

الفصل الأول

سلسلة

ASP.NET

خطوة بخطوة حتى الاحتراف

C# & VB

اعداد المهندس

محمد عمر الحاج خلف

الفصل الأول

استخدام أدوات التحكم القياسية

في هذا الفصل

- ✓ عرض المعلومات
- ✓ قبول دخل المستخدم
- ✓ إرسال بيانات النموذج
- ✓ عرض الصور
- ✓ استخدام أداة التحكم Panel
- ✓ استخدام الروابط التشعبية

استخدام أدوات التحكم الأساسية

في هذا الفصل سنتعلم كيف نستخدم عناصر التحكم الأساسية المضمنة في منصة العمل ASP.NET والتي لا بد منها لإنشاء تطبيقات الانترنت , حيث سنتعلم كيف نعرض المعلومات باستخدام الأدوات Label , Literal , وكيف نقبل ونتعامل مع دخل المستخدم من خلال الأدوات TextBox , CheckBox , RadioButton وكيفية إرسال بيانات النماذج إلى السيرفر بواسطة أدوات الأزرار المتنوعة , بالإضافة لأدوات عرض الصور على صفحات الإنترنت , كما نستعرض كيفية ضم الأدوات واحتوائها داخل أداة Panel بالإضافة لكيفية ربط الصفحات ببعضها من خلال الأداة HyperLink .

عرض المعلومات

منصة عمل ASP.NET تدعم أداتين لعرض البيانات النصية Label , Literal ففي حين أن الأداة Literal تعرض البيانات بأسلوب بسيط , فإن الأداة Label تقدم العديد من المزايا والتنسيقات الإضافية .

استخدام أداة التحكم Label

نستخدم أدوات عرض المعلومات حتى نتمكن من تعديل النص المعروض على الصفحة وقت التنفيذ حيث نستطيع ببساطة أن نحدد النص المراد عرضه من خلال إسناده للخاصية Text لأداة Label , كما يمكن أن نستخدم تنسيقات html لتقوم هذه الأداة بتفسيرهم وتوليد الخرج المطلوب , وبشكل افتراضي تقوم هذه الأداة بإحاطة النص بالوسم , في الكود التالي نعرض أساليب مختلفة لإسناد المعلومات لأدوات Label :

كود ASP.net

```
<div>
  <asp:Label ID="Label1" runat="server" Text="Hi i'm Label1" ></asp:Label>
  <br />
  <asp:Label ID="Label2" runat="server" Text="Label1">
  Hi i'm Label2
  </asp:Label>
  <br />
  <asp:Label ID="Label3" runat="server" Text="<b><i>Hi i'm Label3</i></b>">
  </asp:Label>
</div>
```

الأداة Label تدعم العديد من الخصائص لتنسيق النصوص , نستعرض منها :

- ✓ BackColor : تغيير لون الخلفية للأداة .
- ✓ BorderColor : تعيين لون لإطار يحيط بأداة Label .
- ✓ BorderStyle : تحديد نمط الإطار الذي سيتم عرضه , يمكن أن تأخذ هذه الخاصية أحد القيم التالية : NotSet, None, Dotted, Dashed, Solid, Double, Groove, Outset, Ridge, Inset, .
- ✓ BorderWidth : تحديد ثخانة الإطار .
- ✓ CssClass : تحديد صف CSS يتم تطبيقه على هذه الأداة .
- ✓ Font : تحديد نوع الخط المستخدم .
- ✓ ForeColor : تحديد لون الخط .
- ✓ Style : لتطبيق تصميم معين على أداة Label .
- ✓ ToolTip : نص يتم عرضه كتلميح عند مرور الماوس على أداة Label .

عادة ما يتم استخدام الأداة Label لعرض عناوين لأدوات أخرى لمساعدة المستخدم على القيام بعمله , وبالتالي فإن هذه الأداة تملك الخاصية AssociatedControlID لتحديد الأداة المرتبطة بها وينصح باستخدام هذه الخاصية بشكل عام .

استخدام أداة التحكم Literal

تعمل هذه الأداة بشكل مشابه جدا لعمل الأداة Label إلا أنها لا تقوم بإحاطة النص بالوسم وهذا ما قد نحتاجه في بعض المواقع التي لا يمكن فيها ترجمة أكواد html (كتحديد عنوان الصفحة على المتصفح مثلا) , وبما أن هذه الأداة لا تدعم الوسم فهي أيضا لا تدعم الخصائص التابعة له BackColor , CssClass , Literal , الأداة تدعم خاصية هامة وغير مدعومة بالأداة Label وهي الخاصية Mode حيث يمكنها أن تأخذ أحد القيم الثلاث التالية :

- ✓ PassThrough : تفسير رموز html الموجودة في النص وعرض ماتعنيه .
- ✓ Encode : عرض النص بدون ترجمة رموز html حيث تظهر كما هي .
- ✓ Transform : يتم تفسير فقط الرموز المدعومة من قبل الجهاز الطالب للصفحة .

كود ASP.net

```
<div>
  <asp:Literal ID="L1" runat="server" Mode="PassThrough" Text="<hr />" />
  <asp:Literal ID="L2" runat="server" Mode="Encode" Text="<hr />" />
  <asp:Literal ID="L3" runat="server" Mode="Transform" Text="<hr />" />
</div>
```

عند تنفيذ الصفحة السابقة تقوم Literal1 بعرض خط أفقي , Literal2 تعرض النص <hr /> في حين أن Literal3 تحص الجهاز الطالب للصفحة فإن كان يدعم هذا الوسم فإنه يعرض خط أفقي وإلا فسيتم عرض النص كما هو <hr /> .

قبول دخل المستخدم

تتضمن منصة عمل ASP.NET أدوات تسمح للمستخدم بإدخال وتحديد اختياراته في الموقع , حيث تقدم ثلاث أدوات أساسية هي TextBox , CheckBox , RadioButton و سنتعرف عليهم في الفقرات التالية بالتفصيل .

استخدام أداة التحكم TextBox

تسمح هذه الأداة للمستخدم بإدخال نص وفقاً لثلاثة أنماط تحددتها قيمة الخاصية TextMode وهم:

- ✓ SingleLine : قبول الدخل كسطر واحد فقط .
- ✓ MultiLine : قبول الدخول بأسطر عديدة .
- ✓ Password : حيث يظهر دخل المستخدم كرموز (نجوم *** أو غيره) .

كود ASP.net

```
<div>
  <asp:TextBox ID="TextBox1" runat="server" TextMode="SingleLine"/>
  <asp:TextBox ID="TextBox2" runat="server" TextMode="MultiLine" />
  <asp:TextBox ID="TextBox3" runat="server" TextMode="Password" />
</div>
```

خصائص الأداة TextBox :

- ✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة .
- ✓ AutoCompleteType : تحديد نمط الإكمال التلقائي حيث يكفي أن يدخل المستخدم بضعة حروف حتى يتم إظهار كلمات مطابقة تم إدخالها مسبقاً , يمكن إلغاء تفعيل هذه الخاصية بإسناد القيمة Disabled لها .
- ✓ AutoPostBack : إرسال المحتويات إلى السيرفر بشكل فوري عند أي تغيير بالنص المدخل .
- ✓ Columns : تحديد عدد الأعمدة التي ستظهر .
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .

- ✓ MaxLength : تحديد العدد الأعظمي من المحارف التي يمكن إدخالها (لاتعمل مع Multiline).
- ✓ ReadOnly : لمنع المستخدم من تغيير القيمة الموجودة ب TextBox .
- ✓ Rows : تحديد عدد الأسطر التي ستظهر .
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .
- ✓ Warp : لتحديد إمكانية الالتفاف التلقائي للنص مع النمط MultiLine .

أداة التحكم TextBox تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تدعم الحدث TextChanged والذي يُطلق عندما تتغير محتويات هذه الأداة .

لتوضيح آلية عمل الخاصية AutoPostBack أنشئ صفحة جديدة أضف لها TextBox1 وأعطي الخاصية السابقة القيمة True ثم أضف Label1 ماسنقوم به هو أنه عند كتابة نص ما فإنه سيعرض في الأداة Label1 بمجرد نقل التركيز خارج TextBox1 وهو دليل على عمل postBack آلي للصفحة , أي إرسالها للسيرفر وإعادة النتائج

كود ASP.net

```
<div>
  <asp:TextBox ID="TextBox1" runat="server" AutoPostBack="True"
    onttextchanged="TextBox1_TextChanged"></asp:TextBox>
  <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
```

في الحدث TextChanged الخاص بالأداة TextBox1 اكتب الكود التالي :

كود C#

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    Label1.Text = TextBox1.Text;
}
```

كود VB

```
Protected Sub TextBox1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged

    Label1.Text = TextBox1.Text
End Sub
```

نفذ الصفحة واكتب ماتريد في TextBox1 ثم اضغط على المفتاح Tab من لوحة المفاتيح وشاهد النتيجة , أغلق المستعرض ثم أعد التنفيذ مرة أخرى وقم بكتابة أول حرف من القيمة التي أدخلتها بالتنفيذ الأول لترى كيف تتم عملية الإكمال التلقائي , ألغى تفعيل خاصية الإكمال التلقائي كما هو موضح بالأعلى ثم أعد التنفيذ وأعد كتابة نفس الكلمة لتلاحظ توقف عمل الإكمال التلقائي .

استخدام أداة التحكم CheckBox

تسمح هذه الأداة للمستخدم أن يحدد قبوله أو رفضه لأمر ما, أي أنها تعتبر كسؤال يطرح على المستخدم والإجابة حتماً إما نعم أو لا , مثلا موقع فيسبوك يسألنا إن كنا نريد استقبال رسائل على بريدنا الإلكتروني أم لا , حيث إن الخاصية Checked هي التي تمكننا من معرفة خيار المستخدم كما سنرى بعد قليل .

خصائص أداة التحكم CheckBox

- ✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة
- ✓ AutoPostBack : إرسال خيار التحديد إلى السيرفر بشكل فوري عند أي تغيير بحالته .
- ✓ Checked : تعود بـ True أو False بحسب قبول المستخدم أو رفضه .
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .
- ✓ Text : تحديد النص الذي يظهر بجانب هذه الأداة (السؤال المطروح) .
- ✓ TextAlign : موضع النص بالنسبة للأداة , على يمينها أم يسارها تأخذ إحدى القيمتين Left , Right

أداة التحكم CheckBox تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تُطلق الحدث CheckChanged عند تغيير المستخدم لخياره سواء بالرفض أو القبول .

مثال بسيط : أنشئ صفحة جديدة أضف لها الأدوات CheckBox1,Button1,Label1 أنقر على الزر Button1 مرتين واكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = CheckBox1.Checked.ToString();
}
```

كود VB

```
Protected Sub Button1_Click1(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click

    Label1.Text = CheckBox1.Checked.ToString()
End Sub
```

نفذ وقم بتحديد خيار CheckBox ثم اضغط على الزر , سترى أن الخاصية Checked أعادت True , ألغى تفعيل الخيار واضغط على الزر مرة أخرى .. ماهي النتيجة الآن ؟
كود الصفحة بعد إجراء تعديلات بسيطة بالخصائص كالتالي :

كود ASP.net

```
<div>
    <asp:CheckBox ID="CheckBox1" runat="server" Text=" استقبال رسائل على الجوال ؟ "
        TextAlign="Left"/><br />
    <asp:Button ID="Button1" runat="server" Text=" ok "
        onclick="Button1_Click" /><br />
    <asp:Label ID="Label1" runat="server" Text="" />
</div>
```

ملاحظة

منصة عمل ASP.Net تدعم الأداة CheckBoxList والتي تتيح إمكانية عمل قائمة عناصرها عبارة عن أدوات CheckBox كما سنرى في فصل لاحق .

استخدام أداة التحكم RadioButton

لا تظهر هذه الأداة منفردة لوحدها إنما تكون دائماً ضمن مجموعة حيث بإمكان المستخدم تحديد خيار واحد فقط من ضمن أدوات مجموعة ما .

خصائص أداة التحكم RadioButton

✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة

✓ AutoPostBack : إرسال خيار التحديد إلى السيرفر بشكل فوري عند أي تغيير بحالته .

- ✓ Checked : تعود ب True أو False بحسب قبول المستخدم أو رفضه .
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ GroupName : لضم أداة RadioButton إلى مجموعة , من كل مجموعة لا يمكن تحديد سوى خيار واحد فقط من أدوات RadionButton الموجودة فيها.
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .
- ✓ Text : تحديد النص الذي يظهر بجانب هذه الأداة (الخيار المطروح) .
- ✓ TextAlign : موضع النص بالنسبة للأداة , على يمينها أم يسارها تأخذ إحدى القيمتين Left , Right

أداة التحكم RadioButton تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تُطلق الحدث CheckChanged عند تغيير المستخدم لخياره.

سنقوم بالمثل التالي بعرض بسيط لكيفية عمل هذه الأداة , أنشئ صفحة جديدة أضف عليها الأدوات التالية : RadioButton عدد 3 , Label , button , نعدل الخاصية Text لأدوات RadioButton بحيث تعرض كل أداة لون معين , كما يجب (وهذا هام جدا) أن تأخذ الأدوات الثلاث السابقة نفس قيمة الـ GroupName , نضغط على الزر مرتين ونكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    string color = "";
    if (RadioButton1.Checked)
        color = RadioButton1.Text;
    if (RadioButton2.Checked)
        color = RadioButton2.Text;
    if (RadioButton3.Checked)
        color = RadioButton3.Text;
    Label1.Text = "Your Favorite color is: " + color;
}
```

كود VB

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim color As String = ""
    If RadioButton1.Checked Then
        color = RadioButton1.Text
    End If
    If RadioButton2.Checked Then
        color = RadioButton2.Text
    End If
End Sub
```

```

If RadioButton3.Checked Then
    color = RadioButton3.Text
End If
Label11.Text = "Your Favorite color is: " + color
End Sub

```

أما كود الصفحة فهو كالتالي :

كود ASP.net

```

<div>
Favorite color ?<br />
<asp:RadioButton ID="RadioButton1" runat="server" Text="Red"
    GroupName="colors"/><br />
<asp:RadioButton ID="RadioButton2" runat="server" Text="Green"
    GroupName="colors"/><br />
<asp:RadioButton ID="RadioButton3" runat="server" Text="Blue"
    GroupName="colors"/><br />
<asp:Button ID="Button1" runat="server" Text=" ok "
    onclick="Button1_Click" />
<br />
<asp:Label ID="Label11" runat="server" Text=""/>
</div>

```

قم بحذف الخاصية `GroupName` من الكود السابق ونفذ وجرب أن تختار أكثر من لون .

ملاحظة

منصة عمل ASP.Net تدعم الأداة `RadioButtonList` والتي تتيح إمكانية عمل قائمة عناصرها عبارة عن أدوات `RadioButton` كما سنرى في فصل لاحق .

إرسال بيانات النموذج

منصة عمل ASP.Net تزودنا بثلاث أدوات لإرسال بيانات النموذج إلى السيرفر ، `Button` ، `ImageButton` ، `LinkButton` وهم يؤدون نفس الوظيفة ولكن لكل منهم مظهر وبعض الخصائص التي تميزه عن غيره.

استخدام أداة التحكم `Button`

تقوم هذه الأداة عند النقر عليها بإرسال بيانات الصفحة إلى السيرفر لتتم معالجتها كما يتم تنفيذ الكود الموجود ضمن الحدث `Click` الخاص بهذه الأداة ، يقوم المثال التالي بعرض الوقت الحالي في أداة `Label` عند النقر على الزر `Button` ، أنشئ صفحة جديدة أضف لها `Button,Label` اضغط على الزر مرتين واكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = DateTime.Now.ToString("T");
}
```

كود VB

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

    Label1.Text = DateTime.Now.ToString("T")
End Sub
```

كود الصفحة كاملاً :

كود ASP.net

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

لاحظ عند الضغط على الزر يتم إعادة تحميل الصفحة بالكامل حيث تم إرسالها للسيرفر وقام بعملية المعالجة المطلوبة ثم أعاد النتيجة للمستخدم .

خصائص أداة التحكم Button

- ✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة
- ✓ CommandArgument : تحديد بارامتر يتم إرساله إلى الحدث Command
- ✓ CommandName : تحديد اسم الأمر الذي يتم إرساله إلى الحدث Command
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ OnClientClick : لتحديد كود يتم تنفيذه على جهاز المستخدم عند الضغط على الزر .
- ✓ PostBackUrl : للانتقال إلى صفحة معينة بعد إرسال البيانات إلى السيرفر .
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .

✓ Text : تحديد النص الذي يظهر على هذه الأداة .

✓ UseSubmitBehavior : تمكنا من استخدام الجافا سكربت لإرسال بيانات النموذج .

أداة التحكم Button تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تدعم الحدثين :

✓ Click : يتم إطلاقه عند الضغط على الزر .

✓ Command : يتم إطلاقه عند الضغط على الزر , ويتم تمرير CommandArgument CommandName , إلى هذا الحدث .

استخدام أداة التحكم LinkButton

تؤدي نفس عمل الأداة Button ولكنها تختلف من حيث المظهر فهي تبدو كرابط وليس كزر وبالتالي يمكن تحديد صفحة للانتقال إليها عبر هذه الأداة , لنقم بتنفيذ نفس المثال السابق ولكن نستبدل الأداة Button بالأداة LinkButton , الكود الذي يتم تنفيذه عند الضغط على LinkButton

كود C#

```
protected void LinkButton1_Click(object sender, EventArgs e)
{
    Label11.Text = DateTime.Now.ToString("T");
}
```

كود VB

```
Protected Sub LinkButton1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles LinkButton1.Click
    Label11.Text = DateTime.Now.ToString("T")
End Sub
```

كود الصفحة :

كود ASP.net

```
<div>
  <asp:LinkButton ID="LinkButton1" runat="server"
    onclick="LinkButton1_Click">LinkButton</asp:LinkButton>
<br />
```

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

تقوم الأداة LinkButton وبشكل غير مرئي بتنفيذ كود جافا سكربت وعلى وجه التحديد يتم تنفيذ الطريقة _doPostBack وهو المسؤول عن إرسال البيانات إلى السيرفر , في مثالنا السابق فإن الكود يبدو كالتالي :

كود ASP.net

```
<a id="LinkButton1" href="javascript:__doPostBack('LinkButton1','')">Submit</a>
```

خصائص أداة التحكم LinkButton

- ✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة
- ✓ CommandArgument : تحديد بارامتر يتم إرساله إلى الحدث Command
- ✓ CommandName : تحديد اسم الأمر الذي يتم إرساله إلى الحدث Command
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ OnClientClick : لتحديد كود يتم تنفيذه على جهاز المستخدم عند الضغط على هذه الأداة.
- ✓ PostBackUrl : للانتقال إلى صفحة معينة بعد إرسال البيانات إلى السيرفر .
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .
- ✓ Text : تحديد النص الذي يظهر على هذه الأداة .

أداة التحكم LinkButton تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تدعم التحديثين :

- ✓ Click : يتم إطلاقه عند الضغط على هذه الأداة .
- ✓ Command : يتم إطلاقه عند الضغط على الأداة, ويتم تمرير CommandArgument CommandName , إلى هذا الحدث .

استخدام أداة التحكم ImageButton

تؤدي نفس وظيفة الأداة السابقة حيث تقوم بإرسال بيانات النموذج إلى السيرفر وتظهر كصورة على صفحة الانترنت , لتتعرف بدايةً على خصائص هذه الأداة ثم نطبق مثال توضيحي .

خصائص أداة التحكم ImageButton :

- ✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة
- ✓ AlternateText : نص بديل يتم عرضه في حال عدم عرض الصورة المحددة
- ✓ DescriptionUrl : لتحديد رابط إلى صفحة تحتوي تفاصيل عن الصورة
- ✓ CommandArgument : تحديد بارامتر يتم إرساله إلى الحدث Command
- ✓ CommandName : تحديد اسم الأمر الذي يتم إرساله إلى الحدث Command
- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ ImageAlign : تحديد موقع الصورة بالنسبة لأدوات html أخرى ويمكن أن تأخذ إحدى القيم التالية : AbsBottom, AbsMiddle, Baseline, Bottom, Left, Middle, NotSet, Right, TextTop, Top .
- ✓ ImageUrl : لتحديد مسار واسم الصورة التي سيتم عرضها .
- ✓ OnClientClick : لتحديد كود يتم تنفيذه على جهاز المستخدم عند الضغط على هذه الأداة.
- ✓ PostBackUrl : للانتقال إلى صفحة معينة بعد إرسال البيانات إلى السيرفر .
- ✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .

أداة التحكم ImageButton تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تدعم التحديثين :

- ✓ Click : يتم إطلاقه عند الضغط على هذه الأداة .
- ✓ Command : يتم إطلاقه عند الضغط على الأداة, ويتم تمرير CommandArgument CommandName , إلى هذا الحدث .

مثال : سنقوم في المثال التالي بتحديد إحداثيات مكان ضغط المستخدم على الأداة ImageButton أنشئ صفحة جديدة ضع فيها أداة ImageButton واجعلها تعرض صورة ما عبر الخاصية imageUrl ثم أضف أداتي Label , اضغط على ImageButton مرتين واكتب الكود التالي :

كود C#

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Label11.Text = "X = " + e.X.ToString();
    Label12.Text = "Y = " + e.Y.ToString();
}
```

كود VB

```
Protected Sub ImageButton1_Click(ByVal sender As Object, ByVal e As System.Web.UI.ImageClickEventArgs) Handles ImageButton1.Click
```

```

Label1.Text = "X = " + e.X.ToString()
Label2.Text = "Y = " + e.Y.ToString()
End Sub

```

كود الصفحة :

كود ASP.net

```

<div>
  <asp:ImageButton ID="ImageButton1" runat="server"
    ImageUrl="~/images/img.png"
    onclick="ImageButton1_Click" />
  <br />
  <asp:Label ID="Label1" runat="server" Text=""/>
  <asp:Label ID="Label2" runat="server" Text=""/>
</div>

```

ملاحظة

عند التعامل مع أي أداة تقوم بعرض الصور ينصح بشدة أن نستخدم الخاصية `AlternateText` لتقوم بعرض نص بديل لبعض المستخدمين يلغون خيار عرض الصور في المتصفح بهدف الحصول على سرعة أكبر في التصفح .

إرسال البيانات عبر الصفحات

بشكل افتراضي , عند الضغط على زر ما في الصفحة , يتم إرسالها إلى السيرفر ومعالجة البيانات ثم عرضها على نفس الصفحة , أي يحدث إعادة تحميل للصفحة ببيانات معدلة , الآن سنتعلم كيف نجعل السيرفر يعيدنا إلى صفحة أخرى لعرض النتائج بعد أن نضغط على زر ما في صفحة أولى .

مثال : أنشئ صفحة جديدة سمها `search.aspx` أضف لها `Button` , `TextBox` , أعطي الخاصية `PostBackUrl` القيمة `results.aspx` وهي صفحة ثانية نقوم بإنشائها , هذه الخاصية كما رأينا سابقاً تجعل السيرفر يعيد توجيه المتصفح إلى صفحة محددة عند الضغط على الزر , مانريده من مثالنا هذا أن يقوم المستخدم بإدخال نص ما في `TextBox` ثم الضغط على الزر ليقوم السيرفر بعرض نفس النص ولكن في صفحة أخرى , كود الصفحة `search.aspx` :

كود ASP.net

```

<div>
  <asp:TextBox ID="TxtSearch" runat="server"></asp:TextBox>
  <br />
  <asp:Button ID="BtnSearch" runat="server" Text="search"

```

```

PostBackUrl="~/results.aspx" />
</div>

```

ننتقل الآن إلى الصفحة results.aspx ونضيف عليها فقط أداة Label لنعرض فيها النص المدخل في الصفحة الأولى .

حيث يكون الكود كالتالي :

كود ASP.net

```

<div>
  <asp:Label ID="LblSearch" runat="server" Text="" />
</div>

```

وفي حدث تحميل الصفحة نكتب الكود التالي :

كود C#

```

protected void Page_Load(object sender, EventArgs e)
{
    if (PreviousPage != null)
    {
        TextBox txtSearch = (TextBox)PreviousPage.FindControl("TxtSearch");
        LblSearch.Text = String.Format("Search for: {0}", txtSearch.Text);
    }
}

```

كود VB

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

    If PreviousPage IsNot Nothing Then
        Dim txtSearch As TextBox =
            DirectCast(PreviousPage.FindControl("TxtSearch"), TextBox)
        LblSearch.Text = [String].Format("Search for: {0}", txtSearch.Text)
    End If
End Sub

```


تحليل الكود السابق : تستخدم الخاصية PreviousPage للحصول على مؤشر للصفحة السابقة التي سببت نقلنا إلى هذه الصفحة , الطريقة FindControl تعيد أداة تحكم محددة وبجميع خصائصها الحالية (ونحن نريد استرجاع قيمة الخاصية Text منها) البارامتر الذي تم تمريره هو Id أداة التحكم المطلوبة , ننفذ الكود الصفحة search وندخل نص ما ثم نضغط على الزر لنرى آلية عمل الكود والخاصية PostBackUrl وكيف استطعنا الوصول لبيانات صفحة سابقة من صفحة تالية .

ملاحظة

يوجد أساليب أكثر فاعلية لنقل البيانات بين صفحات الموقع مثل Cookies,Session وغيرهم . وسناقشهم بالتفصيل في فصول لاحقة إن شاء الله .

تحديد الزر الافتراضي

تسمح هذه الخاصية بتحديد زر في الصفحة بحيث يتم تنفيذ الكود داخله بمجرد الضغط على مفتاح Enter من لوحة المفاتيح ودون الحاجة للضغط على الزر من خلال الماوس , تدعى هذه الخاصية defaultButton ويتم تحديدها كخاصية للنموذج .

مثال : أنشئ صفحة جديدة , أضف عليها أداتين Button وأداة Label واحدة , في حدث النقر على الزر الأول اكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = "you click Button1 ";
}
```

كود VB

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

    Label1.Text = "you click Button1 "
End Sub
```

وفي حدث النقر على الزر الثاني :

كود C#

```
protected void Button2_Click(object sender, EventArgs e)
{
    Label1.Text = "you click Button2 ";
}
```

كود VB

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

    Label1.Text = "you click Button2 "
End Sub
```

الآن نأتي إلى بيت القصيد وهي تحديد الخاصية defaultButton التابعة للفورم وإعطائها كقيمة أحد الزرين السابقين كما في الكود التالي :

كود ASP.net

```
<form id="form1" runat="server" defaultbutton="Button2">
<div>
    <asp:Button ID="Button1" runat="server" Text="Button"
onclick="Button1_Click" />
    <br />
    <asp:Button ID="Button2" runat="server" Text="Button"
onclick="Button2_Click" />
    <br />
    <asp:Label ID="Label1" runat="server" Text=""></asp:Label>
</div>
</form>
```

حيث تم تحديد Button2 كزر افتراضي , نفذ الصفحة واضغط على Enter لترى ما يحدث, أعد تنفيذ المثال بعد تعديل الخاصية السابقة وجعلها تساوي القيمة Button1 .

ملاحظة

يمكن استخدام الخاصية defaultButton مع أداة التحكم Panel , سنتعرف على هذه الأداة بالتفصيل في فقرات قادمة ضمن هذا الفصل .

معالجة الحدث Command

أدوات التحكم Buttons بأنواعهم الثلاثة يدعمون الحدثين Command , Click كما رأينا سابقاً , الفرق بين هذين الحدثين هو أنه عند العمل مع Command فيمكننا تمرير اسم أمر ما وبارامتر للأمر أيضاً Command name,argument وهذه الأمور غير مدعومة مع الحدث Click .
 مثال توضيحي : أنشئ صفحة جديدة , أضف عليها أداتي Button وأداة Label واحدة , أضف للأداتين Button الخصائص onCommand, CommandName , CommandArgument وفق التالي :

الخاصية	القيمة
الأداة Button1	
onCommand	Button_Command
CommandName	Colors
CommandArgument	Red
الأداة Button2	
onCommand	Button_Command
CommandName	Colors
CommandArgument	Blue

وبالتالي يصبح كود الصفحة :

```

ASP.net كود
<div>
  <asp:Button ID="Button1" runat="server" Text="Red"
    oncommand="Button_Command"
    CommandName="colors"
    CommandArgument="Red" />
  <asp:Button ID="Button2" runat="server" Text="Blue"
    oncommand="Button_Command"
    CommandName="colors"
    CommandArgument="Blue" />
  <asp:Label ID="Label1" runat="server" Text=""></asp:Label>
</div>
    
```

افتح ملف الكود الخلفي Default.aspx.cs ونكتب فيه الكود التالي :

```

C# كود
protected void Button_Command(object sender, CommandEventArgs e)
    
```

```

{
    if (e.CommandName == "colors")
    {
        switch (e.CommandArgument.ToString())
        {
            case "Red":
                Label1.Text = "RED";
                break;
            case "Blue":
                Label1.Text = "BLUE";
                break;
        }
    }
}

```

كود VB

```

Protected Sub Button_Command(ByVal sender As Object, ByVal e As
CommandEventArgs)

If e.CommandName = "colors" Then
    Select Case e.CommandArgument.ToString()
        Case "Red"
            Label1.Text = "RED"
        Exit Select
        Case "Blue"
            Label1.Text = "BLUE"
        Exit Select
    End Select
End If
End Sub

```

لاحظ أن كلا الزرين يقوم باستدعاء نفس الكود Button_Commaen مع تحديد CommandName للقيمة colors والإختلاف الوحيد بينهما هو البارامتر المرر حيث إن الزر الأول يرسل البارامتر Red في حين أن الزر الثاني يرسل البارامتر Blue , نفذ الصفحة لترى نتيجة العمل , إن العمل مع الحدث Command فعال عندما تؤدي مجموعة من الأزرار نفس الكود ولكن ببارامترات مختلفة حيث يوفر علينا عناء إعادة كتابة الكود لو تم استخدام الحدث Click

عرض الصور

منصة عمل ASP.Net تقدم أداتين لعرض الصور ImageMap , Image تقوم أداة التحكم Image بعرض الصور بأسلوب بسيط , في حين أن الأداة ImageMap تقدم مزايا أكثر تعقيدا , سنتعرف على كلا الأداتين في الفقرات التالية .

استخدام أداة التحكم Image

تستخدم أداة التحكم Image لعرض صورة محددة عبر الخاصية imageUrl , سنتعرف على بعض بعض خصائص هذه الأداة ثم نطبق مثال عليها .

خصائص أداة التحكم Image :

- ✓ AlternateText : نص بديل يتم عرضه في حال عدم عرض الصورة المحددة
- ✓ DescriptionUrl : لتحديد رابط إلى صفحة تحتوي تفاصيل عن الصورة
- ✓ GenerateEmptyAlternateText : إسناد سلسلة فارغة كنص بديل للصورة
- ✓ ImageAlign : تحديد موقع الصورة بالنسبة لأدوات html أخرى ويمكن أن تأخذ إحدى القيم التالية : AbsBottom, AbsMiddle, Baseline, Bottom, Left, Middle, NotSet, Right, TextTop, Top
- ✓ imageUrl : لتحديد مسار واسم الصورة التي سيتم عرضها .

مثال : سنقوم في المثال التالي بجعل أداة Image تقوم بعرض واحدة من ثلاث صور وبشكل عشوائي , أنشئ صفحة جديدة وأضف عليها أداة Image , أنشئ ضمن المشروع مجلد جديد باسم images أضف عليه ثلاث صور ولتكن أسماؤها pic1 , pic2 , pic3 افتح ملف الكود الخلفي أي الصفحة Default.aspx.cs وفي حدث تحميل الصفحة (Page_Load) اكتب الكود التالي :

كود C#

```
protected void Page_Load(object sender, EventArgs e)
{
    Random rnd = new Random();
    switch (rnd.Next(3))
    {
        case 0:
            Image1.ImageUrl = "~/images/pic1.png";
            Image1.AlternateText = "picture1";
            break;
        case 1:
            Image1.ImageUrl = "~/images/pic2.png";
            Image1.AlternateText = "picture2";
            break;
        case 2:
            Image1.ImageUrl = "~/images/pic3.png";
            Image1.AlternateText = "picture3";
            break;
    }
}
```

كود VB

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)
Handles Me.Load

Dim rnd As New Random()
Select Case rnd.[Next](3)
Case 0
Image1.ImageUrl = "~/images/pic1.bmp"
Image1.AlternateText = "picture1"
Exit Select
Case 1
Image1.ImageUrl = "~/images/pic2.jpg"
Image1.AlternateText = "picture2"
Exit Select
Case 2
Image1.ImageUrl = "~/images/pic3.bmp"
Image1.AlternateText = "picture3"
Exit Select
End Select
End Sub

```

نفذ الصفحة السابقة ليتم عرض إحدى الصور الثلاث , اضغط على زر تحديث الصفحة (أو F5) ليتم عرض صورة أخرى وبشكل عشوائي . مما سبق نلاحظ مدى بساطة التعامل مع أداة التحكم Image , في الفقرة التالية سنتعرف على أداة تستخدم أيضاً لعرض الصور ولكن بخيارات ومزايا أكثر تنوعاً .

استخدام أداة التحكم ImageMap

تستخدم هذه الأداة لعرض صورة ما بالإضافة لربطها مع خريطة غير مرئية , بمعنى آخر عند نقر المستخدم على أجزاء مختلفة من الصورة بإمكاننا الاستجابة بردود فعل مختلفة أيضاً بحسب إحداثيات موقع النقر على الصورة , وعلى هذا الأساس بإمكاننا استخدام هذه الأداة كوسيلة للتنقل بين الصفحات مثلاً , حيث نقوم بعرض صورة تحتوي على رسومات معبرة عن صفحات تالية ثم نحدد عند النقر على موقع ما من هذه الصورة الصفحة التي سيتم الانتقال لها , استخدام آخر لهذه الأداة هو تحديد دخل المستخدم حيث بإمكاننا عرض صورة تحتوي على منتجات عديدة ويكون خيار المستخدم بحسب المنتج الذي تم الضغط عليه على الصورة .

أداة التحكم ImageMap ترتبط تلقائياً مع صف HotSpot class والذي يسمح بتحديد مناطق قابلة للنقر على الأداة ImageMap , منصة العمل ASP.Net تقدم ثلاثة صفوف من النمط HotSpot بإمكاننا التعامل معها وهي :

- ✓ CircleHotSpot : تمكنا من تحديد منطقة دائرية على خريطة الصورة
- ✓ PolygonHotSpot : تمكنا من تحديد شكل غير منتظم على خريطة الصورة

✓ RectangleHotSpot : تمكنا من تحديد منطقة مستطيلة على خريطة الصورة

خصائص أداة التحكم ImageMap :

✓ AccessKey : لتحديد مفتاح من لوحة المفاتيح عند الضغط عليه يتم نقل التركيز لهذه الأداة

✓ AlternateText : نص بديل يتم عرضه في حال عدم عرض الصورة المحددة

✓ DescriptionUrl : لتحديد رابط إلى صفحة تحتوي تفاصيل عن الصورة

✓ GenerateEmptyAlternateText : إسناد سلسلة فارغة كنص بديل للصورة

✓ HotSpotMode : تحديد السلوك الذي سيتم تنفيذه عند الضغط على منطقة من الصورة وتأخذ إحدى القيم التالية : Inactive, Navigate, NotSet,PostBack

✓ HotSpots : تمكنا من الحصول على جميع النقاط الساخنة (HotSpots) الموجودة ضمن أداة التحكم ImageMap

✓ ImageAlign : تحديد موقع الصورة بالنسبة لأدوات html أخرى ويمكن أن تأخذ إحدى القيم التالية : AbsBottom, AbsMiddle, Baseline, Bottom, Left, Middle,NotSet, Right, TextTop, Top

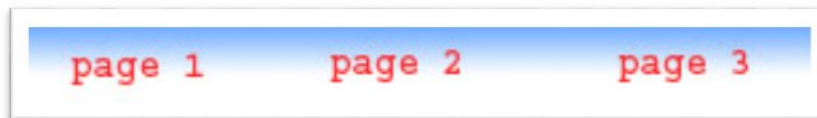
✓ ImageUrl : لتحديد مسار واسم الصورة التي سيتم عرضها .

✓ TabIndex : تحديد رقم يدل على ترتيب الوصول لهذه الأداة عبر مفتاح Tab .

✓ Target : تسمح لنا بفتح الصفحة في نافذة جديدة .

أداة التحكم ImageMap تدعم الطريقة Focus التي تسمح بوضع التركيز عليها عند التنفيذ , كما أنها تدعم الحدث Click والذي يتم إطلاقه عند الضغط على منطقة من هذه الأداة وكانت الخاصية HotSpotMode تأخذ القيمة PostBack.

مثال : سنقوم في المثال بعرض كيفية استخدام أحد جوانب أداة التحكم ImageMap , حيث سنقوم بإنشاء قائمة للتنقل بين صفحات مختلفة , أنشئ مشروع جديد وأضف أربع صفحات ولتكن أسماؤها page1 , page2 , page3 , Default , أضف على الصفحة Default أداة ImageMap واجعلها تعرض صورة مناسبة للغرض السابق , ولتكن كالتالي :



حيث سنقوم بتقسيم أداة الصورة السابقة إلى ثلاث مناطق وكل منطقة تنقلنا إلى صفحة محددة

كود صفحة Default.aspx :

ASP.net كود

```

<div>
  <asp:ImageMap ID="ImageMap1" runat="server"
    ImageUrl="~/images/navigationBar.jpg">

    <asp:RectangleHotSpot NavigateUrl="~/page1.aspx" Left="0" Top="0"
      Right="100" Bottom="30" AlternateText="go to page1" />

    <asp:RectangleHotSpot NavigateUrl="~/page2.aspx" Left="100" Top="0"
      Right="200" Bottom="30" AlternateText="go to page2" />

    <asp:RectangleHotSpot NavigateUrl="~/page3.aspx" Left="200" Top="0"
      Right="300" Bottom="30" AlternateText="go to page3" />

  </asp:ImageMap>
</div>

```

نلاحظ من الكود السابق أن للخاصية RectangleHotSpot الخصائص الفرعية Left , Right , Top , Bottom لنحدد إحداثيات المنطقة (حجم الصورة في المثال 300 X 30) . وكل منطقة تم ربطها بصفحة معينة عبر الخاصية NavigateUrl ليتم نقل المتصفح إليها عند النقر على تلك المنطقة .

مثال آخر : بعد أن تعرفنا على كيفية استخدام الأداة ImageMap للتنقل بين الصفحات المختلفة , سنعرض هنا آلية العمل معPostBack أي البقاء ضمن نفس الصفحة , أضف صفحة جديدة وأضف عليها ImageMap , TextBox , الصورة المطلوبة ليتم عرضها شبيهة بالتالي :



المطلوب تنفيذه هو أنه عند الضغط المنطقة الأولى (ToUpper) يتم تحويل النص المكتوب في أداة TextBox إلى حالة أحرف كبيرة , وعند الضغط المنطقة الثانية (ToLower) يتم تحويل النص المكتوب في أداة TextBox إلى حالة أحرف صغيرة , ويؤدي الضغط على المنطقة الثالثة إلى مسح (Erase) النص الموجود في الأداة TextBox , انقر على الأداة ImageMap مرتين وأضف الكود التالي :

C# كود

```

protected void ImageMap1_Click(object sender, ImageMapEventArgs e)
{
  switch (e.PostBackValue)

```



```

{
    case "ToUpper":
        TextBox1.Text = TextBox1.Text.ToUpper();
        break;
    case "ToLower":
        TextBox1.Text = TextBox1.Text.ToLower();
        break;
    case "Erase":
        TextBox1.Text = String.Empty;
        break;
}
}

```

كود VB

```

Select Case e.PostBackValue
    Case "ToUpper"
        TextBox1.Text = TextBox1.Text.ToUpper()
    Exit Select
    Case "ToLower"
        TextBox1.Text = TextBox1.Text.ToLower()
    Exit Select
    Case "Erase"
        TextBox1.Text = [String].Empty
    Exit Select
End Select

```

في الكود السابق استقبلنا الخاصية PostBackValue والتي يتم تحديد قيمتها بحسب المنطقة التي يتم الضغط عليها في الأداة ImageMap كما سنرى , كود الصفحة :

كود ASP.net

```

<div>

    <asp:ImageMap ID="ImageMap1" runat="server" HotSpotMode="PostBack"
        ImageUrl="~/images/textCase.jpg" onclick="ImageMap1_Click">

        <asp:RectangleHotSpot PostBackValue="ToUpper" Left="0" Top="0" Right="100"
            Bottom="30" AlternateText="ToUpper" />

        <asp:RectangleHotSpot PostBackValue="ToLower" Left="100" Top="0"
            Right="200" Bottom="30" AlternateText="ToLower" />

        <asp:RectangleHotSpot PostBackValue="Erase" Left="200" Top="0" Right="300"
            Bottom="30" AlternateText="Erase" />

    </asp:ImageMap>

    <br />

    <asp:TextBox ID="TextBox1" runat="server" TextMode="MultiLine" />

```

```
</div>
```

نلاحظ في الكود السابق تحديد الخاصية HotSpotMode إلى القيمة PostBack , وفي كل منطقة ساخنة تم تحديد قيمة مختلفة للخاصية PostBackValue لتمييز المنطقة التي تم النقر عليها .

استخدام أداة التحكم Panel

تمكننا هذه الأداة من التعامل مع مجموعة أدوات ASP.Net دفعة واحدة , حيث يتم تطبيق سلوك ما على جميع أدوات التحكم الموجودة ضمن أداة Panel , على سبيل المثال لو تم إخفاء أداة التحكم هذه فسيختفي معها جميع الأدوات المحتويات ضمنها وكذلك لو تم إلغاء تفعيل أداة Panel فسيتم إلغاء تفعيل جميع أدوات التحكم الموجودة ضمنها وهكذا ...

خصائص أداة التحكم Panel :

- ✓ DefaultButton : تمكننا هذه الخاصية من تحديد زر افتراضي ضمن الأداة Panel بحيث يتم تنفيذ الكود داخله بمجرد الضغط على مفتاح Enter من لوحة المفاتيح .
- ✓ Direction : تحديد اتجاه النص والأدوات ضمن Panel يمكن أن تأخذ إحدى القيم التالية
NotSet , LeftToRight , RightToLeft
- ✓ GroupingText : تحديد عنوان لأداة Panel
- ✓ HorizontalAlign : تحديد المحاذاة الأفقية لمحتويات أداة Panel ويمكن أن تأخذ إحدى القيم التالية
NotSet , Justify , Left , Center , Right
- ✓ ScrollBars : تمكننا من إظهار شريط تمرير أفقي أو عمودي (أو كلاهما) لأداة Panel وتأخذ إحدى القيم التالية
None , Horizontal , Vertical , Both , Auto

مثال : سنقوم في التطبيق التالي بإضافة Panel ومن ثم نضيف إليها مجموعة من أدوات التحكم المختلفة لنرى أن تطبيق أمر ما على Panel سيتم تطبيقه على الأدوات داخلها أيضاً , أنشئ صفحة جديدة , أضف أداة تحكم Panel وضع داخلها مجموعة من أدوات التحكم ثم ضع زر واحد خارجها وليكن كود الصفحة شبيه بالآتي :

كود ASP.net

```
<div>
<asp:Panel id="pnlContact" GroupingText="Contact Information" Runat="server">
  <asp:Label id="lblFirstName" Text="First Name:"
    AssociatedControlID="txtFirstName" Runat="server" />
<br />
```

```

<asp:TextBox id="txtFirstName" AutoCompleteType="FirstName" Runat="server"
/>
<br /><br />
<asp:Label id="lblLastname" Text="Last Name:"
AssociatedControlID="txtLastName" Runat="server" />
<br />
<asp:TextBox id="txtLastName" AutoCompleteType="LastName" Runat="server"
/>
<br /><br />
<asp:Button id="btnSubmit" Text="Submit" Runat="server" />
</asp:Panel>

<br />
<asp:Button ID="Button1" runat="server" Text="Visible" onclick="Button1_Click"
/>
</div>

```

أما الكود الذي يتم إضافته لحدث الضغط على الزر Button1 فهو :

كود C#

```

protected void Button1_Click(object sender, EventArgs e)
{
    pnlContact.Visible = !pnlContact.Visible;
}

```

كود VB

```

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs)
    pnlContact.Visible = Not pnlContact.Visible
End Sub

```

نفذ التطبيق السابق لترى تأثير خصائص الأداة Panel على محتوياتها .

إذاً : تعمل أداة التحكم Panel على تجميع أدوات التحكم ذات وظيفة أو صفات مشتركة يحددها أسلوب العمل في مجموعات , بغرض تنظيم الصفحة والاستفادة من المزايا التي تقدمها هذه الأداة .

استخدام أداة التحكم HyperLink

تعد هذه الأداة الطريقة الأكثر شيوعاً وبساطة للتنقل بين الصفحات المختلفة , وعلى النقيض من الأداة LinkButton فهي لاتقوم بإرسال البيانات إلى السيرفر .

خصائص أداة التحكم HyperLink :

- ✓ Enabled : تفعيل أو إلغاء تفعيل هذه الأداة .
- ✓ ImageUrl : لتحديد مسار واسم صورة ليتم عرضها كرابط .
- ✓ NavigateUrl : تحديد عنوان URL يتم الانتقال إليه عند الضغط على هذه الأداة .
- ✓ Text : تحديد النص الذي يظهر على هذه الأداة .
- ✓ Target : تمكنا من فتح صفحة الرابط في نافذة جديدة بالمستعرض .

ملاحظة

لو تم تحديد قيم للخاصيتين ImageUrl , Text معاً , عندئذ سيتم عرض الصورة لأن أولويتها أعلى ولن يتم عرض النص المحدد بالخاصية Text .

مثال : أنشئ صفحة جديدة , أضف أداة HyperLink وحدد خصائصها كالتالي :

كود ASP.net

```
<div>
  <asp:HyperLink ID="HyperLink1" runat="server" ImageUrl="~/images/img.png"
    NavigateUrl="~/Default14.aspx">Go to Page Defaul14 page
  </asp:HyperLink>
</div>
```

نفذ الصفحة السابقة ولاحظ عدم ظهور النص Go to Page Defaul14 page إنما ظهرت الصورة المحددة فقط .

خاتمة الفصل

في هذا الفصل , استعرضنا معاً مجموعة من الأدوات الأساسية لبناء تطبيقات الإنترنت وأهم خصائصها , سنكمل في الفصول اللاحقة التعرف على المزيد من الأدوات الأكثر تنوعاً وتعقيداً مع المزيد من الأمثلة المتنوعة , حيث سنلاحظ ارتفاع مستوى الأمثلة كلما تقدمنا أكثر في هذا الكتاب .

بطاقة شكر :

أتوجه بجزيل الشكر إلى كل اللذين يقدمون الدعم المعنوي والعلمي لإنجاز هذا الكتاب بأفضل مستوى ممكن , وأخص بالشكر الأخت شجون (عضو منتدى فيجوال بيسك للعرب) والتي تفضلت بكتابة أكواد الفيجوال بيسك لهذا الفصل .

للتواصل :

m-hajjkhalf@hotmail.com

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

محمد عمر الحاج خلف – سوريا

+963 932 033250