

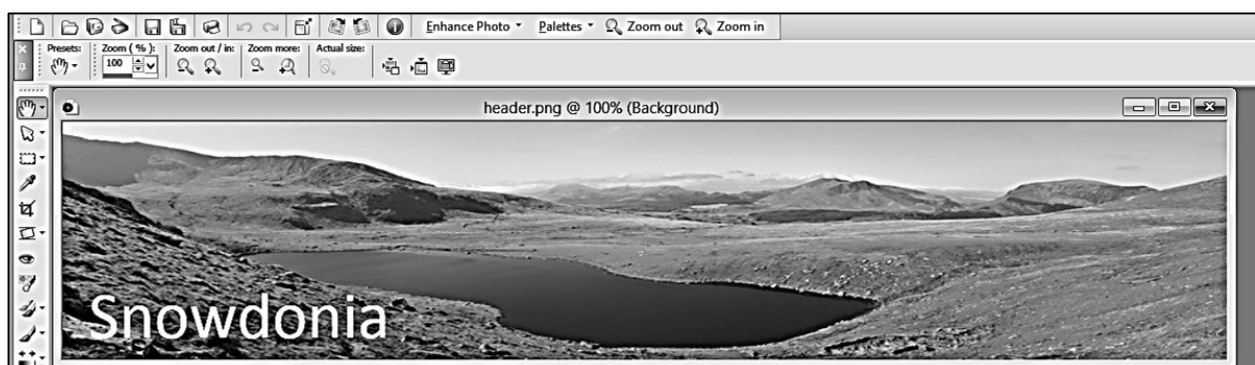
# 5 Snowdonia

In this and the following three chapters we will explore the use of particular programming techniques, before combining these methods to create two substantial web applications: for on-line shopping and an on-line booking system.

We begin by creating a menu system to link a series of pages in a simple website to display information. The text and images will be included directly in the HTML code for each page, but in later chapters you will see how these could be stored in a database linked to the website. The website will provide tourist information about Snowdonia. Each page will have the same header image and drop down menu options, to give the website a consistent appearance.

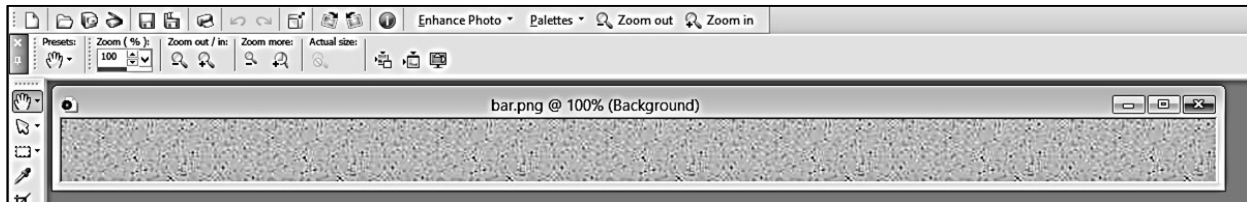


Before starting to code the website, we need to prepare some resources. Use the Internet to obtain a suitable image for the page header. As this will be wide and narrow, a panorama might be suitable. Load the image into a graphics utility package such as Paintshop or Photoshop, then add the title word 'Snowdonia' in a contrasting colour:

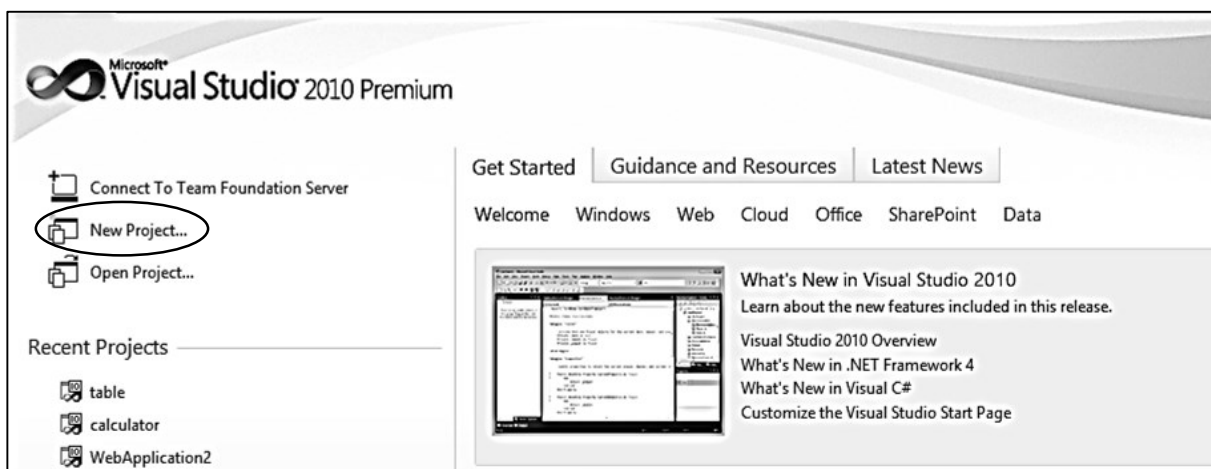


The image should be approximately 1100 pixels wide and 220 pixels high. Don't worry about the exact height, as this can be adjusted later when it is displayed on the page. Save the header in .JPG or .PNG format.

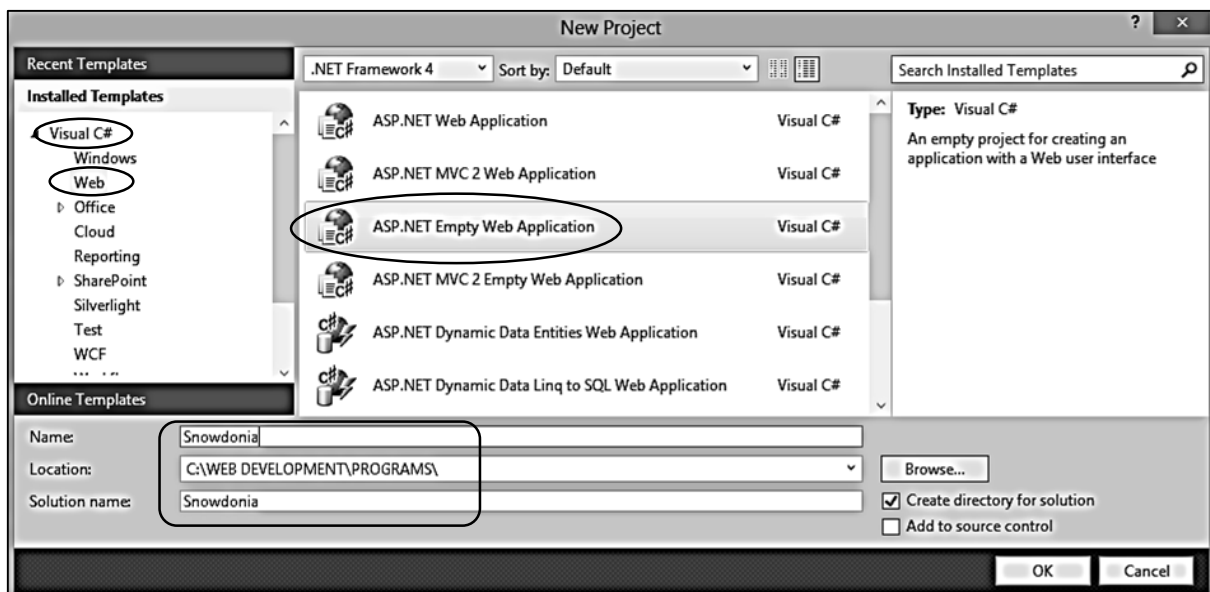
A panel will be needed for the menu bar. A texture similar to sandstone would be suitable. Create the panel using your graphics software, and store this as another .JPG or .PNG image:



Open Microsoft Visual Studio 2010 and select **New Project**.



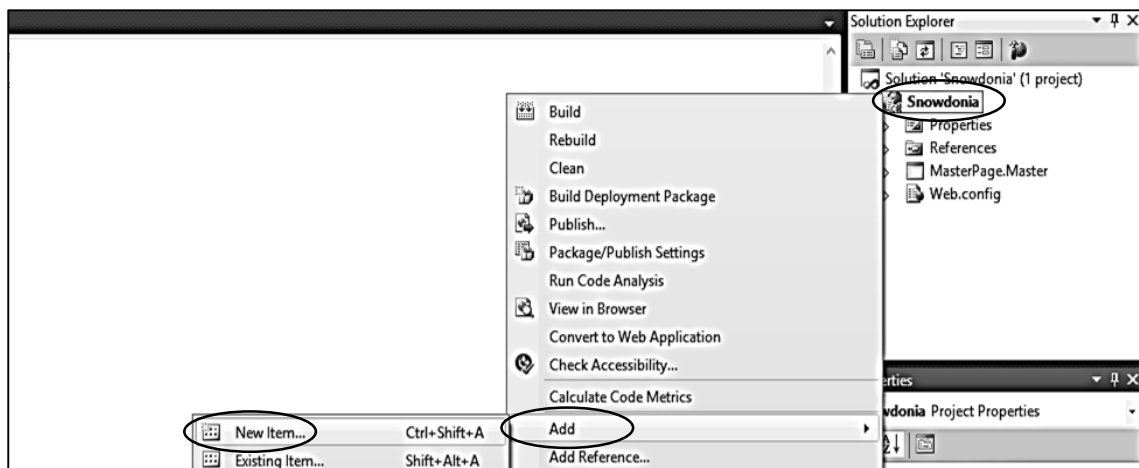
Select **Visual C#** as the programming language, and **Web** as the project type. Click the ASP.NET Empty Web Application option. Give the project the name '**Snowdonia**', and select a suitable storage location on your computer system.



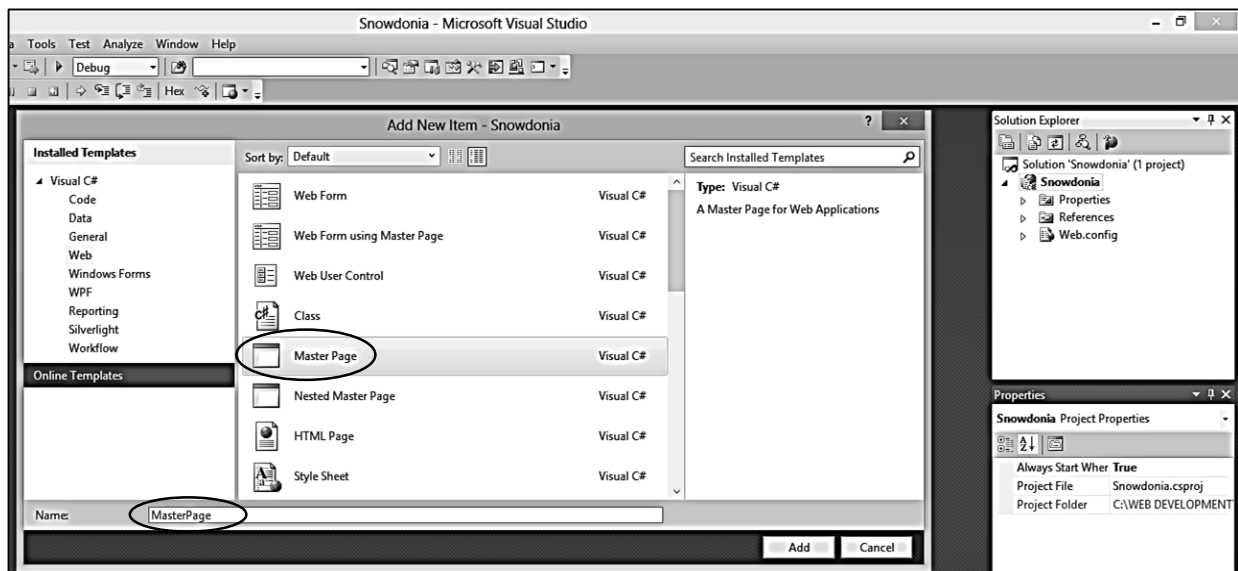
We will begin by creating a **Master Page**. This will contain the components which appear on every page of the web site: the header image, menu bar and a footer bar. The master page can then be used as a template for creating as many web pages as we wish. Each page can have different content added to build the site.



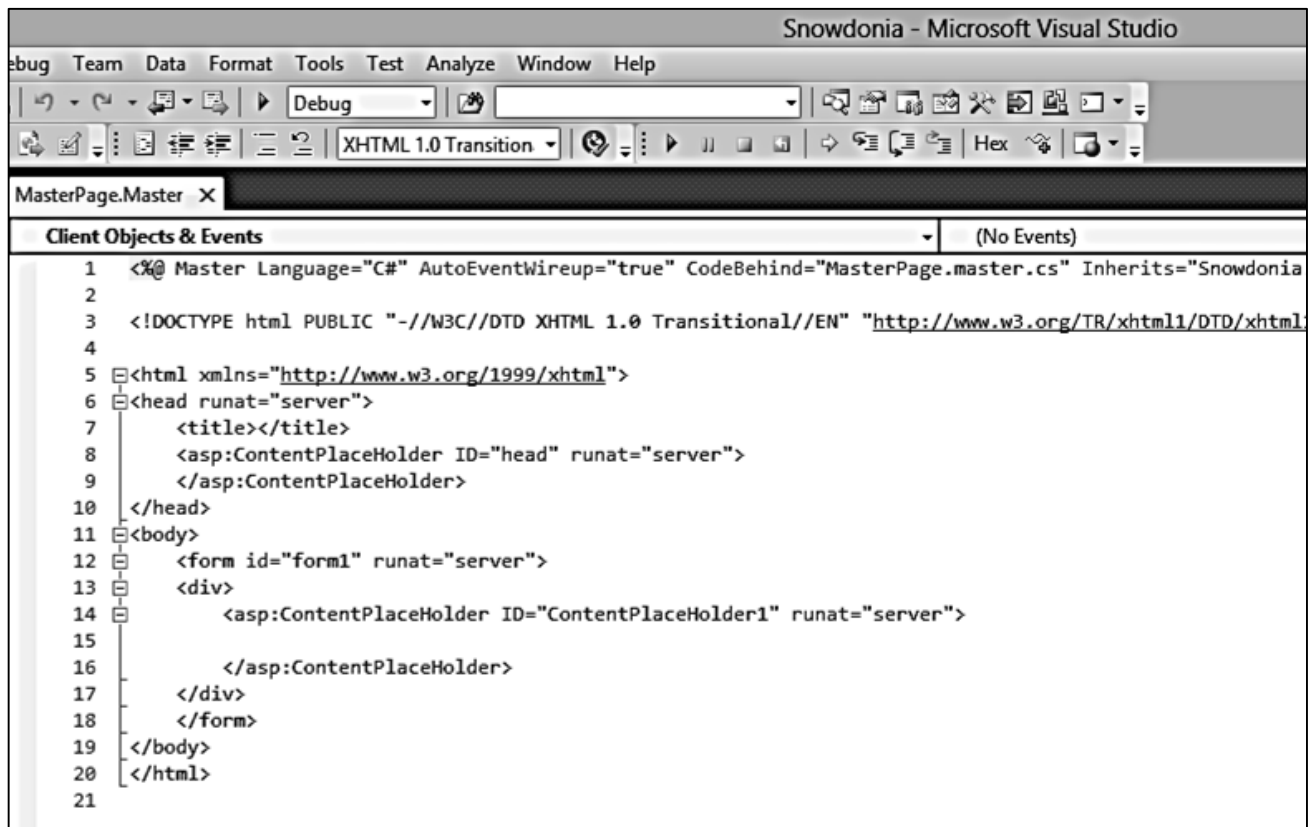
Go to the **Solution Explorer** window and right click on the project title **Snowdonia**. Select **Add / New item**:



Select **Master Page**, and give this the name **MasterPage**:



Visual Studio will create an HTML master page, ready for us to add our own code.



```

Snowdonia - Microsoft Visual Studio
Debug Team Data Format Tools Test Analyze Window Help
Debug
XHTML 1.0 Transitional
Hex
MasterPage.Master X
Client Objects & Events (No Events)
1 <%@ Master Language="C#" AutoEventWireup="true" CodeBehind="MasterPage.master.cs" Inherits="Snowdonia
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7     <title></title>
8     <asp:ContentPlaceHolder ID="head" runat="server">
9     </asp:ContentPlaceHolder>
10 </head>
11 <body>
12     <form id="form1" runat="server">
13     <div>
14         <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
15
16         </asp:ContentPlaceHolder>
17     </div>
18     </form>
19 </body>
20 </html>
21

```

This code looks complicated, but it has quite a simple underlying structure:

```

<html>
<head>
    <title></title>
</head>
<body>
    <div>
        <asp:ContentPlaceHolder>

        </asp:ContentPlaceHolder>
    </div>
</body>
</html>

```

The page is written in **Hypertext Markup Language** (HTML). The **<div>** marker, standing for **division**, allows a particular section of the web page to be identified. It can be used, for example, when particular formatting needs to be applied to just one section of a page.

The **<asp:ContentPlaceHolder>** tag is not part of the standard HTML language, but belongs to the .ASP programming system. It marks the area where individual page content will be added when pages are created from the Master Page template. We will leave this area blank at the present time.

Click the **Design** button at the bottom of the code window. The screen display will change to show a preview of the master page we are creating. At present there is just a shaded rectangle present, representing the place holder where individual page content can be added later.



Click the **Source** button to return to the program code. Add a title '**Snowdonia**' to the <head> section of the page:

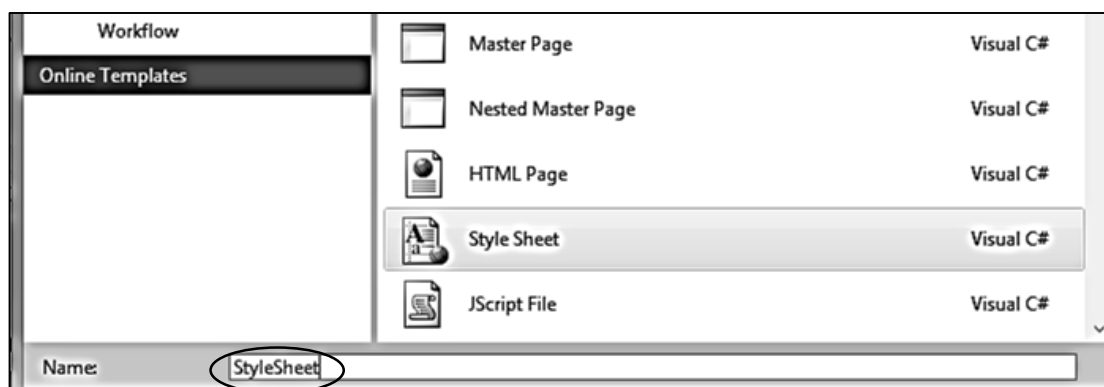
```
<head runat="server">
  <title>Snowdonia</title>
  <asp:ContentPlaceHolder ID="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
```

Insert an identification name "**wrapper**" for the division in the <body> section:

```
<body>
  <form id="form1" runat="server">
    <div id="wrapper">
```

We will carry out the formatting of the web site using a **style sheet**. The style sheet system allows great flexibility to select fonts, colours and other features to give pages an attractive and professional appearance.

Go to the **Solution Explorer** window at the top right of the screen and right click the project name **Snowdonia**. Select **Add / New item**. Click on the **Style Sheet** option, and give the name **StyleSheet**:



Open the style sheet. You will see that an entry for **body** is already present. Create another entry for the **wrapper** division and type the formatting commands as shown:

```

body
{
    background-color: #E9E9E9;
}

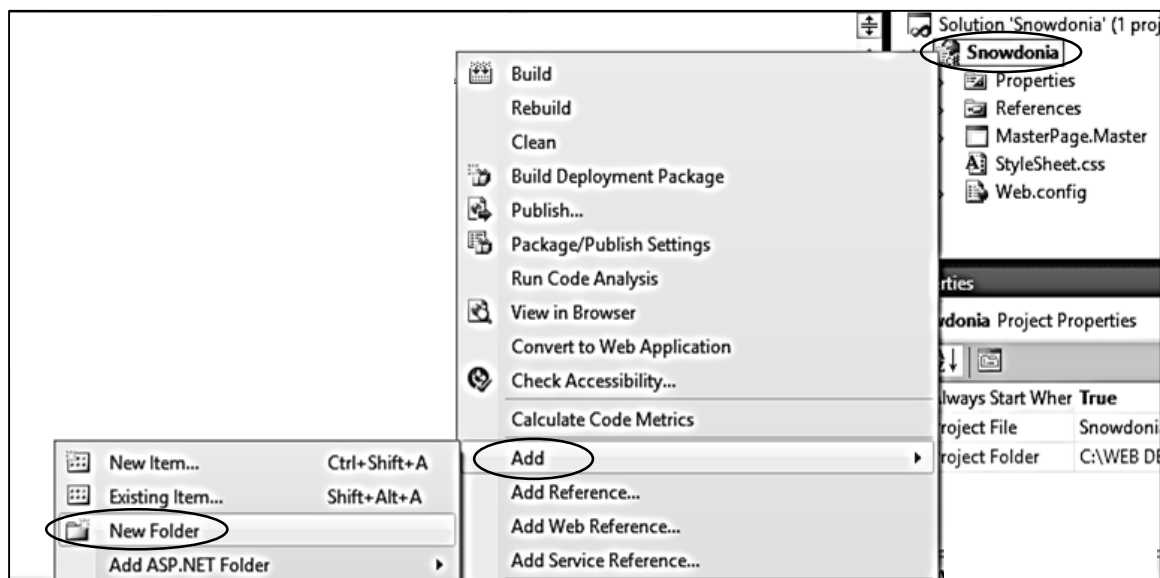
#wrapper
{
    width: 1080px;
    margin:0 auto;
    padding: 10px;
    border: 5px solid #DEDEDE;
    background-color: #FFFFFF;
}

```

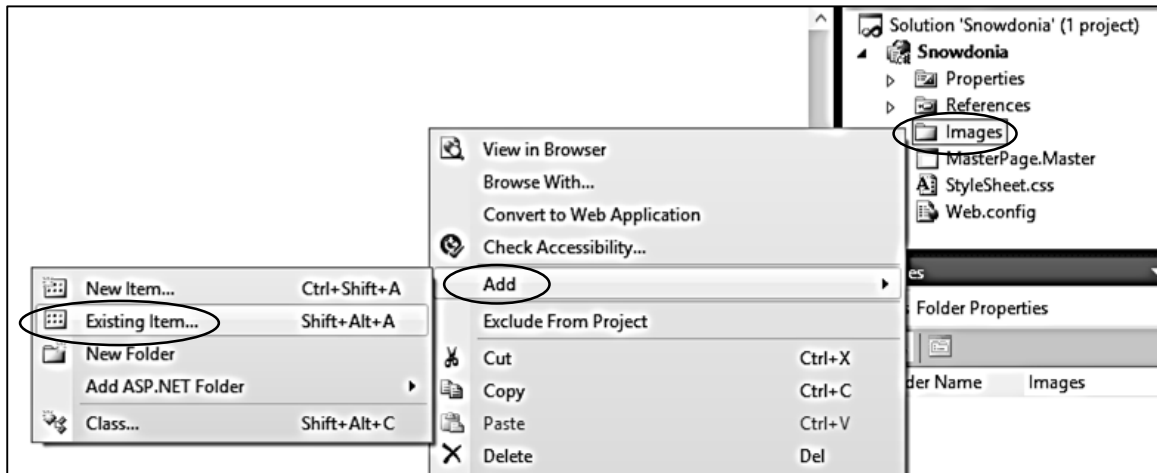
We have done a series of things here:

- The colour of the web page has been set to white using the **background-color** command inside the **wrapper** section. This is done with the colour code #FFFFFF. Charts showing the HTML code values for a wide range of colours can be found on the Internet.
- The width of the page is set to 1080 pixels. The page is given a grey border (colour code #DEDEDE) with a width of 5 pixels.
- We have set the page to be centred on the monitor screen using the **margin:0 auto** command
- Any screen area outside of the web page is set to grey (colour code #E9E9E9).
- We have used the **padding** command to leave a clear area of 10 pixels inside the web page before any text or images are displayed.

The next step is to add the header and menu bar graphics images to the project. Begin by right clicking the project name **Snowdonia** and selecting **Add / New folder**.



Rename the folder as **Images**. Right click the Images folder, then select **Add / Existing item**:



Locate the header and menu bar images which you created earlier, and add these to the Images folder.

Return to the MasterFile code page. Include a line within the **<head>** section to give the location of the style sheet:

```
<head runat="server">
  <title>Snowdonia</title>
  <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
  <link rel="stylesheet" type="text/css" href="StyleSheet.css" />
</head>
```

Add another division within **<body>** which we will call **header**:

```
<body>
  <form id="form1" runat="server">
    <div id="wrapper">
      <div id="header">
      </div>
      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
```

Return to the style sheet and add commands to display the image for the page header:

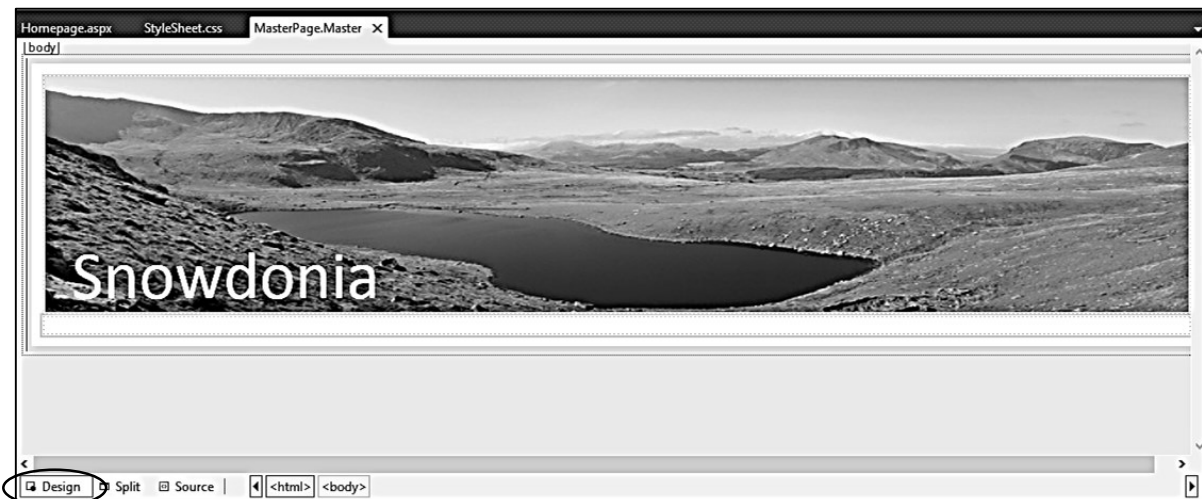
```
body
{
    background-color: #E9E9E9;
}

#header
{
    height: 220px;
    border: 3px solid #E3E3E3;
    background-image: url(Images/header.png);
    background-repeat: no-repeat;
}

#wrapper
{
```

In this section we have created a panel with a height of 220 pixels, surrounded by a grey border with a width of 3 pixels. We then ask for the header image to be displayed on the panel.

Return to the Master Page and use the **Design** button to check that the image is displayed correctly, as shown below. If necessary, change the **height** value of the **header** panel on the style sheet.



The next step is to create an interactive menu with drop-down item lists.



Return to the MasterPage HTML code by clicking the **Source** button, and add two new divisions underneath **header**. Call these **navigation** and **nav**.

```

<div id="header">
</div>

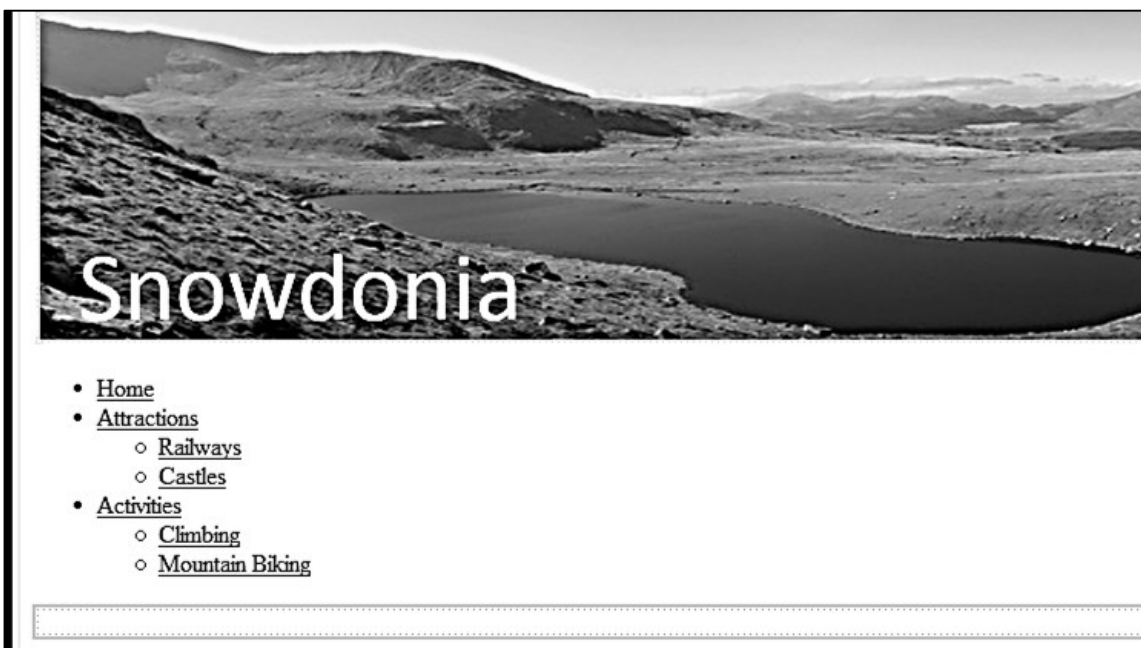
<div id="navigation">
  <ul id="nav">
    <li><a href="#">Home</a></li>
    <li><a href="#">Attractions</a>
      <ul>
        <li><a href="#">Railways</a></li>
        <li><a href="#">Castles</a></li>
      </ul>
    </li>
    <li><a href="#">Activities</a>
      <ul>
        <li><a href="#">Climbing</a></li>
        <li><a href="#">Mountain Biking</a></li>
      </ul>
    </li>
  </ul>
</div>

```

Inside the navigation division we create the menu, first of all as a bullet point list. We do this using some new HTML tags:

- <ul> This tag, standing for **unordered list**, tells the computer that a bullet point list is to be displayed.
- <li> This tag, standing for **list item**, produces a bullet point.
- <a href> Indicates to the computer that a list item can be clicked by the mouse to move to another location or web page.

Use the **Design** option to view the page that we have produced so far:



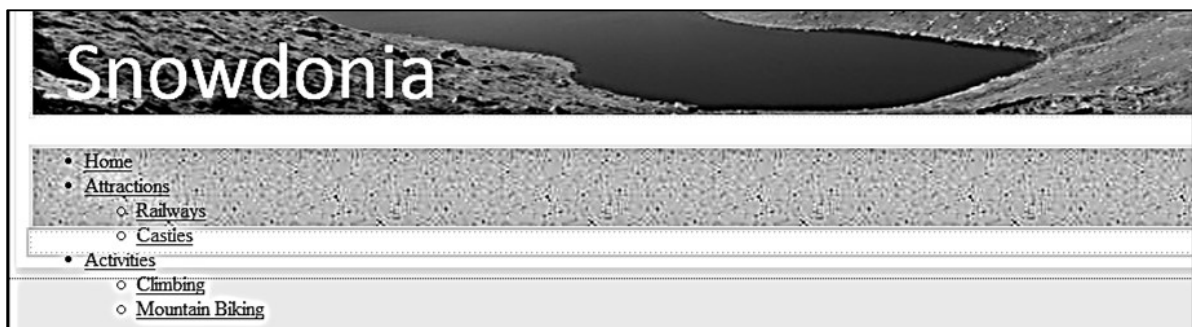
Notice that we have three main headings: **Home, Attractions and Activities**. Within the **Attractions** category are two further choices: **Railways** and **Castles**, and within the **Activities** category are **Climbing** and **Mountain Biking**.

We now need to format the menu using the style sheet, so that the menu items appear in a horizontal row below the header image.

Go to the end of the style sheet and add another section of code for the **navigation** division after the existing code:

```
#navigation
{
    height: 60px;
    border: 3px solid #E3E3E3;
    margin-top: 20px;
    background-image: url(Images/bar.png);
    text-shadow: 0.15em 0.15em #333;
}
```

Return to the Design view. The background bar for the menu is now in place:



Go to the style sheet and add commands for the **nav** division. These will format the list items to create the horizontal menu, as shown below.

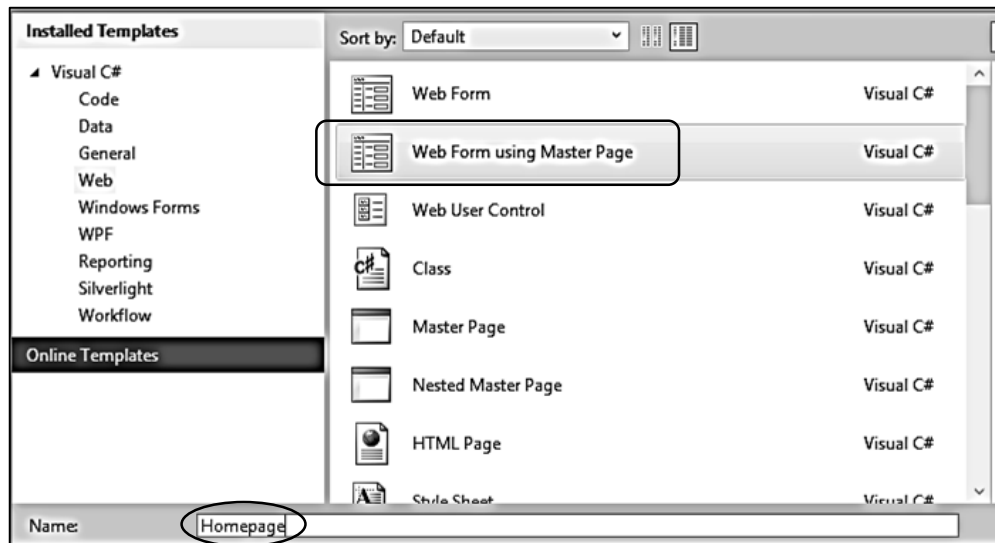
These commands cause the menu items to be displayed as rectangular blocks alongside one another, rather than as simple text in a vertical list. The last of these commands, **#nav a:hover**, will make the text of a menu item change from white to light blue colour when the mouse pointer hovers above it.

```
#nav
{
    list-style: none;
}
#nav ul
{
    list-style: none;
    margin: 0;
    padding: 0;
    width: auto;
    display: none;
}
#nav li
{
    font-size:24px;
    float:left;
    position: relative;
    width:220px;
    height: 40px;
}
#nav li ul li
{
    background-image: url(Images/bar.png);
    background-repeat: no-repeat;
    background-size: cover;
    border: 3px solid #e3e3e3;
    padding-left:10px;
}
#nav a:link, #nav a:active, #nav a:visited
{
    display:block;
    color: #fff;
    text-decoration: none;
}
#nav a:hover
{
    color: #ADD8E6;
}
```

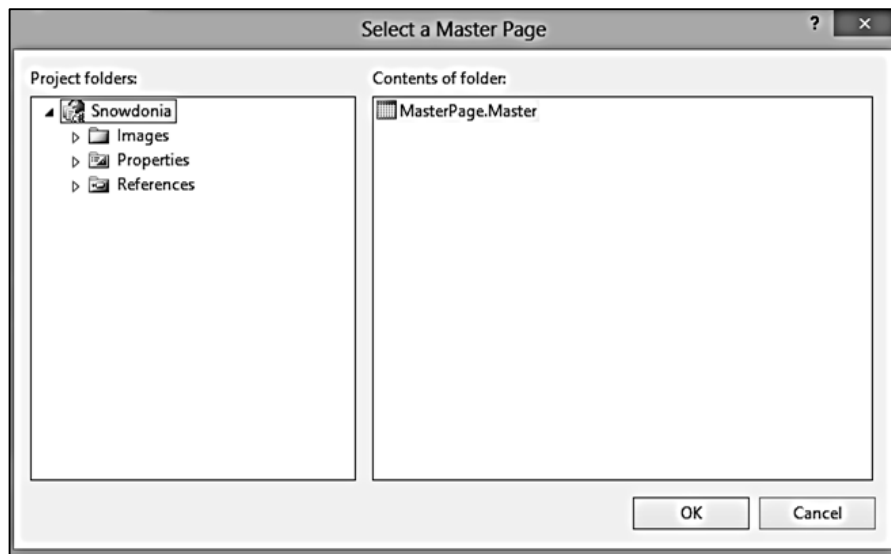
To test the work we have done so far, it is best to create an actual web page rather than just use the **Display** preview option. The Master Page cannot be run directly in a web browser, so we will first create a homepage for the website which incorporates the Master Page template.

Right click the **Snowdonia** project icon in the **Solution Explorer** window, then select **Add / New item**.

Choose the option: **Web Form using Master Page**, and give this the name **Homepage**:



Click OK to incorporate the MasterPage template we have created:



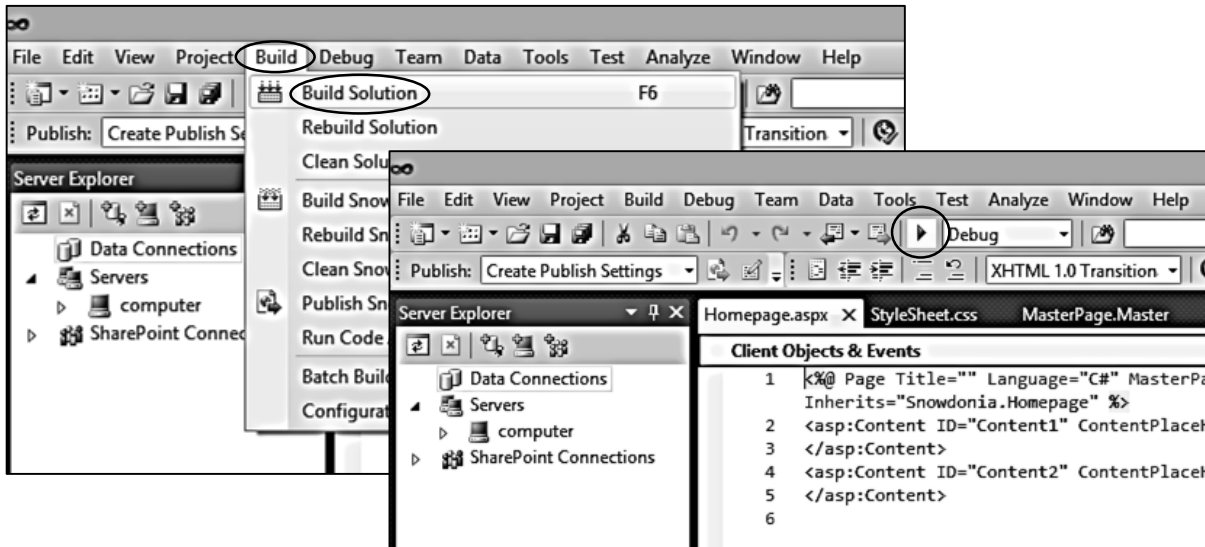
An empty Homepage.aspx file will be produced. We will be able to add content to this later.

```

Homepage.aspx X StyleSheet.css MasterPage.Master
Client Objects & Events (No Events)
1 <%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master" AutoEventWireup="true"
  Inherits="Snowdonia.Homepage" %>
2 <asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
3 </asp:Content>
4 <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
5 </asp:Content>
6

```

To run the Homepage, first select the **Build / Build Solution** menu option, then click the green arrow icon.



The homepage should run in the web browser. This website is actually running on a local server which is part of the Microsoft Visual Studio installation on your computer. Once a web site project is completed, the files can be uploaded to an external server and made fully accessible on the Internet.



Notice that the menu items change colour from white to blue as the mouse is moved over them, but the drop-down menu items do not yet appear. To activate the drop-down menus, we will add some Javascript code.

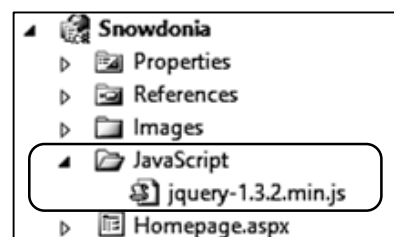
Close the web browser. Go to the **Debug** menu option and click **Stop Debugging**.

Javascript programming is greatly simplified by importing a utility package called jQuery. This is available for free download from the jQuery Foundation at the web site <http://jquery.com/>

Obtain a copy of jQuery. It is recommended that you download a version 1 of jQuery in compressed form.

Make a new folder in the **Snowdonia** project and name this **JavaScript**.

Right click the JavaScript folder and select



**Add / Existing item.** Add the jQuery file.

Add the code shown below to the **<head>** section of the Master Page.

Note: the example given assumes that you have downloaded version **1.3.2.min** of **jquery**. Alter the name of the **jquery** file, if necessary, to correspond with the version file you have installed:

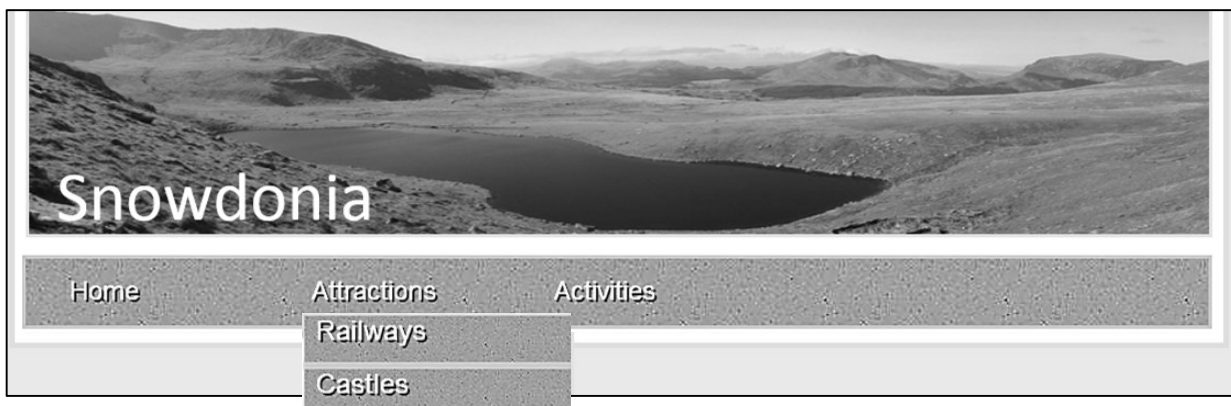
```
<head runat="server">
  <title>Snowdonia</title>
  <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>

    <script type="text/javascript" src="JavaScript/jquery-1.3.2.min.js"></script>
    <script type="text/javascript">

      function mainmenu() {
        $(" #nav ul ").css({ display: "none" });
        $(" #nav li").hover(function () {
          $(this).find('ul:first').css({ visibility: "visible",
            display: "none" }).show(400);
        },
        function () {
          $(this).find('ul:first').css({ visibility: "hidden" });
        });
      }
      $(document).ready(function () {mainmenu();});
    </script>

    <link rel="stylesheet" type="text/css" href="StyleSheet.css" />
  </head>
```

Return to the **Homepage** code and use the **Build / Build Solution** and **Run** menu options. The page should now produce drop down menus for the **Attractions** and **Activities** options:



Return to the HTML code for Master Page, stop debugging, and add **<div>** tags around the **Content Place Holder** in the **<body>** area. Also create a new division for the footer.

```

        <li><a href="#">Climbing</a></li>
        <li><a href="#">Mountain Biking</a></li>
    </ul>
</li>
</ul>
</div>

<div id="content_area">

    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
    </asp:ContentPlaceHolder>

</div>

<div id="footer">

</div>

</div>
</form>

```

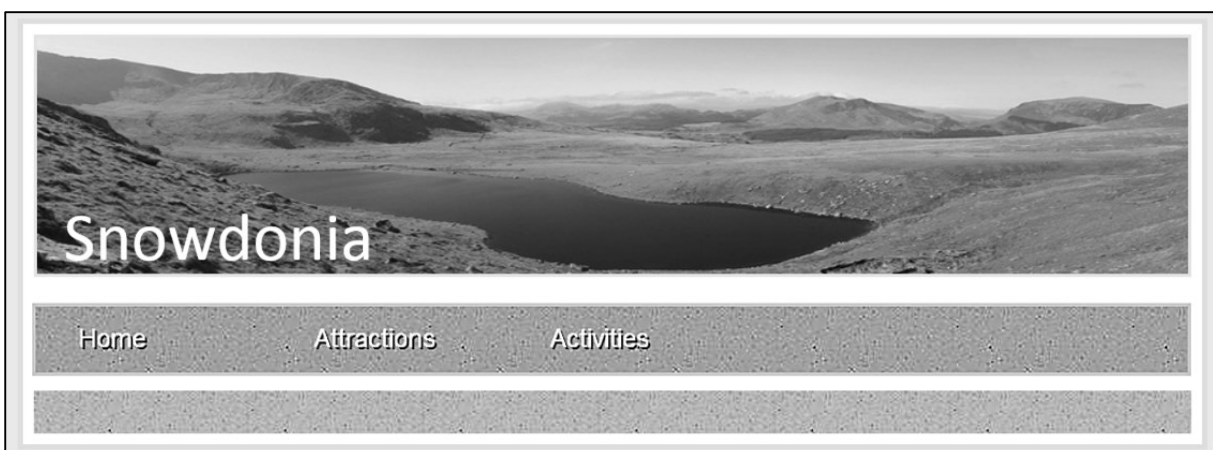
Go to the style sheet and add a footer section:

```

#footer
{
    width: auto;
    height: 40px;
    margin-top: 20px;
    background-image: url(Images/bar.png);
}

```

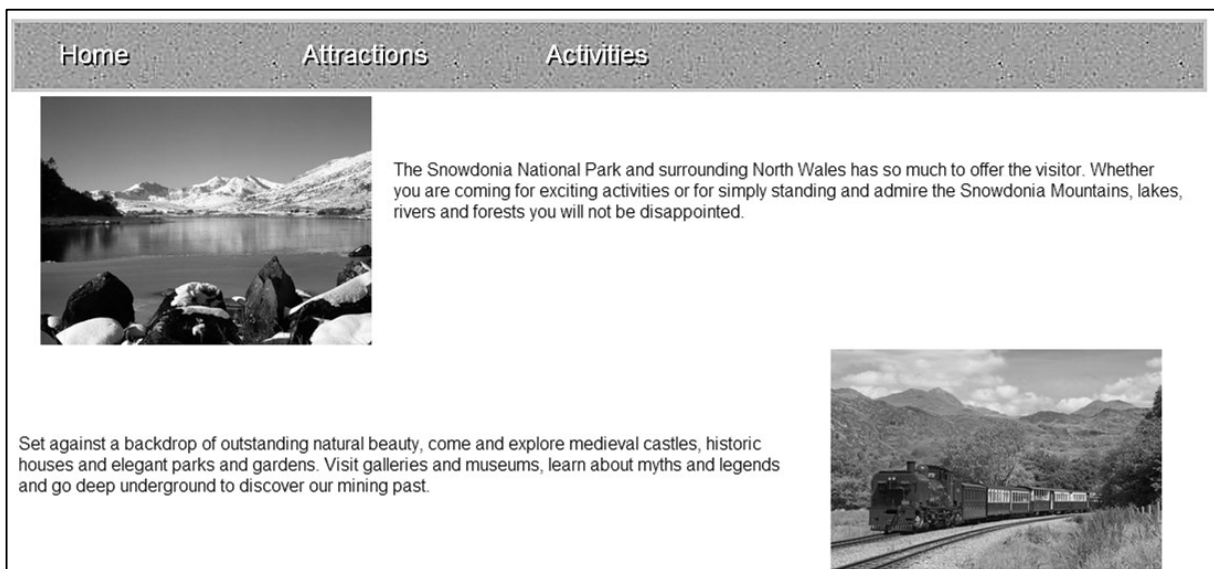
This will again make use of the background bar image. Run the Homepage to check that this is displayed below an empty content area.



One final task is to set the font style for the pages. A sans-serif font would be suitable. Add a line to the **body** group on the style sheet.

```
body
{
    background-color: #E9E9E9;
    font-family: Arial, Helvetica, sans-serif;
}
```

The Master Page is now complete, and we can move to the Homepage to add content. Use the Internet to select some suitable pictures and sections of text which describe features of Snowdonia. For example:



Upload the pictures to the **Images** folder of the project.

Go to Homepage and locate the lines of code which begin and end **ContentPlaceHolder1**. Insert a division called homepage into this area.

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
    runat="server">
    <div id="homepage">
    </div>
</asp:Content>
```

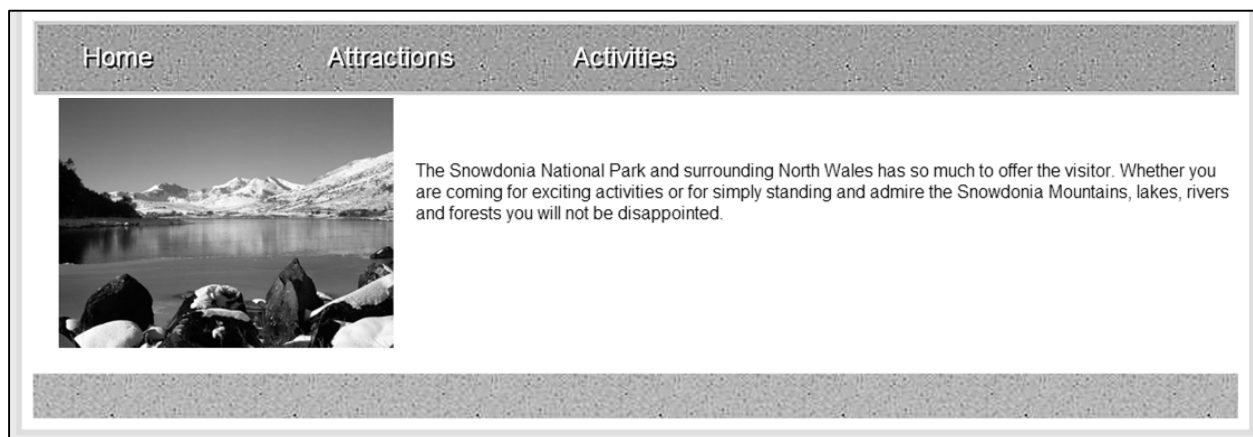
Between these lines we will construct a **table** to display images and text. A table is made up of horizontal rows and vertical columns. To start a new **table row**, we use the HTML tag **<tr>**. To add a data item for a column of the table, we use the HTML tag **<td>**, meaning **table data**.



Add the table, insert the first row and column, and add an image and text. Here, we are using an image called `mountain.jpg` which is stored in the Images folder of the project.

```
<div id="homepage">
  <table>
    <tr>
      <td>
        
        <br /><br /><br />
        The Snowdonia National Park and surrounding North Wales has so much
        to offer the visitor. Whether you are coming for exciting activities
        or for simply standing and admiring the Snowdonia Mountains, lakes,
        rivers and forests you will not be disappointed.
      </td>
    </tr>
  </table>
</div>
```

Run the Homepage and check that the picture and text are displayed:




Another row can now be added to the table, this time with the text on the left, followed by the image on the right. In this example, the picture file is `train.jpg` and is set to `float: right`.


```
        or for simply standing and admire the Snowdonia Mountains, lakes,
        rivers and forests you will not be disappointed.
      </td>
    </tr>
  <tr>
    <td>
      
      <br /><br /><br /><br />
      Set against a backdrop of outstanding natural beauty, come and explore
      medieval castles, historic houses and elegant parks and gardens. Visit
      galleries and museums, learn about myths and legends and go deep
      underground to discover our mining past.
    </td>
  </tr>
```

You may add further rows of text and pictures to the Homepage if you wish.


Home
Attractions
Activities



The Snowdonia National Park and surrounding North Wales has so much to offer the visitor. Whether you are coming for exciting activities or for simply standing and admire the Snowdonia Mountains, lakes, rivers and forests you will not be disappointed.



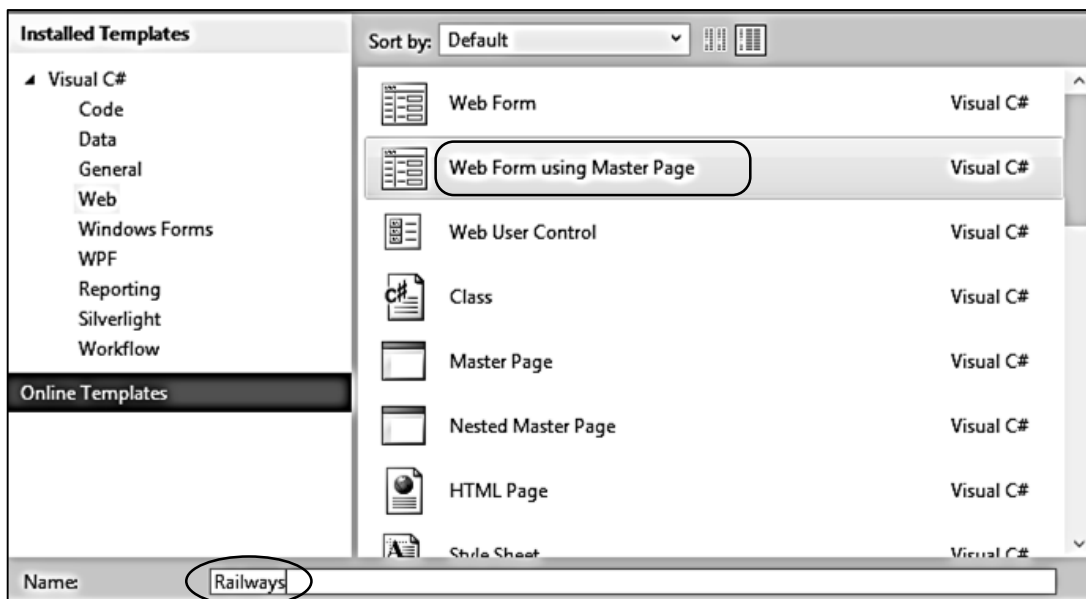
Set against a backdrop of outstanding natural beauty, come and explore medieval castles, historic houses and elegant parks and gardens. Visit galleries and museums, learn about myths and legends and go deep underground to discover our mining past.



Snowdonia has an entire range of accommodation from 5-star hotels to bunkhouses. It has a lot of mainly family run Guesthouses. Do not forget that there are quaint cottages if you wish to self-cater. If you are of a more adventurous frame of mind, there are lots of high quality camp and caravan sites.

We can now link further pages to the web site for the different menu options. We will first create a page for **Railways**.

Exit from the web site and use the menu option **Debug / Stop Debugging**. Right click the Snowdonia project icon and select **Add / New item**. Choose **Web Form using Master Page**, and give the name **Railways**.



Within the **ContentPlaceHolder1** section of the **Railways** page, create a division called **railways**.

Add the word 'Railways' within the division:

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
            runat="server">
    <div id="railways">
        Railways
    </div>
</asp:Content>
```

We will now link this page to the menu system. Return to the Master Page and locate the list items for the menu. Change the **<a href>** entries for the **Home** and **Railways** menu options, so that they show the names of the corresponding web pages to display:

```
<ul id="nav">
    <li><a href="Homepage.aspx">Home</a></li>
    <li><a href="#">Attractions</a>
        <ul>
            <li><a href="Railways.aspx">Railways</a></li>
            <li><a href="#">Castles</a></li>
        </ul>
    </li>
```

Run the Homepage and select the Railways menu option. The railways page should open.



Clicking the **Home** menu option will take you back to the Home Page.

You may like to develop the website further by adding more pages and content.