الفصل الثاني

سلسلة

ASP.NET

خطوة بخطوة حتى الاحتراف

C# & VB

اعداد المهندس

محمد عمر الحاج خلف

الفصل الثاني

استخدام أدوات التحكم المتقدمة

في هذا الفصل

- ✓ **رفع الملفات إلى الموقع**
- ✓ **عرض التقويم**
- ✓ **عرض الإعلانات**
- ✓ **تقسيم الصفحة لمناظير عديدة**
- ✓ **تقسيم العمل لمجموعة خطوات**

استخدام أدوات التحكم الغنية

في الفصل السابق قمنا بدراسة الأدوات القياسية والتي لابد من استخدامها في معظم تطبيقات ASP.Net , سنتناول في هذا الفصل دراسة أدوات تحكم أكثر تخصصاً وتعرف مجموعة الأدوات هذه باسم Rich controls .

في القسم الأول من هذا الفصل ستتعلم كيف تسمح للمتصفحين بأن يرفعوا ملفاتهم على الموقع , كأن يقوموا مثلاً برفع صور , ملفات فيديو ... إلخ , بعد ذلك ستتعلم استخدام أداة التقويم كطريقة تسهل على المستخدم إدخال التواريخ , كما نستعرض في هذا الفصل كيفية التعامل مع أداة التحكم AdRotator والتي تستخدم عادةً لعرض الإعلانات على الموقع وسنناقش حالتين : أولاً أن تكون الإعلانات في قاعدة بيانات , ثانياً أن تكون مخزنة في ملف XML , لاحقاً نتطرق إلى أداة التحكم MultiView التي تقوم بإظهار وإخفاء مناطق من الصفحة , وتقسيم الصفحة إلى ألسنة عديدة (taps) , وفي نهاية الفصل نعرض كيفية التعامل مع أداة التحكم Wizard والتي تمكننا من تقسيم عمل ما إلى مجموعة من الخطوات وعرضها على المستخدم كمجموعة واجهات متتالية .

رفع الملفات إلى الموقع

نستخدم أداة التحكم FileUpload لرفع الملفات إلى الموقع , ونقوم بتحديد المكان الذي سيتم تخزين ملفات المستخدمين فيه , إما مجلد داخل الموقع أو ضمن قاعدة بيانات الموقع وسنستعرض كلا الحالتين .

خصائص أداة التحكم FileUpload

- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ FileBytes : للحصول على محتويات الملف كمصفوفة بايتات .
- ✓ FileContent : للحصول على محتويات الملف كمجرى stream .
- ✓ FileName : الحصول على اسم الملف .
- ✓ HasFile : تعيد True عندما يتم رفع الملف .
- ✓ PostedFile : تعيد الملف المرفوع مغلف بغرض من الصف HttpPostedFile.

كما أن أداة التحكم FileUpload تدعم الطرائق التالية :

- ✓ Focus : تسمح بوضع التركيز على هذه الأداة .
- ✓ SaveAs : تقوم بتخزين الملف (المراد رفعه) على الموقع .

تقوم الخاصية PostedFile بتغليف الملف المرفوع بغرض HttpPostedFile , هذا الأمر يسمح بالحصول على معلومات إضافية حول الملف .

خصائص الصف HttpPostedFile

- ✓ ContentLenght : لمعرفة حجم الملف بالبايتات .
- ✓ ContentType : لمعرفة نوع الملف (اللاحقة).
- ✓ FileName : لمعرفة اسم الملف .
- ✓ InputStream : الحصول على الملف كمجرى stream .

كما أن الصف HttpPostedFile يدعم الطريقة SaveAs والتي تقوم بتخزين الملف على الموقع .

لاحظ أن الصف HttpPostedFile يقدم بعض الخصائص المتوفرة مسبقاً مع أداة التحكم FileUpload والتي يمكن التعامل معها دون الحاجة للصف HttpPostedFile , على سبيل المثال للحصول على اسم الملف يمكن اتباع أحد الأسلوبين :

- ✓ FileUpload.FileName
- ✓ HttpPostedFile.FileName

كما أن الطريقة SaveAs موجودة في الصف HttpPostedFile على الرغم من توافرها مع أداة التحكم FileUpload بشكل صريح .

مثال

أنشئ صفحة جديدة وأضف عليها الأدوات FileUpload1 , Button1, Label1 , أنشئ مجلد جديد داخل ملفات الموقع باسم uploads لنقوم برفع الملفات إليه , في حدث الضغط على الزر اكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        if (FileUpload1.HasFile)
        {
            string path = "~/uploads/" + FileUpload1.FileName;
            FileUpload1.SaveAs(MapPath(path));
            Label1.Text = "File Uploaded successfuly...";
        }
    }
    catch (Exception ex) { }
}
```

كود VB

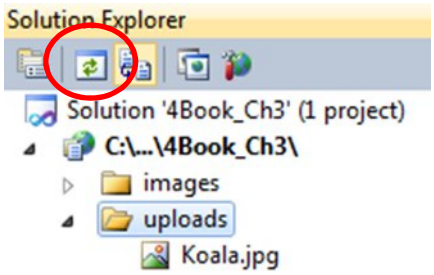
```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click

    If (FileUpload1.HasFile) Then
        Dim filePath As String = "~/uploads/" & FileUpload1.FileName
        FileUpload1.SaveAs(MapPath(filePath))
    End If
End Sub
```

كود الصفحة يجب أن يكون كالتالي :

كود ASP.net

```
<div>
    <asp:FileUpload ID="FileUpload1" runat="server" />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Upload"
                onclick="Button1_Click" />
    <br />
    <asp:Label ID="Label1" runat="server" Text=""/>
</div>
```

نفذ التطبيق السابق وحاول رفع ملف ما (صورة مثلاً) , أغلق التنفيذ , يجب أن يكون الملف قد أضيف للمجلد Upload , إن لم تره اضغط على تحديث ملفات الموقع كالتالي :



في المثال السابق يستطيع المستخدم أن يرفع أي نوع كان من الملفات , سنقوم في المثال التالي بتحديد الأنماط التي نسمح للمستخدم بأن يقوم برفعها (فقط بعض أنواع الصور).

مثال

أنشئ صفحة جديدة وأضف عليها الأدوات FileUpload1 , Button1, Label1 , أنشئ مجلد جديد داخل ملفات الموقع باسم uploads لنقوم برفع الملفات إليه , في حدث الضغط على الزر اكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        if (FileUpload1.HasFile)
        {
            if (CheckFileType(FileUpload1.FileName))
            {
                string path = "~/uploads/" + FileUpload1.FileName;
                FileUpload1.SaveAs(MapPath(path));
                Label1.Text = "File Uploaded successfuly...";
            }
        }
    }
    catch (Exception ex) { }
}
```

كود VB

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click

    If (FileUpload1.HasFile) Then
        If (CheckFileType(FileUpload1.FileName)) Then
            Dim filePath As String = "~/uploads/" & FileUpload1.FileName
            FileUpload1.SaveAs(MapPath(filePath))
        End If
    End If
End Sub
```

قم باستدعاء فضاء الأسماء الخاص بالدخل والخرج IO , حيث نضيف الكود التالي أعلى الصفحة:

كود C#

```
using System.IO;
```

كود VB

```
Imports System.IO
```

أما الدالة CheckFileType المسؤولة عن فحص نوع الملف فهي :

كود C#

```
bool CheckFileType(string fileName)
{
    string ext = Path.GetExtension(fileName);
    switch (ext.ToLower())
    {
        case ".gif":
            return true;
        case ".png":
            return true;
        case ".jpg":
            return true;
        case ".jpeg":
            return true;
        default:
            return false;
    }
}
```

كود VB

```
Function CheckFileType(ByVal fileName As String) As Boolean
    Dim ext As String = Path.GetExtension(fileName)
    Select Case ext.ToLower()
        Case ".gif"
            Return True
        Case ".png"
            Return True
        Case ".jpg"
            Return True
        Case ".jpeg"
            Return True
        Case Else
            Return False
    End Select
End Function
```

كود الصفحة السابقة :

كود ASP.net

```
<div>
    <asp:FileUpload ID="FileUpload1" runat="server" />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Upload"
                onclick="Button1_Click" />
```

```
        <br />
        <asp:Label ID="Label1" runat="server" Text=""/>
    </div>
```

نفذ التطبيق السابق وحاول أن ترفع ملف ما مغاير للأنواع المحددة , ستلاحظ عدم رفع الملف , أعد التجربة مع أحد أنواع الصور السابقة لترى قبول عملية الرفع .

رفع الملفات إلى قاعدة بيانات

حيث نقوم بتخزين الملفات في قاعدة بيانات عوضاً عن مجلد عادي , هذا الأسلوب يسبب كبر هائل بحجم قاعدة البيانات وبالتالي صعوبة وبطء في نقلها والتعامل معها , كما أننا نحتاج لتحويل الملفات إلى بيانات ثنائية (0,1) لنتمكن من تخزينها في الجدول , الأسلوب الأفضل والمتبع في المواقع هو رفع الملفات إلى مجلد عادي وتخزين أسماء الملفات المرفوعة ضمن قاعدة البيانات مع تخزين معلومات أخرى عنها كتاريخ الرفع مثلاً , على العموم لن ندخل بهذه التفاصيل الآن على اعتبار أننا لم نتطرق لموضوع قواعد البيانات , سيكون لنا عودة لمناقشة كيفية رفع الملفات بأسلوب احترافي في فصل المواضيع المتقدمة .

ملاحظة

يتم قبول رفع الملفات ذات الأحجام أصغر من 4 ميغابايت (عد للمثال الأول وحاول رفع ملف أكبر من هذا الحجم لترى عدم قبول العملية) , ولتغيير الحجم الأقصى المسموح به لكل ملف , نفتح الملف Web.config ونضيف كود السطر الرابع :

كود XML

```
<configuration>
    <system.web>
        <compilation debug="true" targetFramework="4.0"/>
        <httpRuntime maxRequestLength="10240"/>
    </system.web>
</configuration>
```

حيث تم تحديد الحجم الأعظمي بـ 10240 كيلو بايت (أي 10 ميغا بايت) .

عرض التقويم (أداة التحكم Calendar)

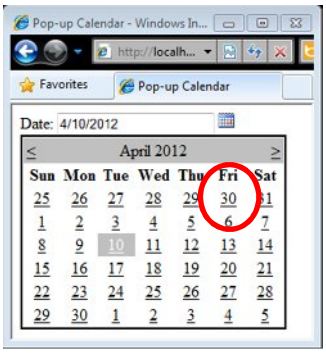
وفرت منصة عمل ASP.Net الأداة Calendar والتي تقوم بتمكين المستخدم من تحديد التواريخ بسهولة ويسر .

خصائص أداة Calendar

- ✓ DayNameFormat : تحديد أسلوب ظهور أسماء أيام الأسبوع (اسم اليوم بالكامل , اسم مختصر ...) ويمكن أن يأخذ إحدى القيم التالية : FirstLetter, FirstTwoLetters, Full, Short, and Shortest.
- ✓ NextMonthText : تحديد نص الرابط الذي ينقلنا للشهر التالي عند الضغط عليه
- ✓ NextPrevFormat : تحديد تنسيق نصي الشهر التالي والسابق (أسماء كاملة , مختصرة) وتأخذ إحدى القيم : CustomText, FullMonth, and ShortMonth.
- ✓ PrevMonthText : تحديد نص الرابط الذي ينقلنا للشهر السابق عند الضغط عليه
- ✓ SelectedDate : لتحديد يوم أو للحصول على اليوم محدد .
- ✓ SelectedDates : لتحديد مجموعة من الأيام أو الحصول عليها .
- ✓ SelectionMode : تحديد أسلوب التواريخ , نختار يوم أو أسبوع كامل أو شهر بالكامل ويأخذ القيم : Day, DayWeek, DayWeekMonth, and None.
- ✓ SelectMonthText : تحديد نص الرابط الذي يقوم بتحديد الشهر كاملا عند الضغط عليه
- ✓ SelectWeekText : تحديد نص الرابط الذي يقوم بتحديد الأسبوع كاملا عند الضغط عليه
- ✓ ShowDayHeader : عرض أو إخفاء أسماء الأيام
- ✓ ShowNextPrevMonth : عرض أو إخفاء روابط الأشهر اللاحقة والسابقة
- ✓ ShowTitle : عرض أو إخفاء شريط العنوان الخاص بالأداة Calendar
- ✓ TitleFormat : لتحديد تنسيق شريط العنوان حيث يمكن أن يظهر اسم الشهر فقط أو يظهر معه العام الحالي أيضاً , ويأخذ : Month ,MonthYear. .
- ✓ TodaysDate : لتحديد تاريخ اليوم الحالي , بشكل افتراضي تأخذ تاريخ اليوم الحالي من السيرفر .
- ✓ VisibleDate : لتحديد الشهر الذي سيتم عرضه , بشكل افتراضي يتم عرض الشهر الذي يحتوي على تاريخ اليوم الحالي المحدد بالخاصية السابقة.

كما أن الأداة Calendar تدعم الأحداث التالية :

- ✓ SelectionChanged : يتم إطلاقه عند اختيار تاريخ جديد .
- ✓ VisibleMonthChanged : يتم إطلاقه عند الضغط على رابط الشهر التالي أو السابق .



عرض أداة التقويم بشكل منبثق

سنقوم في هذه الفقرة بتطبيق الأسلوب المتبع في المواقع والمنتديات الاحترافية , وهو أن تكون أداة التقويم بشكل افتراضي غير ظاهرة للمستخدم , إنما يدل على وجودها رمز صغير يعبر عن التقويم فإن شاء المستخدم يضغط على هذا الرمز لتنبثق له أداة Calendar فيختار التاريخ منها , وإن شاء كتبه بشكل يدوي دون الاستعانة بهذه الأداة , يتم أداء العمل السابق بالاستعانة بلغة الجافا سكربت وتقنية الـ css .

أنشئ صفحة جديدة , وأضف عليها أدوات التحكم Label1 , TextBox1 , Calendar1 ومن قسم أدوات html أضف الأداة img واربطها مع صورة مناسبة كما هو موضح بالأعلى وفي حدث النقر عليها سنقوم باستدعاء كود جافاسكربت المسؤول عن إظهار الأداة Calendar1 , لن أقوم بشرح أكود جافا سكربت ولا css فهي ليست موضوعنا إنما سأكتفي بوضع كود الصفحة كاملاً :

كود ASP.net

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default4.aspx.cs"
        Inherits="Default4" %>

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

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

<script type="text/javascript">
    function displayCalendar() {
        var datePicker = document.getElementById('datePicker');
        datePicker.style.display = 'block';
    }
</script>

<style type="text/css">
#datePicker
{
    display:none;
    position:absolute;
    border:solid 2px black;
    background-color:white;
}
</style>

    <title>Pop-up Calendar </title>
</head>
<body>
    <form id="form1" runat="server">
```

```
    <div>
    <asp:Label id="Label1" Text=" Date:" AssociatedControlID="TextBox1"
              Runat="server" />
    <asp:TextBox id="TextBox1" Runat="server" />
    <img src="images/Calendar.png" onclick="displayCalendar()" />
    <div id="datePicker">
        <asp:Calendar ID="Calendar1" runat="server"
            onselectionchanged="Calendar1_SelectionChanged"></asp:Calendar>
    </div>
    </div>
    </form>
</body>
</html>
```

أما الكود المضاف للحدث SelectionChanged التابع للأداة Calendar1 فيقوم بإسناد التاريخ المحدد إلى الأداة TextBox1 بالأسلوب التالي :

كود C#

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    TextBox1.Text = Calendar1.SelectedDate.ToString("d");

}
```

كود VB

```
Protected Sub Calendar1_SelectionChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Calendar1.SelectionChanged

    TextBox1.Text = Calendar1.SelectedDate.ToString("d")
End Sub
```

عرض الإعلانات (أداة التحكم AdRotator)

تمكننا الأداة AdRotator من عرض الإعلانات على صفحات الموقع , ويتم اختيار الإعلان الذي سيعرض بشكل عشوائي حيث تخزن الإعلانات في ملف XML أو في جدول ضمن قاعدة بيانات .

خصائص أداة التحكم AdRotator

- ✓ AdvertisementFile : لتحديد ملف XML الذي يحتوي على الإعلانات .

- ✓ AlternateTextField : لتحديد اسم حقل النص البديل (اسم وسم XML أو عمود ضمن جدول) الذي ستظهر محتوياته في حال فشل عرض صورة الإعلان , القيمة الافتراضية لهذا الحقل هي : AlternateText .
- ✓ DataMember : لربط عضو بيانات محدد ضمن مصدر البيانات DataSource .
- ✓ DataSource : لتحديد مصدر بيانات الإعلانات بشكل برمجي .
- ✓ DataSourceID : لتحديد مصدر بيانات الإعلانات بشكل تصريحي .
- ✓ ImageUrlField : لتحديد اسم حقل صورة الإعلان (اسم وسم XML أو عمود ضمن جدول) ,القيمة الافتراضية ImageUrl .
- ✓ KeywordFilter : تحديد فلتر للإعلانات عبر كلمة مفتاحية وحيدة (نستعرضها لاحقا بالتفصيل) .
- ✓ NavigateUrlField : لتحديد اسم حقل روابط التنقل (اسم وسم XML أو عمود ضمن جدول) , القيمة الافتراضية NavigateUrl .
- ✓ Target : تمكننا من فتح نافذة جديدة عند النقر على الإعلان .

كما أن أداة التحكم AdRotator تقوم بإطلاق الحدث AdCreated عند اختيارها لإعلان ما وقبل أن يتم عرضه .

مثال

أنشئ صفحة جديدة وأضف عليها أداة AdRotator , ثم أنشئ ملف XML وسمه Ads.xml (لإنشاء ملف XML : من القائمة Websilte اختر Add New Item ثم XML File) , قم بإنشاء مجلد لنخزن صور الإعلانات داخله وليكن اسمه images وضع فيه ثلاث صور ولتكن اسماؤها pic1 , pic2 , pic3 , الآن افتح الملف Ads.xml واكتب فيه الكود التالي :

كود XML

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>

  <Ad>
    <ImageUrl>~/images/pic1.png</ImageUrl>
    <Width>300</Width>
    <Height>50</Height>
    <NavigateUrl>http://www.vb4arab.com</NavigateUrl>
    <AlternateText>Advertisement 1</AlternateText>
    <Impressions>50</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>~/images/pic2.png</ImageUrl>
    <Width>300</Width>
    <Height>50</Height>
```

```
    <NavigateUrl>http://www.yahoo.com</NavigateUrl>
    <AlternateText>Advertisement 2</AlternateText>
    <Impressions>25</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>~/images/pic3.png</ImageUrl>
    <Width>300</Width>
    <Height>50</Height>
    <NavigateUrl>http://www.google.com</NavigateUrl>
    <AlternateText>Advertisement 3</AlternateText>
    <Impressions>25</Impressions>
  </Ad>
</Advertisements>
```

في الكود السابق تم إضافة ثلاثة إعلانات , وتحديد صورة كل إعلان وقياساتها 200*200 والنص البديل في حال عدم عرض الصورة والرابط الذي سيتم الانتقال إليه عند الضغط على الإعلان , أما الخاصية Impressions فهي تحدد أهمية الإعلان كنسبة مئوية فالإعلان الأول أولوية عرضه هي 50% أما الثاني والثالث فكل منهما 25% , أسماء الوسوم التي تمت كتابتها في ملف Ads.xml هي نفس الأسماء الافتراضية للخصائص AlternateTextField , ImageUrlField , NavigateUrlField , فلو قمت بتغيير القيم الافتراضية لهذه الخصائص فعليك تعديل أسماء وسوم ملف xml السابق .

بعد أن انتهينا من تجهيز ملف الإعلانات Ads.xml , نعود إلى الصفحة التي أنشأناها ونربط الأداة AdRotator1 مع ملف xml السابق عبر الخاصية AdvertisementFile , ليصبح كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
    <asp:AdRotator ID="AdRotator1" runat="server"
                   AdvertisementFile="~/Ads.xml" />
</div>
```

نفذ الصفحة السابقة , قم بتحديث الصفحة عدة مرات لترى كيف يتم عرض الإعلانات بشكل عشوائي مع الأخذ بعين الإعتبار الأولوية المحددة لكل إعلان عبر الخاصية Impressions .

فلترة الإعلانات

لنفرض السيناريو التالي : لديك أداتين AdRotator1, AdRotator2 وملف xml يحتوي على بيانات خمسة إعلانات, أول ثلاثة إعلانات لمواقع تعليمية , الإعلانان الرابع والخامس لمواقع

رياضية , وتريد أن يتم عرض يتم عرض الإعلانات التعليمية بالأداة AdRotator1 والإعلانات الرياضية تعرض عبر الأداة AdRotator2 , فكيف يتم ذلك ؟ ببساطة نقوم بعمل فلترة للإعلانات ويتم هذا على مرحلتين :

أولاً : إضافة الوسم <Keyword> لكل إعلان في ملف xml بحيث تأخذ الإعلانات التعليمية قيمة ما (ولتكن Teaching) , ومجموعة الإعلانات الرياضية تأخذ قيمة أخرى (ولتكن Sporting) , أي أن ملف Ads2.xml يكون بالشكل التالي :

كود XML

```xml
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
<!-- Teaching Advertisements -->
  <Ad>
    <ImageUrl>~/images/pic1.png</ImageUrl>
    <Width>200</Width>
    <Height>200</Height>
    <NavigateUrl>http://www.vb4arab.com</NavigateUrl>
    <AlternateText> vb4arab </AlternateText>
    <Keyword>Teaching</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>~/images/pic2.png</ImageUrl>
    <Width>200</Width>
    <Height>200</Height>
    <NavigateUrl>http://www.arabTeam2000.com</NavigateUrl>
    <AlternateText> arabTeam2000 </AlternateText>
    <Keyword>Teaching</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>~/images/pic3.png</ImageUrl>
    <Width>200</Width>
    <Height>200</Height>
    <NavigateUrl>http://www.AspWorkShopes.com</NavigateUrl>
    <AlternateText> AspWorkShopes </AlternateText>
    <Keyword>Teaching</Keyword>
  </Ad>
  <!-- Sporting Advertisements -->
  <Ad>
    <ImageUrl>~/images/pic4.png</ImageUrl>
    <Width>200</Width>
    <Height>200</Height>
    <NavigateUrl>http://www.kooora.com</NavigateUrl>
    <AlternateText> kooora </AlternateText>
    <Keyword>Sporting</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>~/images/pic5.png</ImageUrl>
    <Width>200</Width>
    <Height>200</Height>
    <NavigateUrl>http://www.hihi2.com</NavigateUrl>
```

```
    <AlternateText> hihi2 </AlternateText>
    <Keyword>Sporting</Keyword>
  </Ad>
</Advertisements>
```

ثانياً : تحديد الفلتر الذي ستطبقه كلاً من الأداتين AdRotator1, AdRotator2 ويتم هذا عبر الخاصية KeywordFilter , بحيث نسند لكل أداة قيمة أحد الفلاتر السابقة المحددة في ملف xml وبالتالي يكون كود الصفحة كما يلي :

كود ASP.net

```
<div>
    <asp:AdRotator ID="AdRotator1" runat="server"
                   AdvertisementFile="~/Ads2.xml" KeywordFilter="Teaching" />
    <br />
    <asp:AdRotator ID="AdRotator2" runat="server"
                   AdvertisementFile="~/Ads2.xml" KeywordFilter="Sporting" />
</div>
```

قم بتنفيذ الصفحة السابقة وأعد تحميلها عدة مرات لترى أن الأداة AdRotator1 تعرض فقط الإعلانات التعليمية , في حين AdRotator2 تعرض فقط الإعلانات الرياضية .

في الفقرات السابقة تعلمنا كيف نتعامل مع الإعلانات المخزنة في ملف XML , يوجد أسلوب آخر وهو تخزين بيانات الإعلانات في جدول ضمن قاعدة البيانات , سنقوم بعرض تلك الطريقة في فصل المواضيع المتقدمة , بالإضافة لموضوع تتبع الإعلانات ومعرفة عدد مرات مشاهدة كل إعلان وغيرها من الأمور التي سنناقشها بالتفصيل في ذلك الفصل .

عرض مناظير مختلفة ضمن الصفحة (أدوات التحكم MultiView , View)

تسمح أداة التحكم MuliView بإخفاء وإظهار مناطق مختلفة من الصفحة , هذه الأداة مفيدة عندما تحتاج لإنشاء صفحة بألسنة عديدة (tabs) , كما أنها تستخدم لتقسيم نموذج طويل إلى عدة نمادج أصغر , هذه الأداة تقوم باحتواء مجموعة أدوات من النوع View , بحيث تعرض View واحد فقط بحسب مايحدده المستخدم و المنظور (View) الذي يتم عرضه يدعى ActiveView , في حين أن محتويات باقي المناظير تبقى مخفية , إذاً يتم عرض View واحد فقط في نفس الوقت .

خصائص أداة التحكم MultiView

- ✓ ActiveViewIndex : لتحديد رقم المنظور View الذي سيتم عرضه , يبدأ الترقيم من الصفر .
- ✓ Views : للحصول على مجموعة المناظير المحتواة ضمن الأداة MultiView .

الأداة MultiView تدعم الطرائق التالية :

- ✓ GetActiveView : للحصول على المنظور المعروض حالياً .
- ✓ SetActiveView : لتحديد المنظور الذي سيتم عرضه .

أخيراً , الأداة MultiView تدعم الحدث ActiveViewChanged والذي يطلق عند اختيار منظور ما من المناظير المحتواة داخلها .

أداة التحكم View لاتدعم أي خصائص أو طرائق خاصة بها , وذلك لأنها لاتستخدم إلا ضمن أداة أخرى تحتويها , على كلٍ فإن الأداة View تدعم الأحداث التالية :

- ✓ Activate : والذي يتم إطلاقه عندما نختار هذا المنظور من مجموعة مناظير الأداة MultiView ليتم عرضه .
- ✓ Deactivate : يطلق عندما نختار منظور آخر , أي عندما يلغى عرض هذا المنظور .

مثال

لنقم بتطبيق المثال التالي لتوضيح مبدأ عمل أدوات التحكم السابقة , أنشئ صفحة جديدة وأضف عليها ثلاثة أزرار (Buttons) , ثم أضف أداة تحكم MuliView وضع داخلها ثلاثة أدوات View واكتب داخل كل أداة View عبارة تدل عليه وذلك لتمييزه عند اختياره , وبالتالي سيصبح كود الصفحة كالتالي :

كود ASP.net

```
<div>
    <asp:Button ID="Button1" runat="server" Text="View1"
                onclick="Button1_Click" />
    <asp:Button ID="Button2" runat="server" Text="View2"
                onclick="Button2_Click" />
    <asp:Button ID="Button3" runat="server" Text="View3"
                onclick="Button3_Click" />
    <br/ />
    <asp:MultiView ID="MultiView1" runat="server">
        <asp:View ID="View1" runat="server">
        Hi , i'm View1 .
        </asp:View>
        <asp:View ID="View2" runat="server">
        Hi , i'm View2 .
        </asp:View>
```

```
        <asp:View ID="View3" runat="server">
        Hi , i'm View3 .
        </asp:View>
    </asp:MultiView>
</div>
```

وفي حدث النقر على الأزرار السابقة اكتب الكود التالي (كل كود لزر على الترتيب) :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 0;
}

protected void Button2_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 1;
}

protected void Button3_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 2;
}
```

كود VB

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click

    MultiView1.ActiveViewIndex = 0
End Sub

Protected Sub Button2_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button2.Click

    MultiView1.ActiveViewIndex = 1
End Sub

Protected Sub Button3_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button3.Click

    MultiView1.ActiveViewIndex = 2
End Sub
```

من خلال الكود السابق حددنا View مختلف ليتم عرضه عند النقر على أحد الأزرار السابقة وذلك عبر الخاصية ActiveViewIndex التابعة لأداة التحكم MultiView1 كما رأينا مسبقاً , نفذ الصفحة السابقة وانقر على الأزرار , بإمكانك إضافة ما تشاء من أدوات التحكم داخل المنظور , قمنا هنا بالإكتفاء بكتابة نص بسيط لتطبيق المثال بسهولة ويسر .

عرض نموذج بأقسام متعددة

نستخدم عادة أداة التحكم MultiView لتقسيم نموذج كبير إلى عدة نماذج فرعية أصغر , بإمكاننا إضافة Button داخل كل منظور بيحث ينقلنا إلى منظور آخر عند الضغط عليه , وذلك بالاستفادة من الأوامرالتالية التي تدعمها الأداة MultiView :

- ✓ NextView : لعرض المنظور التالي للمنظور المعروض حالياً ضمن MultiView
- ✓ PrevView : لعرض المنظور السابق للمنظور المعروض حالياً ضمن MultiView
- ✓ SwitchViewByID : لتحديد المنظور الذي سيتم عرضه , ويتم تحديده من خلال CommandArgument الخاص بالأداة Button
- ✓ SwitchViewByIndex : لتحديد المنظور الذي سيتم عرضه , ويتم تحديده من خلال CommandArgument الخاص بالأداة Button

إذاً , وبمنتهى البساطة إن أردت الانتقال إلى المنظور التالي فعليك إضافة Button إلى المنظور الحالي واسند للخاصية CommandName القيمة NextView , كذلك الأمر إن أردت الانتقال إلى منظور سابق حيث تستبدل القيمة السابقة بالقيمة PrevView .

مثال

في المثال التالي سنقوم بتصميم نموذج لإدخال البيانات وهو مقسم إلى ثلاثة مناظير (واجهات – views) , المنظور الأول يطلب إدخال الاسم , المنظور الثاني يطلب إدخال العمر , أما المنظور الثالث فيقوم بعرض المعلومات التي أدخلتها في الواجهتين السابقتين . أنشئ صفحة جديدة وأضف عليها أداة MultiView وضع داخلها ثلاث أدوات View , أضف داخل View1 أداة Label واجعل قيمة الخاصية Text تساوي "your Name" , ثم أضف أداة TextBox1 و Button1 واجعل قيمة الخاصية CommandName التابعة للزر تساوي NextView , أضف داخل View2 أداة Label2 واجعل قيمة الخاصية Text تساوي "your Age" , ثم أضف أداة TextBox2 و Button2 واجعل قيمة الخاصية CommandName التابعة للزر تساوي NextView , نأتي للمنظور الثالث , أضف داخله أداتين Label3, Label4 ثم انتقل للحدث Activate الخاص بهذا المنظور واكتب الكود التالي :

كود C#

```
protected void View3_Activate(object sender, EventArgs e)
{
    Label3.Text = TextBox1.Text;
    Label4.Text = TextBox2.Text;
}
```

كود VB

```
Protected Sub View3_Activate(ByVal sender As Object, ByVal e As
System.EventArgs) Handles View3.Activate

    Label3.Text = TextBox1.Text
    Label4.Text = TextBox2.Text
End Sub
```

كود الصفحة بالشكل التالي :

كود ASP.net

```
<div style="border-style: solid; border-width: thin">

    <asp:MultiView ID="MultiView1" runat="server" ActiveViewIndex="0">
        <asp:View ID="View1" runat="server">
            <asp:Label ID="Label1" runat="server" Text="Your Name : " />
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="Button1" runat="server" Text="next"
                        CommandName="NextView" />
        </asp:View>
        <asp:View ID="View2" runat="server">
            <asp:Label ID="Label2" runat="server" Text="Your Age : " />


            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="Button2" runat="server" Text="next"
                        CommandName="NextView" />
        </asp:View>
        <asp:View ID="View3" runat="server" onactivate="View3_Activate">


            your name :
            <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
            <br />
            your age :
```

```
            <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
        </asp:View>
    </asp:MultiView>

</div>
```

حاول التعديل على المثال السابق بإضافة أزرار تسمح بالإنتقال إلى مناظير سابقة .

أداة التحكم Wizard

أداة التحكم Wizard تشبه من حيث المبدأ الأداة MultiView حيث تقوم أيضاً بتقسيم نموذج كبير إلى عدة نماذج فرعية أصغر , بالإضافة إلى أن هذه الأداة تدعم مزايا عديدة غير متوفرة بنظيرتها MultiView , أداة التحكم Wizard تحتوي داخلها على مجموعة من الأدوات WizardStep , يمكن عرض WizardStep واحد فقط في نفس الوقت .

خصائص أداة التحكم Wizard

- ✓ ActiveStep : للحصول على أداة WizardStep الفعالة الآن .
- ✓ ActiveStepIndex : لتحديد أداة WizardStep التي نريد تفعليها , ويمكن استخدامها لمعرفة WizardStep الفعال حالياً .
- ✓ CancelDestinationPageUrl : لتحديد عنوان لصفحة يتم نقل المستخدم إليها اذا ضغط على زر إلغاء الأمر (Cancel).
- ✓ DisplayCancelButton : إظهار أو إخفاء الزر Cancel .
- ✓ DisplaySideBar : إظهار أو إخفاء الشريط الجانبي والذي يقوم بعرض قائمة بخطوات الأداة Wizard .
- ✓ FinishDestinationPageUrl : لتحديد عنوان لصفحة يتم نقل المستخدم إليها عند ضغطه على الزر إلغاء الأمر Finish .
- ✓ HeaderText : لتحديد العنوان الذي يظهر أعلى الأداة Wizard .
- ✓ WizardSteps : للحصول على جميع الأدوات WizardStep الموجودين ضمن الأداة Wizard .

أداة التحكم Wizard تتألف من ثلاث مناطق :

- ✓ المنطقة العلوية وندعوها بالترويسة Header
- ✓ المنطقة الجانبية وندعوها بالشريط الجانبي SideBar
- ✓ المنطقة المتبقية والتي تحتوي على الخطوات ندعوها بمنطقة الإبحار Navigation Area

أداة التحكم Wizard تدعم القوالب التالية :

- ✓ HeaderTemplate : لتحديد مظهر منطقة الترويسة
- ✓ SideBarTemplate : لتحديد مظهر منطقة الشريط الجانبي
- ✓ StartNavigationTemplate : لتحديد مظهر منطقة الإبحار لأول خطوة فقط
- ✓ FinishNavigationTemplate : لتحديد مظهر منطقة الإبحار لآخر خطوة فقط
- ✓ StepNavigationTemplate : لتحديد مظهر منطقة الإبحار لجميع الخطوات باستثناء الأولى والأخيرة .

أداة التحكم Wizard تدعم الطرائق التالية :

- ✓ GetHistory() : تمكننا من الحصول على مجموعة الخطوات WizardStep التي اجتزناها.
- ✓ GetStepType() : للحصول على نوع خطوة معينة , حيث توجد الأنواع التالية للخطوات WizardStep : Start, Finish, Step, Auot, Complete .
- ✓ MoveTo() : للانتقال إلى WizardStep معينة .

كما أن أداة التحكم Wizard تدعم الأحداث التالية :

- ✓ ActiveStepChanged : يتم إطلاقه عندما تتغير الخطوة الفعالة .
- ✓ CancelButtonClick : يتم إطلاقه عند الضغط على الزر Cancel .
- ✓ FinishButtonClick : يتم إطلاقه عند الضغط على الزر Finish .
- ✓ NextButtonClick : يتم إطلاقه عند الضغط على الزر Next .
- ✓ PreviousButtonClick : يتم إطلاقه عند الضغط على الزر Previous .
- ✓ SideBarButtonClick : يتم إطلاقه عند الضغط على زر SideBar .

إذاً , وكما رأينا فإن أداة التحكم Wizard تحتوي داخلها على مجموعة خطوات WizardStep .

أداة التحكم WizardStep تمتلك الخصائص التالية :

- ✓ AllowReturn : السماح للمستخدم بالعودة لهذه الخطوة من خطوة لاحقة , أو منعه من ذلك .
- ✓ Name : للحصول على اسم الخطوة الحالية .
- ✓ StepType : لتعيين نوع الخطوة الحالية أو استعادته , حيث يمكن أن تكون الخطوة من أحد الأنواع : Start, Finish, Step, Auot, Complete .
  - Start : اذا تم تعيين نمط الخطوة بهذه القيمة , عندئذٍ الزر Previous لن يظهر في هذه الخطوة .

- Step : يظهر كلا الزرين Next , Previous في هذه الخطوة .
- Finish : يظهر الزرين Previous , Finish .
- Complete : لن يظهر أي زر في الخطوة .
- Auto : وهو الخيار الأنسب , حيث يتم إظهار الأزرار المناسبة لكل خطوة بحسب موقعها من مجموعة الخطوات .

✓ Title : لتعيين عنوان للخطوة أو لاستعادته , عنوان الخطوة يتم عرضه في منطقة الشريط الجانبي SideBar .

✓ Wizard : تمكننا من استعادة أداة التحكم Wizard الحاوية لهذه الخطوة .

كما أن أداة التحكم WizardStep تطلق الحدثين التاليين :

✓ Activate : يُطلق عند تفعيل هذه الخطوة (أي عندما يتم عرضها) .

✓ Deactivate : يُطلق عند إلغاء تفعيل هذه الخطوة (أي عند الانتقال إلى خطوة أخرى) .

مثال

سنقوم في المثال التالي بإنشاء Wizard يتألف من أربع خطوات , الخطوة الأولى عبارة عن مقدمة , الخطوتين الثانية والثالثة لإدخال بيانات المستخدم , الخطوة الرابعة تعرض البيانات التي قام المستخدم بإدخالها , فيما يلي عرض للكود النهائي للصفحة كاملةً , مع إضافة صفوف CSS لإعطاء منظر أكثر جمالية لأداة Wizard (يمكن حذف هذه التنسيقات بدون مشكلة) , كود الصفحة :

كود ASP.net

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default9.aspx.cs"
Inherits="Default9" %>

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
<style type="text/css">
    .wizard
    {
        border:solid 1px black;


        font:14px Verdana,Sans-Serif;
        width:400px;
        height:300px;
    }
    .header
```

```
    {
        color:gray;
        font:bold 18px Verdana,Sans-Serif;
    }
    .sideBar
    {
        background-color:#eeeeee;
        padding:10px;
        width:100px;
    }
    .sideBar a
    {
        text-decoration:none;
    }
    .step
    {
        padding:10px;
    }
</style>
<title>Show Wizard</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Wizard id="Wizard1" HeaderText="Product Survey"
            OnFinishButtonClick="Wizard1_FinishButtonClick"
            CssClass="wizard" HeaderStyle-CssClass="header"
            SideBarStyle-CssClass="sideBar"
            StepStyle-CssClass="step"
            Runat="server">
        <WizardSteps>
        <asp:WizardStep ID="WizardStep1" Title="Introduction">
        Please complete our survey so that we can improve our
        products.
        </asp:WizardStep>
        <asp:WizardStep ID="WizardStep2" Title="Step 1">
            <asp:Label id="lblSSN" Text="Social Security Number:"
                AssociatedControlID="txtSSN" Runat="server" />
            <br />
            <asp:TextBox id="txtSSN" Runat="server" />
        </asp:WizardStep>
        <asp:WizardStep ID="WizardStep3" Title="Step 2" StepType="Finish">
            <asp:Label id="lblPhone" Text="Phone Number:"
                AssociatedControlID="txtPhone" Runat="server" />
            <br />
            <asp:TextBox id="txtPhone" Runat="server" />
        </asp:WizardStep>
        <asp:WizardStep ID="WizardStep4" Title="Summary" StepType="Complete">
            <h1>Summary</h1>
            Social Security Number:
            <asp:Label id="lblSSNResult" Runat="server" />
            <br /><br />
            Phone Number:
            <asp:Label id="lblPhoneResult" Runat="server" />
        </asp:WizardStep>
        </WizardSteps>
```

```
        </asp:Wizard>
    </div>
    </form>
</body>
</html>
```

أما في الحدث FinishButtonClick التابع لأداة التحكم Wizard1 فنكتب الكود التالي (وهو لإظهار البيانات التي أدخلها المستخدم في خطوات سابقة) :

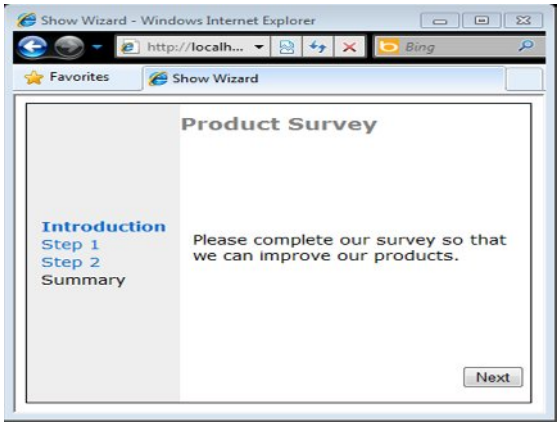
كود C#

```
protected void Wizard1_FinishButtonClick(object sender,
 WizardNavigationEventArgs e)

{
    lblSSNResult.Text = txtSSN.Text;
    lblPhoneResult.Text = txtPhone.Text;
}
```

كود VB

```
Protected Sub Wizard1_FinishButtonClick(ByVal sender As Object, ByVal e As
 System.Web.UI.WebControls.WizardNavigationEventArgs) Handles
 Wizard1.FinishButtonClick

    lblSSNResult.Text = txtSSN.Text
    lblPhoneResult.Text = txtPhone.Text
End Sub
```

نتيجة تنفيذ الصفحة السابقة ستكون بالشكل التالي :



خاتمة الفصل

إلى هنا نأتي لنهاية هذا الفصل , حيث استعرضنا فيه كيفية رفع الملفات إلى الموقع , وكيفية التعامل مع أداة التقويم واستخدامها كأداة منبثقة بالاستعانة بأكواد جافا سكريبت , ثم رأينا معاً مبدأ الإعلانات في المواقع وتعاملنا معها من خلال ملفات XML , ثم توجهنا إلى تقسيم نماذج العمل حيث كانت لنا وقفة مع أداة التحكم MultiView وأسلوب عرضها لمناظير عديدة بغية تقسيم النماذج الطويلة إلى نماذج فرعية أصغر وأبسط , وفي نهاية الفصل تعلمنا معاً كيفية العمل بأداة التحكم Wizard وأنواع الخطوات المختلفة .

أعيد وأذكر بأن هناك العديد من الفقرات والأمثلة الهامة جداً والمتعلقة بهذا الفصل والتي لا مجال لعرضها الآن بسبب احتوائها على معلومات لم ندرسها بعد , إنما سنناقشها بالتفصيل في فصول قادمة إن شاء الله .

للتواصل :

m-hajjkhalaf@hotmail.com

http://www.facebook.com/mohammed.hajkhalaf

محمد عمر الحاج خلف – سوريا

+963 932 033250