6 Mountain Bike Club

In the previous chapter we created a web site with images programmed into HTML page code using commands such as:

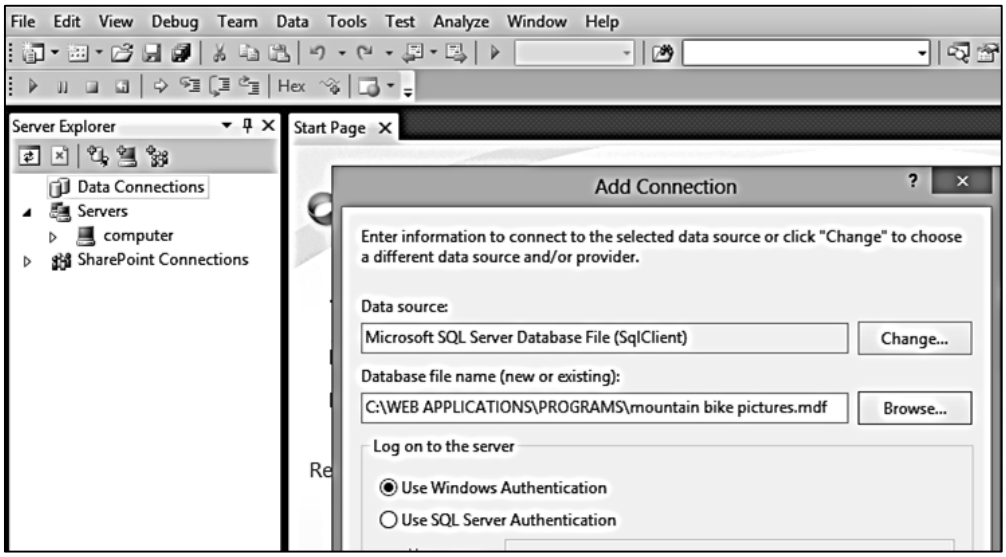
```
<img src="Images/train.jpg" />
```

In this section we will take a different approach by loading images interactively via a web page, storing them in a database table, and then loading and displaying them on another web page. This approach, forming the basis of a content management system, has advantages in allowing the content of a web site to be easily updated without the need to alter the HTML code.

We will create web pages which could be used by a Mountain Biking Club to upload and display photographs of club activities.

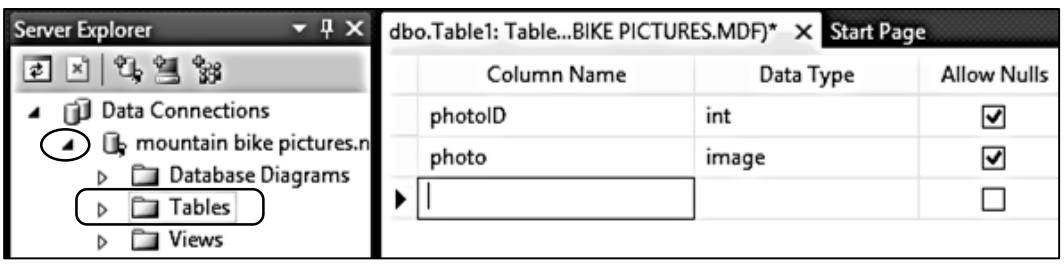
We will begin by creating a database. Open Visual Studio and select ***View*** / ***Server Explorer***.

Right click the ***Data Connections*** icon and choose ***Add connection***. Make sure that ***Microsoft SQL Server Database File (SqlClient)*** is selected, choose a location on your computer for the new database, then give the name '***mountain bike pictures.mdf***'.

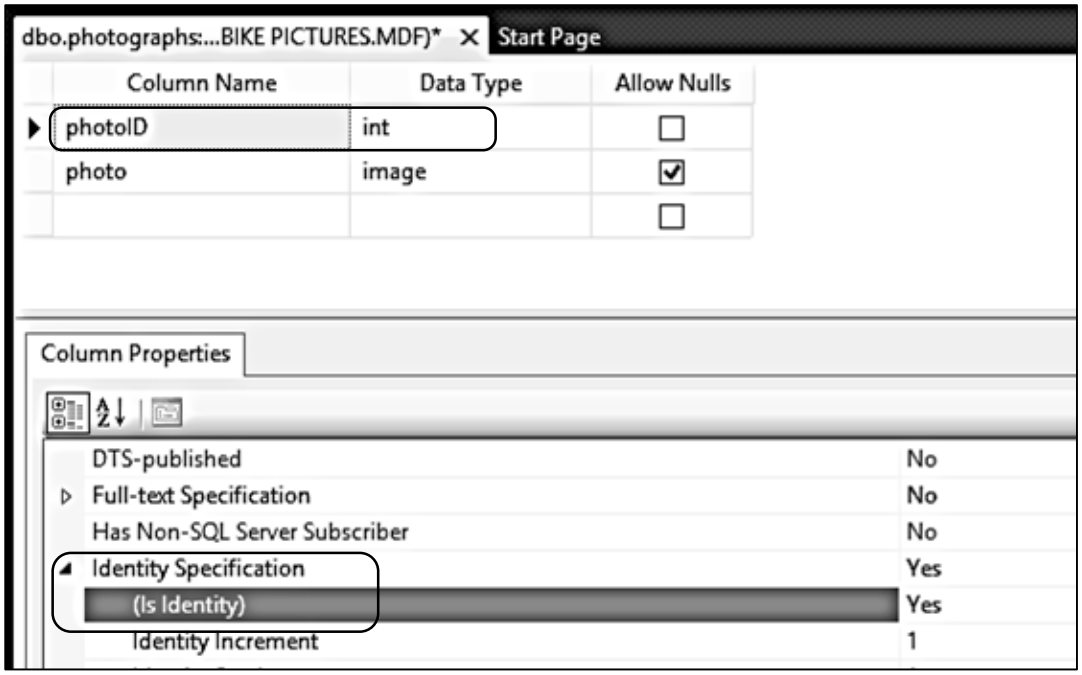


Open the database by clicking the small triangle icon. Right click the ***Tables*** icon and select '***Add New Table***'.

Insert two fields: **photoID** as an ***integer*** number, and **photo** as an ***image*** data type.



We will set ***photoID*** to be an auto number, which can be generated by the computer when a photograph is added to the database. Select the ***photoID*** row in the list of fields, then locate the ***Identity Specification*** section in the ***properties*** window below. Click the small arrow icon to display further options, and set '***Is Identity***' to '**Yes**'.

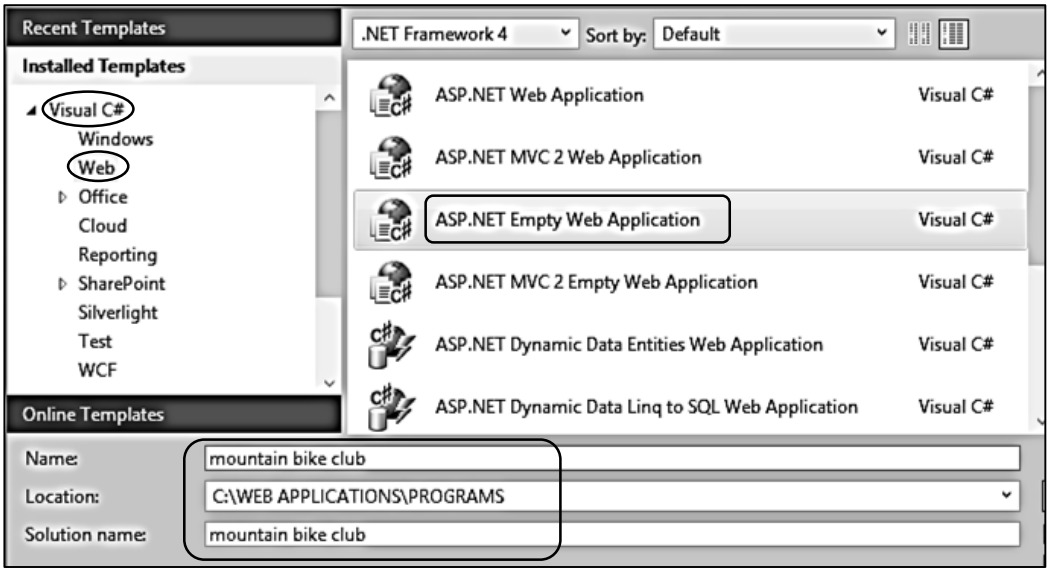


Close the database table by clicking the cross symbol on the tab. Name the table '**photographs**'.

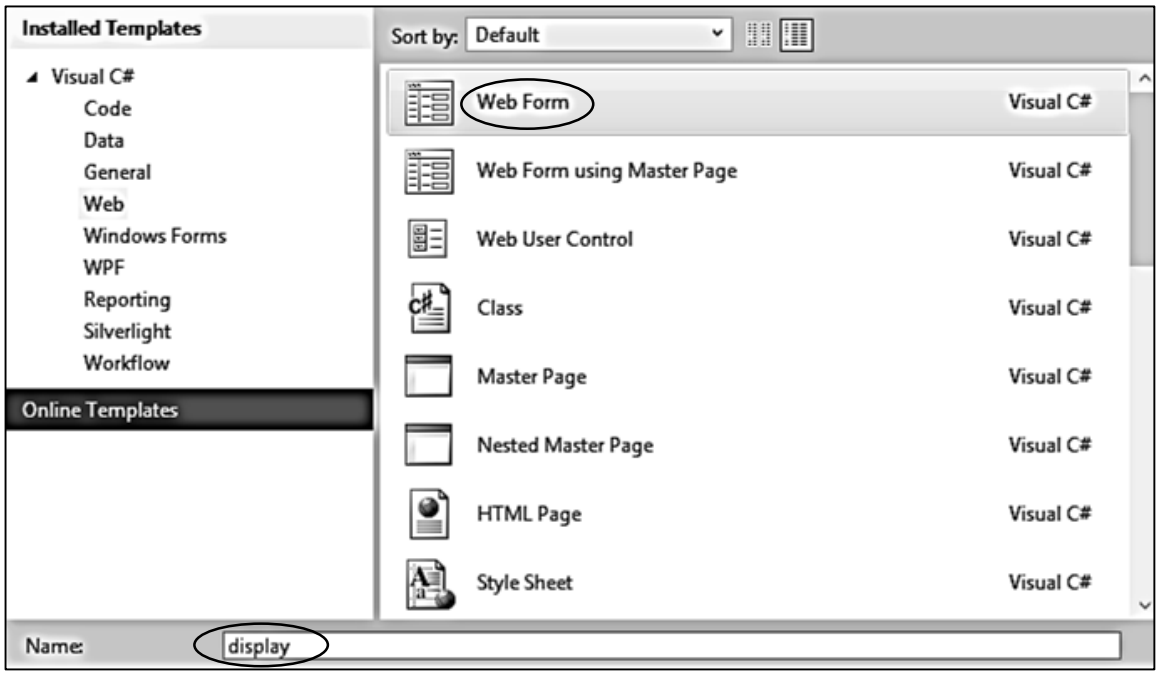
The next stage is to create a web site. This will have two pages: the first to display photographs uploaded by members of the club, and a second page to allow further photographs to be added to the database.

Go to the ***View*** option on the main menu bar and select ***Start page***.

Click ***New Project***. Select ***Visual C# / Web*** as the project type, then choose ***ASP.NET Empty Web Application***. Give the name '**mountain bike club**' and select a location where the project can be stored on your computer.



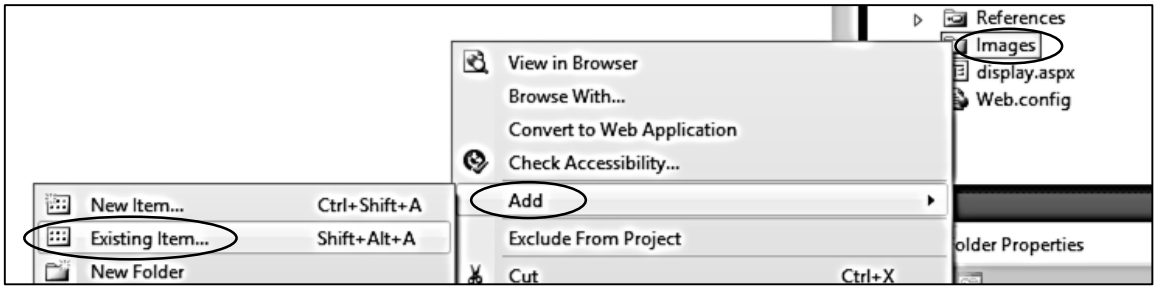
Go to the ***Solution Explorer*** window and right click the ***mountain bike club*** project icon. Select ***Add/ New Item***. Choose ***Web Form***, and give the name '**display**'. Click '**Add**' to create the page.



Before working on the page, we should prepare some resources for the web site. Use a graphics package such as ***Photoshop*** or ***Paintshop*** to produce a page header with suitable wording. This should have a width of around 1000 pixels. Save the image in .PNG or .JPG format using the name '***header***'.

Return to Visual Studio ***Solution Explorer*** window, and right click the ***mountain bike club*** project icon. Select ***Add / New Folder***, then give the name '**Images**' to the folder which is created.

Right click the ***Images*** folder and select ***Add / Existing Item***.



Use the file selection dialogue to find and load the ***header*** image.

Double click the page name '***display.aspx'*** in the ***Solution Explorer*** window to open the HTML code window.
Add the title '**Mountain Bike Club**' in the ***<head>*** section.
Go to the ***<body>*** section and add *** *** and ***<img>*** tags. Give the image file name and location, and set the width to 1000 pixels.

```
<head runat="server">
    <title>Mountain Bike Club</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <center>
       <img src="Images/header.png" width="1000" />
    </center>
    </div>
    </form>
</body>
</html>
```

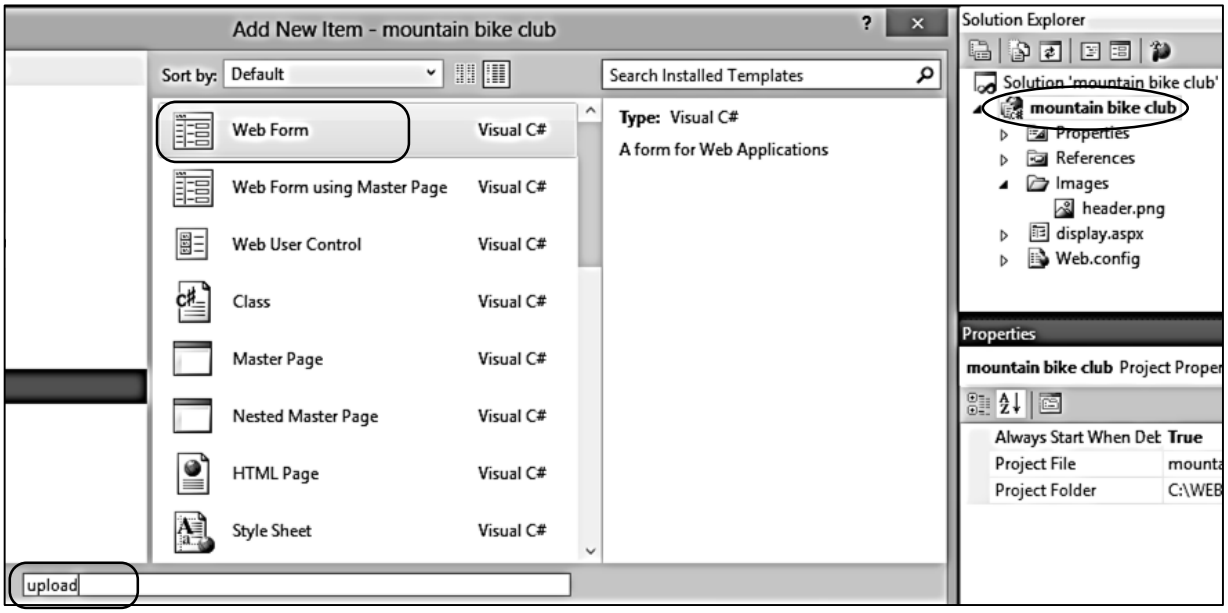
Use the ***Design*** button at the bottom left of the window to go to the page preview. The header image should be displayed.

Open the ***Toolbox*** from the ***View*** drop down list of the main menu bar. Click on the right edge of the image and press the ***Enter*** key to move the cursor downwards. Select the ***Button*** object and drag this to the design window. Use the ***Properties*** window to set the button text to '**Upload photo**', and change the button **ID** to '***btnUpload***'.

Return to the HTML code page by clicking the ***Source*** button. Check that code has been added correctly for the button.

```
<div>
  <center>
      <img src="Images/header.png" width="1000" />
    <br />
    <br />
    <asp:Button ID="btnUpload" runat="server" Text="Upload photo" />
  </center>
```

We will now create the upload page. Right click the ***mountain bike club*** project icon and select Add / New Item. Choose ***Web Form*** and give the name '**upload**'.



Open the HTML page for ***upload.aspx***. Add a title in the ***<head>*** section, and add code to the ***<body>*** section to produce a centred heading displaying the words '**Upload Photograph**' in Arial font.

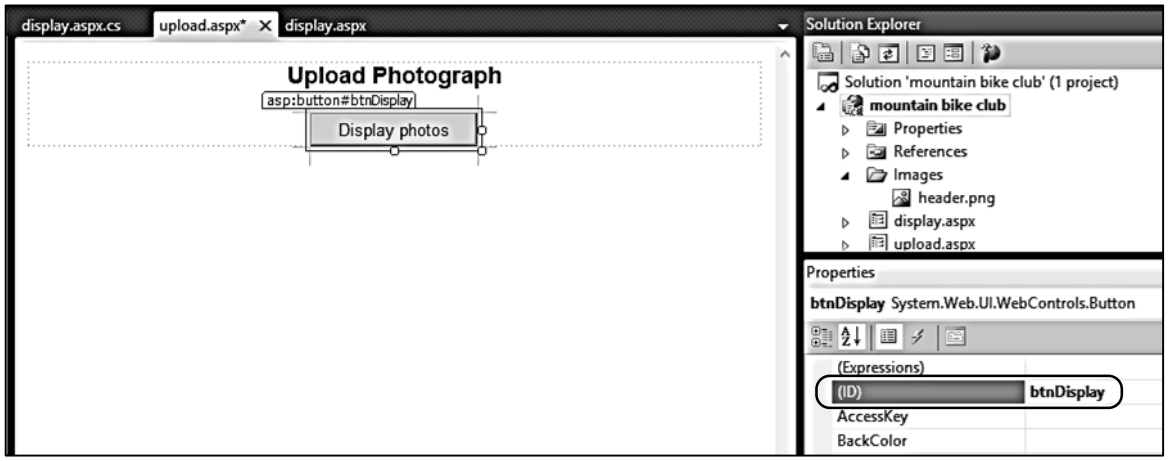
```
<head runat="server">
    <title>Mountain Bike Club</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
      <center>
         <h3 style="font-family: Arial, Helvetica, sans-serif">
             Upload Photograph
         </h3>
      </center>
    </div>
    </form>
</body>
```

We can now link this page to the button created earlier. Open the design preview for the ***display.aspx*** page and double click the '***Upload photo***' button. A button_click method will be produced. Add a line of code to open the upload page.

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

Using similar tecniques, we will add a button to the ***upload.aspx*** page which will link back to the display page.

Go to the ***upload.aspx*** page display. Press the ***Enter*** key after the heading to move the cursor downwards, then add a ***button*** component from the ***Toolbox***. Change the button ***ID*** to '**btnDisplay**', and set the Text property to 'Display photos'.



Double click the button and add code to the ***button_click*** method:

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

Build and run the web site. Check that it is possible to switch between the two pages correctly by means of the buttons. Close the web browser, stop debugging, and return to the ***upload.aspx*** page. Click the ***Source*** button and add HTML code which will allow an image file to be selected.

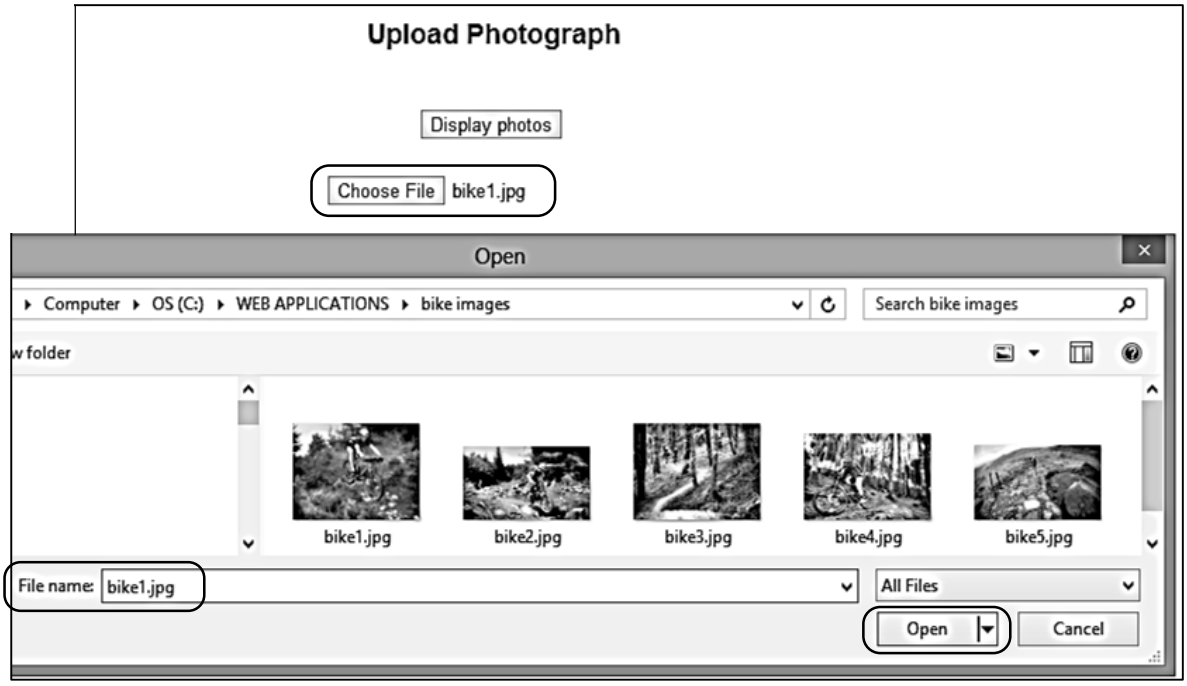
```
<center>
    <h3 style="font-family: Arial, Helvetica, sans-serif">
        Upload Photograph
    </h3>
    <br />
    <asp:Button ID="btnDisplay" runat="server" onclick="btnDisplay_Click"
             Text="Display photos" />

    <br />
    <br />
    <asp:FileUpload ID="FileUpload1" runat="server" />

</center>
```

Use the Internet to find some suitable mountain biking photographs, then store these in a folder on your computer.

Build and run the ***Mountain Bike Club*** web site. Go to the ***Upload Photograph*** page. A '***Choose File***' button should be present. Click this button and search for the photographs you have saved. Select one of the photographs, and check that the file name appears next to the ***Choose File*** button.



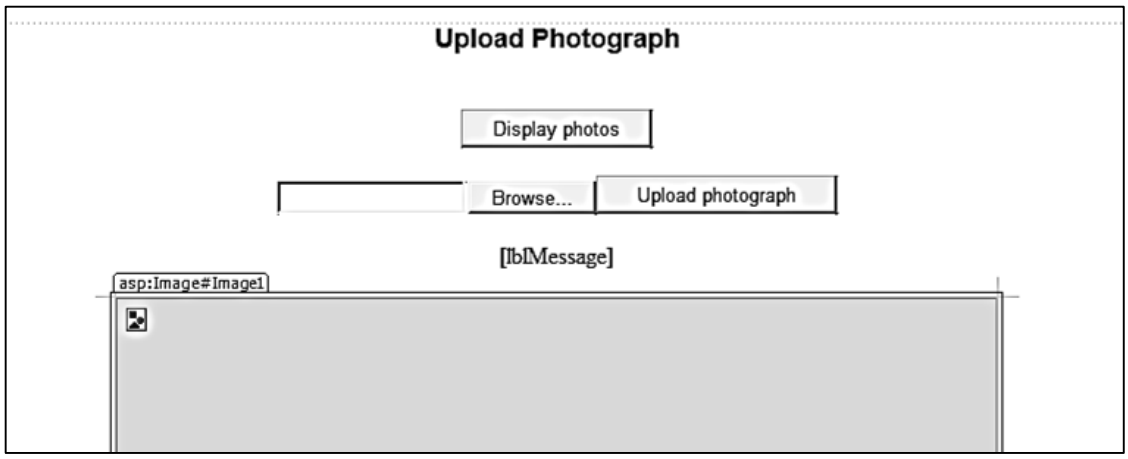
Close the web browser, return to the ***upload.aspx*** page and stop debugging. We will now add some further screen components: a ***button*** to upload the chosen image file to the database, a blank ***label*** which can be used to display error messages if necessary, and an ***image box*** to display the uploaded photograph.

```
<center>
    <h3 style="font-family: Arial, Helvetica, sans-serif">
        Upload Photograph
    </h3>
    <br />
    <asp:Button ID="btnDisplay" runat="server" onclick="btnDisplay_Click"
             Text="Display photos" />
    <br />
    <br />
    <asp:FileUpload ID="FileUpload1" runat="server" />

    <asp:Button ID="btnPhotoUpload" runat="server" Text="Upload photograph" />
    <br />
    <br />
    <asp:Label ID="lblMessage" runat="server"></asp:Label>
    <br />
    <br />
    <asp:Image ID="Image1" runat="server" Width="600px" />

 </center>
```

Change to the ***Design*** preview screen and check that all the components are displayed.



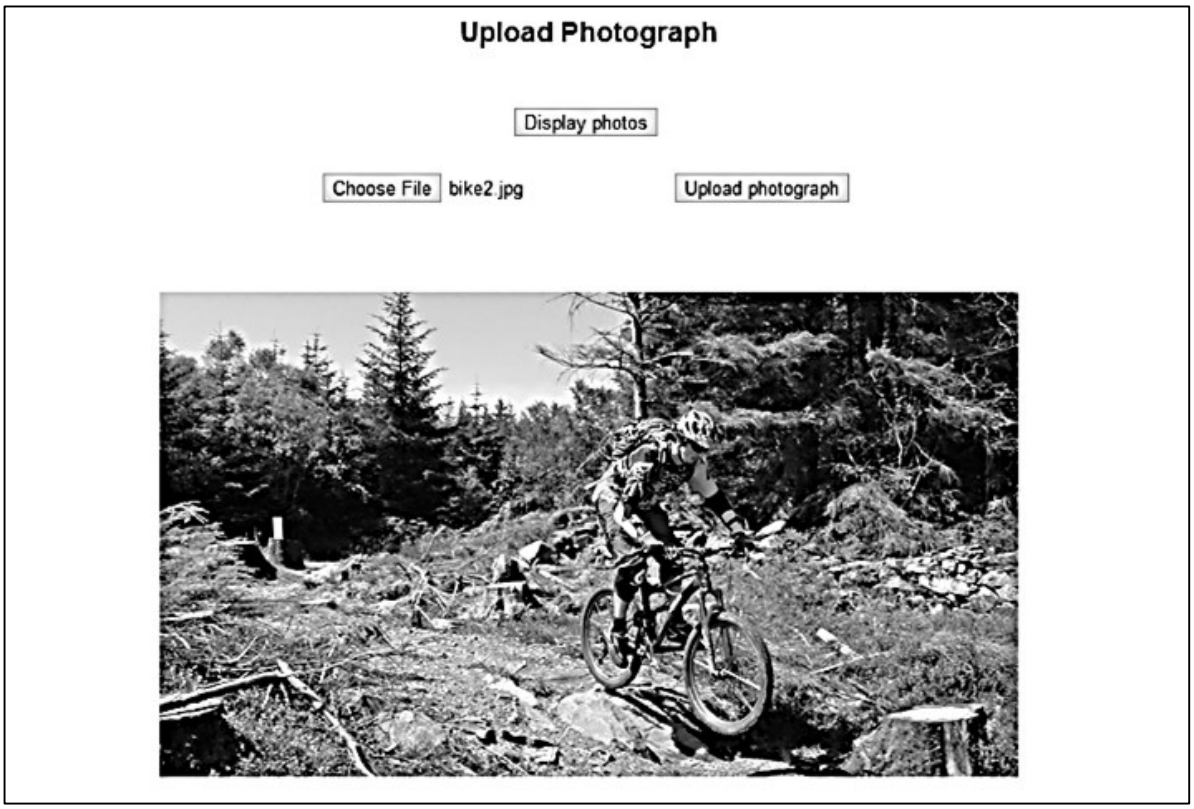
Double click the ***Upload photograph*** button to create a ***button_click( )*** method.

We will begin by setting up error trapping procedures. The ***try..catch*** block will give a message if a file error occurs. The ***if..else*** block will warn the user if the ***upload*** button is clicked before a picture file has been selected.

The processing code will store the selected picture file in the ***Images*** folder of the project, and then display the picture in the ***image box*** on the web page.

```
protected void btnPhotoUpload_Click(object sender, EventArgs e)
{
    lblMessage.Text = " ";
    try
    {
        if (FileUpload1.HasFile)
        {
            FileUpload1.SaveAs(Server.MapPath("Images").ToString() + @"\" +
                FileUpload1.FileName);
            Image1.ImageUrl = @"Images\" + FileUpload1.FileName;
        }
        else
        {
            lblMessage.Text = "Please Select Image File";
        }
    }
    catch
    {
        lblMessage.Text = "File error";
    }
}
```

Build and run the web page. Check that the error message appears if the ***upload*** button is clicked without first selecting a file. Select a picture file and check that this is displayed when the ***upload*** button is clicked. (Please note: the Firefox browser may not display the photograph unless the default security settings in Firefox are adjusted for a local server.)



Close the web browser and return to the C# code by selecting the tab for the page **upload.aspx.cs**. Add '***using Data***' and '***using SqlClient***' directives, and the location of the database.

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

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

namespace mountain_bike_club
{
    public partial class upload : System.Web.UI.Page
    {
        string databaseLocation =
               "C:\\WEB APPLICATIONS\\PROGRAMS\\mountain bike pictures.mdf;";

        protected void Page_Load(object sender, EventArgs e)
        {

        }
```

Note: the ***databaseLocation*** string should be set to the address of your database if this is different.

The final step is to add code to transfer the image file to the database. A message is displayed if the upload is successful.

Please note that the command:

```
SqlConnection cnTB = new SqlConnection( .. .. User Instance=True");
```

should be entered as a single line of code with no line breaks.

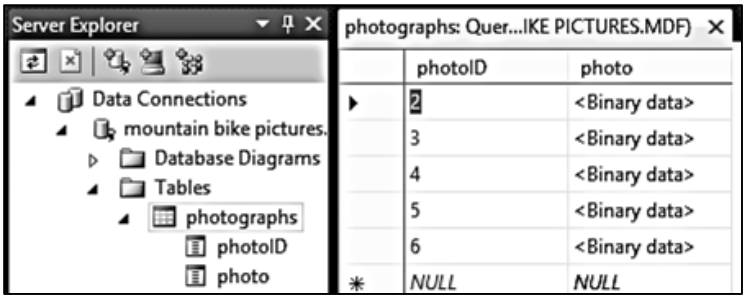
```
protected void btnPhotoUpload_Click(object sender, EventArgs e)
{
    lblMessage.Text = " ";
    try
    {
        if (FileUpload1.HasFile)
        {
            FileUpload1.SaveAs(Server.MapPath("Images").ToString() + @"\" +
                FileUpload1.FileName);
            Image1.ImageUrl = @"Images\" + FileUpload1.FileName;

            byte[] picbyte = FileUpload1.FileBytes;
            SqlConnection cnTB = new SqlConnection(@"Data Source=.\SQLEXPRESS;
              AttachDbFilename=" + databaseLocation + "Integrated Security=True;
              Connect Timeout=30; User Instance=True");
            cnTB.Open();
            string query = "INSERT INTO photographs(photo) VALUES(@pic)";
            SqlParameter picparameter = new SqlParameter();
            picparameter.SqlDbType = SqlDbType.Image;
            picparameter.ParameterName = "pic";
            picparameter.Value = picbyte;
            SqlCommand cmd = new SqlCommand(query, cnTB);
            cmd.Parameters.Add(picparameter);
            int result = cmd.ExecuteNonQuery();
            cnTB.Close();
            if (result > 0)
            {
                lblMessage.Text = "Photo Uploaded Successfully";
            }

        }
        else
        {
            lblMessage.Text = "Please Select Image File";
        }
    }
    catch
    {
        lblMessage.Text = "File error";
    }
}
```

Build and run the web page. Use the ***file selection*** and ***upload*** buttons to transfer your collection of photographs to the database.

Close the web page and stop debugging. Go to the *Server Explorer* window and check that entries have appeared in the table by right clicking on '***photographs***' and selecting '**Show Table Data**'. Values for ***photoID*** will have been inserted automatically by the computer. The images are not actually displayed, but appear with the label ***<Binary data>***



We can now work on the web page to display the photographs.

Open the HTML code page ***display.aspx.*** Add a ***label*** component, which will be used as a place holder for the photograph images.

```
<center>
    <img src="Images/header.png" width="1000" />
    <br />
    <br />
    <asp:Button ID="btnUpload" runat="server" Text="Upload photo"
        onclick="btnUpload_Click" />

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

</center>
```

Select the C# code page ***display.aspx.cs*** from the ***Solution Explorer.*** You may need to click the small triangle icon alongside ***display.aspx*** to reveal the file name. Add '***using Data***' and '***using SqlClient***' directives, and give the database location.

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

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

namespace mountain_bike_club
{
    public partial class display : System.Web.UI.Page
    {
        string databaseLocation =
            "C:\\WEB APPLICATIONS\\PROGRAMS\\mountain bike pictures.mdf;";

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

A ***DataSet*** needs to be created. We then add code to the ***Page_Load( )*** method to access the database and load the records into this DataSet. Note that the command

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

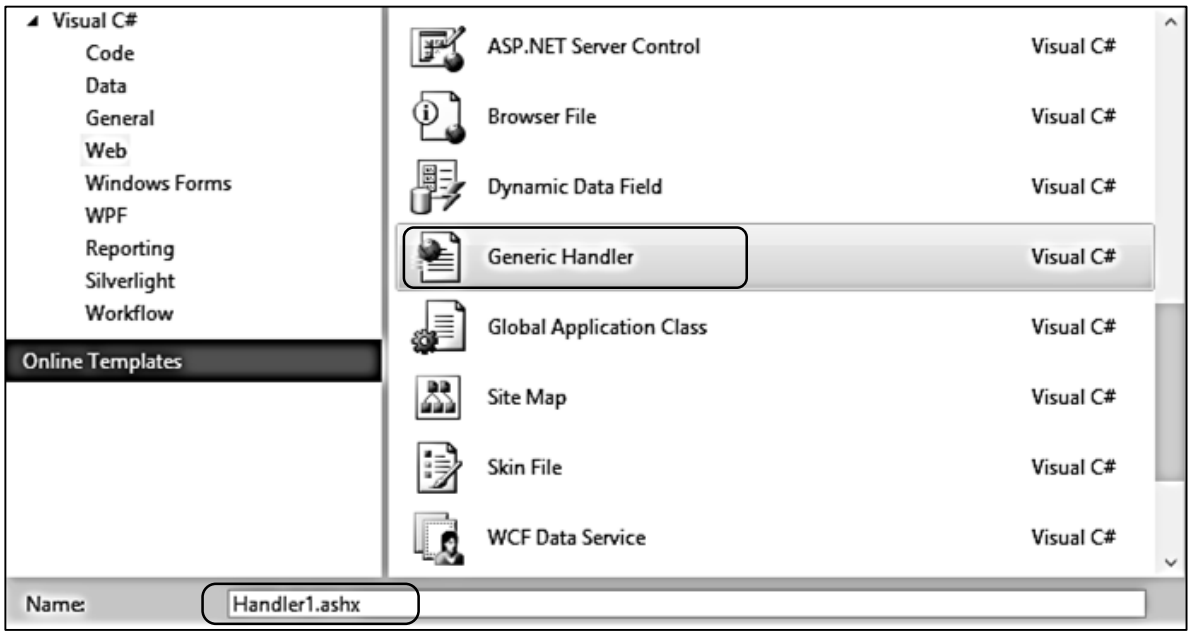
should be entered as a single line with no line breaks.

```
string databaseLocation =
  "C:\\WEB APPLICATIONS\\PROGRAMS\\mountain bike pictures.mdf;";

DataSet dsPicture = new DataSet();

protected void Page_Load(object sender, EventArgs e)
{
   SqlConnection cnTB = new SqlConnection(@"Data Source=.\SQLEXPRESS;
      AttachDbFilename=" + databaseLocation + "Integrated Security=True;
      Connect Timeout=30; User Instance=True");
   cnTB.Open();
   SqlCommand cmPicture = new SqlCommand();
   cmPicture.Connection = cnTB;
   cmPicture.CommandType = CommandType.Text;
   cmPicture.CommandText = "SELECT * FROM photographs";
   SqlDataAdapter daPicture = new SqlDataAdapter(cmPicture);
   daPicture.Fill(dsPicture);
   cnTB.Close();
}
```

Unfortunately, working with images from a database is more complicated than accessing text and number data. It is necessary to create a graphics ***Handler*** file. To do this, right click the ***mountain bike club*** project icon in the ***Solution Explorer*** window and select ***Add*** / ***New Item***.
Choose the ***Generic Handler*** option, and accept the name '**Handler1**'.



Open the ***Handler1.ashx*** file which has been created.

It is necessary to give the location of the database on your computer, and also to replace the code in the ***ProcessRequest( )*** method. Note that the two commands:

```
conn = new System.Data.SqlClient.SqlConnection(..  ..)
sqlcmd = new System.Data.SqlClient.SqlCommand(..   ..)
```

need to be entered as single lines of code with no line breaks.

```
public class Handler1 : IHttpHandler

{
    string databaseLocation =
        "C:\\WEB APPLICATIONS\\PROGRAMS\\mountain bike pictures.mdf;";

    public void ProcessRequest(HttpContext context)
    {
        System.Data.SqlClient.SqlDataReader rdr = null;
        System.Data.SqlClient.SqlConnection conn = null;
        System.Data.SqlClient.SqlCommand sqlcmd = null;
        try
        {
            conn = new System.Data.SqlClient.SqlConnection(@"Data
                 Source =.\SQLEXPRESS; AttachDbFilename=" + databaseLocation +
                 "Integrated Security=True; Connect Timeout=30; User
                 Instance=True");
            sqlcmd = new System.Data.SqlClient.SqlCommand("SELECT photo
                 FROM photographs WHERE photoID=" +
                 context.Request.QueryString["imgid"], conn);
            conn.Open();
            rdr = sqlcmd.ExecuteReader();
            while (rdr.Read())
            {
                context.Response.ContentType = "image/jpg";
                context.Response.BinaryWrite((byte[])rdr["photo"]);
            }
            if (rdr != null)
                rdr.Close();
        }
        finally
        {
            if (conn != null)
                conn.Close();
        }
    }

    public bool IsReusable
    {
        get
        {
            return false;
        }
    }
}
```

We can now finish the C# code on the ***display.aspx.cs*** page to display the pictures. Return to the Page_Load( ) method and add the lines shown below.

The code first of all counts the number of photo records which have been loaded, then carries out a loop to process each photograph.

For each record, the ***photoID*** is obtained and this is passed to the ***Handler1*** file. The handler then accesses the correct record in the database using this ***photoID***, accesses the binary image data and converts it into **. JPG** format, then displays it using the HTML ***<img>*** tag.

```
protected void Page_Load(object sender, EventArgs e)
{
    SqlConnection cnTB = new SqlConnection(@"Data Source=.\SQLEXPRESS;
        AttachDbFilename=" + databaseLocation + "Integrated Security=True;
        Connect Timeout=30; User Instance=True");
    cnTB.Open();
    SqlCommand cmPicture = new SqlCommand();
    cmPicture.Connection = cnTB;
    cmPicture.CommandType = CommandType.Text;
    cmPicture.CommandText = "SELECT * FROM photographs";
    SqlDataAdapter daPicture = new SqlDataAdapter(cmPicture);
    daPicture.Fill(dsPicture);
    cnTB.Close();

    int countRecords = dsPicture.Tables[0].Rows.Count;
    string s = "";
    string photoID;
    s += "<table border=0 cellspacing=20 cellpadding=10>";
    for (int i = 0; i < countRecords; i++)
    {
        s += "<tr>";
        DataRow drPicture = dsPicture.Tables[0].Rows[i];
        s += "<td>";
        photoID = Convert.ToString(drPicture[0]);
        s += "<img src='Handler1.ashx?imgid=" + photoID +
            "' width='600' border='1' >";
        s += "</td>";
        s += "</tr>";
    }
    s += "</table>";
    Label1.Text = s;

}
```

Build and run the program. The set of photographs should be displayed below the header on the ***display.aspx*** page.