



### فصل دوم :

## ایجاد و استفاده از User controls

مقدمه :

ASP.NET برای ایجاد کنترل های وب دو مدل را ارائه داده است: کنترل های کاربر (User controls) و کنترل های سفارشی (Custom controls).

کنترل های کاربر همانند یک وب فرم استاندارد نوشته می شوند. بنابراین ایجاد آنها با استفاده از محیط ویژوال نسبتا ساده است. کنترل های کاربر به صورت ساده ای روی صفحه قرار گرفته و خواص آنها را نمی توان از طریق پنجره ی خواص تنظیم کرد. همچنین امکان اضافه کردن آنها به Toolbox نیز وجود ندارد. به اشتراک گذاشتن آنها صرفا از طریق قرار دادن فایل های یک کنترل کاربر در دایرکتوری برنامه صورت می گیرد.

اما بر خلاف کنترل های کاربر که همانند فرم های وب نوشته می شوند ، کنترل های سفارشی کامپایل شده و با فرمت بایناری به شکل dll ارائه می گردند. کنترل های سفارشی بدون کمک طراح ویژوال ساخته می شوند و بنابراین نوشتن آنها مشکل تر است. چون کامپایل شده هستند قابلیت افزودن به Toolbox را داشته و بنابراین نحوه ی استفاده از آنها همانند یک کنترل متداول وب و مانند TextBox و غیره است. همچنین تنظیم خواص آنها به کمک پنجره ی خواص نیز میسر است.

هنگام بکار گیری کنترل های کاربر و یا کنترل های سفارشی بحث کارآیی اهمیت خاصی ندارد. زیرا هر دو از فضای نام Sytem.Web.UI.Control مشتق می شوند و کارآیی یکسانی خواهند داشت. در هنگام انتخاب بین یکی از این دو می توان به نکات زیر توجه کرد:



- آیا کنترل مختص به یک برنامه ی خاص نوشته می شود و یا عمومی است و قابل توزیع برای برنامه های دیگر؟ اگر کنترل برای یک برنامه ی خاص نوشته می شود و این برنامه حاوی Layout های استاتیک بسیاری است مانند Header یک سایت و یا Footer آن ، استفاده از کنترل های کاربر معنا پیدا می کند. اگر کنترل عمومی است و باید قابلیت توزیع داشته باشد ، Layout آن دینامیک است مانند اغلب کنترل های استاندارد وب ، کنترل های سفارشی مناسب تر خواهند بود.

بنابراین می توان نتیجه گرفت که :

- ۱- سادگی در نوشتن : کنترل های کاربر برنده ی آن خواهند بود.
- ۲- اضافه شدن به تول باکس : کنترل های سفارشی این قابلیت را دارند.
- ۳- توزیع : کنترل های سفارشی به صورت dll و یک فایل باینری کامپایل می شوند و توزیع ساده تری دارند.
- ۴- Layout : کنترل های کاربر برای Layout های استاتیم بسیار مناسب هستند در حالیکه کنترل های سفارشی برای Layout های پیچیده مناسبند.
- ۵- کارآیی : در هر دو یکسان است.

### کنترل های کاربر:

توانایی یک کنترل کاربر بسیار بیشتر از بکار گیری آن صرفا به عنوان Header و Footer تمام صفحات یک وب سایت است. آنها می توانند هر دو کنترل های HTML و وب را در خود جای داده و همچنین دارای خواص و متدهای پابلیک نیز باشند. بنابراین این کنترل ها شبیه به کنترل های ActiveX در برنامه های معمول تحت ویندوز می توانند در ASP.NET عمل کنند. اغلب اوقات از کنترل های سفارشی برای ساخت قسمت های ورود اطلاعات ، Login نمودن و جستجو کردن سایت ها بکار برده می شوند.

کنترل های کاربر را می توان در هر دو زمان طراحی و همچنین اجرای برنامه ، در صفحات بکاربرد. بنابراین در این مورد بسته به موضوع کنترل کاربر می توان بر مبنای حالت هایی مانند ورود اطلاعات و یا نمایش اطلاعات دینامیک تصمیم گیری کرد.

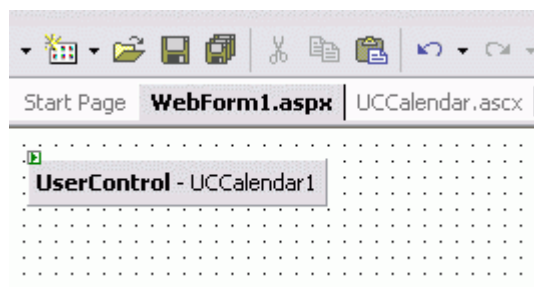
یکی از موارد مهم همراه با کنترل های کاربر توانایی استفاده ی آنها از موتور Caching وابسته به ASP.NET می باشد. در این صورت قسمتی از صفحه که کنترل کاربر در آن قرار می گیرد و حاوی اطلاعات استاتیک است ، بسیار سریعتر بارگذاری شده و یکی از موارد بهبود کارایی برنامه به شمار می رود (در این مورد در طی یک فصل جداگانه بحث خواهد شد) .

بنابراین به صورت خلاصه یک کنترل کاربر ASP.NET مجموعه ای از یک یا چند کنترل سرور یا HTML است که در مجموع عملیات خاصی را ارائه می دهند. یک کنترل کاربر می توان به سادگی توسعه ی توانایی های یک کنترل موجود باشد برای مثال تقویمی با یک تکست باکس که روز انتخاب شده در آن تکست باکس نمایش داده شود و یا ترکیبی از چند کنترل و از حاصل بر هم کنش آنها یک کار واحد صورت گیرد.

مثال ۱:

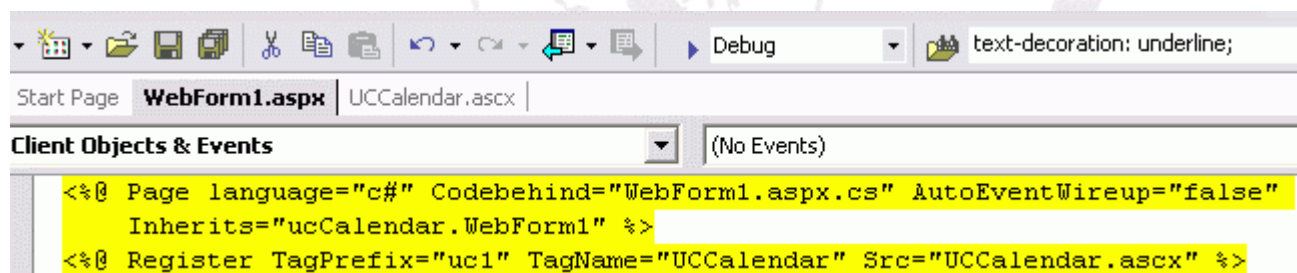
### ایجاد یک کنترل کاربر مشتق شده از کنترل تقویم

- ۱- در ابتدا یک پروژه ی C# ASP.NET Web Application جدید را شروع کنید.
- ۲- از منوی پروژه گزینه ی Add new item را انتخاب نموده و آیتم Web User Control را برگزینید. نام UCCalendar را وارد کرده و سپس روی Open کلیک کنید. این مورد سبب می شود تا فایل با پسوند .ascx به پروژه اضافه شود. این فایل است که طراحی ظاهر کنترل کاربر در آن انجام می گیرد. یک فایل .ascx را مستقلا نمی توان در مرور گر مشاهده کرد و باید درون یک Container قرار گیرد ( مانند یک فرم وب ) تا قابل مشاهده شود.
- ۳- این فایل .ascx را از قسمت Solution explorer برنامه که در سمت راست صفحه قرار دارد ، بر روی فرم برنامه drag کنید. شکل زیر نمایش داده خواهد شد.



شکل ۱ - پس از drag کردن یک کنترل کاربر بر روی فرم ، شکل نهایی آن به صورت فوق خواهد بود.

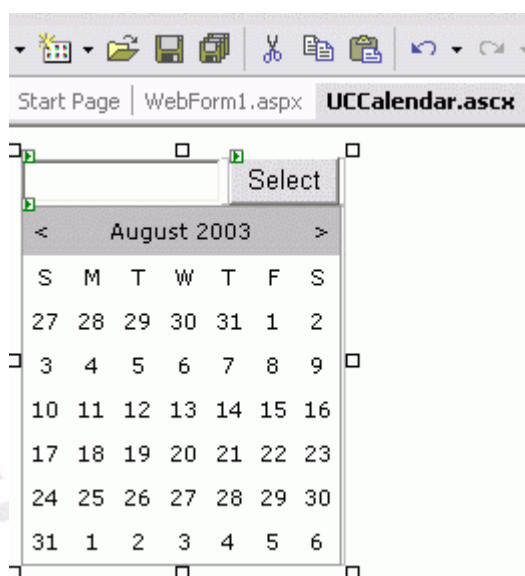
باید خاطر نشان کرد که کنترل هیچگونه نمای ظاهری نخواهد داشت. این مورد می تواند در حالتیکه کنترل فضای زیادی را لازم دارد در مورد قرار گیری سایر کنترل ها روی صفحه در هنگام طراحی مشکل چیدن آنها را ایجاد کند. بنابراین می توان اندازه ی آنها را با تنظیم ارتفاع و عرض یک جدول حاوی آن، مناسب با کار و اندازه ی واقعی آن در آورد. در پایین صفحه قسمت سمت راست ، روی دکمه HTML کلیک کنید تا سورس HTML مانند صفحه ظاهر شود (شکل ۲).



شکل ۲ - نحوه ی رجیستر کردن یک کنترل کاربر به صورت خودکار توسط VS.NET در شکل نمایان است.

در خط سوم عبارت @Register را مشاهده می نمایید. این مورد کنترل را بر روی صفحه ثبت می نماید. هنگامیکه کنترل رجیستر شد ، هر تعداد از آنها می توان به سادگی بر روی فرم قرار داد.

توسط TagName نام کنترل مشخص شده است و توسط Src نام فایل کنترل بیان می شود. دو کنترل Calendar و TextBox را روی فرم کنترل کاربر قرار دهید. با استفاده از کد نویسی کار تکمیل و تمام می شود.



شکل ۳- نمای کنترل کاربر جدید ما در حالت طراحی.

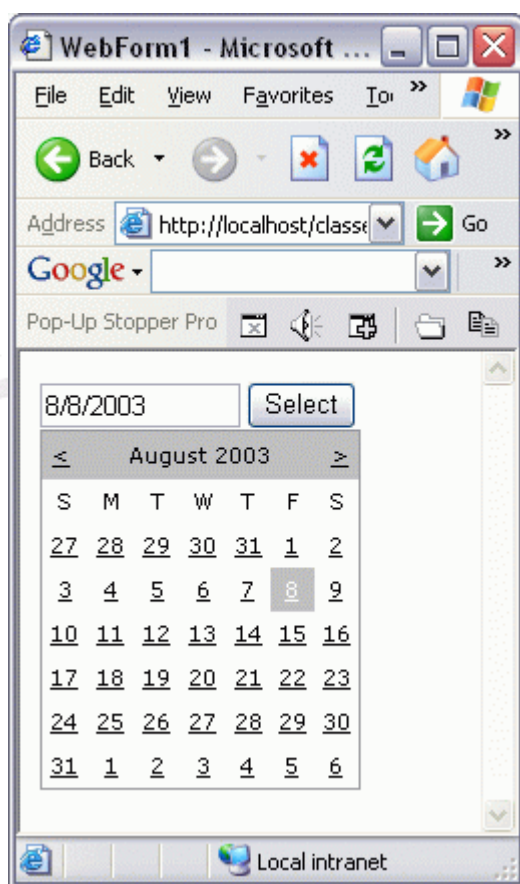
روی تقویم (با نام calDate) دوبار کلیک نموده و کد زیر را اضافه کنید.

```
protected void calDate_SelectionChanged(Object sender, EventArgs e)
{
    System.DateTime dtDate = calDate.SelectedDate;
    txtDate.Text = dtDate.ToShortDateString();
}
```

روی دکمه ای (با نام btnSelect) که در صفحه گذاشته شده است دوبار کلیک نمایید و کد زیر را در روال رخداد آن بنویسید:

```
private void btnSelect_Click(object sender, System.EventArgs e)
{
    if ( calDate.Visible == false )
    {
        calDate.Visible = true;
    }
    else
    {
        calDate.Visible = false;
    }
}
```

اکنون کار کد نویسی کنترل جدید ما به پایان رسیده است. دکمه ی F5 را بفشارید تا برنامه اجرا شود. با کلیک کردن بر روی دکمه ، تقویم نمایش داده شده و با انتخاب روزها در تقویم ، روز انتخاب شده در تکست باکس نمایش داده می شود (شکل ۴). به این صورت کنترل جدیدی مشتق شده از چند کنترل موجود با توانایی بیشتر خلق گردیده است.

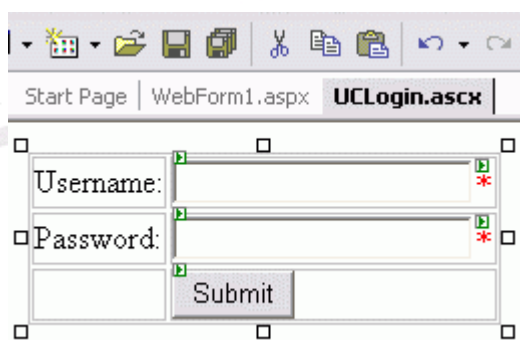


شکل ۴- اجرای نهایی برنامه در مرورگر وب.

مثال ۲ :

## ایجاد یک کنترل کاربر برای Login نمودن

در ابتدا یک پروژه ی C# ASP.NET Web Application جدید را شروع کنید. از منوی پروژه گزینه ی Add new item را انتخاب نموده و آیتم Web User Control را برگزینید. فایل UCLLogin.ascx را به پروژه اضافه نمایید. آنرا به صورت زیر طراحی نمایید ( ستاره ها در شکل همان Required field validators هستند ) :

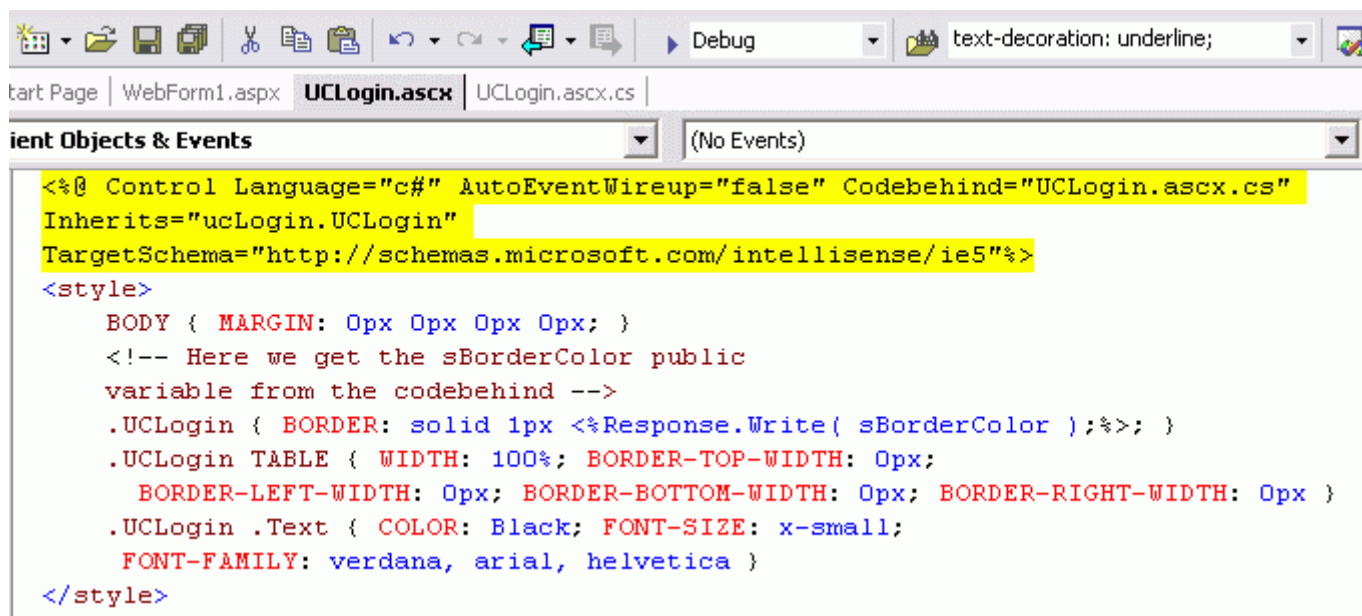


شکل ۵- طراحی کنترل لاگین

بعد از اینکه کنترل ایجاد شد دو متغیر عمومی در سطح کلاس آن تعریف شده اند. این متغیرها حاوی مقادیر تنظیم شده در خواص عمومی get و set روتین ها می باشد. همچنین متغیر sBorderColor نیز عمومی است زیرا ما این متغیر را در سورس HTML صفحه با استفاده از تگ های <% %> مورد استفاده قرار داده ایم.

```
//Declare our global variables to hold our property values. The  
//sBorderColor variable must be public because we are referencing  
//it in the HTML.  
private string sUsername;  
public string sBorderColor;
```

سپس برای استفاده از این متغیرهای عمومی در سورس HTML صفحه به صورت زیر عمل می کنیم:



```
<%@ Control Language="c#" AutoEventWireup="false" Codebehind="UCLogin.ascx.cs"
Inherits="ucLogin.UCLogin"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5"%>
<style>
    BODY { MARGIN: 0px 0px 0px 0px; }
    <!-- Here we get the sBorderColor public
    variable from the codebehind -->
    .UCLogin { BORDER: solid 1px <%Response.Write( sBorderColor );%>; }
    .UCLogin TABLE { WIDTH: 100%; BORDER-TOP-WIDTH: 0px;
        BORDER-LEFT-WIDTH: 0px; BORDER-BOTTOM-WIDTH: 0px; BORDER-RIGHT-WIDTH: 0px }
    .UCLogin .Text { COLOR: Black; FONT-SIZE: x-small;
        FONT-FAMILY: verdana, arial, helvetica }
</style>
```

شکل ۶- نحوه ی استفاده از متغیرهای عمومی در سورس HTML صفحات.

روتین های get و set مقادیر متغیرهای عمومی را تنظیم می کنند. خواص کنترل کاربر را می توان در زمان طراحی مشخص کرد و یا در CodeBehinde صفحه در زمان اجرا.

```
//Our exposed public properties
public string Username
{
    get {return sUsername;}
    set {sUsername = value;}
}

public string BorderColor
{
    get {return sBorderColor;}
    set {sBorderColor = value;}
}
```

و در نهایت در آینده می توان روتین کلیک بر روی دکمه را به صورت مناسب پیاده سازی کرد:

```
private void btnSubmit_Click(object sender, System.EventArgs e)
{
    //Logic to handle the submit goes here.
}
```

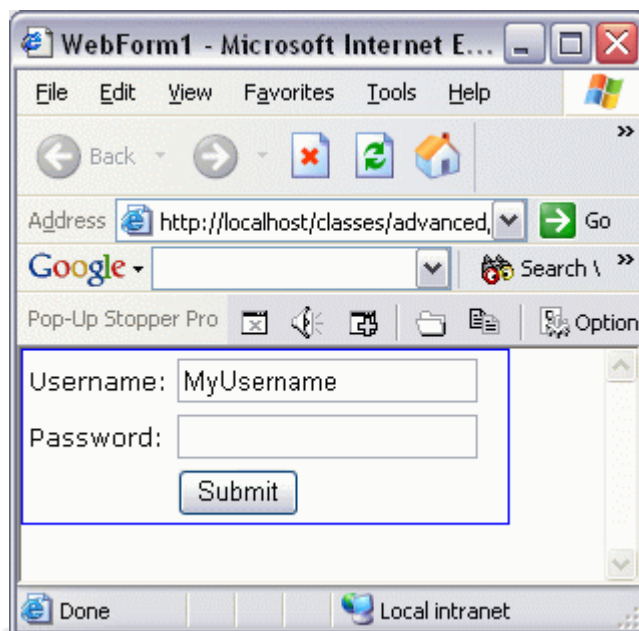


اکنون کنترل ایجاد شده را روی وب فرم اصلی برنامه drag کنید. برای اینکه بتوان رنگ حاشیه ی جدول و نام یوزر پیش فرض را همانطور که پیش بینی کرده ایم تنظیم نماییم به صورت زیر در سورس HTML صفحه عمل می شود.

```
<uc1:uclogin id="UCLogin1" BorderColor="Blue"
Username="MyUsername" runat="server"></uc1:uclogin>
```

رخدادی که در اینجا فراخوانی می شود OnPreRender است. این رخداد جزو آخرین رخدادهایی است که هنگام خلق یک صفحه در ASP.NET فراخوانی می شود. اگر نیاز به نمایش آن نباشد به سادگی می توان `this.visibility = false;` را تنظیم کرد تا مخفی شود و به کاربران غیرمجاز نمایش داده نشود. اگر برنامه را اجرا کنید ملاحظه خواهید کرد که خاصیت Text این کنترل مقدار دهی شده و اینکار در رخداد OnPreRender انجام پذیرفته است. اگر اینکار در Page\_Load انجام می شد بدلیل مکانیزم سیکل طول عمر یک صفحه ی ASP.NET ، محتویات تکست باکس بدلیل اطلاعات موجود در ViewState پاک می شد.

```
//Catches the OnPreRender event of the page lifecycle
protected override void OnPreRender( EventArgs e )
{
    //If the developer specifies the username in the public property,
    //use that one. Otherwise, we could get the username from
    //another source (i.e. A cookie on the client or a context
    //object attached to the user).
    if ( sUsername != null )
    {
        txtUsername.Text = sUsername;
    }
    else
    {
        //Logic to get the username goes here.
    }
}
```



شکل ۷- اجرای نهایی برنامه و نمایش کنترل لاگین ( به رنگ حاشیه و نام پیش فرض دقت کنید ).



### تمرین:

۱- سه وب فرم جدید به پروژه ای تازه اضافه نمایید. اکنون محتویات صفحات جدید را هر مطلب دلخواهی می توانید در نظر بگیرید. می خواهیم دو کنترل کاربر یکی به نام UCHheader و دیگری را به نام UCFooter خلق نماییم. در کنترل UCFooter که در پایین تک تک صفحات قرار خواهد گرفت کپی رایت سایت را به همراه یک تصویر نمایش می دهیم. در کنترل UCHheader آرم سایت به همراه چند لینک به تمام صفحات تعریف شده قرار خواهد گرفت. برای نظم بخشیدن به تمام صفحات حتما از کنترل Html Table استفاده نمایید.

