Part قرار دارند به

صورت زیر خواهد بود:

```
TagPrefix="Mysite" Register @%>

"TagName="Panel_right" Src="Panel_right.ascx

<%.

Register TagPrefix="Mysite" @%>

<%. "TagName="Panel_logo" Src="Panel_logo.ascx

<%. "Inherits="site.template

<HTML>

<head/> My site <head>
```

```

<body>

<table>

  <tr>

    <"colspan="2                                td>

      "mysite:panel_logo                      id="panel_logo1>

        <mysite:panel_logo/><"runat="server

                                          <td/>

                                          <tr/>

                                          <td>

```

و بلافاصله پس از خط آخر کدهای قسمت اصلی قرار دارند:

of the page, must put Codes that are for main part]

[.here

parSI
WWW.PARSIBOOK.4T.COM

و بلافاصله پس از این کدها، کدهای HTML مربوط به ادامه

قالب قرار دارند:

```
<td/>

<td>

" id="panel_right1"      mysite:panel_right>

<mysite:panel_right/><"runat="server

<td/>

<tr>

<tr/>

<table/>

<body/>

<HTML/>
```

هم اکنون این دو قسمت بالایی و پایینی را در دو کنترل جداگانه

تعریف می کنیم. اولین کنترل را سگمنت بالایی می نامیم: (

کپی برداری بدون ذکر نام منبع مجاز نیست

(segment_top.ascx

narsi e-book

---segment_top.ascx---

TagPrefix="Mysite" Register @/.>

</ "TagName="Panel_logo" Src="Panel_logo.ascx

Control Language="vb" @/.>

AutoEventWireup="false"

"Codebehind="segment_top.ascx.vb

" =Inherits="site.segment_top" TargetSchema

</ "http://schemas.microsoft.com/intellisense/ie5

<HTML>

www.narsi.com

```

<head/>           My           site           <head>

<body>

<table>

    <tr>

        <"colspan="2           td>

        "mysite:panel_logo           id="panel_logo1>

            <mysite:panel_logo/><"runat="server

            <td/>

        <tr/>

    <td>

```

از آنجایی که کد HTML فوق شامل پنل راست نیست،

directive مربوطه را نیز از آن حذف می کنیم. همچنین این

قسمت دارای code behind خاصی نیست و از آن صرف نظر

می کنیم.

کنترل دوم را سگمنت پایینی می نامیم و شامل کدهای قسمت

پایینی قالب است (segment_down.ascx):

کی برداری بدون ذکر نام منی

```
---segment_down.ascx---

"Register          TagPrefix="Mysite          @/.>

"TagName="Panel_right"    Src="Panel_right.ascx

</

Language="vb"          Control          @/.>

AutoEventWireup="false"

"Codebehind="segment_down.ascx.vb

" =Inherits="site.segment_down" TargetSchema

</ "http://schemas.microsoft.com/intellisense/ie5

<td/>
```

www

```

<td>
" id="panel_right1"      mysite:panel_right>
<mysite:panel_right/><"runat="server
<td/>
<tr>
<tr/>
<table/>
<body/>
<HTML/>

```

همانطور که دیده می شود، directive حذف شده در این

سگمنت آورده شده است. همچنین کدهای HTML نیز در

هیچکدام از دو کنترل فوق بصورت کامل نمی باشند (به عبارتی

well formed نیست) و در واقع بصورت تنها، اشتباه و بدون

کاربرد می باشد. بنابراین توجه داشته باشید که این دو

سگمنت با هم بکار برده می شوند. همچنین VS.NET در بعضی

شرایط خود اقدام به تکمیل این کدها می کند که در صورت

مشاهده این عمل، خود اقدام به حذف کدهای اضافی

نمایید. انجام یک کار در هنگام ایجاد سگمنت ها مورد اجباری

است و آن اینکه از آنجا که پس از استفاده از صفحه قالب ایجاد

شده، امکان دسترسی به خصوصیات کنترل‌های موجود در این

سگمنت ها نیست، باید این خصوصیات را به این سگمنت ها

انتقال داد. بطور مثال در پنل راست خصیصه ای بود که با آن

صفحه جاری را معین می کردیم. حال این خصیصه را به صورت

زیر به عنوان یک خصیصه در سگمنت پایینی (چون این کنترل در

این سگمنت قرار دارد) تعریف می کنیم. ابتدا باید کنترل

panel_right را بصورت زیر در درون codebehind تعریف

کنیم، زیرا بصورت خودکار این تعریف ایجاد نمی شود.

---segment_down.ascx.vb---

Panel_right WithEvents Panel_right1 As Protected

سپس این خصیصه را به کنترل سگمنت پایین اضافه می کنیم:

```
Integer WriteOnly Property current_page() As
(Set(ByVal Value As Integer
Value = Panel_right1.current_page
End Set
Property End
```

parsi e-book
WWW.PARSIBOOK.4T.COM

بدین ترتیب این خصیصه را به کنترل پایینی اضافه کرده ایم.

حال صفحه قالب ما در تمامی حالات فقط از شکل زیر برخوردار

است:

```
---template.aspx---

Language="vb"                                Page           @/>

AutoEventWireup="false"

</.                                "Codebehind="template.aspx.vb

Register                                TagPrefix="mysite"           @/>

                                "TagName="segment_down

</.                                "Src="segment_down.ascx

"Register                                TagPrefix="mysite           @/>
```

```

TagName="segment_top" Src="segment_top.ascx
</
"

id="Segment_top1"      mysite:segment_top>
<mysite:segment_top      /><"runat="server
"mysite:segment_down      id="Segment_down1">
      <mysite:segment_down /><"runat="server

```

فقط اینکه در صفحات مورد استفاده از قالب ها، page

directive باید مربوط به خود آن صفحه باشد.

حال به عنوان نمونه صفحه aboutus.aspx را با این قالب

جدید پیاده سازی می کنیم:

```

---aboutus.aspx---

"Page      Language="vb      @/>

```

```

AutoEventWireup="false"

</.          "Codebehind="aboutus.aspx.vb

TagPrefix="mysite"          Register          @/.>

TagName="segment_down"

</.          "Src="segment_down.ascx

Register          TagPrefix="mysite"          @/.>

Src="segment_top.ascx "TagName="segment_top

</.          "

"mysite:segment_top          id="Segment_top1          >

<mysite:segment_top          /><"runat="server

...-۱-...

...-۲-...

id="Segment_down1"          mysite:segment_down          >

```

```
</> "runat="server" current_page=" 3
```

```
<mysite:segment_down
```

همانطور که ملاحظه می شود، متن قسمت اصلی در میان دو

سگمنت قرار می گیرد. همچنین مقدار خصیصه

current_page نیز در سگمنت پایینی معین شده است. حال

فرض کنید می خواهیم یک پنل تبلیغات در سمت چپ قرار

دهیم. فقط کافی است در سگمنت بالایی کدهای زیر را در

انتهای آن اضافه کنیم: (فرض کنید کنترل تبلیغات به اسم

panel_adv.ascx طراحی شده است)

```
TagPrefix="Mysite" Register @/>
```

```
</ "TagName="Panel_adv" Src="Panel_adv.ascx
```

```
runat="server" mysite:panel_adv id="panel_adv1">  
  
</myite:panel_adv/></>
```

و فقط همین! دیگر نیازی به تغییر در هیچکدام از فایلها نیست و صفحه "درباره ما" با شکلی جدید به نمایش خواهد آمد. در این شرایط اگر ظاهر سایت نیز به کلی تغییر کند طوری که تمام

کدهای HTML عوض گردند، حداکثر تغییر در کنترل های سگمنت بالایی و پایینی است. یعنی حداکثر کدهای HTML این دو تغییر خواهند کرد و تمامی صفحات از قالب جدید پیروی خواهند کرد.

در دو طراحی آخر فقط به این مورد می توان اشاره کرد که در

صورت استفاده از قالب بندی به یکی از این دو روش، در هر صورت تمامی فایل های سایت شما باید تبدیل به aspx شوند.

شاید این از نظر برخی ناخوشایند باشد. بطور مثال تمامی

صفحات با محتوای ثابت که می توانند به صورت HTML نیز

باشند، باید به صورت aspx در آیند و از آنجا که هیچ برنامه

نویسی سمت سرور در این فایلها نیست، در واقع کار بی موردی

انجام داده ایم و به سرور بار اضافی تحمیل کرده ایم. خوب این

بستگی به طراح سایت دارد که کدام مدل برای وی بیشتر به

صرفه باشد. چیزی که اینجا مشخص است، برای سایت های با

حجم و ترافیک محدود شاید چند صفحه ثابت ارائه شده

بصورت aspx آنچنان بر عملکرد سایت موثر نباشد. در ضمن

قابل ذکر است که این یک مدل و شیوه است و اجباری بر اجرای

آن بر روی تمامی صفحات نیست. در آخر به عنوان نمونه ای از

کارهای انجام شده با این روش می توان به سایت

www.hudhud-dubai.com و (بزودی) اشاره کرد.

کپی برداری بدون ذکر نام منبع مجاز نیست
parsī e-book



parsi e-book
WWW.PARSIBOOK.4T.COM