

الفصل الثاني

سلسلة

# ASP.NET

خطوة بخطوة حتى الاحتراف

C# & VB

إعداد المهندس

محمد عمر الحاج خلف

---

في هذا الفصل

- ✓ عرض التقويم
- ✓ تقسيم الصفحة لمناظير عديدة
- ✓ تقسيم العمل لمجموعة خطوات

## استخدام أدوات التحكم الغنية

في الفصل السابق قمنا بدراسة الأدوات القياسية منها 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 لأن يقوموا مثلاً برفع ، ملفات فيديو ... إلخ ، بعد ذلك ستتعلم استخدام أداة التقويم كطريقة تسهل على إدخال التواریخ ، كما نستعرض في هذا الفصل كيفية التعامل مع أداة التحكم وسنقاش حالتين : أولاً أن تكون AdRotator

في قاعدة بيانات ، ثانياً أن تكون مخزنة في ملف XML قوم بإظهار وإخفاء مناطق من الصفحة ، وتقسيم الصفحة إلى أنسنة عديدة ( tabs ) ، وفي نهاية الفصل نعرض كيفية التعامل مع أداة التحكم Wizard من تقسيم عمل ما إلى مجموعة من الخطوات وعرضها على المستخدم كمجموعة واجهات متتالية .

حديد المكان الذي سيتم تخزينه FileUpload مبين فيه ، إما مجلد داخل الموقع و ضمن قاعدة بيانات الموقع وسنستعرض كلا الحالتين .

FileUpload  
Enabled : تفعيل أو إلغاء تفعيل هذه الأداة . ✓  
FileBytes : للحصول على محتويات الملف كمصفوفة بايتات . ✓  
. stream : للحصول على محتويات الملف كمجرى FileContent ✓  
: FileName ✓  
. HasFile : تعيين True عندما يتم رفع الملف . ✓  
. HttpPostedFile : تعيين الملف المرفوع مغلق بعرض PostedFile ✓

FileUpload تدعم الطرائق التالية :

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

تقوم الخاصية PostedFile بتنزيل الملف المرفوع بـ HttpPostedFile ، هذا الأمر يسمح بالحصول على معلومات إضافية حول الملف .

## HttpPostedFile

لمعرفة حجم الملف بالبايتات . ContentLength ✓

( ) : ContentType ✓

: FileName ✓

. stream : InputStream ✓

يدعم الطريقة SaveAs والتي تقوم ب تخزين الملف على الموقع .

يقدم بعض الخصائص المتوفرة مسبقاً م HttpPostedFile

والتي يمكن التعامل معها دون الحاجة للصنف HttpPostedFile ، على سبيل المثال للحصول على اسم الملف يمكن اتباع أحد الأسلوبين :

✓ FileUpload.FileName

✓ HttpPostedFile.FileName

على الرغم من توافرها مع أداة HttpPostedFile

كما أن الطريقة SaveAs

FileUpload بشكل صريح .

أنشئ صفحة جديدة وأضف إليها الأدوات FileUpload1 , Button1, Lab جيد داخل ملفات الموقع باسم uploads لنقوم برفع الملفات إليه ، في حدث الضغط على الزر

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 successfully...";
        }
    }
    catch (Exception ex) { }
}
```

VB

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

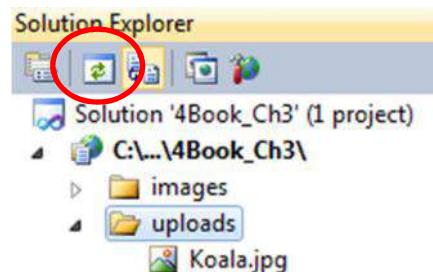
    If (FileUpload1.HasFile) Then
        Dim filePath As String = "~/uploads/" & FileUpload1.FileName
        FileUpload1.SaveAs(MapPath(filePath))
    End If
End Sub
```

يجب أن يكون كالتالي :

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



في المثال السابق يستطيع المستخدم أن يرفع بتحديد الأنماط التي نسمح للمستخدم بأن يقوم برفعها ( فقط بعض أنواع الصور ).

أنشئ صفحة جديدة وأضف إليها الأدوات FileUpload1 , Button1, Lab جيد داخل ملفات الموقع باسم uploads لنقوم برفع الملفات إليه ، في حدث الضغط على الزر

## 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 successfully...";
            }
        }
    }
    catch (Exception ex) { }
}
```

## 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
```

IO ، حيث نضيف الكود التالي أعلى الصفحة:

## C#

```
using System.IO;
```

## VB

```
Imports System.IO
```

سؤولة عن فحص نوع الملف فهي : CheckFileType

## 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;
    }
}
```

## 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
```

## 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>

```

يق السابق وحاول أن ترفع ملف ما مغایر لأنواع المحددة ، ستلاحظ عدم رفع الملف ،  
الصور السابقة لنرى 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 أنها تحتاج لتحويل بيانات ثنائية ( , ) لنتمكّن من تخزينها في الجدول ، الأسلوب الأفضل والمتبّع في الواقع 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 تيار أننا لم نتطرق لموضوع قواعد البيانات ، سيكون لنا عودة لمناقشة كيفية رفع الملفات بأسلوب احترافي في فصل المواضيع المتقدمة .

يتم قبول رفع الملفات ذات الأحجام ذات الأحجام أصغر من ميغابايت ( عد للمثال الأول وحاول رفع ملف أكبر من هذا الحجم لنرى عدم قبول العملية ) ، ولتحجيم الحجم الأقصى المسموح به ونصيف كود السطر الرابع : Web.config

#### XML

```

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

```

حيث تم تحديد الحجم الأعظمي بـ

كيلو بايت ( أي ميغا بايت ) .

#### عرض التقويم ( Calendar )

Calendar والتي تقوم بتمكين المستخدم من تحديد التواریخ ASP.Net بسهولة ويسر .

## 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 الذي ينقلنا للشهر عند الضغط عليه ( NextPrevFormat ✓ : تحديد تنسيق نصي الشهر مختصرة) وتأخذ إحدى القيم CustomText, FullMonth, and ShortMonth. .

PrevMonthText ✓ : تحديد الذي ينقلنا للشهر عند الضغط عليه SelectedDate ✓ : لتحديد يوم .

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 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.

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 المتبوع في الموقع 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 لتتحقق له أداة Calendar فيختار التاريخ منها ، وإن شاء كتبه بشكل يدوي دون الاستعانة بهذه الأداة ، يتم أداء العمل السابق بالاستعانة بلغة الجافا سكريبت وتقنية

. CSS

أنشئ صفحة جديدة ، وأضف عليها أدوات التح **Label1 , TextBox1 , Calendar1** واربطها مع صورة مناسبة كما هو موضح بالأعلى **html** النقر عليها سنقوم باستدعاء كود جافاسكريبت المسؤول عن إظهار الأداة **Calendar1** فهي ليست موضوعنا إنما سأكتفي بوضع كود الصفحة كاملاً : **css**

```
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" />

<div id="datePicker">
    <asp:Calendar ID="Calendar1" runat="server"
        onselectionchanged="Calendar1_SelectionChanged"></asp:Calendar>
</div>
</div>
</form>
</body>
</html>

```

فيقوم بإسناد التاريخ Calendar1

SelectionChanged

: TextBox1

C#

```

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

```

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

```

( AdRotator )

يتم اختيار الإعلان الذي

أو في جدول ضمن قاعدة بيانات XML .

AdRotator

سيعرض بشكل عشوائي حيث

AdRotator

. XML الذي يحتوي على الإعلانات . لتحدي AdvertismentFile ✓

XML : لتحديد اسم حقل النص البديل ( AlternateTextField ) ✓  
 ظهر محتوياته في حال فشل عرض صورة الإعلان ، القيمة الافتراضية ( ) . لهذا الحقل هي : AlternateText ✓

- . DataSource : لربط عضو بيانات محدد ضمن مصدر البيانات DataMember ✓
- . DataSource : لتحديد مصدر بيانات الإعلانات بشكل برمجي . DataSource ✓
- . DataSourcesID : لتحديد مصدر بيانات الإعلانات بشكل تصريحي . DataSourcesID ✓
- XML : لتحديد اسم حقل صورة الإعلان ( اسم وسم ImageUrlField ) ✓  
 ، القيمة الافتراضية ( ImageUrl ) .

KeywordFilter : تحديد فلتر للإعلانات عبر كلمة مفتاحية وحيدة (نستعرضها لاحقاً بالتفصيل) .

XML : لتحديد اسم حقل روابط التنقل ( اسم وسم NavigateUrlField ) ✓  
 جدول ) ، القيمة الافتراضية ( NavigateUrl ) .  
 Target : تمكنا من فتح نافذة جديدة ✓

Ads.xml XML وسمه AdCreated AdRotator  
أن يتم عرضه .  
عند اختيارها لإعلان ما وقبل

نشئ صفحة جديدة وأضف إليها أداة AdRotator ( XML File Add New Item Websilte : XML )  
مجلد لخزن صور الإعلانات داخله ولتكن اسمه images وضع فيه ثلاثة صور ولتكن اسماؤها Ads.xml واتكتب فيه الكود التالي : pic1 , pic2 , pic3

## 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>

```

#### \* تحديد صورة كل إعلان وقياساتها

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 % أما الثاني والثالث فكل منهما % أسماء الوسوم التي تمت كتابتها في ملف Ads.xml هي نفس الأسماء الافتراضية للخصائص ImageUrlField AlternateTextField NavigateUrlField ، فلو قمت بتغيير القيم الافتراضية لهذه الخصائص فعليك تعديل أسماء وسوم xml .

بعد أن انتهينا من تجهيز ملف الإعلانات Ads.xml ، نعود إلى الصفحة التي أنشأناها AdRotator1 xml السابق عبر الخاصية AdRotator1 ، ليصبح كود الصفحة :

#### ASP.net

```

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

```

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

لفرض السيناريو التالي : لديك أداتين AdRotator1, AdRotator2 xml يحتوي على بيانات خمسة إعلانات، أول ثلاثة إعلانات لموقع تعليمية ، الإعلان الرابع والخامس لموقع

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

حيث تأخذ الإعلانات التعليمية قيمة xml **<Keyword>** :  
 الإعلانات الرياضية تأخذ قيمة أخرى ( Sporting ) ( Teaching )  
 يكون بالشكل التالي : Ads2.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 ستطبّقه كلاً من الأداتين AdRotator1, AdRotator2 ويتم هذا عبر الخاصية KeywordFilter بحيث نسند لكل قيمة أحد الفلتر السابق المحددة في ملف xml وبالتالي يكون كود الصفحة كما يلي :

### 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 الإعلانات التعليمية ، في حين AdRotator2 تعرض فقط الإعلانات الرياضية .

في الفقرات السابقة تعلمنا كيف نتعامل مع الإعلانات 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 ( MultiView , View )

إظهار مناطق مختلفة من الصفحة ، هذه الأداة مفيدة عندما تحتاج لإنشاء صفحة بألسنة عديدة ( tabs ) ، كما أنها تستخدم لتقسيم نموذج طويل إلى عدة نماذج أصغر ، هذه الأداة تقوم باحتواء مجموعة أدوات من النوع View ، بحيث تعرض View فقط بحسب ما يحدده المستخدم و ActiveView ( View ) الذي يتم عرضه يدعى View حين أن محتويات باقي المناظير تبقى مخفية ، إذاً يتم عرض View .

### MultiView

لتحديد رقم المنظور View الذي سيتم عرضه ، يبدأ الترقيم من ActiveViewIndex ✓

. Views ✓ للحصول على مجموعة المناظير المحتواة ضمن الأداة MultiView .

MultiView تدعم الطرائق التالية :

. GetActiveView ✓ للحصول على المنظور المعروض حالياً .

. SetActiveView ✓ لتحديد المنظور الذي سيتم عرضه .

أخيراً ، الأداة MultiView والذي يطلق عند اختيار ActiveViewChanged منظور ما من المناظير المحتواة داخلها .

آخرى تحتويها ، على كلِّ فإن الأداة View تدعم الأحداث التالية :

. Activate ✓ والتي 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 .

لنقم بتطبيق المثال التالي لتوضيح مبدأ عمل أدوات التحكم السابقة ، أنشئ صفحة جديدة وأضف MultiView وضع داخلها ثلاثة أدوات ليها ثلاثة أزرار ( Buttons )

عبارة تدل عليه وذلك لتمييزه عند اختياره ، وبالتالي سيصبح View View :

## 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>
```

لزر على الترتيب ( )

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;  
}
```

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
```

View مختلف ليتم عرضه عند النقر على أحد الأزرار السابقة وذلك  
عبر الخاصية ActiveViewIndex كما رأينا مسبقاً ، نفذ

قمنا هنا بالإكفاء بكتابة نص بسيط لتطبيق المثال بسهولة ويسر .

سيم نموذج كبير إلى عدة نماذج فرعية أصغر ، بإمكاننا MultiView داخل كل منظور ببحث ينقلنا إلى منظور آخر عند الضغط عليه ، وذلك بالاستفادة من الأوامر التالية التي تدعمها الأداة MultiView :

MultiView	NextView	✓
MultiView	PrevView	✓
SwitchViewByID	SwitchViewByID	✓
Button	CommandArgument	
SwitchViewByIndex	SwitchViewByIndex	✓
Button	CommandArgument	

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

في المثال التالي سنقوم بتصميم نموذج لإدخال البيانات وهو مقسم إلى ثلاثة مناظير ( واجهات - views ) ، المنظور الأول يطلب إدخال الاسم ، المنظور الثاني يطلب إدخال الـ Label View1 View third يقوم بعرض المعلومات التي أدخلتها في الواجهتين السابقتين . أنشئ صفحة جديدة وأضف عليها أدلة MultiView وضع داخلها ثلاث أدوات Label TextBox1 View1 واجعل قيمة الخاصية Text "your Name" واجعل قيمة الخاصية CommandName NextView واجعل قيمة الخاصية Text "your Age" واجعل قيمة الخاصية CommandName Button View2 واجعل قيمة الخاصية Text "your Age" واجعل قيمة الخاصية CommandName TextBox2 واجعل قيمة الخاصية CommandName Button View2 ، نأتي للمنظور الثالث ، أضف داخله أداتين Label3, Label4 NextView الخاص بها المنظور واتكتب الكود التالي : Activate

## C#

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

## 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
```

## 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="Label14" runat="server" Text="Label"></asp:Label>
</asp:View>
</asp:MultiView>

</div>

```

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

## **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 WizardStep حتي داخلها على مجموعة من الأدوات Wizard MultiView يمكن عرض WizardStep .

## **Wizard**

WizardStep : ActiveStep ✓  
 . ActiveStepIndex ✓  
 : لتحديد أداة WizardStep التي نريد تفعيلها ، ويمكن استخدامها  
 الفعال حالياً . WizardStep  
 CancelDestinationPageUrl ✓  
 : لتحديد عنوان لصفحة يتم نقل المستخدم إليها اذا ضغط ( Cancel ) .

. Cancel : إظهار أو إخفاء الزر CancelCancelButton ✓  
 DisplaySideBar ✓  
 : إظهار أو إخفاء الشريط الجانبي والذي يقوم بعرض قائمة بخطوات . 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

## **Wizard**

✓ المنطقه العلوية وندعوها بالترويسه Header  
 ✓ المنطقه الجانبية وندعوها بالشريط الجانبي SideBar  
 قيه والتي تحتوي على الخطوات ندعوها بمنطقه الإبحار Navigation Area ✓

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 لأخر خطوة فقط ✓
- StepNavigationTemplate : لتحديد مظهر منطقة الإبحار لجميع الأولى والأخيرة . ✓

Wizard تدعم الطرائق التالية :

- WizardStep : GetHistory() ✓  
اجتنزناها.
- GetStepType() ✓ : للحصول على نوع خطوة معينة ، حيث توجد الأنواع التالية . Start, Finish, Step, Aut, Complete : WizardStep
- WizardStep : MoveTo() ✓ معينة .

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 .

WizardStep تمتلك الخصائص التالية :

- AllowReturn : السماح للمستخدم بالعودة لهذه الخطوة من خطوة لاحقة ، أو منعه من ✓
- Name : للحصول على اسم الخطوة الحالية . ✓
- StepType : لتعيين نوع الخطوة الحالية أو استعادتها ، حيث يمكن أن تكون الخطوة من . Start, Finish, Step, Aut, Complete : ✓
- Start : اذا تم تعيين Start •  
هذا القيمة ، عندئذِ الزر Previous لن يظهر في هذه الخطوة .

- Step : يظهر كلا الزرين Next , Previous في هذه الخطوة .
- Previous , Finish : يظهر الزرين .
- 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 لاستعادتها ، عنوان الخطوة يتم عرضه في منطقة الشريط .
- ✓ Wizard :Wizard الحاوية لهذه الخطوة .
- ✓ Wizard :Wizard

تطلق الحدفين التاليين : WizardStep

- ✓ Activate : يُطلق عند تفعيل هذه الخطوة ( أي عندما يتم عرضها ) .
- ✓ Deactivate : يُطلق عند إلغاء تفعيل هذه الخطوة ( أي عند الانتقال . )

#### Wizard يتتألف من أربع

، الخطوتين الثانية والثالثة لإدخال بيانات المستخدم ، الخطوة الرابعة تعرض البيانات التي قام المستخدم بإدخالها ، فيما يلي عرض للكود النهائي لصفحة CSS ( لإعطاء منظر أكثر جمالية لأداة Wizard ( يمكن حذف هذه التسقيفات بدون مشك ) ) :

#### 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>

```

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

فنكتب الكود التالي ( وهو Wizard1 FinishButtonClick لإظهار البيانات التي أدخلها المستخدم في خطوات سابقة ) :

C#

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

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

VB

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

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

نتيجة تنفيذ الصفحة السابقة ستكون بالشكل التالي :



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 الواقع 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

### Wizard

أعيد وأذكر بأن هناك العديد من الفقرات والأمثلة الهامة جداً وال المتعلقة بهذا الفصل والتي لا مجال لعرضها الآن بسبب احتوائها على معلومات لم ندرسها بعد ، إنما سنناقشه بالتفصيل في فصول

[m-hajjkhala@hotmai.com](mailto:m-hajjkhala@hotmai.com)

<http://www.facebook.com/mohammed.hajjkhala>

– سوريا –