

الفصل السابع

سلسلة

ASP.NET

خطوة بخطوة حتى الاحتراف

C# & VB

اعداد المهندس

محمد عمر الحاج خلف

الفصل السابع

استخدام قوائم التكرار

في هذا الفصل

- ✓ مقدمة إلى قوائم التكرار
- ✓ الخصائص المشتركة لقوائم التكرار
- ✓ استخدام الأداة DropDownList
- ✓ استخدام الأداة RadioButton
- ✓ استخدام الأداة ListBox
- ✓ استخدام الأداة CheckBoxList
- ✓ استخدام الأداة BulletedList

استخدام أدوات قوائم التكرار

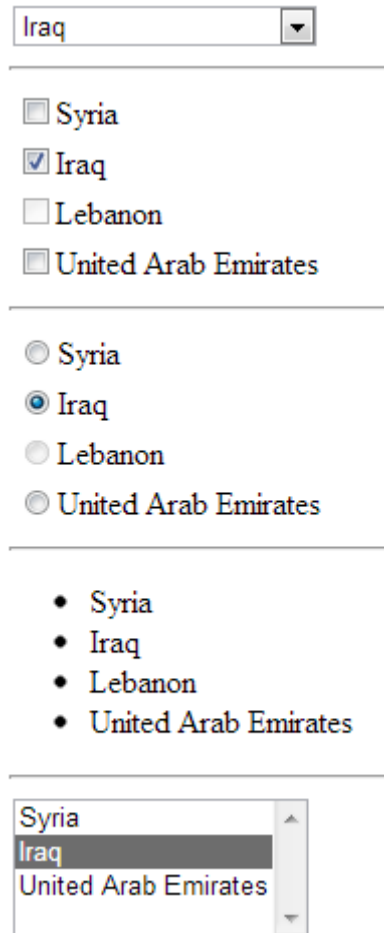
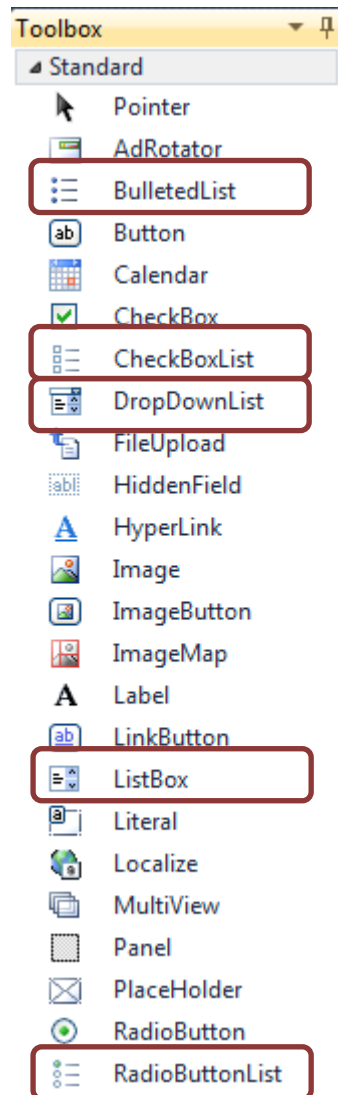
تستخدم هذه الأدوات لعرض قوائم بسيطة متعددة الخيارات ، على سبيل المثال نستخدم الأداة RadioButtonList لعرض قائمة ، كل عنصر فيها عبارة عن Radio Button ، كما نستخدم الأداة BulletedList لعرض قائمة من الروابط التشعبية Links ، وهكذا ...

في هذا الفصل سنتعلم كيفية التعامل مع أدوات القوائم الخمس وهي : DropDownList , RadioButtonList , CheckBoxList , BulletedList , ListBox . حيث نستعرض معا كيفية ربط هذه الأدوات مع قواعد البيانات ، بالإضافة إلى أمور أخرى ضمن هذا الفصل .

التصريح عن خيارات أدوات القوائم

يتم إضافة العناصر (الخيارات) إلى أدوات القوائم باستخدام الصف ListItem والذي نقوم بالتعامل معه وكأنه أداة تحكم مستقلة عبر الوسم <asp:ListItem> ، سنقوم في البداية بعرض مثال بسيط للتوضيح حيث نقوم بإنشاء صفحة جديدة ونضيف إليها أدوات القوائم الخمس . DropDownList , CheckBoxList , RadioButtonList , BulletedList , ListBox .

وسنقوم بإضافة أربع عناصر (خيارات) لكل منها كما يظهر في الصورة التالية :



ويكون كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
  <asp:DropDownList ID="DropDownList1" runat="server">
    <asp:ListItem Text="Syria" Value="sy" />
    <asp:ListItem Text="Iraq" Value="ir" Selected="True" />
    <asp:ListItem Text="Lebanon" Value="lb" Enabled="false" />
    <asp:ListItem Text="United Arab Emirates" Value="ae" />
  </asp:DropDownList>

  <asp:CheckBoxList ID="CheckBoxList1" runat="server" >
    <asp:ListItem Text="Syria" Value="sy" />
    <asp:ListItem Text="Iraq" Value="ir" Selected="True" />
    <asp:ListItem Text="Lebanon" Value="lb" Enabled="false" />
    <asp:ListItem Text="United Arab Emirates" Value="ae" />
  </asp:CheckBoxList>

  <asp:RadioButtonList ID="RadioButtonList1" runat="server">
    <asp:ListItem Text="Syria" Value="sy" />
    <asp:ListItem Text="Iraq" Value="ir" Selected="True" />
    <asp:ListItem Text="Lebanon" Value="lb" Enabled="false" />
    <asp:ListItem Text="United Arab Emirates" Value="ae" />
  </asp:RadioButtonList>

  <asp:BulletedList ID="BulletedList1" runat="server">
    <asp:ListItem Text="Syria" Value="sy" />
    <asp:ListItem Text="Iraq" Value="ir" Selected="True" />
    <asp:ListItem Text="Lebanon" Value="lb" Enabled="false" />
    <asp:ListItem Text="United Arab Emirates" Value="ae" />
  </asp:BulletedList>

  <asp:ListBox ID="ListBox1" runat="server">
    <asp:ListItem Text="Syria" Value="sy" />
    <asp:ListItem Text="Iraq" Value="ir" Selected="True" />
    <asp:ListItem Text="Lebanon" Value="lb" Enabled="false" />
    <asp:ListItem Text="United Arab Emirates" Value="ae" />
  </asp:ListBox>
</div>
```

نلاحظ من خلال المثال السابق أننا قمنا باستخدام العنصر <listItem> ليعبر عن خيار ضمن أداة القوائم التي ينتمي إليها ، ولكل عنصر <listItem> أربع خصائص :

- ❖ Text : نحدد من خلال هذه الخاصية النص الذي يعبر عن هذا العنصر ضمن القائمة .
- ❖ Value : لتحديد قيمة غير مرئية مرتبطة بهذا العنصر وغالبا ماتكون قيمة فريدة كالمفتاح الرئيسي .
- ❖ Enabled : حيث بالإمكان إلغاء تفعيل العنصر عبر هذه الخاصية .
- ❖ Selected : تمكنا من تحديد العنصر الذي نريده بشكل افتراضي .

بالعودة إلى المثال السابق ، لقد قمنا بعمل قائمة مؤلفة من أربعة عناصر ، حيث إن أسماء الدول تم إسنادها للخاصية Text وهي ظاهرة للعيان على عكس رموز الدول فهي غير مرئية لأنها مسندة للخاصية Value ، كما تم تحديد القيمة المختارة افتراضيا (العراق) وذلك عبر الخاصية Selected والتي يؤدي اسنادها إلى أكثر من خيار ضمن بعض الأدوات إلى حدوث مشكلة كما هو الحال في الأداة DropDownList ، كما أن وجودها لامتني له في الأداة BulletedList حيث يتم تجاهلها ، أخيرا نجد الخيار (لبنان) لايمكن اختياره أو أنه غير ظاهر أساسا وسبب ذلك إلغاء تفعيل الخاصية Enabled كما هو ظاهر في كود الصفحة .

ربط أدوات القوائم مع قواعد البيانات

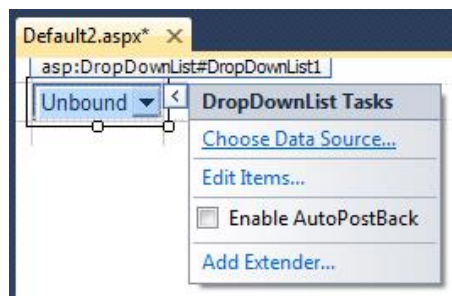
تتم الفائدة الفعلية من أدوات القوائم بربطها مع قواعد البيانات وبالتالي الحصول على قوائم ديناميكية العناصر ، أدوات القوائم الخمس لها أسلوب مماثل بالربط مع قواعد البيانات ، لنقم بداية ببناء قاعدة بيانات بسيطة (كما تعلمنا في الفصل السابق) حيث ننشئ فيها جدول واحد باسم Marks وله البنية التالية :

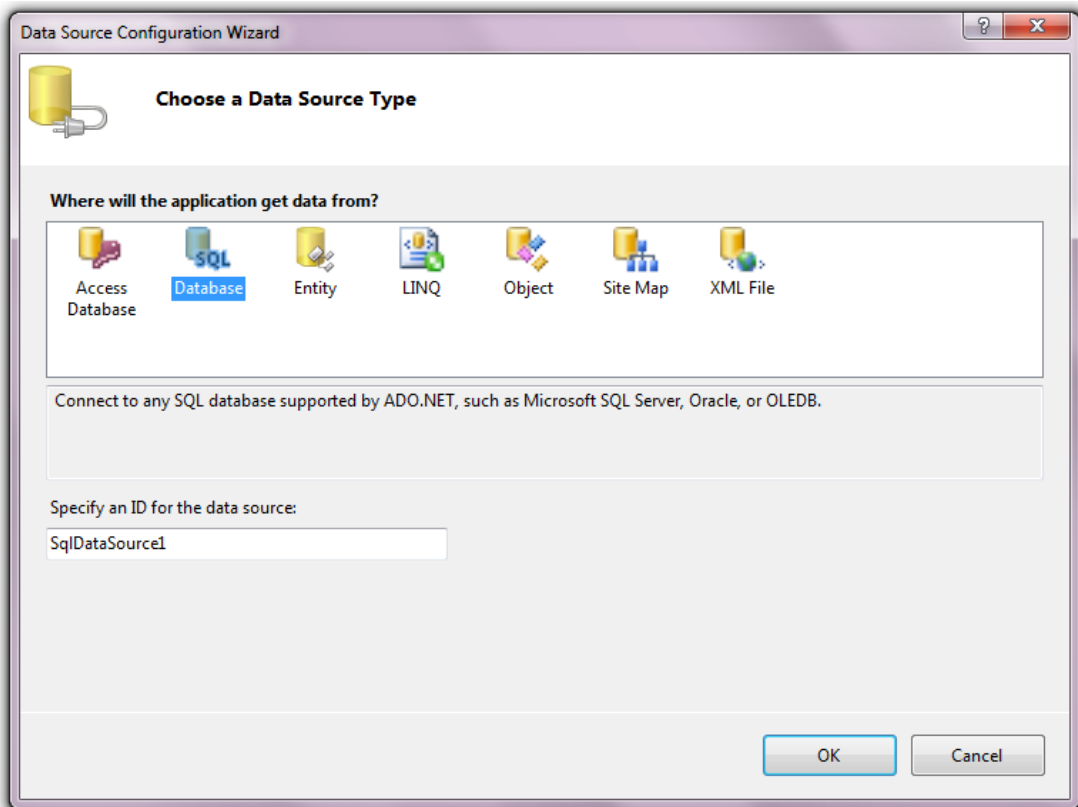
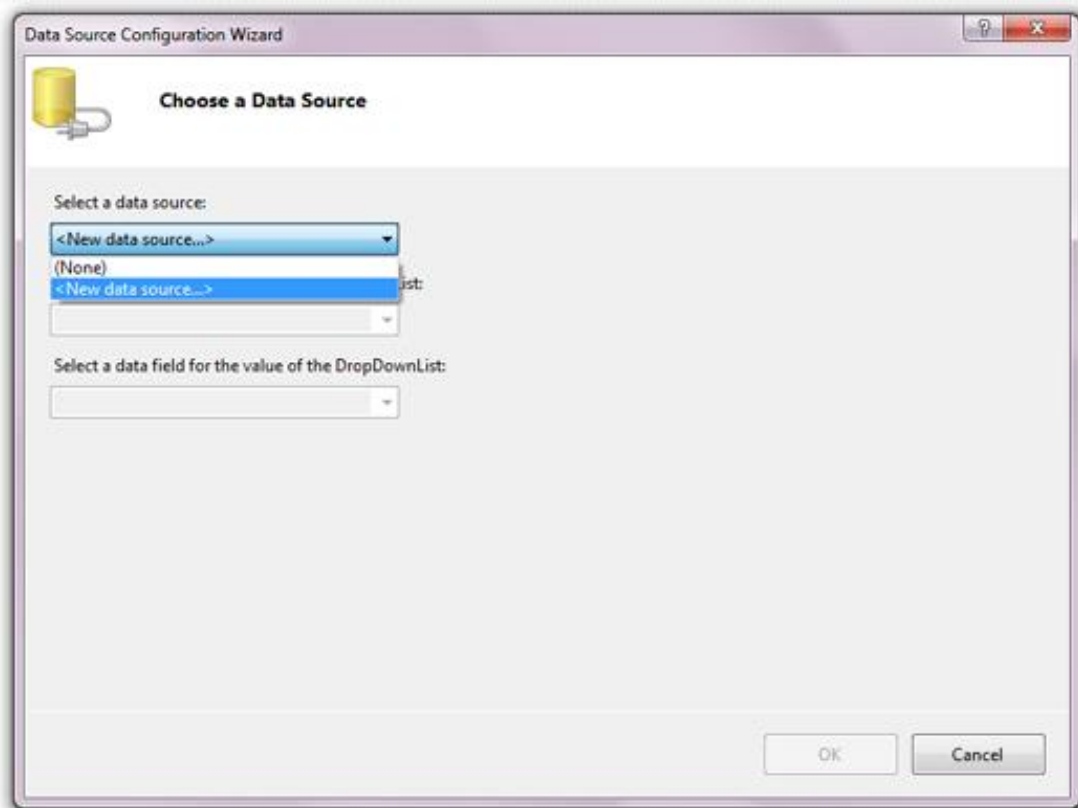
اسم الحقل	نمط البيانات	وصف الحقل
id	int	مفتاح رئيسي
name	Nvarchar(50)	لتخزين اسم الطالب
mark	int	لتخزين علامة الطالب

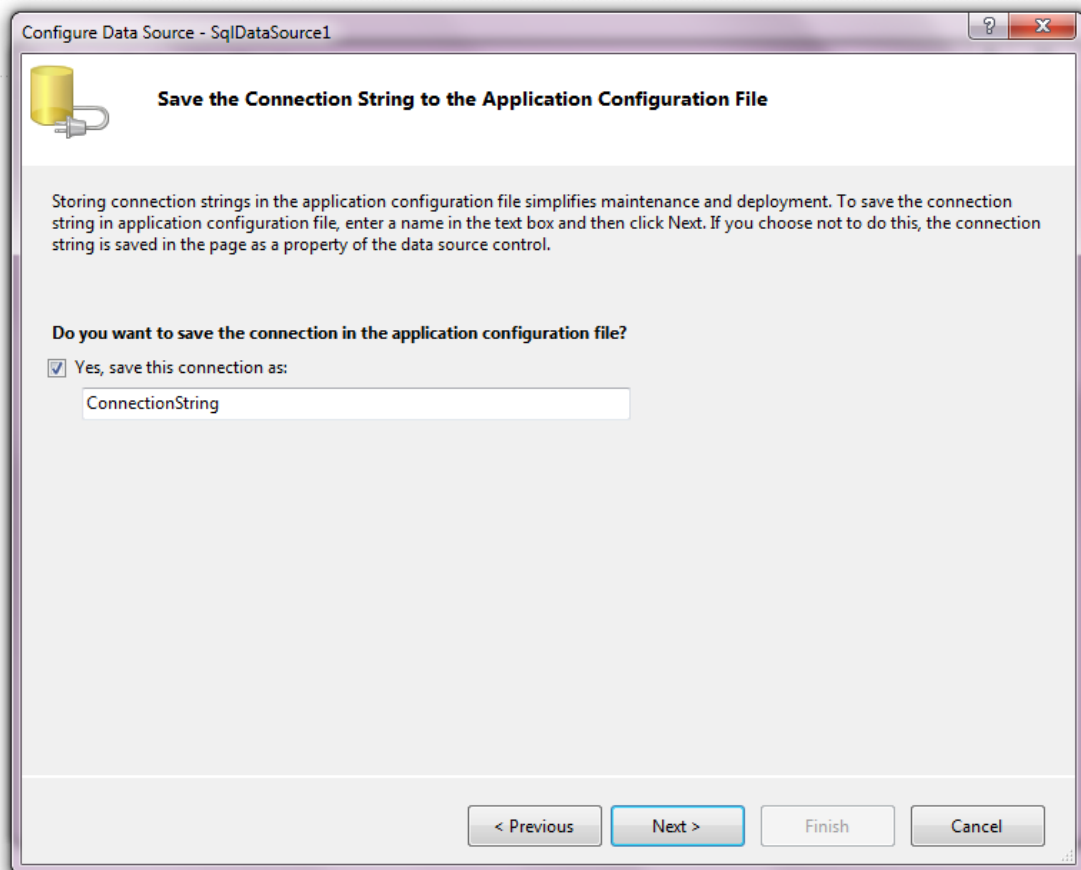
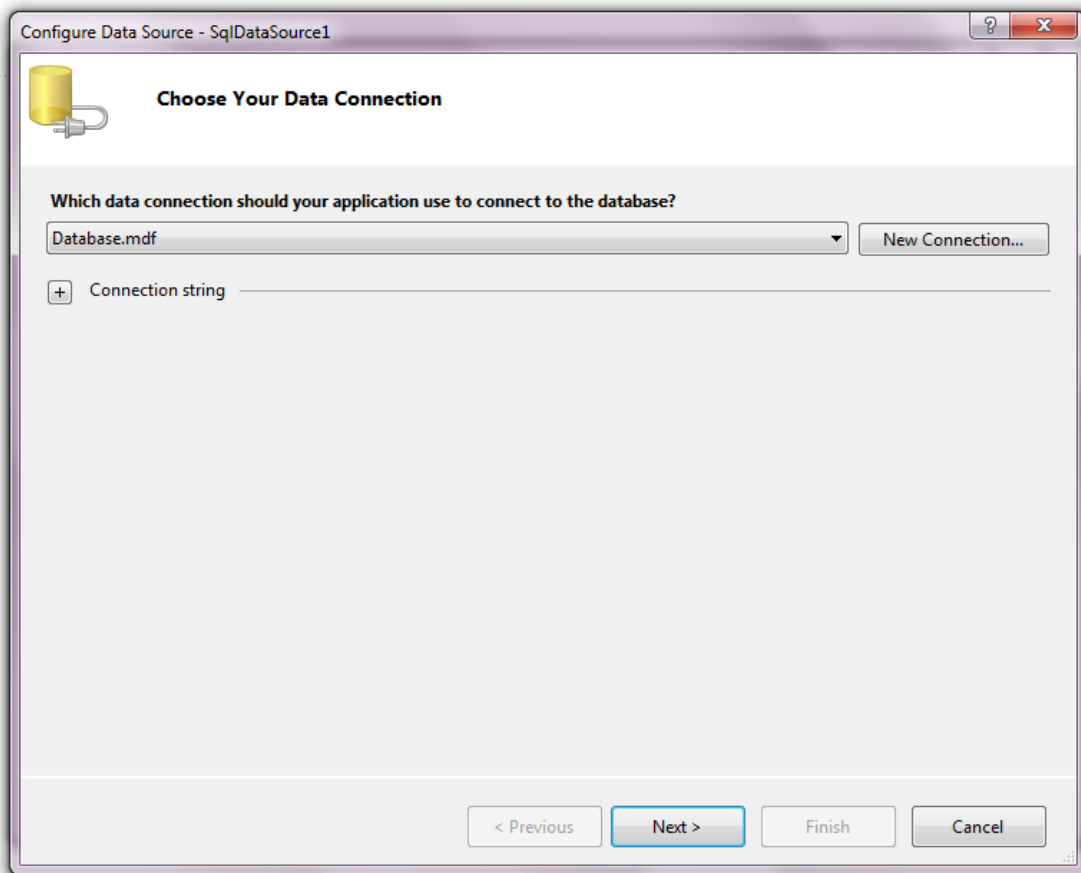
نقوم بعد ذلك بملئه ببعض البيانات لتطبيق المثال عليها .

Column Name	Data Type	id	name	mark
id	int	1	Mohammed	86
name	nvarchar(100)	2	Batowl	92
mark	int	3	Iman	79
		4	Rahaf	77
		* NULL	NULL	NULL

نقوم الآن بإضافة صفحة جديدة ، ونضيف فيها أداة DropDownList ، ونربطها مع الجدول السابق كما توضح الصور التالية :







Configure Data Source - SqlDataSource1

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure
 Specify columns from a table or view

Name:

Columns:

<input checked="" type="checkbox"/>	*
<input type="checkbox"/>	id
<input type="checkbox"/>	name
<input type="checkbox"/>	mark

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
SELECT * FROM [marks]

< Previous Next > Finish Cancel

Configure Data Source - SqlDataSource1

Test Query

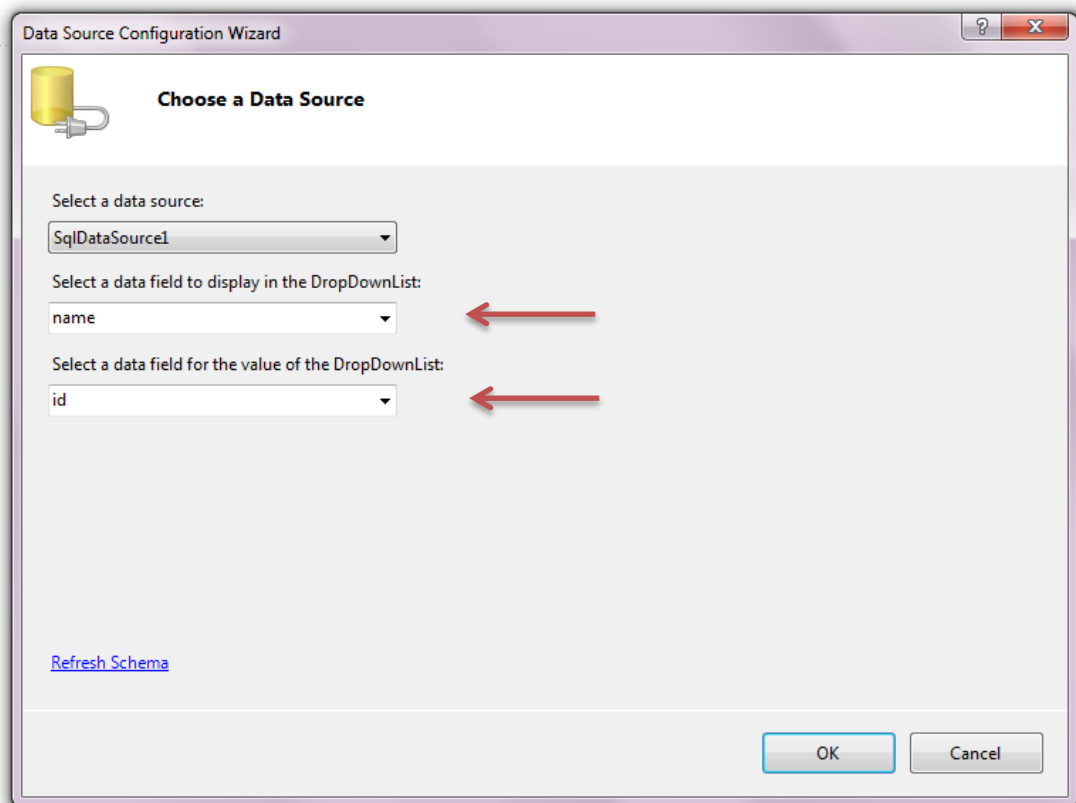
To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

id	name	mark
1	Mohammed	86
2	Batowl	92
3	Iman	79
4	Rahaf	77

Test Query

SELECT statement:
SELECT * FROM [marks]

< Previous Next > Finish Cancel



وبهذا نكون قد انتهينا من ربط الأداة DropDownList مع جدول علامات الطلاب ، حيث سيكون اسم الطالب هو النص الظاهر (الخاصية DataTextField) في حين يبقى رقمه قيمة مخفية (الخاصية DataValueField) ، لنستعرض كود الصفحة حيث يجب أن يكون مشابها للتالي :

كود ASP.net

```
<div>
  <asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [marks]" />
</div>
```

الآن نستطيع وبشكل مشابه ربط باقي أدوات القوائم مع الجدول السابق (لاداعي لتكرار جميع الخطوات السابقة حيث يمكن الاستفادة من وجود الأداة SqlDataSource وبالتالي نربط أداة القوائم معها عبر الخاصية DataSourceID ثم نقوم بظبط قيم الخاصيتين DataTextField و DataValueField كما يظهر الكود التالي :

```

<div>
  <asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:CheckBoxList ID="CheckBoxList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:BulletedList ID="BulletedList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:ListBox ID="ListBox1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:RadioButtonList ID="RadioButtonList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="name"
    DataValueField="id" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %"
    SelectCommand="SELECT * FROM [marks]" />
</div>

```

وتكون نتيجة تنفيذ الصفحة السابقة مشابهة للصورة المجاورة .

تحديد خيار المستخدم

كما رأينا فإن أدوات القوائم تقدم أسلوب عرض رائع للخيارات ولكن الشيء الأهم هو معرفة ماقام المستخدم باختياره ، أدوات القوائم توفر ثلاث خصائص نستطيع استخدام أيها لمعرفة خيار المستخدم وهي :

- ❖ SelectedIndex : للحصول على دليل العنصر المختار .
- ❖ SelectedValue : للحصول على قيمة العنصر المختار .
- ❖ SelectedItem : للحصول على العنصر المختار .

يمكن استخدام الخاصيتين selectedIndex , selectedValue في عمليات الاسناد أيضا وليس فقط في عملية الإرجاع (أي تدعمان الطريقتين Get , Set) .

وتجدر الإشارة إلى أن الخاصيتين `selectedItem` , `selectedValue` تعيدان معلومات فقط عن أول عنصر مختار وذلك في حال قام المستخدم بتحديد أكثر من خيار في الأدوات التي تسمح بذلك .

لنقم الآن بإضافة صفحة جديدة ، نضيف إليها أداة `DropDownList` ونربطها مع الجدول `Marks` كما تعلمنا في فقرة سابقة . ثم نضيف زر و أداة `Label` لنسند لها القيمة التي سيقوم المستخدم باختيارها ، كما هو موضح في الصورة المجاورة .

Rahaf

في حدث الضغط على الزر نكتب الكود التالي :

كود C#

```
Label1.Text = DropDownList1.SelectedItem.Text;
```

كود VB

```
Label1.Text = DropDownList1.SelectedItem.Text
```

كود ASP.net

```
<div>
  <asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="SqlDataSource1" DataTextField="name"
    DataValueField="id">
  </asp:DropDownList>
  <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
    Text="select" />
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %">
    SelectCommand="SELECT * FROM [marks]"></asp:SqlDataSource>
  <br />
  <hr />
  <br />
  <asp:Label ID="Label1" runat="server"></asp:Label>
</div>
```

استخدام أداة التحكم `DropDownList`

تسمح أداة التحكم `DropDownList` للمستخدم باختيار قيمة وحيدة ، وهي من أكثر أدوات القوائم انتشارا على صفحات الانترنت ، يمكن أن تكون عناصرها ثابتة أو متغيرة عبر ربطها

مع مصدر بيانات كما رأينا في فقرات سابقة ، لاشيء جديد نضيفه على هذه الأداة بعد المعلومات التي ناقشناها سابقا ولكن سنستعرض ضمن هذه الفقرة مثالين جديرين بالمتابعة والتطبيق .

Column Name	Data Type
id	int
country	nvarchar(50)
city	nvarchar(50)

سنقوم بداية بإنشاء جدول جديد ضمن قاعدة البيانات وذلك لنتمكن من تطبيق المثالين التاليين ، الجدول سيحتوي أسماء بعض الدول وبعض المدن التي تنتمي لكل دولة ، ويمكن أن تكون له البنية التالية :

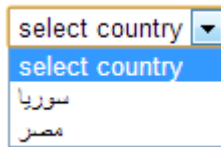
اسم الحقل	نمط البيانات	وصف الحقل
id	int	مفتاح رئيسي
country	Nvarchar(50)	لتخزين اسم الدولة
city	Nvarchar(50)	لتخزين اسم المدينة

يمكن أن نعطيها الاسم cities ، ونقوم بملئه ببعض البيانات كما هو موضح في الصورة التالية :

id	country	city
2	سوريا	دمشق
3	سوريا	حلب
5	سوريا	حمص
6	مصر	القاهرة
7	مصر	الإسكندرية

وبهذا نكون أصبحنا جاهزين لتطبيق الأمثلة .

سنقوم من خلال المثال الأول بتعلم كيفية إضافة خيار ما إلى مجموعة الخيارات التي نقوم بجلبها من مصدر البيانات ، حيث توضح الصورة المجاورة وجود الخيارين (سوريا ، مصر) وقد تم جلبهما من قاعدة البيانات في حين إن (select country) هو عبارة عن نص توضيحي للمستخدم ونضيفه في حيز التصميم وليس لقاعدة البيانات علاقة به .



تتيح الخاصية AppendDataBoundItems إمكانية إضافة عناصر (خيارات) إلى أدوات القوائم مع الإبقاء على العناصر التي تم إحضارها من مصدر البيانات ، أضف صفحة جديدة وأضف إليها أداة DropDownList ثم اجعل الخاصية AppendDataBoundItems تأخذ القيمة True ، ثم قم بربطها مع الجدول cities بحيث تعرض أسماء الدول فقط وبدون تكرار

وبعد ذلك نقوم بإضافة عنصر ListItem إلى DropDownList حيث نجعل قيمة النص الظاهر له "select country" أما القيمة المخفية المرتبطة به فلا داعي لها في مثالنا هذا وبالتالي يصبح كود الصفحة كاملا بالشكل التالي :

كود ASP.net

```

<div>
  <asp:DropDownList ID="DropDownList1" runat="server"
    AppendDataBoundItems="true"
    DataSourceID="SqlDataSource1"
    DataTextField="country"
    DataValueField="country" >
    <asp:ListItem Text="select country" Value="" />
  </asp:DropDownList>

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT DISTINCT [country] FROM [cities]" />
</div>

```

قم بتنفيذ الصفحة السابقة ولاحظ ظهور "select country" كأول خيار ضمن مجموعة الخيارات ، قم بتعديل المثال السابق واجعل قيمة الخاصية AppendDataBoundItems تساوي False وأعد تنفيذ الصفحة لتلاحظ عدم ظهور "select country" ضمن مجموعة

الخيارات ، (أعد تفعيل الخيار السابق لنكمل المثال) . بالإمكان ربط الأداة DropDownList مع أداة التحقق RequiredFieldValidator وذلك للتأكد من أن المستخدم قد اختار قيمة ما وسيتم تجاهل الخيار "select country" لأننا لم نقم بتحديد قيمة للخاصية value التابعة له وهذا مانريده (أي أن لا يتم اعتبار select country خيار كباقي الخيارات) ، أضف أداة التحقق RequiredFieldValidator إلى الصفحة السابقة واربطها مع الأداة DropDownList ثم قم بإضافة زر إلى الصفحة السابقة حيث سيصبح كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
  <asp:DropDownList ID="DropDownList1" runat="server"
    AppendDataBoundItems="true"
    DataSourceID="SqlDataSource1"
    DataTextField="country"
    DataValueField="country" >
    <asp:ListItem Text="select country" Value="" />
  </asp:DropDownList>

  <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
    ControlToValidate="DropDownList1" ErrorMessage="(Required)" />
  <br />

  <asp:Button ID="Submit" runat="server" Text="Button" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %">
    SelectCommand="SELECT DISTINCT [country] FROM [cities]"
    onselecting="SqlDataSource1_Selecting" />
</div>
```

قم بتنفيذ الصفحة السابقة ، واضغط على الزر دون أن تحدد دولة من الأداة DropDownList ولاحظ ظهور رسالة تطالبك باختيار قيمة ما .

المثال الثاني :

سنقوم من خلال هذا المثال بإضافة أداتين DropDownList حيث تقوم الأولى بعرض أسماء الدول ، في حين تقوم الثانية بعرض أسماء المدن الموجودة ضمن الدولة المحددة في الأداة DropDownList الأولى ، فيما يلي صورة توضيحية لناتج هذا المثال .



سيناريو العمل : يقوم المستخدم باختيار دولة ضمن الأداة DropDownList1 ، نقوم بإرسال هذه القيمة إلى السيرفر ، يقوم السيرفر بتنفيذ استعلام لجلب مدن الدولة المختارة ، ثم إعادة النتائج ليتم عرضها ضمن DropDownList2 .

خطوات العمل :

أنشئ صفحة جديدة وأضف إليها أداتين DropDownList ، قم بربط الأداة DropDownList1 مع الجدول cities بحيث تقوم بعرض أسماء الدول وبدون تكرار ، فيما يلي صورة توضيحية

Configure Data Source - SqlDataSource1

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure

Specify columns from a table or view

Name: cities

Columns:

*

id

country

city

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT DISTINCT [country] FROM [cities]

< Previous Next > Finish Cancel

نقوم بعد ذلك بربط الأداة DropDownList2 مع الجدول السابق أيضا ولكن سنقوم بجعل الاستعلام يستقبل بارامتر وهو اسم الدولة المحددة بالأداة DropDownList1 ليتم جلب المدن على أساس هذا الاختيار ، الصور التالية توضح خطوات العمل

Configure Data Source - SqlDataSource2

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure
 Specify columns from a table or view

Name: cities

Columns:

- *
- id
- country
- city

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
 SELECT [city] FROM [cities] WHERE ([country] = @country)

< Previous Next > Finish Cancel

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: country

Operator: =

Source: Control

Parameter properties

Control ID: DropDownList1

Default value:

SQL Expression: [country] = @country2

Value: DropDownList1.SelectedValue

Add

Remove

SQL Expression	Value
[country] = @country	DropDownList1.SelectedValue

OK Cancel

ونتابع باقي الخطوات بالشكل التقليدي ، نعود إلى الأداة DropDownList1 ونفعل الخاصية AutoPostBack (نسندها للقيمة True) وذلك ليتم إرسال القيمة الجديدة إلى السيرفر في كل مرة يقوم فيها المستخدم بتغيير خياره ، كود الصفحة يصبح بالشكل التالي :

ASP.net كود

```
<div>

    <asp:DropDownList ID="DropDownList1" runat="server"
        AutoPostBack="True"
        DataSourceID="SqlDataSource1"
        DataTextField="country" DataValueField="country">
    </asp:DropDownList>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
        SelectCommand="SELECT DISTINCT [country] FROM [cities]">
    </asp:SqlDataSource>

    <asp:DropDownList ID="DropDownList2" runat="server"
        DataSourceID="SqlDataSource2"
        DataTextField="city" DataValueField="city">
    </asp:DropDownList>

    <asp:SqlDataSource ID="SqlDataSource2" runat="server"
        ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
        SelectCommand="SELECT [city] FROM [cities]
            WHERE ([country] = @country)" >
        <SelectParameters>
            <asp:ControlParameter
                ControlID="DropDownList1"
                Name="country"
                PropertyName="SelectedValue"
                Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>

</div>
```

يتضح من خلال الكود السابق أن الأداة ControlParameter هي التي تقوم بتجهيز البارامتر المدعو country وتسليمه إلى الاستعلام ، حيث يتم جلب قيمة هذا البارامتر من الأداة المحددة عبر الخاصية ControlID وهي DropDownList1 والخاصية التي تعينها هي القيمة المختارة لذلك تم تحديد "PropertyName="SelectedValue" ، نفذ الصفحة السابقة ، قم باختيار دولة ما لتظهر أسماء مدنها .

استخدام الأداة RadioButtonList

إن أسلوب عمل الأداة RadioButtonList يشبه إلى حد كبير أسلوب الأداة DropDownList حيث إن كلا الأدوات يسمح للمستخدم بانتقاء خيار واحد فقط من ضمن قائمة الخيارات المتاحة،

يتم من خلال هذه الأداة إظهار كل خيار على شكل RadioButton ، ولنا إمكانية اختيار اتجاه إظهار عناصر القائمة أفقيا أو عموديا ، من خصائص هذه الأداة :

- ❖ RepeatDirection : تحديد اسلوب عرض عناصر القائمة أفقي أو عمودي وتأخذ إحدى القيمتين Horizontal أو Vertical .
- ❖ RepeatColumns : لتحديد عدد الأعمدة في حال تم اختيار أسلوب العرض الأفقي عبر الخاصية السابقة (Horizontal) .
- ❖ RepeatLayout : تحدد أن كان سيتم تنسيق عناصر القائمة ضمن جدول HTML أم لا حيث تأخذ إحدى القيمتين Table , Flow .

مثال : سنقوم خلال هذا المثال بربط أداة RadioButtonList مع الجدول cities السابق ليتم عرض قائمة المدن ، ثم نقوم بتحديد خيار المستخدم .

دمشق حاب
 حمص القاهرة
 الإسكندرية

حاب

إنشئ صفحة جديدة ، أضف إليها أداة RadioButtonList ثم قم بربطها مع الجدول cities بحيث يتم عرض قائمة المدن ، أضف زر Button وأداة Lable ، وفي حدث النقر على الزر اكتب الكود التالي :

كود C#

```
Label1.Text = RadioButtonList1.SelectedItem.Text;
```

كود VB

```
Label1.Text = RadioButtonList1.SelectedItem.Text
```

وبالتالي يكون كود الصفحة كالتالي :

كود ASP.net

```
<div>  
  <asp:RadioButtonList ID="RadioButtonList1" runat="server"  
    DataSourceID="SqlDataSource1" DataTextField="city"  
    DataValueField="city"  
    RepeatColumns="2" RepeatDirection="Horizontal">  
  </asp:RadioButtonList>  
  
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"  
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"  
    SelectCommand="SELECT [city] FROM [cities]" />  
  <asp:Button ID="Button1" runat="server" Text="Submit"  
    onclick="Button1_Click" />  
</div>
```

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

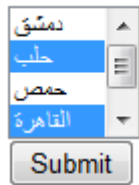
استخدام الأداة ListBox

تسمح هذه الأداة للمستخدم بانتقاء أكثر من خيار واحد ، حيث تظهر هذه الأداة على شكل صندوق ويدخله عدد من الخيارات ، ومن أهم خصائصها

- ❖ Rows : لتحديد عدد الخيارات التي يتم عرضها قبل دون أن يتم عرض شريط التمرير (scroll bar) .
- ❖ SelectionMode : لتحديد إمكانية اختيار أكثر من قيمة أو قيمة واحدة فقط وتأخذ إحدى القيمتين Single , Multiple .

مثال :

في المثال التالي نستعرض كيفية التعامل مع الأداة ListBox مع السماح للمستخدم باختيار أكثر من قيمة ومن ثم الحصول برمجيا على هذه القيم التي اختارها .



حلب
القاهرة

أنشئ صفحة جديدة ، أضف عليها أداة ListBox ثم اربطها مع الجدول cities بحيث تقوم بعرض أسماء المدن ، ثم اعط الخاصية SelectionMode القيمة Multiple وذلك لتمكين المستخدم من اختيار أكثر من قيمة ، واجعل الخاصية Rows تساوي القيمة ٤ وبذلك ستظهر أربع خيارات أما الخيارات الأخرى فنشاهدها من خلال تحريك شريط التمرير . أضف أداة Label وزر Button وفي حدث الضغط على الزر نكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    foreach (ListItem item in ListBox1.Items)
        if (item.Selected)
            Label1.Text += "<li>" + item.Text;
}
```

كود VB

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Button1.Click
    For Each item As ListItem In ListBox1.Items
        If item.Selected Then
            Label1.Text += "<li>" + item.Text
        End If
    Next
End Sub
```

ويكون كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
  <asp:ListBox ID="ListBox1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="city" DataValueField="city"
    SelectionMode="Multiple"
    Rows="4" />
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [city] FROM [cities]" />
  <br />
  <asp:Button ID="Button1" runat="server" Text="Submit"
    onclick="Button1_Click" />
  <hr />
  <asp:Label ID="Label1" runat="server" Text="" />
</div>
```

نفذ الصفحة السابقة ، جرب اختيار أكثر من قيمة (استخدم المفتاح Ctrl من لوحة المفاتيح لتتمكن من تحديد أكثر من قيمة) ، ثم اضغط على الزر لتحصل على قائمة الخيارات التي انتقيتها . إن الخصائص SelectedItem ، SelectedIndex ، SelectedValue والتي استعرضناها في بداية الفصل تعيد معلومات عن أول قيمة مختارة من مجموعة القيم ، لذلك فإننا لم نستخدمهم في المثال السابق ، كما لايجوز استخدامهم عندما نسمح للمستخدم بانتقاء أكثر من قيمة . إن الأداة ListBox قليلة الاستخدام ضمن صفحات الانترنت وذلك لأنها لا تظهر للمستخدم بشكل واضح إمكانية اختيار أكثر من قيمة ، لذلك غالبا ما يتم التعامل مع الأداة CheckBox عوضا عنها .

استخدام الأداة CheckBoxList

تقوم الأداة CheckBoxList بعرض قائمة الخيارات على شكل مجموعة من الأدوات CheckBox ، ويمكن للمستخدم اختيار أكثر من خيار واحد ، وبإمكاننا تحديد أسلوب عرض قائمة الخيارات بشكل أفقي أو عمودي بشكل مشابه لأسلوب عرض الأداة RadioButtonList ومن أهم خصائص الأداة CheckBoxList :

- ❖ RepeatDirection : تحديد أسلوب عرض عناصر القائمة أفقي أو عمودي وتأخذ إحدى القيمتين Horizontal أو Vertical .
- ❖ RepeatColumns : لتحديد عدد الأعمدة في حال تم اختيار أسلوب العرض الأفقي (Horizontal) عبر الخاصية السابقة.
- ❖ RepeatLayout : تحدد أن كان سيتم تنسيق عناصر القائمة ضمن جدول HTML أم لا حيث تأخذ إحدى القيمتين Table , Flow .

دمشق حلب
 حمص القاهرة
 الإسكندرية

حمص
الإسكندرية

مثال :

في المثال التالي سنقوم بعرض قائمة المدن من الجدول cities ومن ثم السماح للمستخدم باختيار أكثر من مدينة ، وأخيرا الكود الذي يمكننا من الحصول على خيارات المستخدم .

إنشئ صفحة جديدة ، أضف إليها أداة CheckBoxList ثم قم بربطها مع الجدول cities بحيث يتم عرض قائمة المدن ، أضف زر Button وأداة Lable ، وفي حدث النقر على الزر اكتب الكود التالي :

كود C#

```
protected void Button1_Click(object sender, EventArgs e)
{
    foreach (ListItem item in CheckBoxList1.Items)
        if (item.Selected)
            Label1.Text += "<li>" + item.Text;
}
```

كود VB

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    For Each item As ListItem In CheckBoxList1.Items
        If item.Selected Then
            Label1.Text += "<li>" + item.Text
        End If
    Next
End Sub
```

ويكون كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
    <asp:CheckBoxList ID="CheckBoxList1" runat="server"
        DataSourceID="SqlDataSource1"
        DataTextField="city" DataValueField="city"
        RepeatColumns="2"
        RepeatDirection="Horizontal" >
    </asp:CheckBoxList>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
        SelectCommand="SELECT [city] FROM [cities]" />
    <asp:Button ID="Button1" runat="server" Text="Submit"
        onclick="Button1_Click" />
    <hr />
    <asp:Label ID="Label1" runat="server" Text="" />
</div>
```

استخدام الأداة BulletedList

تقوم هذه الأداة بعرض قائمة الخيارات بأحد الأساليب الثلاثة التالية :

- ❖ Text : يتم عرض كل خيار كنص عادي ، وهو الخيار الافتراضي .
- ❖ LinkButton : يتم عرض كل خيار كأداة LinkButton حيث يكون متاح استخدام جميع خصائص هذه الأداة .
- ❖ HyperLink : يتم عرض كل خيار كرابط تشعبي يقودنا إلى عنوان نقوم بتحديدده .

من أهم خصائص الأداة BulletedList :

- ❖ DisplayMode : لتحديد ماهية عناصر القائمة وتأخذ إحدى القيم التالية : Text , HyperLink , LinkButton ، وقد تم توضيح كل خيار منهم .
- ❖ BulletStyle : لتحديد تنسيق إظهار عناصر القائمة (تعداد رقمي ، تعداد أبجدي ، روماني ..) ، وتأخذ إحدى القيم التالية :
 ١. NotSet : • , • , • , • وهي الخيار الافتراضي .
 ٢. Numbered : 1. , 2. , 3. , ...
 ٣. LowerAlpha : a. , b. , c , ...
 ٤. UpperAlpha : A. , B. , C. , ...
 ٥. LowerRoman : i. , ii , iii , iv , v , ...
 ٦. UpperRoman : I. , II. , III. , IV. , V , ...
 ٧. Disc : • , • , • , •
 ٨. Circle : ○ , ○ , ○ , ○
 ٩. Square : ▪ , ▪ , ▪ , ▪
 ١٠. CustomImage : لتحديد صورة يتم عرضها بجانب كل خيار ضمن القائمة .
- ❖ BulletImageUrl : لتحديد مسار الصورة التي سيتم عرضها ، ويجب أن تكون قيمة الخاصية BulletStyle تساوي CustomImage .
- ❖ FirstBulletNumber : لتحديد رقم بداية ترقيم عناصر القائمة ، ويعمل هذا الخيار إذا كان أسلوب عرض عناصر القائمة قابل للعد (أي أن الخاصية BulletStyle تأخذ إحدى القيم التالية , UpperAlpha , LowerAlpha , Numbered , UpperRoman , LowerRoman) .

قبل البدء بتطبيق الأمثلة على هذه الأداة ، نقوم ببناء جدول جديد ضمن قاعدة البيانات ولتكن له البنية التالية :

اسم الحقل	نمط البيانات	وصف الحقل
id	int	مفتاح رئيسي
website	Nvarchar(50)	اسم موقع انترنت
url	Nvarchar(50)	عنوان الموقع على الانترنت

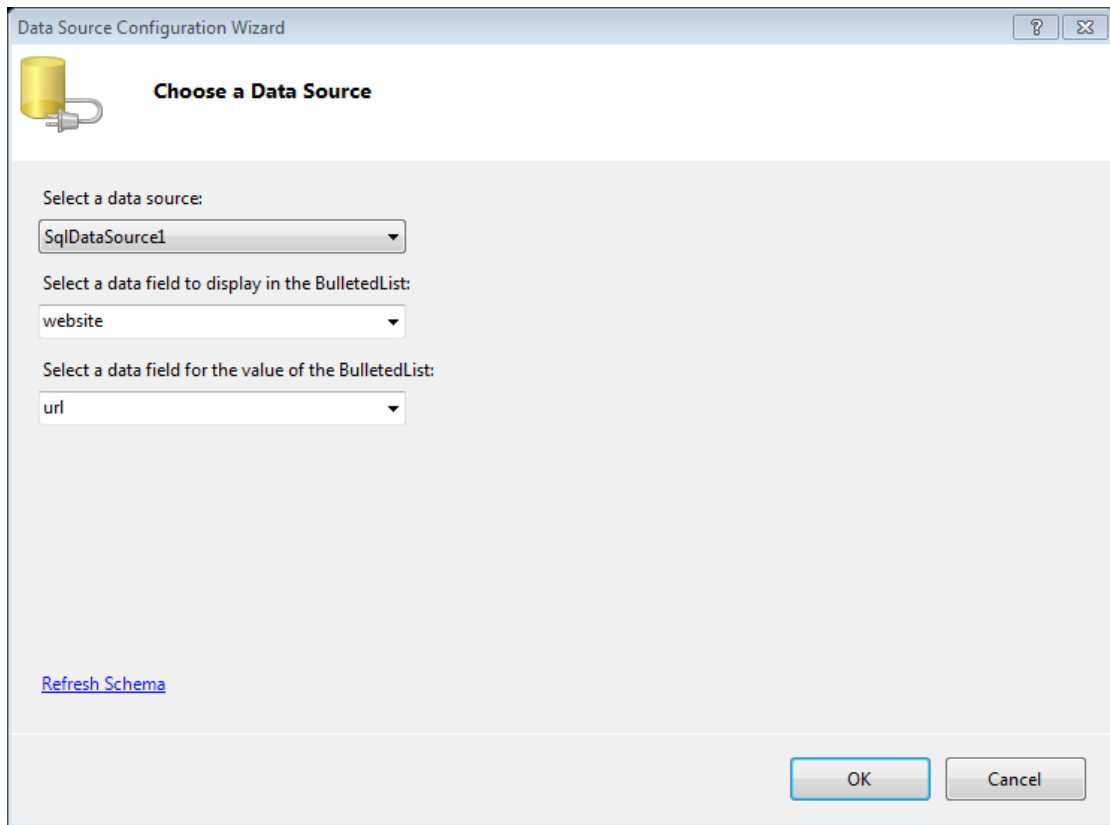
Column Name	Data Type
id	int
website	nvarchar(50)
url	nvarchar(50)

يمكن أن نسميه MyWebSites ، نقوم بعد ذلك بملئه ببعض البيانات ، على سبيل المثال :

id	website	url
1	Google	http://www.google.com
2	FaceBook	http://www.facebook.com
3	Yahoo	http://www.yahoo.com

ننشئ صفحة جديدة و عليها أداة التحكم BulletedList ونقوم بربطها مع الجدول السابق كما تعلمنا ضمن الفقرات السابقة .

نحدد بعد ذلك الحقل website كنص ظاهر ، في حين يكون الحقل url هو القيمة المرتبطة



كود الصفحة يكون بالشكل التالي :

```
ASP.net كود

<div>
  <asp:BulletedList ID="BulletedList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="website"
    DataValueField="url" >
  </asp:BulletedList>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [MyWebSites]" />
</div>
```

عند تنفيذ الصفحة السابقة ستظهر قائمة كل عنصر فيها عبارة عن نص عادي وذلك لأن نمط عناصر الأداة BulletedList الافتراضي هو Text كما أوضحنا سابقا . قم الآن بتعديل قيمة الخاصية BulletStyle وتنفيذ الصفحة لتتعرف على أسلوب العرض لكل منها .

لنقم الآن بتعديل قيمة الخاصية DisplayMode وجعلها تساوي القيمة HyperLink وبالتالي سيتم عرض عناصر القائمة كروابط إلى الصفحات المحددة ضمن الجدول MyWebSites .

- [Google](#)
- [FaceBook](#)
- [Yahoo](#)

ويكون كود الصفحة بالشكل التالي :

كود ASP.net

```
<div>
  <asp:BulletedList ID="BulletedList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="website"
    DataValueField="url"
    DisplayMode="HyperLink" >
  </asp:BulletedList>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [MyWebSites]" />
</div>
```

لاحظ من خلال الكود السابق أن الخاصية DataTextField هي التي تحدد النص الذي سيتم عرضه ، في حين أن الخاصية DataValueField هي التي تحدد عنوان الصفحة التي سيتم الانتقال إليها عند الضغط على هذا الخيار .

نفس الأمر تماما ينطبق لوقمت بتعديل قيمة الخاصية DisplayMode إلى LinkButton بدلا من HyperLink مع الاستفادة من الخصائص التي توفرها الأداة LinkButton والتي قد تم شرح كيفية استخدامها في فصل سابق .

ملاحظة

تختلف الأداة BulletedList عن باقي أدوات القوائم بأنها لاتدعم الخصائص SelectedIndex , SelectedItem , SelectedValue .

الخاتمة

إلى هنا نأتي إلى نهاية هذا الفصل والذي تعلمنا فيه كيفية التعامل مع أدوات القوائم المختلفة ، والتي يدعم بعضها إمكانية اختيار قيمة وحيدة فقط كالأداة DropDownList والأداة RadioButtonList ، في حين يسمح بعضها الآخر للمستخدم باختيار أكثر من قيمة كالأداة ListBox و الأداة CheckBoxList ، وأخيرا استعرضنا معا الأساليب المختلفة التي يمكن أن تظهر بها الأداة BulletedList فإما قائمة نصية بسيطة أو قائمة من الروابط التشعبية ، إن أدوات القوائم شائعة الاستخدام في صفحات الانترنت وخاصة بأسلوبها الديناميكي عند ربطها مع قواعد البيانات وهذا ماتعلمناه خلال هذا الفصل .

لتحميل جميع أجزاء هذه السلسلة المتوفرة حتى الآن :

<https://www.mediafire.com/folder/u44z9omx9ajjv/asp.net>

http://www.4shared.com/folder/i34_ICo0/ASP.html

كما يتم طرح هذه الفصول و مقالات أخرى على منتدى فيجوال بيسك لكل العرب

<http://www.vb4arb.com>

للتواصل :

m-hajkhalaf@hotmail.com

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

محمد عمر الحاج خلف – سوريا