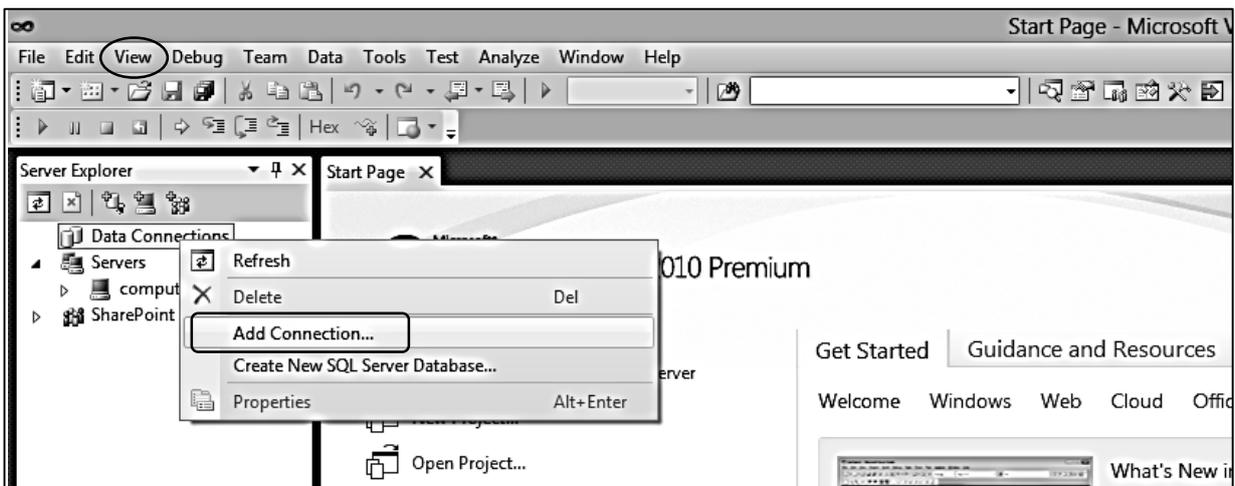


3 Customer records

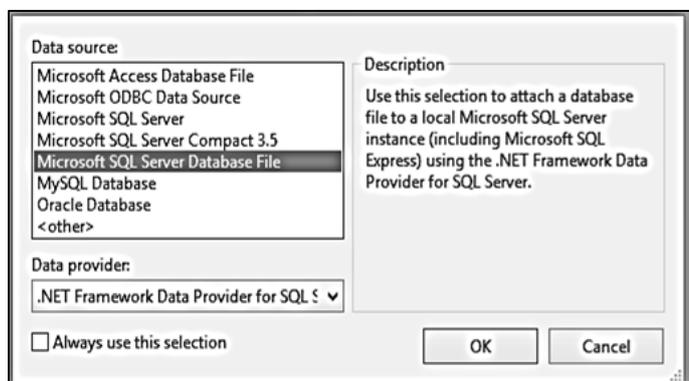
In this program we will investigate how records in a database can be displayed on a web page, and how new records can be entered on a web page and uploaded to the database.

Open Visual Studio. Go to the **Server Explorer** window. If this is not visible, open it by selecting **View** from the top menu bar, then click on the **Server Explorer** list item.

Right click on **Data Connections**, then select **Add Connection**.

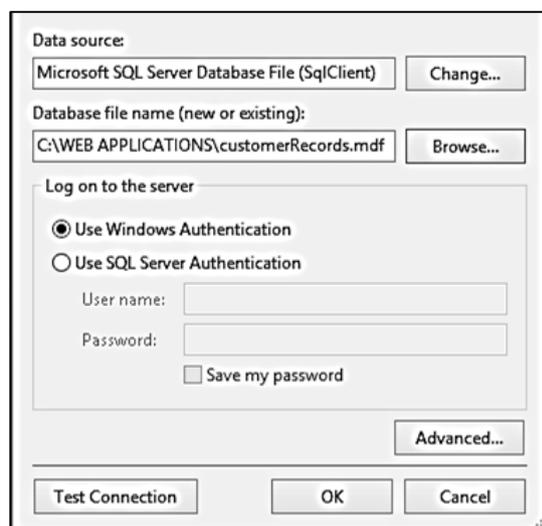


If you are offered a choice of **Data Source**, select '**Microsoft SQL Server Database File**' then click to continue.

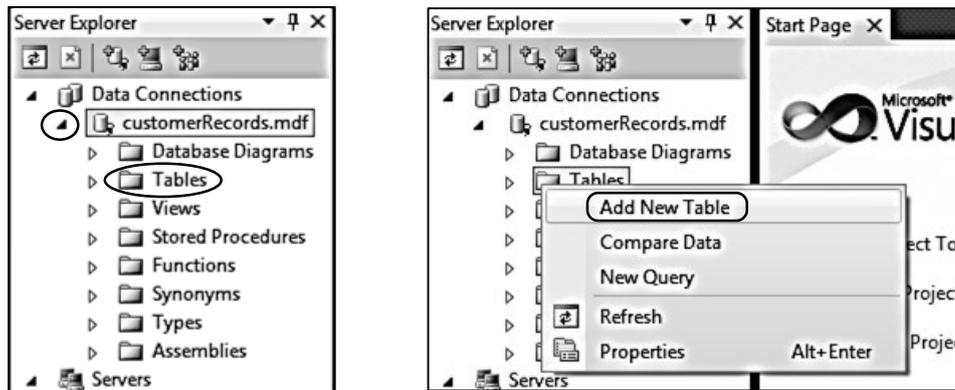


Use the '**Browse**' button to select the location on your computer where you wish to save the database. Enter the database name '**customerRecords.mdf**'.

Click '**OK**', then confirm that you wish to create a new database file.



An icon for the **customerRecords** database should appear. Click the small arrow to the left of this icon to open a list of options. Right click on **Tables** and select '**Add New Table**'.



The table design view will open. Add the series of data fields shown below. Enter the **Column Names** and select the **Data Types** from the drop down lists. **CustomerID** will be an **integer** (whole number), whilst the other fields will contain text. The numbers in brackets for the **nchar** and **nvarchar** data types indicate the maximum number of characters which we will store in these fields. The values should be set to allow more than enough space for the expected data entries.

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
Surname	nvarchar(50)	<input checked="" type="checkbox"/>
Forename	nvarchar(50)	<input checked="" type="checkbox"/>
Title	nchar(10)	<input checked="" type="checkbox"/>
Address	nvarchar(50)	<input checked="" type="checkbox"/>
Town	nvarchar(50)	<input checked="" type="checkbox"/>
Postcode	nchar(10)	<input checked="" type="checkbox"/>
Phone	nchar(20)	<input checked="" type="checkbox"/>
	nchar(20)	<input type="checkbox"/>
	ntext	
	numeric(18, 0)	
	nvarchar(50)	

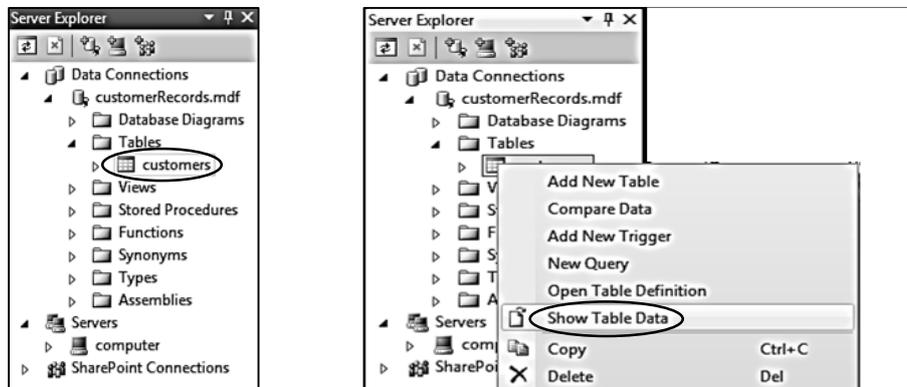
It will be convenient to set the **CustomerID** field to be an auto-number which is allocated automatically by the computer when a new record is added. To do this, select the **CustomerID** field and scroll down the **Column Properties** list to find the '**Identity Specification**' entry. Click the small arrow icon to open further options, then set the '**Is Identity**' property to '**Yes**'.

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
Surname	nvarchar(50)	<input checked="" type="checkbox"/>
Forename	nvarchar(50)	<input checked="" type="checkbox"/>
Title	nchar(10)	<input checked="" type="checkbox"/>

Column Properties	
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1

When the list of fields is completed, close the list window by clicking the cross in the top right hand corner. Confirm to save the table, and give the name '*customers*'.

Right click on the *customers* table icon and select '*Show Table Data*'.

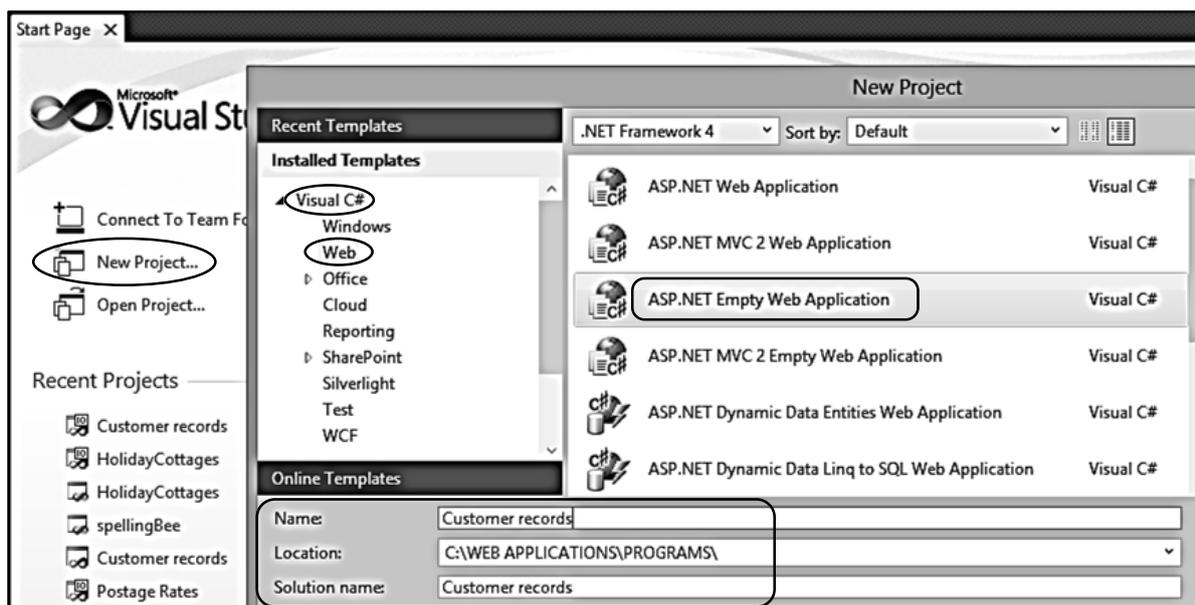


The empty *customers* table will open. Add test data for about four customers. Do not enter values for the *CustomerID* field – these will be added automatically by the computer.

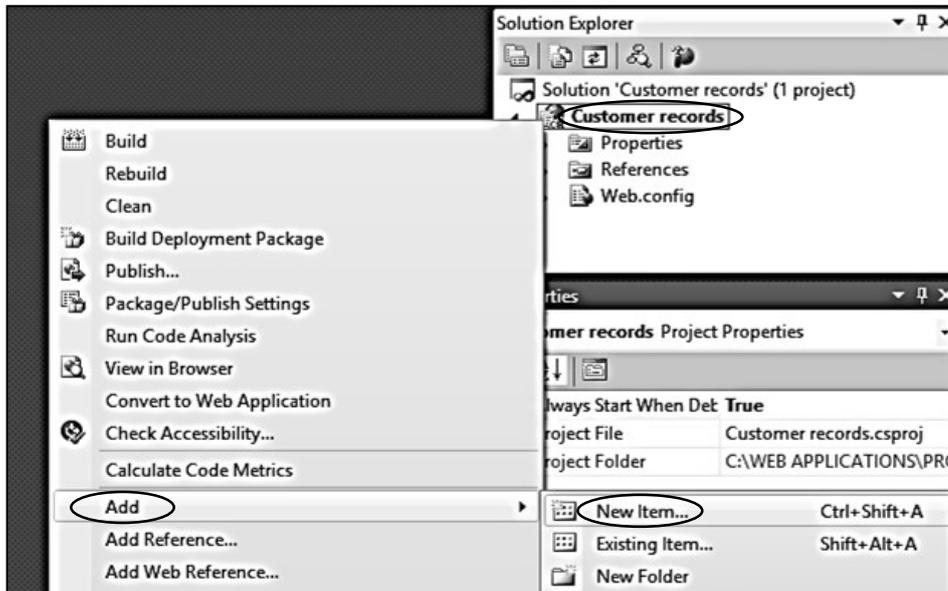
CustomerID	Surname	Forename	Title	Address	Town	Postcode	Phone
1	Roberts	David	Mr	12 Moelwyn Terrace	Ffestiniog	LL38 7RT	01766 879332
2	Johnson	Steven	Mr	Ty Gwyn	Bala	LL29 6WR	01356 792144
3	Green	Janet	Ms	7 High Street	Tywyn	LL32 9QB	01452 567199
4	Davidson	Sally	Ms	7 Oak Tree Crescent	Chester	CH2 6TK	01266 823451
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Close the customers window. We will now create a web page to display the records you have entered.

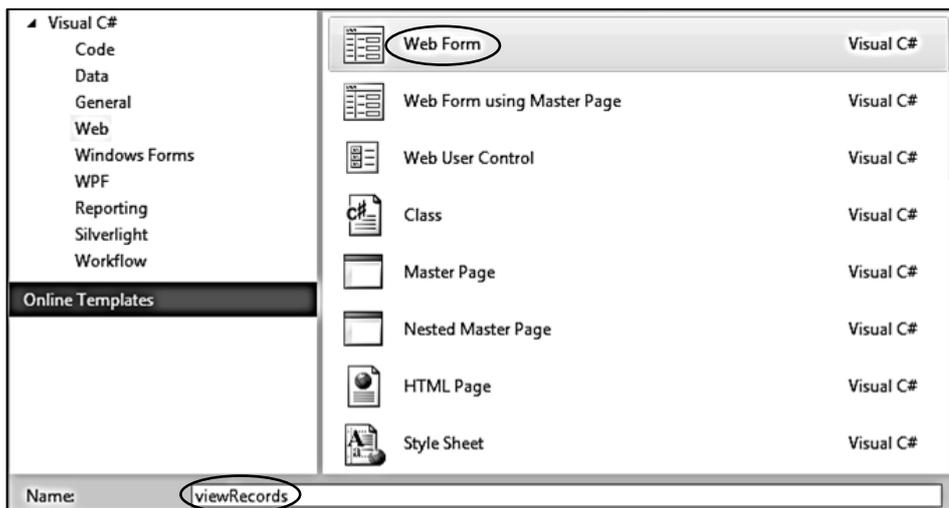
From the Visual Studio start page, select *New Project*. Choose *Visual C#/Web* as the project type. Select *ASP.NET Empty Web Application*. Use the *Browse* button to identify a location where the project can be stored. Give the name '*Customer records*' for the project.



Go to the **Solution Explorer** window and right click the **Customer records** project icon. Select **Add/New Item**.



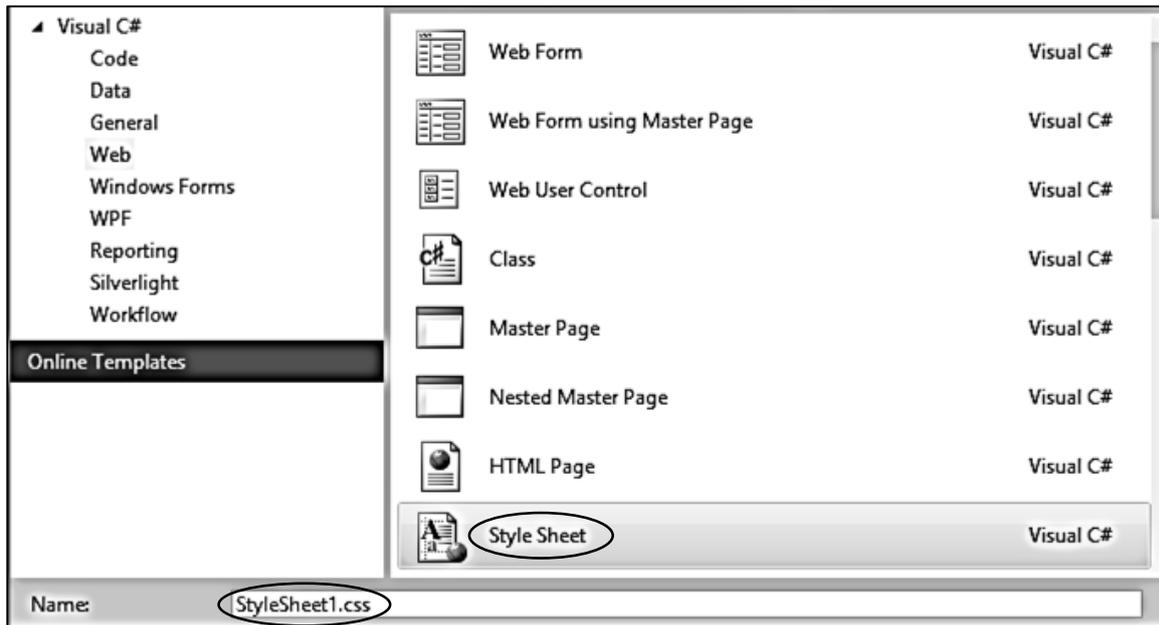
Select **Web Form** and give the name '**viewRecords**'.



The HTML window for **viewRecords** page will open. Add the window title '**Customer records**'. Give the name '**content**' to the **division**, and insert an **<h1>** heading.

```
<head runat="server">
  <title>Customer records</title>
</head>
<body>
  <form id="form1" runat="server">
    <div id="content">
      <h1>Customer records</h1>
    </div>
  </form>
</body>
```

We will use a style sheet to format the text output. Go again to the **Solution Explorer** window and right click the **Customer records** project icon. Select **Add/New Item**, then **Style Sheet**. Accept the name **'StyleSheet1'**.



We will add some basic formatting commands to the style sheet. These will set the font to **Arial** and select a large font size for the **<h1>** heading. We also set the width of the content area to 1000pixels and centre this on the screen.

```
body
{
    font-family: Arial, Helvetica, sans-serif;
}

h1
{
    font-size: large
}

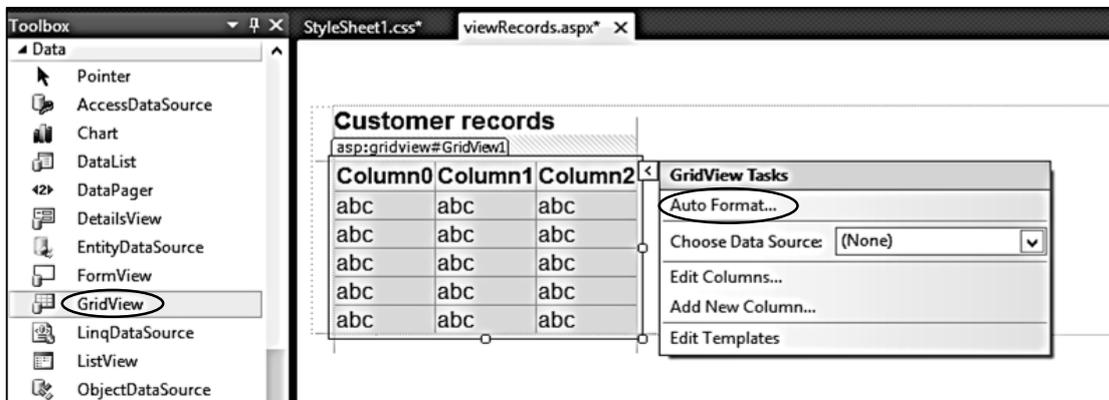
#content
{
    width: 1000px;
    background-color: White;
    margin-left: auto;
    margin-right: auto;
    color: Black;
    margin-top: 30px;
}
```

Return to the **viewRecords** HTML page and add a link to the style sheet.

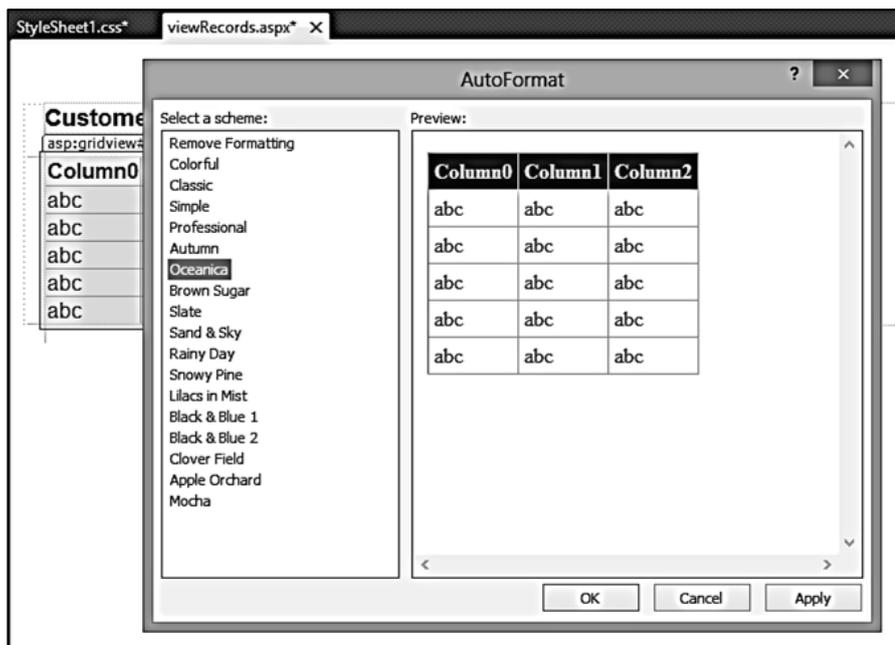
```
<head runat="server">
  <title>Customer records</title>
  <link rel="stylesheet" type="text/css" href="StyleSheet1.css" />
</head>
```

Click the **Design** button to change to the page layout view. Click the mouse at the right of the heading '**Customer records**' and press the '**enter**' key to move downwards. A new paragraph area will be created.

Go to the **View** option on the main menu and select **Toolbox**. Open the **Data** section of the Toolbox and identify the **GridView** component. Drag and drop the **grid view** onto the web page below the heading. This table will be used to display our customer records.

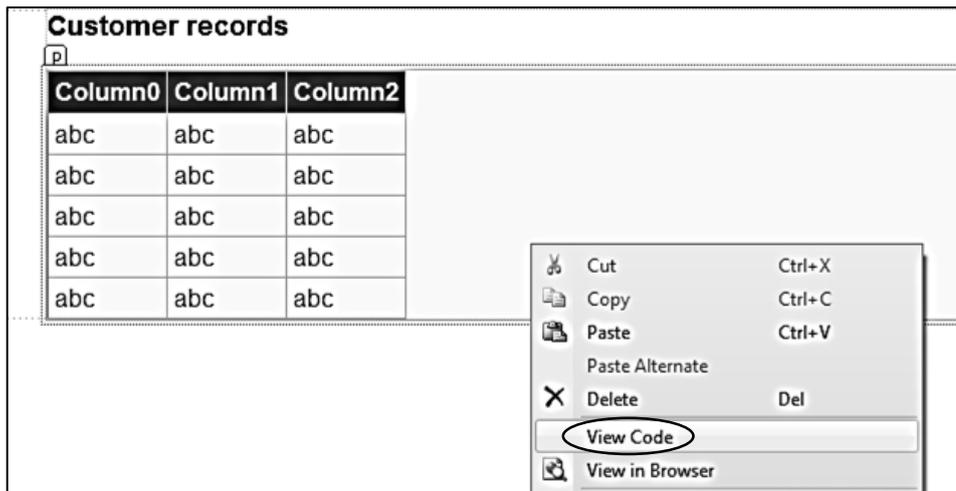


Click the small arrow icon at the top right hand corner of the table to open a list of options, then select **Auto Format**.



A range of styles are available for the table display. You are free to choose any of the styles. For example, **Oceanica** will work well. Click OK to continue.

Right click in an empty area of the page to open an options menu. Select '**View Code**'.



Add '**using System.Data**' and '**using System.Data.SqlClient**' directives to the list at the top of the code page. Also add the location of the customer database which you created earlier.

```
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using System.Data.SqlClient;
using System.Data;

namespace Customer_records
{
    public partial class viewRecords : System.Web.UI.Page
    {
        string databaseLocation="C:\\WEB APPLICATIONS\\customerRecords.mdf;";

        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

The next stage is to add code to the **Page_Load()** method to access the database table and transfer the customer records to the grid view on the web page. The code required is shown on the next page.

Please note that the command:

```
SqlConnection cnTB = new SqlConnection(@"Data Source=.\SQLEXPRESS;
AttachDbFilename=" + databaseLocation + "Integrated Security=True;
Connect Timeout=30; User Instance=True");
```

should be entered as one single line of code without line breaks.

```
protected void Page_Load(object sender, EventArgs e)
{
    DataSet dsCustomer = new DataSet();

    SqlConnection cnTB = new SqlConnection(@"Data Source=. \SQLEXPRESS;
        AttachDbFilename=" + databaseLocation + "Integrated Security=True;
        Connect Timeout=30; User Instance=True");
    cnTB.Open();
    SqlCommand cmCustomer = new SqlCommand();
    cmCustomer.Connection = cnTB;
    cmCustomer.CommandType = CommandType.Text;
    cmCustomer.CommandText = "SELECT * FROM customers";
    SqlDataAdapter daCustomer = new SqlDataAdapter(cmCustomer);
    daCustomer.Fill(dsCustomer);
    cnTB.Close();

    GridView1.DataSource = dsCustomer;
    GridView1.DataBind();
}
```

The purpose of this quite complex section of code is to:

- define a '**dsCustomer**' **data set** ready to store the customer records,
- open a connection to the database,
- carry out the SQL command '**SELECT * FROM customers**' to load the records from the **customers** table,
- transfer the records into the **dsCustomer** data set,

then finally:

- link the **dsCustomer** data set to the **grid view** component so that the records are displayed.

Go to the menu bar at the top of the Visual Studio page and select Build/Build Solution. If no error messages appear, click the green arrow icon below the menu bar to run the web page. The table of customers should be displayed.

localhost:57461/viewRecords.aspx

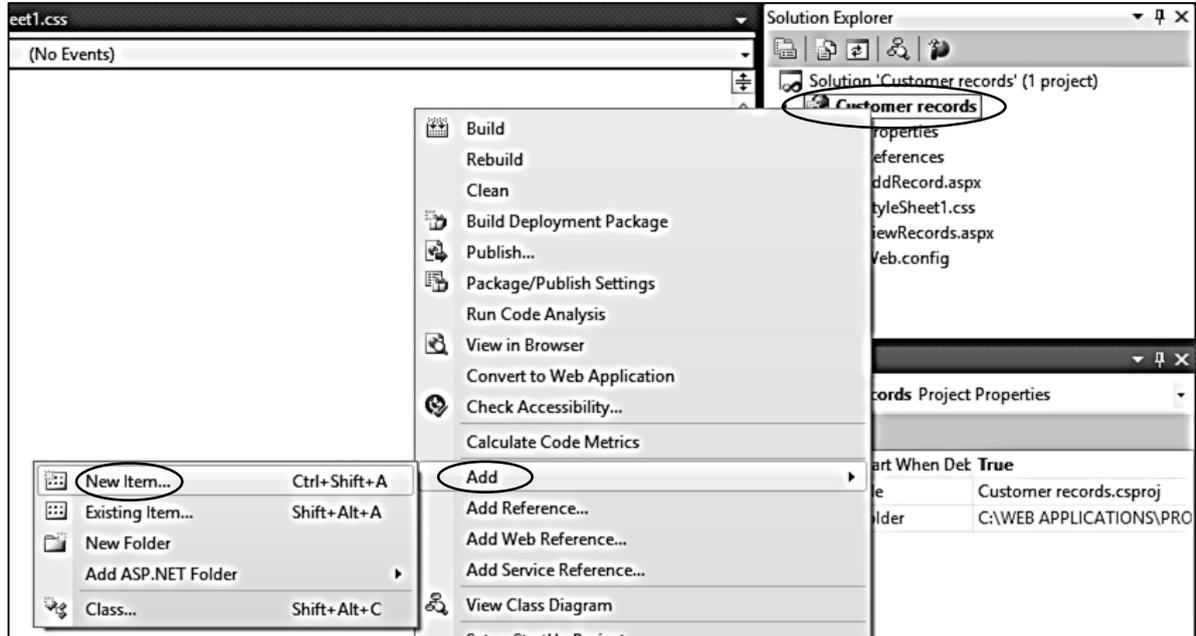
Customer records

CustomerID	Surname	Forename	Title	Address	Town	Postcode	Phone
1	Roberts	David	Mr	12 Moelwyn Terrace	Ffestiniog	LL38 7RT	01766 879332
2	Johnson	Steven	Mr	Ty Gwyn	Bala	LL29 6WR	01356 792144
3	Green	Janet	Ms	7 High Street	Tywyn	LL32 9QB	01452 567199
4	Davidson	Sally	Ms	7 Oak Tree Crescent	Chester	CH2 6TK	01266 823451

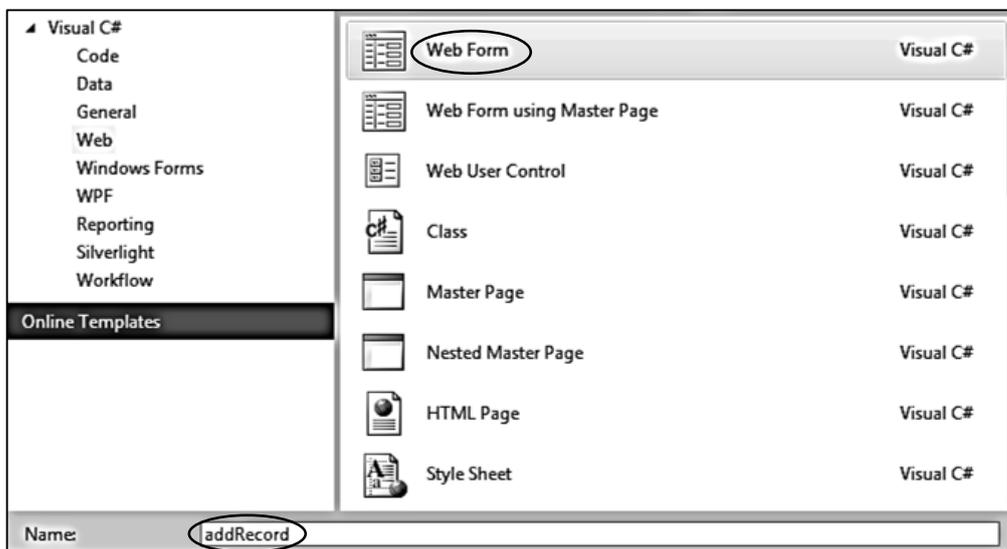
Close the web browser and return to **Visual Studio**. Click the **Debug** menu option to stop debugging.

A further option we can offer to the user is to add customer records directly to the database from the web site.

Begin by creating another web page. Go to the *Solution Explorer* window, right click on the *Customer records* project icon and select *Add/New Item*.

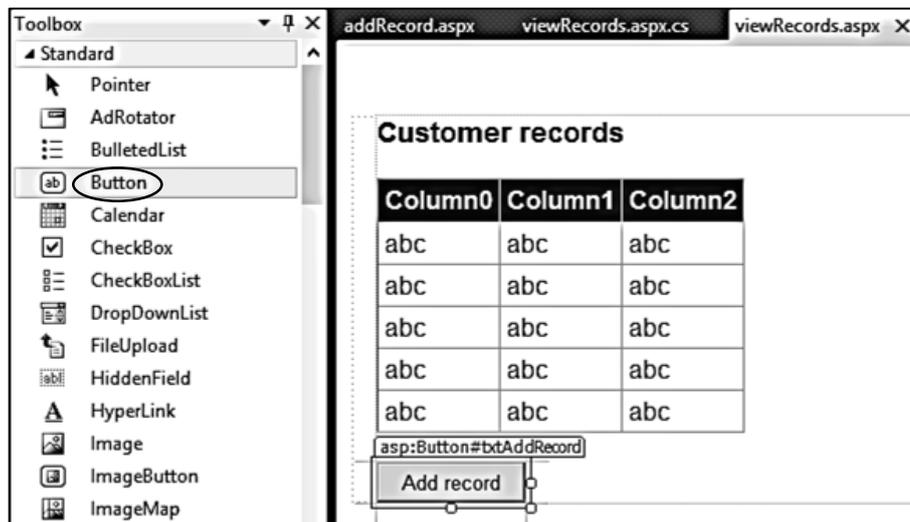


Choose *Web Form* and give the page name '*addRecord*'.



Return to *viewRecords* design screen to add a button which will link to the *addRecord* page.

Click the mouse to the right of the grid view table and press 'enter' to create a new paragraph. Locate the **Button** component in the **Toolbox** and drag this to the page below the table. Go to the **Properties** window and change the name of the button by setting the **ID** value to '*btnAddRecord*'. Change the **Text** property to '*Add record*'.



Double click the button to create a *button_click()* method. Add a line of code which will open the *addRecord* web page.

```
protected void txtAddRecord_Click(object sender, EventArgs e)
{
    Response.Redirect("AddRecord.aspx");
}
```

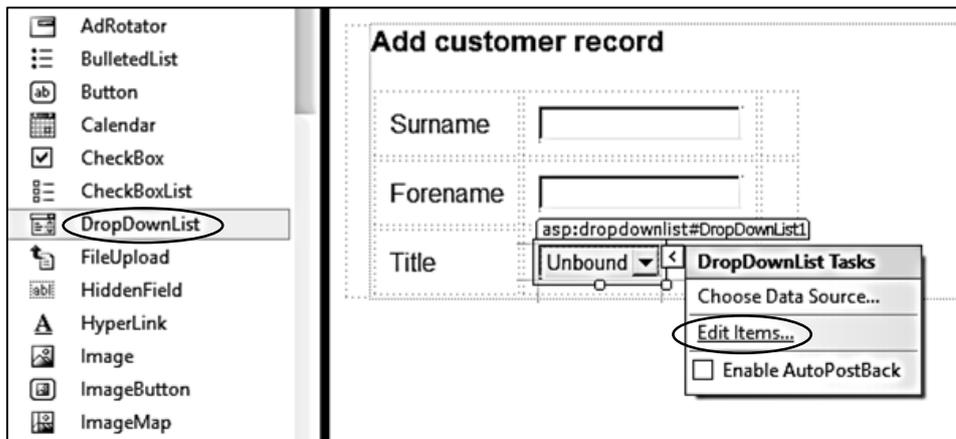
Build and run the web page. Check that clicking the button opens the blank *addRecord* page correctly.

Close the browser, return to Visual Studio and stop debugging.

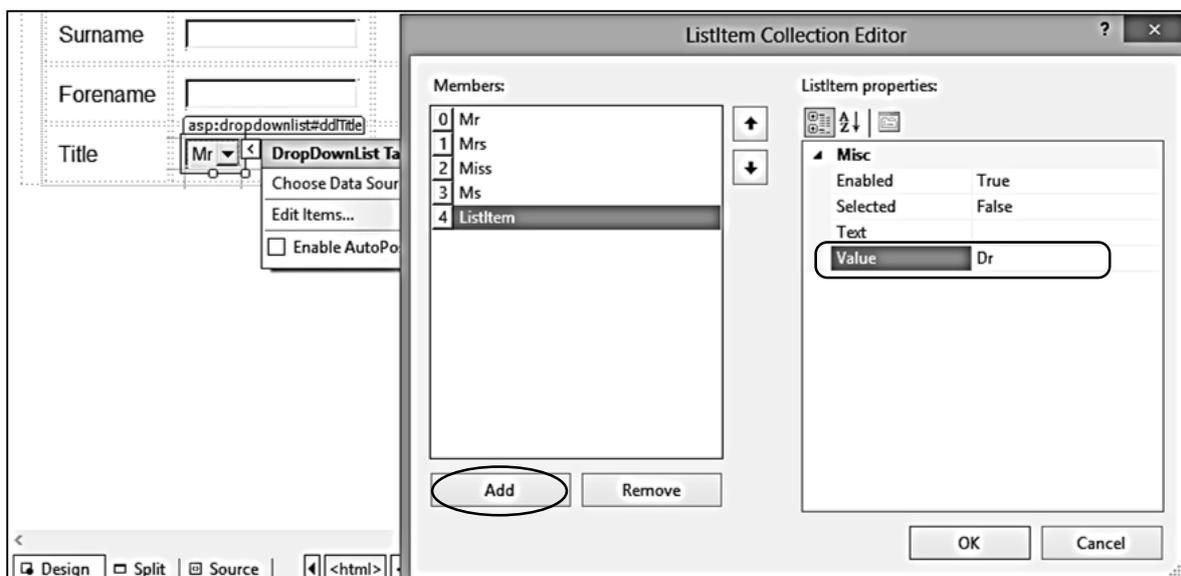
Select the *addRecord* HTML page. Add a title '*Customer records*' and specify the style sheet location. Insert a **<h1>** heading '*Add customer record*', and name the division as '*content*'.

```
<head runat="server">
    <title>Customer records</title>
    <link rel="stylesheet" type="text/css" href="StyleSheet1.css" />
</head>
<body>
    <form id="form1" runat="server" >
        <div id="content">
            <h1>Add customer record</h1>
        </div>
    </form>
</body>
</html>
```


Locate the **DropDownList** component in the **Toolbox**. Drag and drop this in the second column of the table beside the caption **'Title'**. Rename the component as **'ddlTitle'**.



Click the small arrow icon to the right of the drop down list to show further options. Select **Edit Items**. Add a series of title options such as: **Mr, Mrs, Miss, Ms, Dr**. To do this, click the **Add** button and enter the required text in the **Value** field of the right hand pane. When all the required titles have been entered, click the **OK** button.



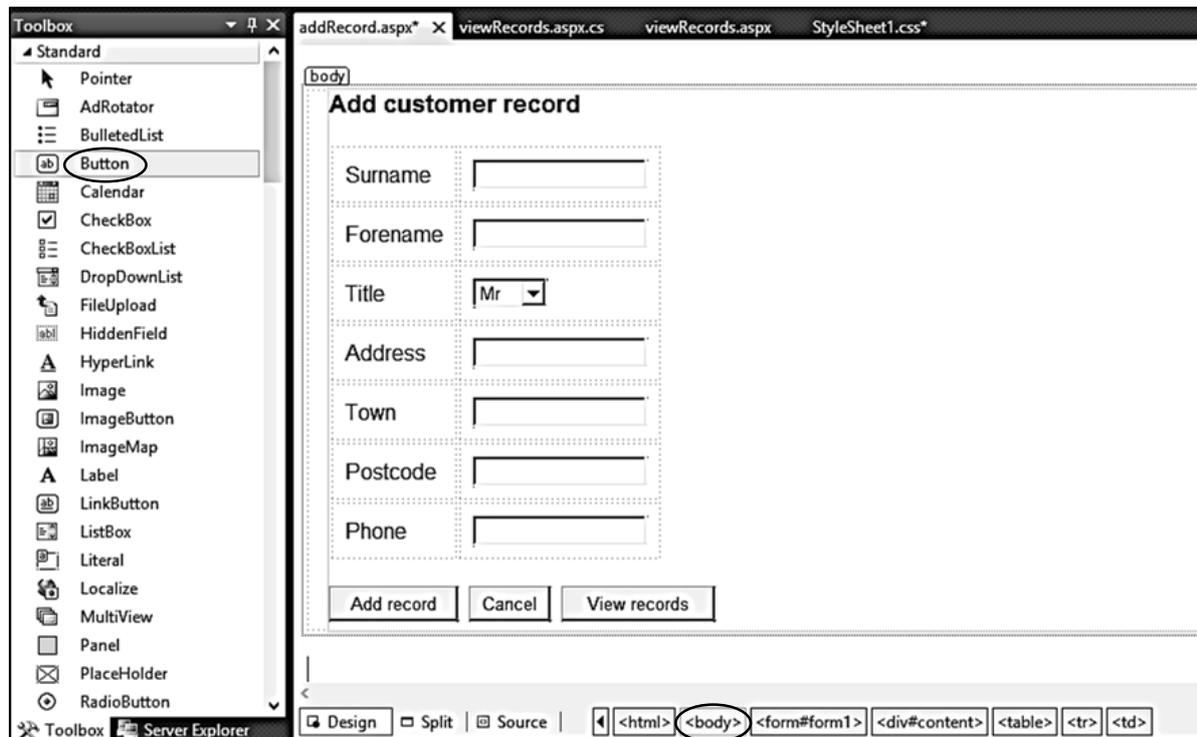
Return to the HTML page by clicking the **Source** button. Notice that the set of customer titles which you entered have been added as **Listitem** values. The empty **<td>** groups containing only ** ** space characters can be removed.

Add four further rows of code to the table to allow the user to input address, town, postcode and phone number for the customer.

```
<tr>
  <td>
    Title</td>
  <td>
    <asp:DropDownList ID="ddlTitle" runat="server">
      <asp:ListItem Value="Mr"></asp:ListItem>
      <asp:ListItem Value="Mrs"></asp:ListItem>
      <asp:ListItem Value="Miss"></asp:ListItem>
      <asp:ListItem Value="Ms"></asp:ListItem>
      <asp:ListItem Value="Dr"></asp:ListItem>
    </asp:DropDownList>
  </td>
</tr>
<tr>
  <td>
    Address</td>
  <td>
    <asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>
  </td>
</tr>
<tr>
  <td>
    Town</td>
  <td>
    <asp:TextBox ID="txtTown" runat="server"></asp:TextBox>
  </td>
</tr>
<tr>
  <td>
    Postcode</td>
  <td>
    <asp:TextBox ID="txtPostcode" runat="server"></asp:TextBox>
  </td>
</tr>
<tr>
  <td>
    Phone</td>
  <td>
    <asp:TextBox ID="txtPhone" runat="server"></asp:TextBox>
  </td>
</tr>
</table>
</div>
</form>
```

Return to the page layout view by using the **Design** button. Click the **<body>** tag at the bottom edge of the page to select the content area. Drag the bottom margin downwards to create space below the table.

Drag and drop a button to a position below the table, then type a **space** character. Repeat this twice more to create a row of three buttons.



Set the **Text** properties of the buttons to **'Add record'**, **'Cancel'**, and **'View records'**. Give the buttons the corresponding names **'btnAddRecord'**, **'btnCancel'**, and **'btnViewRecords'**.

Double click the **View records** button and add a line of code to the event handler which will re-load the page displaying the table of records.

```
protected void btnViewRecords_Click(object sender, EventArgs e)
{
    Response.Redirect("viewRecords.aspx");
}
```

Return to the **Design** screen and double click the **Cancel** button to create a **button_click()** method. Add a call to a new method, **clearText Boxes()**.

```
protected void btnCancel_Click(object sender, EventArgs e)
{
    clearTextBoxes();
}
```

Insert the `clearTextBoxes()` method immediately below the `button_click()` method.

```
protected void btnCancel_Click(object sender, EventArgs e)
{
    clearTextBoxes();
}

private void clearTextBoxes()
{
    txtSurname.Text = "";
    txtForename.Text = "";
    txtAddress.Text = "";
    txtTown.Text = "";
    txtPostcode.Text = "";
    txtPhone.Text = "";
}
```

Build and run the web site. Check that it is possible to move between the two pages by clicking the **View records/Add record** buttons. Check also that any entries typed into the text boxes on the Add record form can be cleared by clicking the **Cancel** button.

It only remains to produce the code necessary to save a record into the database.

Go to the C# code for the **AddRecord** page. Add the `'using Data'` and `'using SqlClient'` directives, and give the database location.

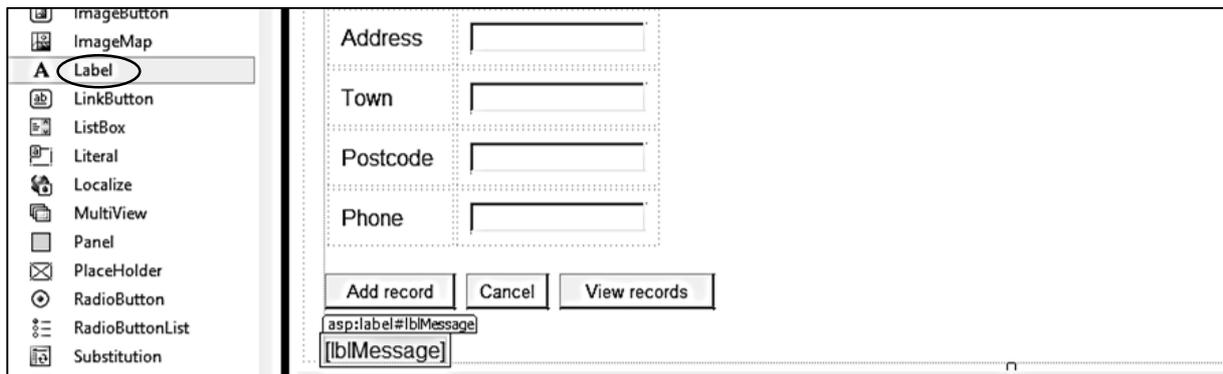
```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using System.Data.SqlClient;
using System.Data;

namespace Customer_records
{
    public partial class addRecord : System.Web.UI.Page
    {
        string databaseLocation="C:\\WEB APPLICATIONS\\customerRecords.mdf;";

        protected void Page_Load(object sender, EventArgs e)
        {
        }
    }
}
```

Return to the AddRecord layout page and add a **Label** below the row of buttons. We will use this to give a confirmation message that the record has been saved, or a warning if a file error has occurred.



Set the name of the label by changing the **ID** property to **'lblMessage'**, and leave its **Text** property blank. Double click the **Add record** button to create a **button_click()** method. Add code to this as shown. Note: the line beginning:

```
SqlConnection cnTB = new SqlConnection(.. ..);
```

should be entered as a single line of code without line breaks.

```
protected void btnAddRecord_Click(object sender, EventArgs e)
{
    SqlConnection cnTB = new SqlConnection(@"Data Source=. \SQLEXPRESS;
    AttachDbFilename=" + databaseLocation + "Integrated Security=True;
    Connect Timeout=30; User Instance=True");
    try
    {
        cnTB.Open();
        SqlCommand cmCustomer = new SqlCommand();
        cmCustomer.Connection = cnTB;
        cmCustomer.CommandType = CommandType.Text;
        cmCustomer.CommandText = "INSERT INTO customers(Surname, Forename, "
            + "Title, Address, Town, Postcode, Phone) VALUES ('"
            + txtSurname.Text + "','" + txtForename.Text + "','"
            + ddlTitle.Text + "','" + txtAddress.Text + "','"
            + txtTown.Text + "','" + txtPostcode.Text + "','"
            + txtPhone.Text + "')";
        cmCustomer.ExecuteNonQuery();
        cnTB.Close();
        clearTextBoxes();
        lblMessage.Text = "Record saved";
    }
    catch
    {
        lblMessage.Text = "File error";
    }
}
```

Notice that we use a **try..catch** block to prevent a program crash in the event of a file error.

The computer begins by making a connection to the database. The SQL command '**INSERT INTO customers**' is used to upload the new record. This command shows the fields which are to be saved into the record and the corresponding sources of data - these are the text boxes and drop down list on the web page. The **CustomerID** field is not included in the command, as this is an auto-number which is generated automatically by the computer itself.

Uploading is carried out by the **ExecuteNonQuery()** method. 'Non query' refers to the fact that no data needs to be returned from the database.

If the record upload is successful, the text boxes are cleared and a '**Record saved**' message is given.

Build and run the web page. Add a customer record and click the **Add record** button. Check that a '**Record saved**' message is received.

Add customer record

Surname

Forename

Title ▼

Address

Town

Postcode

Phone

Click the **View records** button. Check that the new record has been added to the table correctly.

Customer records

CustomerID	Surname	Forename	Title	Address	Town	Postcode	Phone
1	Roberts	David	Mr	12 Moelwyn Terrace	Ffestiniog	LL38 7RT	01766 879332
2	Johnson	Steven	Mr	Ty Gwyn	Bala	LL29 6WR	01356 792144
3	Green	Janet	Ms	7 High Street	Tywyn	LL32 9QB	01452 567199
4	Davidson	Sally	Ms	7 Oak Tree Crescent	Chester	CH2 6TK	01266 823451
5	Young	Ann	Ms	5 Hillcrest Avenue	Wrexham	CH18 6KL	01721 872445