

Navigation

Asst. Prof. Dr. Özgü Can

Navigation

- Enables users **to find their way** around the site.
- Navigation Controls:
 - Menu
 - TreeView
 - SiteMapPath

Navigation

- A well-organized site is **easy for users to navigate**.
- **Web.sitemap**
 - Used by navigation controls
 - XML-based file
 - Helps you to define the logical structure of your site.
 - You can have **multiple site map files with different names**, with **a configuration setting** in the **web.config** file that exposes these additional files to the system.

<a> Element

- Letting a user move from one page to another.
- **href** attribute
 - Enables to define the address of a page or other resource you want to link to.
 - Such as text, an image, or other HTML.

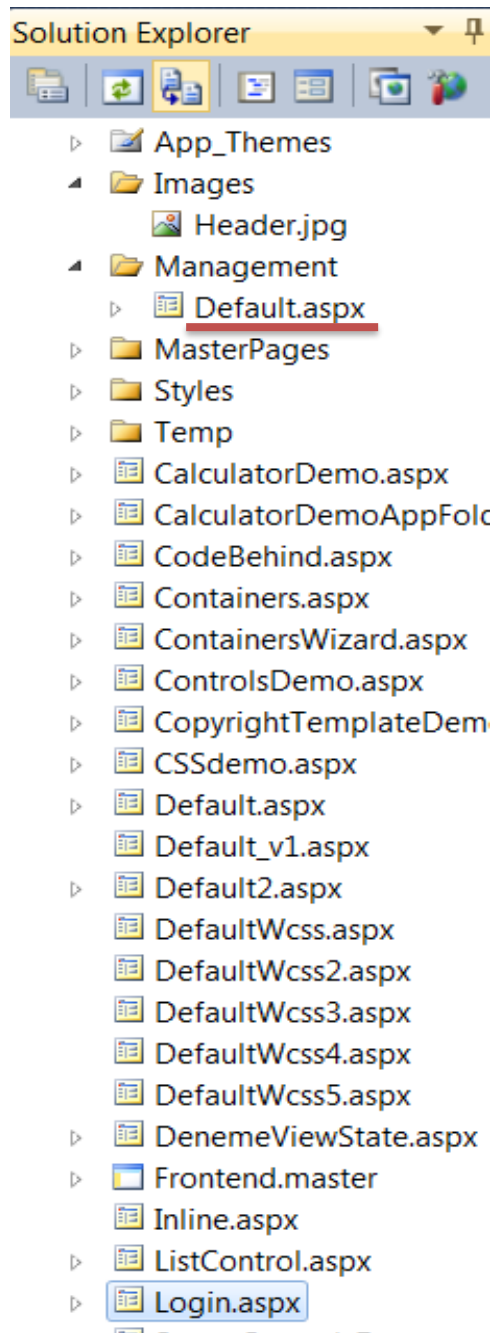
```
<a href="Login.aspx">You can log in here</a>
```

URL

- **U**niform **R**esource **L**ocator
- Used to **uniquely identify a resource**.
- Used in different places:
 - **href** attribute
 - **link** attribute
 - **src** attribute
- URLs can be expressed in:
 1. Relative URL
 2. Absolute URL

Relative URL

- Points to another resource **relative to the location** where the URL is used.



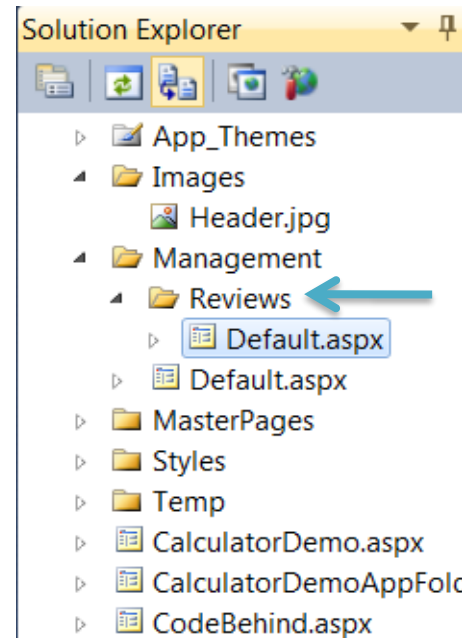
Login.aspx to **Default.aspx** in the **Management** folder:

```
<a href="Management/Default.aspx">Management</a>
```

Default.aspx in the **Management** folder to **Header.jpg**:

```

```



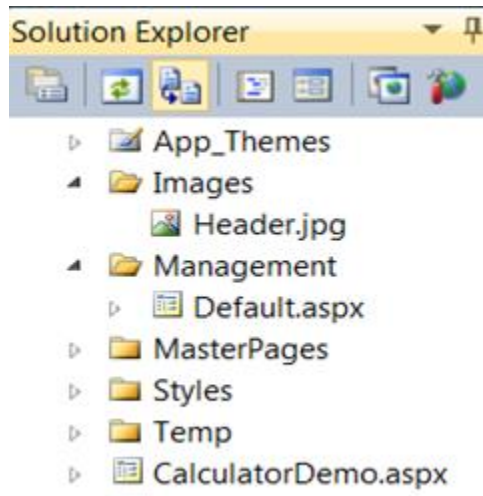
From **Default.aspx** in **Reviews** to **Header.jpg**:

```

```

Root-Based Relatives

- Always start with a leading forward slash (/) to indicate **the root of the site**.



Default.aspx in Management:

```
<a href="/Management/Default.aspx">Management</a>
```



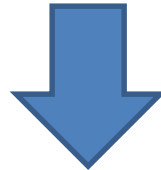
The **root** of the website

Relative URLs in Server-Side Controls

- You can use the tilde (~) character to point to the current root of the site.

- When you create a new site:

<http://localhost:51236/MyEmptyWebSite1>

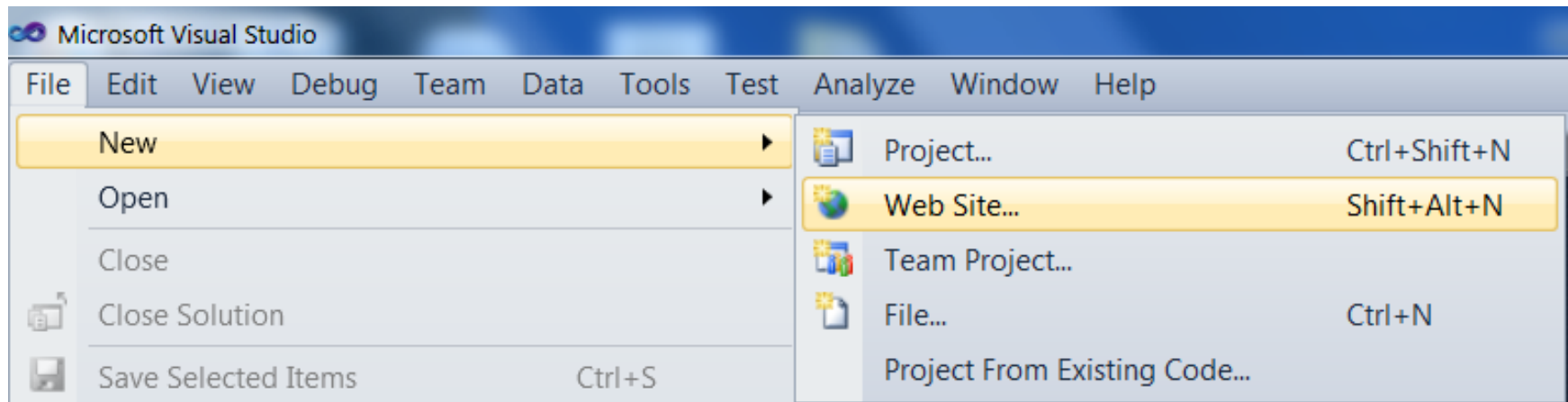


<http://www.mywebsite.com>

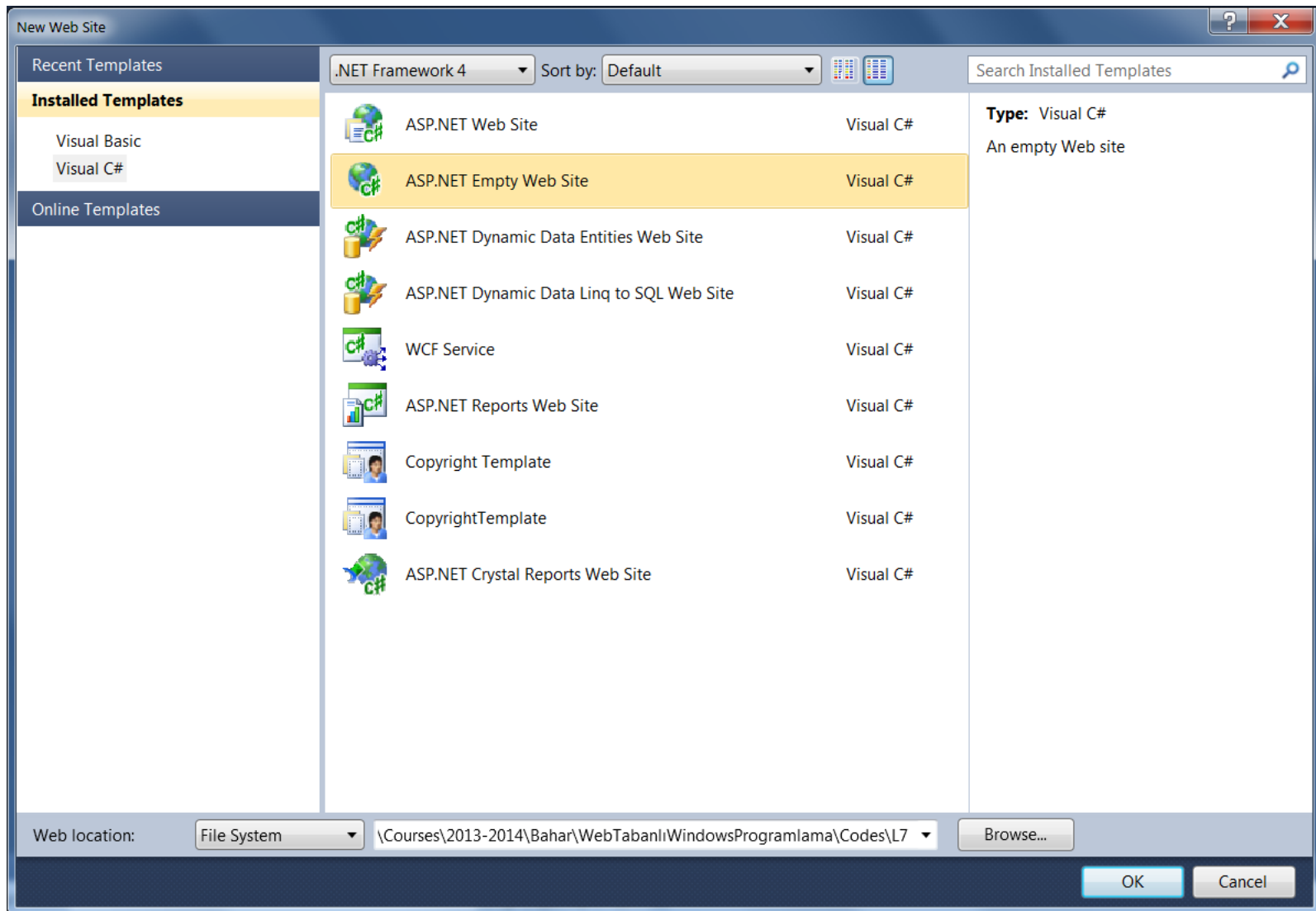
- To stop creating a separate folder:
 - Set **Virtual Path** to /

Example - Virtual Path

- File → New Web Site

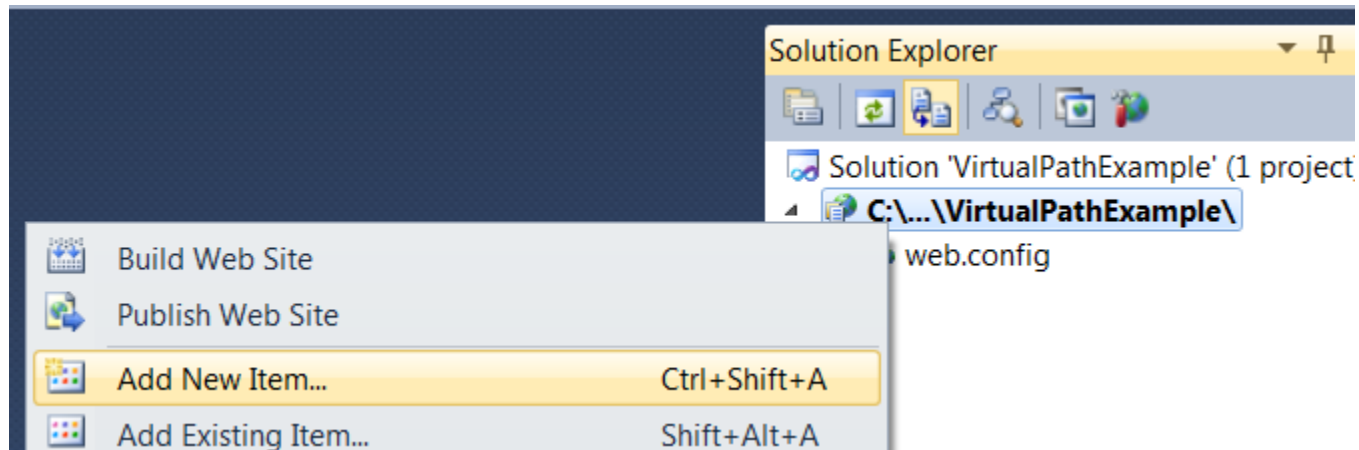


Example - Virtual Path

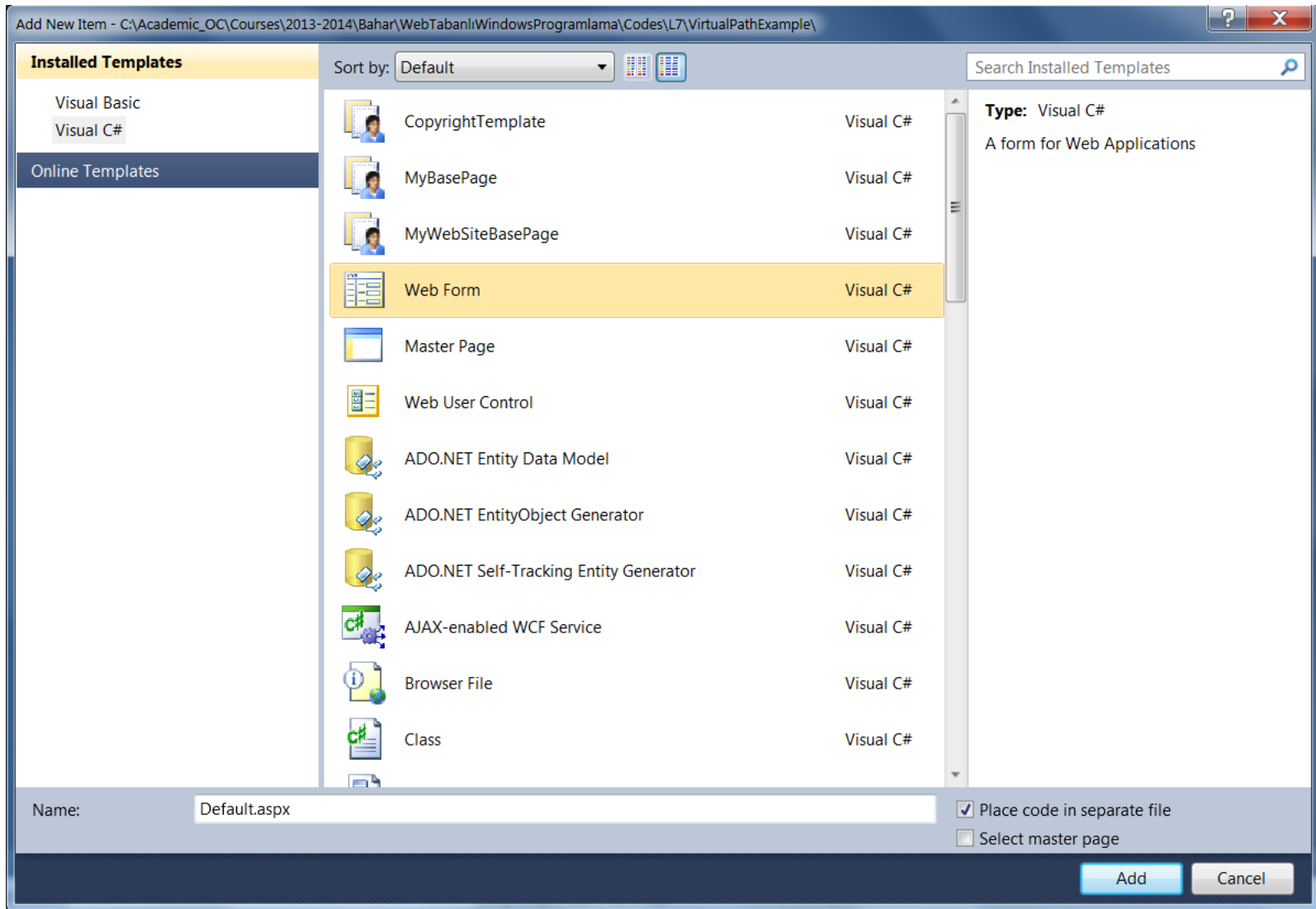


Example - Virtual Path

- Add New Item

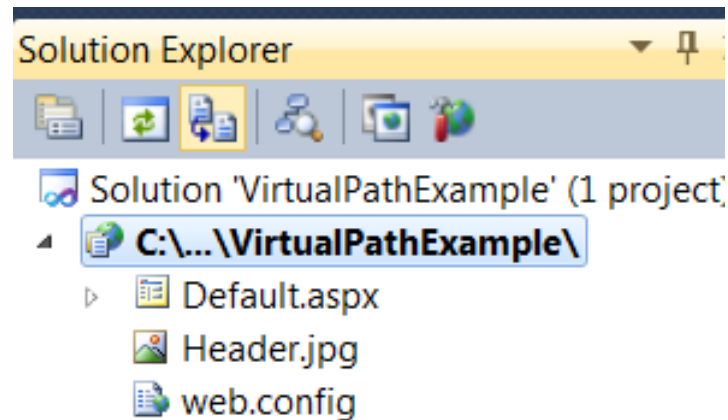


Example - Virtual Path



Example - Virtual Path

- Add **Header.jpg**



Example - Virtual Path

In `Default.aspx` source code:

```
<div>
    <asp:Image ID="Image1" runat="server" ImageUrl="Header.jpg"/><br />
    <asp:Image ID="Image2" runat="server" ImageUrl="/Header.jpg"/><br />
    <asp:Image ID="Image3" runat="server" ImageUrl="~/Header.jpg"/><br />
</div>
```

Example - Virtual Path

- View in browser



Example - Virtual Path

- Note the address bar of the browser reads something like:

<http://localhost:64026/VirtualPathExample/Default.aspx>

- The website is located **in a separate folder** under the web server called **localhost**.
- The second image is looked for at:

<http://localhost:64026/Header.jpg>



Doesn't exist!

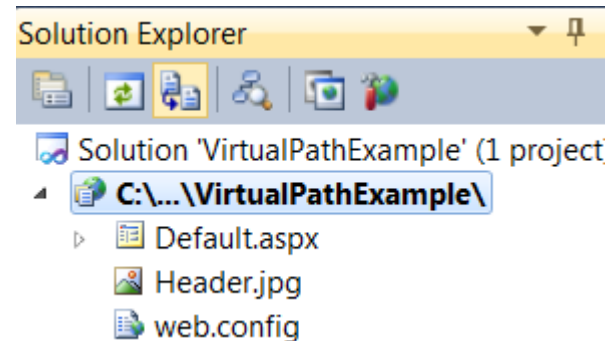
Example - Virtual Path

In the browser → View source code:

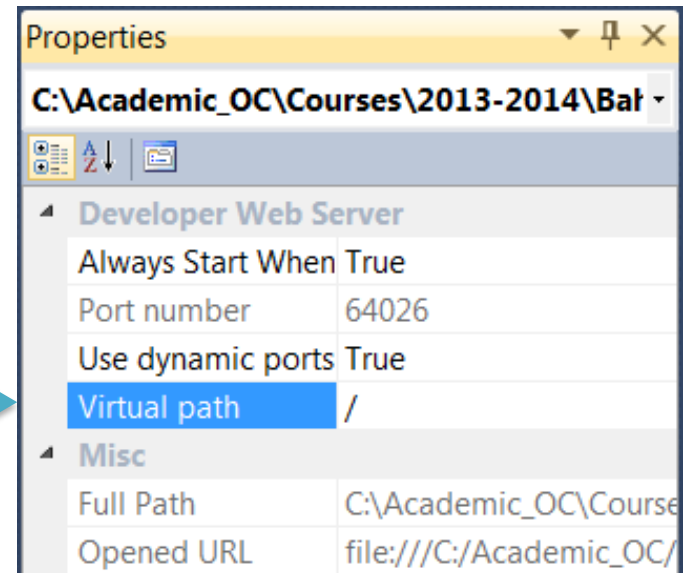
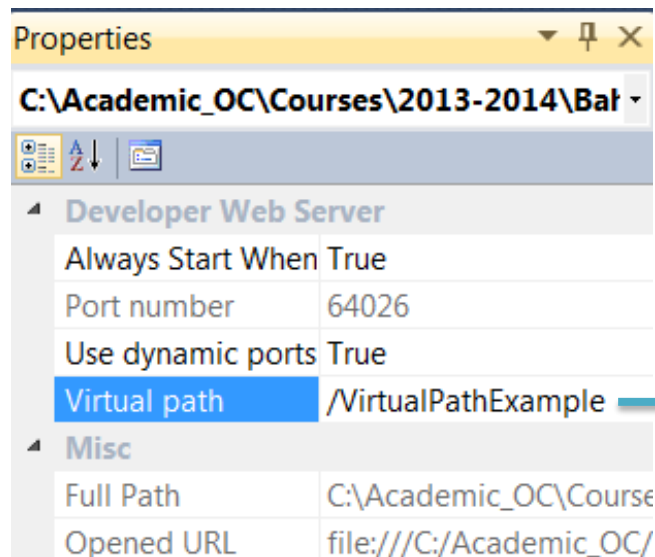
```
<br />  
<br />  
<br />
```

Example - Virtual Path

- In the Solution Explorer:



– Web site's Properties



Example - Virtual Path

- View in browser:

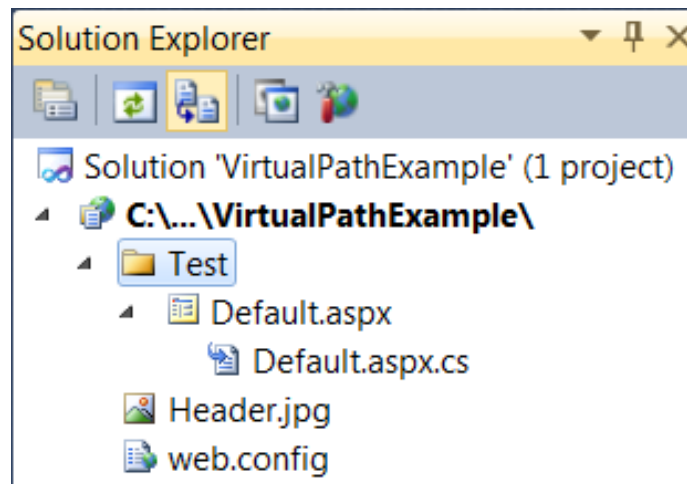


Example - Virtual Path

- The address bar of the browser:

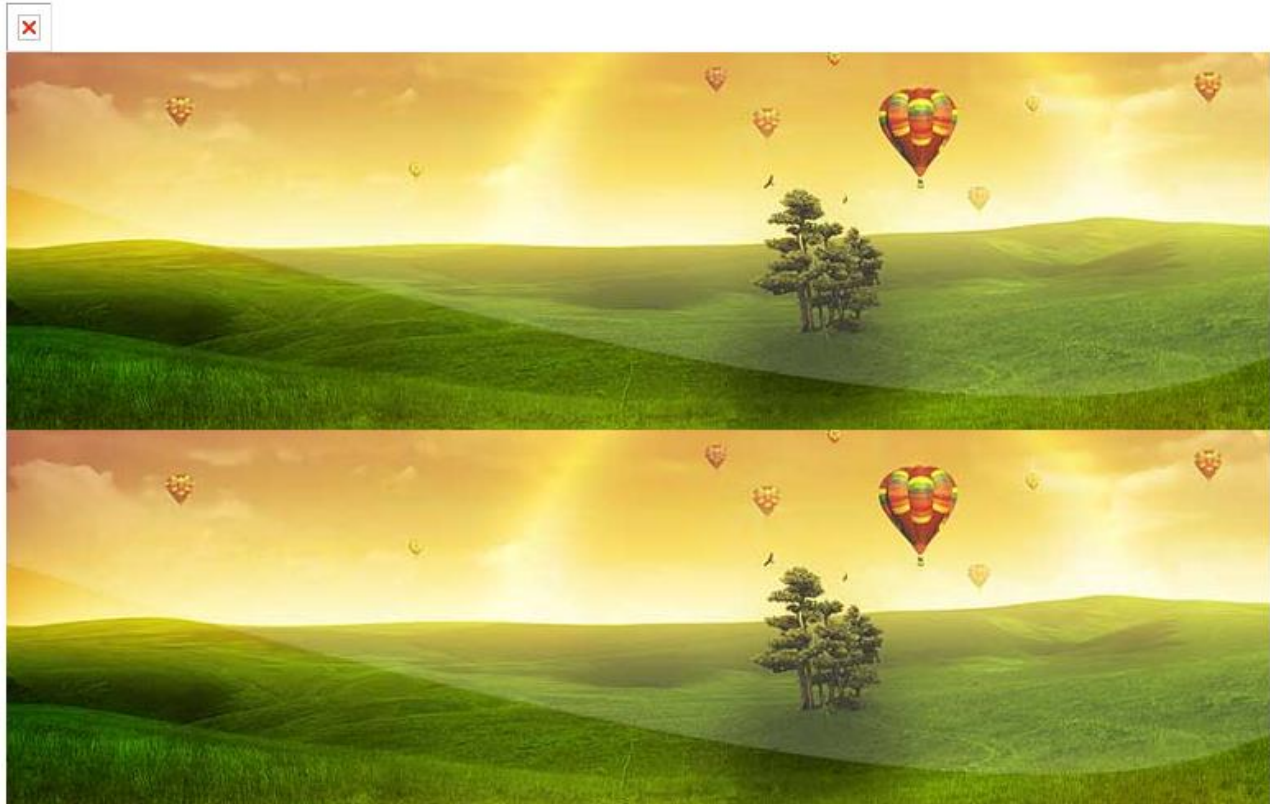
`http://localhost:64026/Default.aspx`

- Create a new folder: **Test**
- Drag **Default.aspx** into **Test** folder



Example - Virtual Path

- View in browser:

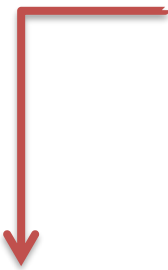


Example - Virtual Path

- The address bar of the browser:

`http://localhost:64026/Test/Default.aspx`

- View source in the browser:



```
<br />  
<br />  
<br />
```

Relative to the current document → Broken image!

Absolute URL

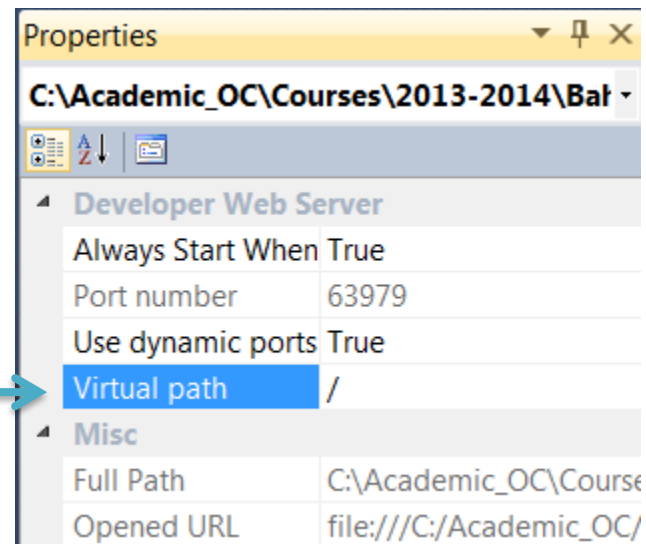
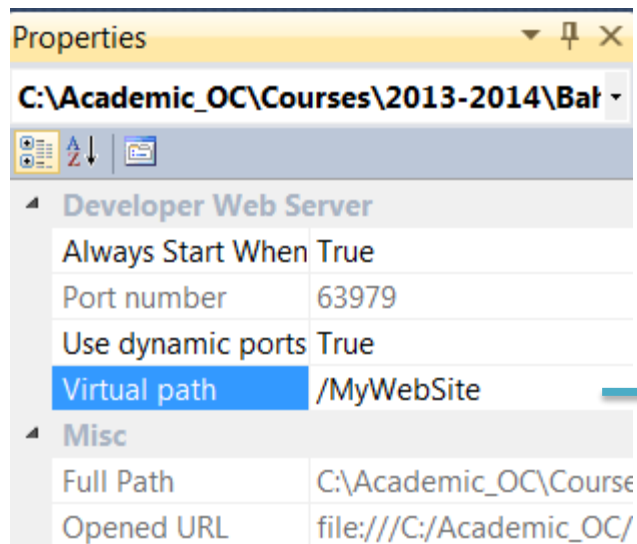
- Refers to a resource by its **full path**.
- Refers to a resource **outside** your own web site.
 - **http://** prefix
- Fixed location.
- Use → When referring to resources outside your web.

Web.sitemap

- XML-based file that describes the *logical structure* of your web site.
- You can have multiple site map files with different names, with a configuration setting in the web.config file.
 - In most cases, a single site map file will be sufficient.

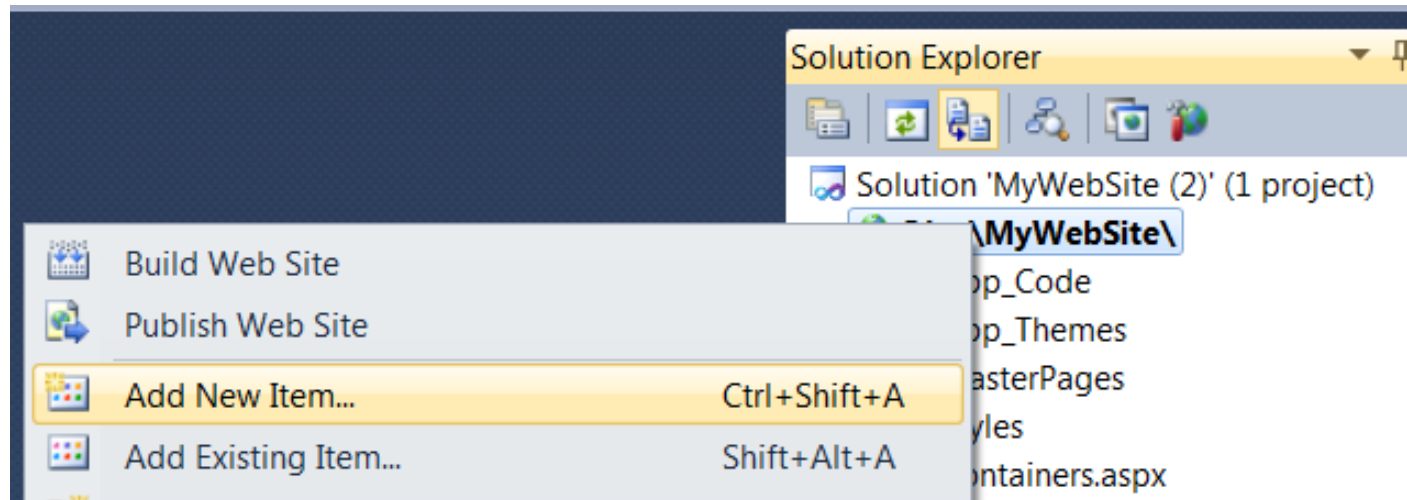
Example - Web.sitemap

- Open **MyWebSite**
- Set **Virtual Path** property

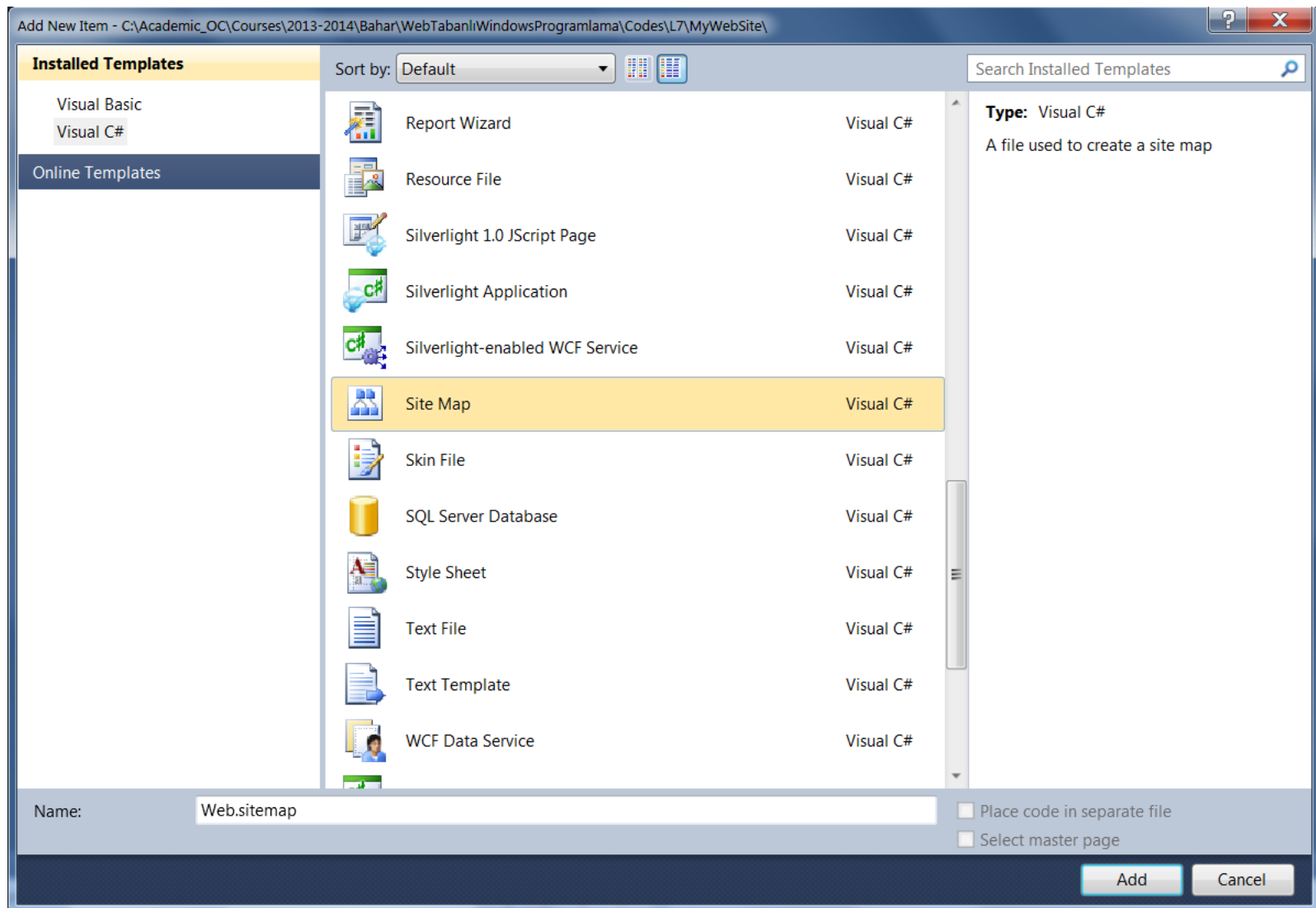


Example - Web.sitemap

- Add New Item



Example - Web.sitemap



Example - Web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="/" title="Home" description="Home" >
    <siteMapNode url="/Default.aspx" title="Home" description="Go to the homepage" />
    <siteMapNode url="/Reviews/Default.aspx" title="Reviews" description="Reviews published on this site" >
      <siteMapNode url="/Reviews/AllByGenre.aspx" title="By Genre" description="All reviews grouped by genre" />
      <siteMapNode url="/Reviews/All.aspx" title="All Reviews" description="All reviews published on this site" />
    </siteMapNode>

    <siteMapNode url="/About/Default.aspx" title="About" description="About my site" >
      <siteMapNode url="/About/Contact.aspx" title="Contact" description="Contact me" />
      <siteMapNode url="/About/AboutMe.aspx" title="About Me" description="About myself" />
    </siteMapNode>

    <siteMapNode url="/Login.aspx" title="Login" description="Log in to the web site" />
  </siteMapNode>
</siteMap>
```

Root element

Web.sitemap

- **siteMapNode** can have **many child nodes**.
- There can only be **one siteMapNode under siteMap** element.

url → Points to **valid page** in your web site

title → Used in the navigation controls to display the name of the page

description → Used as a **tooltip** for the navigation elements

To be able to work with the **Web.sitemap** file → **SiteMapDataSource** control should be added

Example - Menu Control

- Open master page → **Frontend.master**
- Remove the text **Menu goes here**

```
<div id="PageWrapper">
  <div id="Header"><a href="/" runat="server"></a></div>
  <div id="MenuWrapper">Menu goes here.</div>
  <div id="MainContent">
    <asp:ContentPlaceHolder id="cpMainContent" runat="server">

    </asp:ContentPlaceHolder>
  </div>
  <div id="Sidebar">Select a theme<br/>
    <asp:DropDownList ID="ThemeList" runat="server" AutoPostBack="True"
      onselectedindexchanged="ThemeList_SelectedIndexChanged">
      <asp:ListItem>Monochrome</asp:ListItem>
      <asp:ListItem>DarkGrey</asp:ListItem>
    </asp:DropDownList>
  </div>
  <div id="Footer">Footer goes here.</div>
</div>
```

Example - Menu Control

- From the **Navigation** category of the Toolbox, drag a **Menu** and drop it between the **MenuWrapper** div tags.

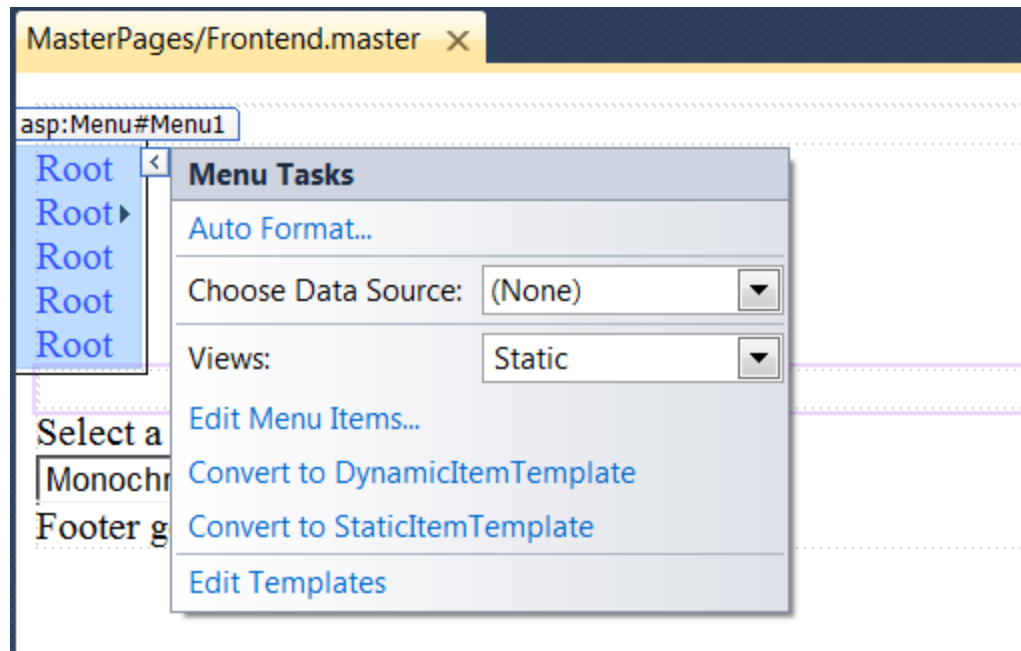
```
<div id="MenuWrapper">  
    <asp:Menu ID="Menu1" runat="server">  
    </asp:Menu>  
</div>
```

- Add **CssClass**

```
<div id="MenuWrapper">  
    <asp:Menu ID="Menu1" runat="server" CssClass="MainMenu">  
    </asp:Menu>  
</div>
```

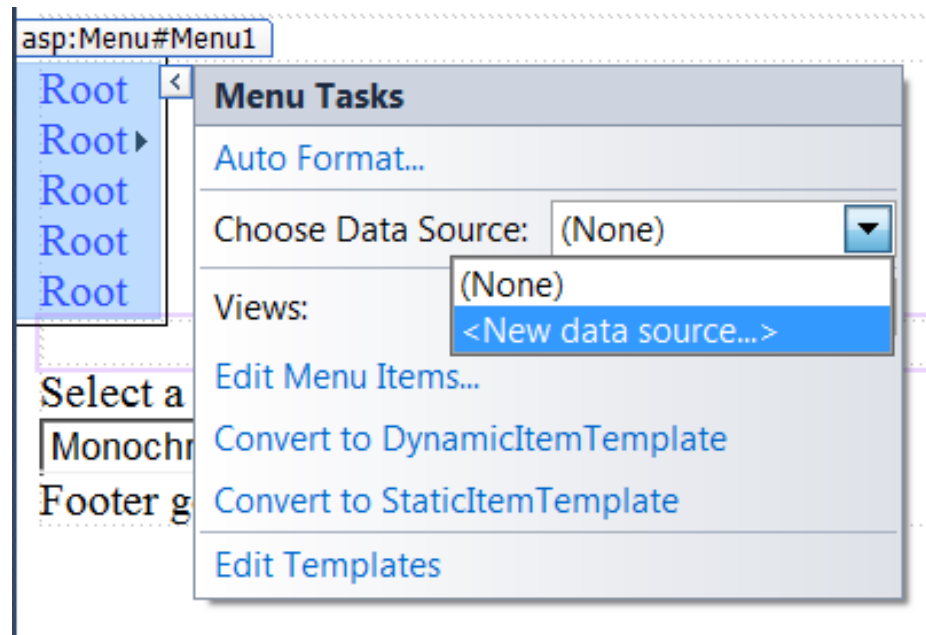

Example - Menu Control

- In Design View → Click the **Menu** control's Smart Tasks panel



Example - Menu Control

Choose Data Source → <New data source>





Choose a Data Source Type

Where will the application get data from?



Site Map



XML File

Connect to the site navigation tree for this application (requires a valid sitemap file at the application root).

Specify an ID for the data source:

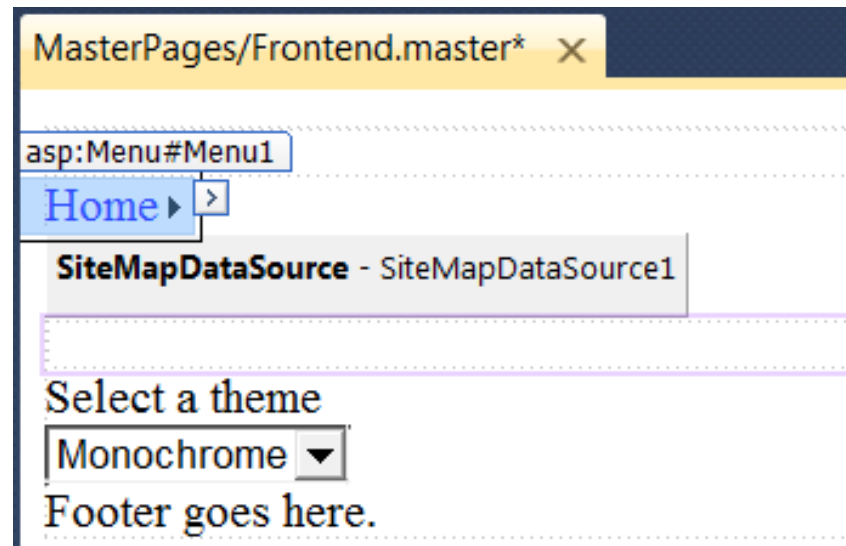
SiteMapDataSource1

OK

Cancel

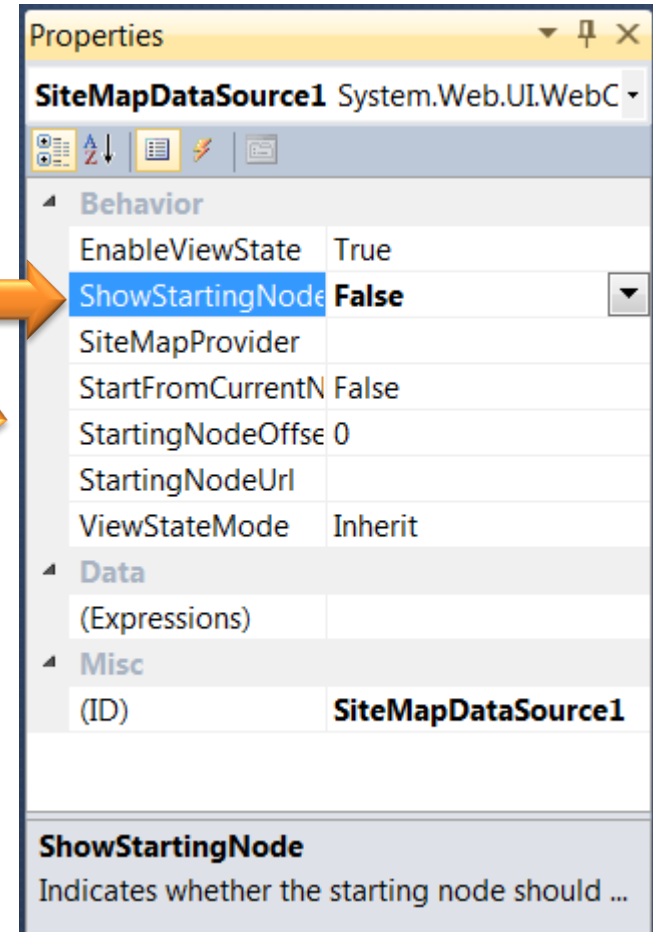
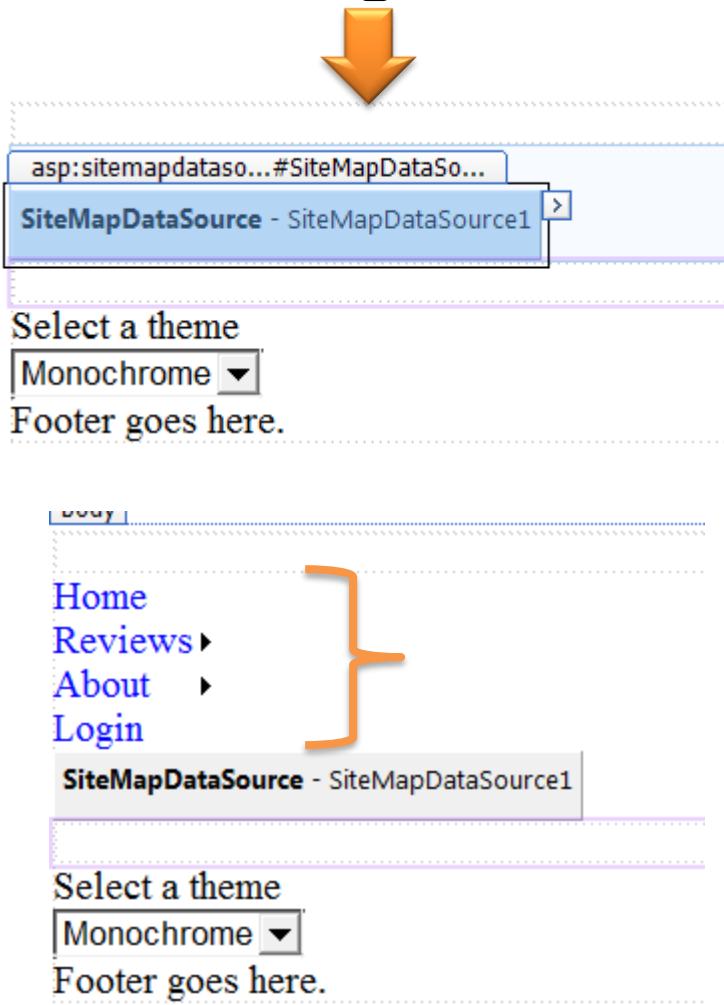
Example - Menu Control

- In Design View → **Menu** control shows the top-level element, **Home**.



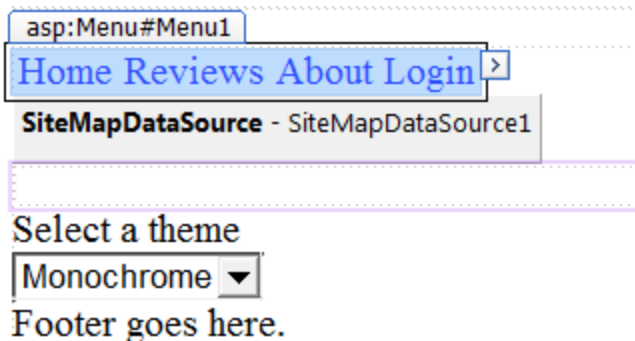
Example - Menu Control

Click SiteMapDataSource



Example - Menu Control

- Click the **Menu** control:
 - **StaticEnableDefaultPopOutImage = False**
 - **Orientation = Horizontal**

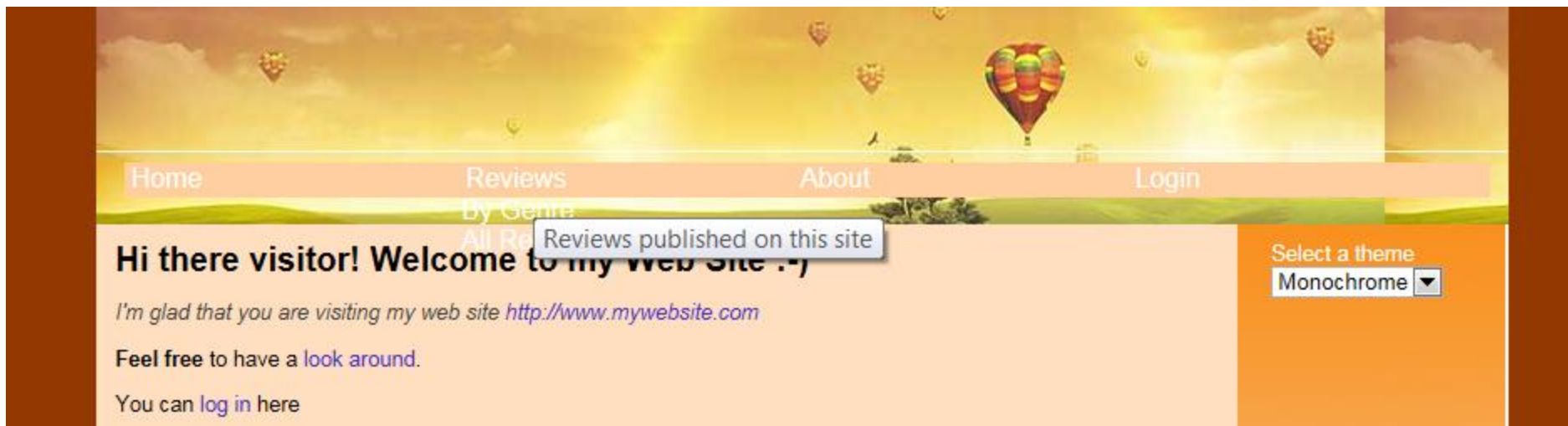


```
<div id="MenuWrapper">  
  <asp:Menu ID="Menu1" runat="server" CssClass="MainMenu"  
    DataSourceID="SiteMapDataSource1" Orientation="Horizontal"  
    StaticEnableDefaultPopOutImage="False">  
  </asp:Menu>  
  <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server"  
    ShowStartingNode="False" />  
</div>
```

Example - Menu Control

View `Default.aspx` in browser

Choose **Monochrome**



Example - Styling Menu Control

- Open **Monochrome.css**
- Change the look of main menu items.


```
ul.level1
{
    /* Defines the appearance of main menu items*/
    font-size: 14px;
    font-weight: bold;
    height: 19px;
    line-height: 19px;
}

ul.level1.selected
{
    /* Defines the appearance of active menu items */
    background-color: #FF5300;
}

a.level1
{
    /* Adds some white space to the left of the main menu item text */
    margin-left: 5px;
}

a.level2
{
    /* Defines the appearance of the sub menu items */
    background-color: #BCE75A;
    padding-left: 8px;
}

a.level1:hover, a.level2:hover
{
    /* Defines the hover style for the main and sub items*/
    background-color: #FF5300;
}
```

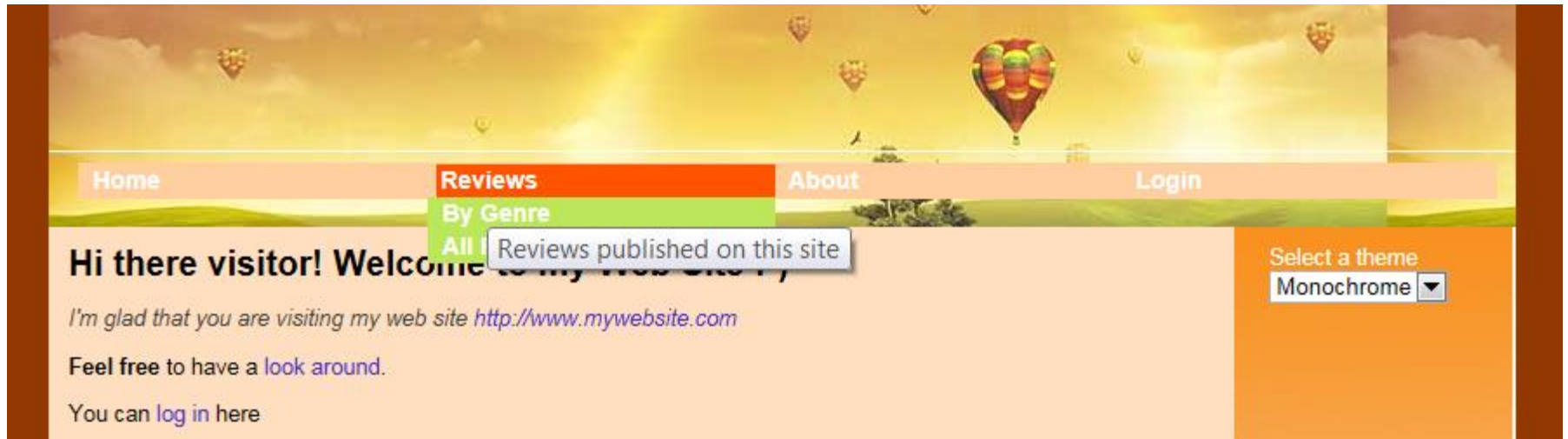
Example - Styling Menu Control

- Create the following folders and Web Forms using **BasePage** template:

| FOLDER | FILE NAME | TITLE |
|----------|-----------------|--------------------------|
| /Reviews | Default.aspx | My Favorite Reviews |
| /Reviews | All.aspx | All Reviews |
| /Reviews | AllByGenre.aspx | Reviews Grouped by Genre |
| /About | Default.aspx | About this Site |
| /About | Contact.aspx | Contact Us |
| /About | AboutMe.aspx | About Me |

Example - Styling Menu Control

View `Default.aspx` in browser



TreeView

- Displays a hierarchical list of items.
- The data used by the **TreeView** control **is not limited** to the **Web.sitemap** file.
- You can also bind it to regular XML files and even create a **TreeView** or its items (called **nodes**) programmatically.

TreeView Properties

| PROPERTY | DESCRIPTION |
|----------------------|--|
| CssClass | Enables you to set a CSS <code>class</code> attribute that applies to the entire control. |
| CollapseImageUrl | The image that collapses a part of the tree when clicked. The default is an icon with a minus symbol on it. |
| ExpandImageUrl | The image that expands a part of the tree when clicked. The default is an icon with a plus symbol on it. |
| CollapseImageToolTip | The tooltip that is shown when a user hovers over a collapsible menu item. |
| ExpandImageToolTip | The tooltip that is shown when a user hovers over an expandable menu item. |
| ShowExpandCollapse | Determines whether the items in the <code>TreeView</code> can be collapsed and expanded by clicking an image in front of them. |
| ShowLines | Determines whether lines are used to connect the individual items in the tree. |
| ExpandDepth | Determines the level at which items in the tree are expanded when the page first loads. The default setting is <code>FullyExpand</code> , which means all items in the tree are visible. Other allowed settings are numeric values to indicate the level to which to expand. |

Example - TreeView Control

- Open the master page → **Frontend.master**
- Add a **TreeView** control by dragging it from the Toolbox.

```
<div id="MenuWrapper">
    <asp:Menu ID="Menu1" runat="server" CssClass="MainMenu"
        DataSourceID="SiteMapDataSource1" Orientation="Horizontal"
        StaticEnableDefaultPopOutImage="False">
    </asp:Menu>
    <asp:TreeView ID="TreeView1" runat="server">
    </asp:TreeView>
    <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server"
        ShowStartingNode="False" />
</div>
```

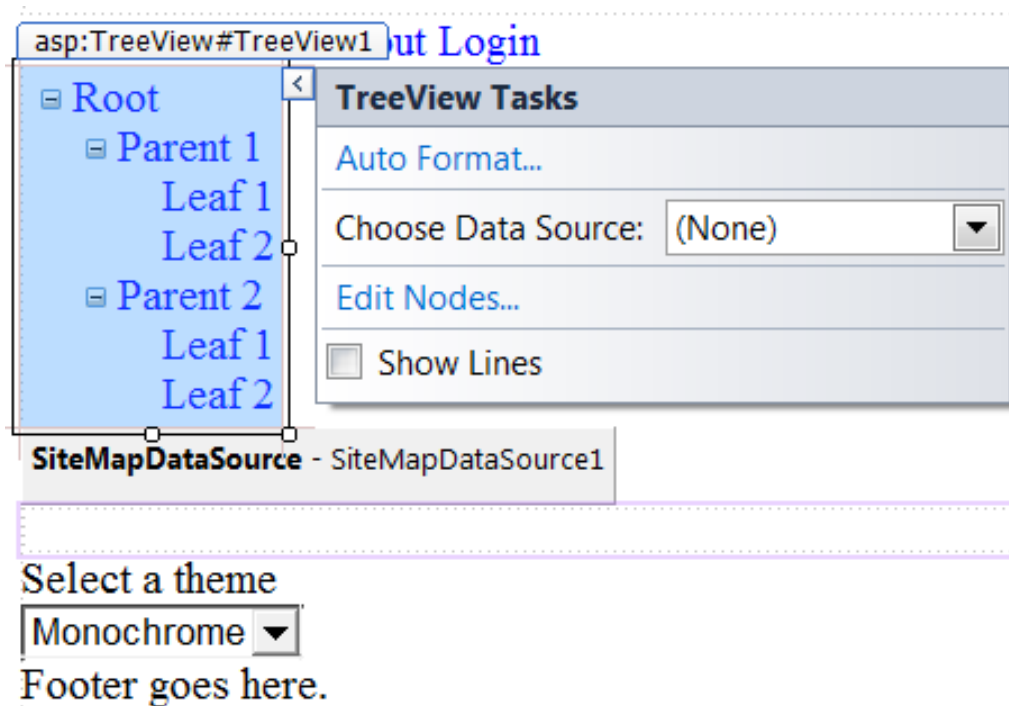
Example - TreeView Control

Add **<LevelStyles>** element

```
<asp:TreeView ID="TreeView1" runat="server">  
  <LevelStyles>  
    <asp:TreeNodeStyle CssClass="FirstLevelMenuItems"/>  
  </LevelStyles>  
</asp:TreeView>
```

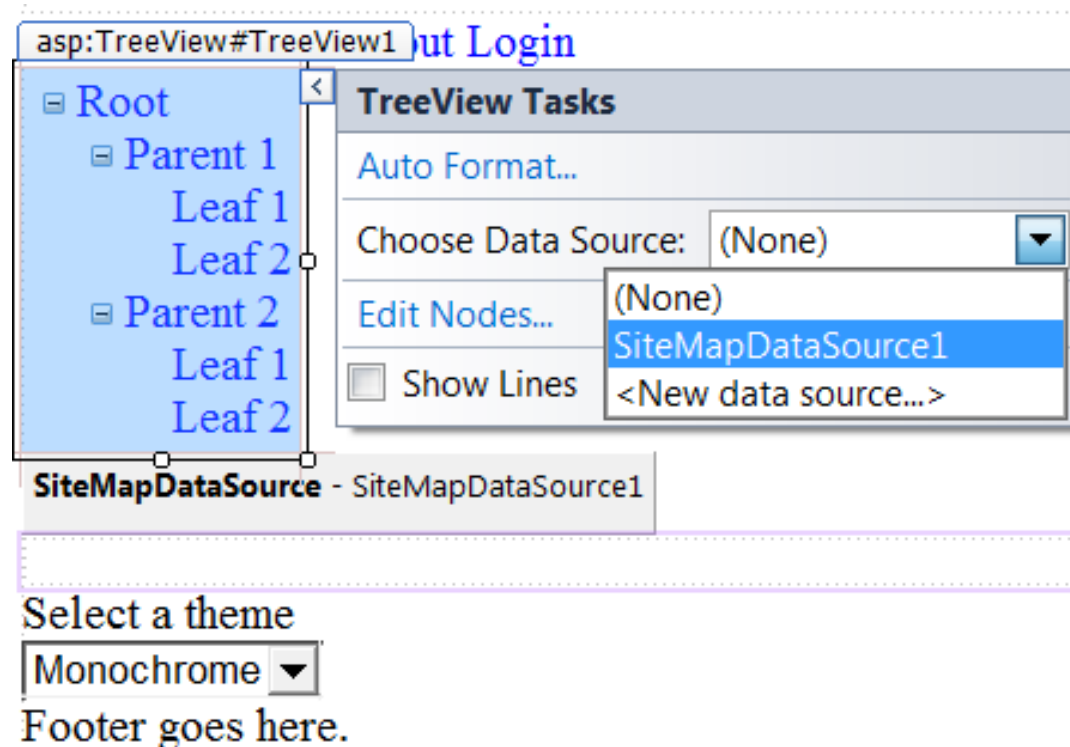
Example - TreeView Control

- Switch to Design View



Example - TreeView Control

Choose Data Source → SiteMapDataSource1



Example - TreeView Control

- In **TreeView** control Properties:
 - **ShowExpandCollapse = False**



True



False

In Frontend.master.cs:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        string selectedTheme = Page.Theme;
        HttpCookie preferredTheme = Request.Cookies.Get("PreferredTheme");
        if (preferredTheme != null)
        {
            selectedTheme = preferredTheme.Value;
        }
        if (!string.IsNullOrEmpty(selectedTheme) && ThemeList.Items.FindByValue(selectedTheme) != null)
        {
            ThemeList.Items.FindByValue(selectedTheme).Selected = true;
        }
    }

    switch (Page.Theme.ToLower())
    {
        case "darkgrey":
            Menu1.Visible = false;
            TreeView1.Visible = true;
            break;
        default:
            Menu1.Visible = true;
            TreeView1.Visible = false;
            break;
    }
}
```



Add the code

Example - TreeView Control

- View in browser



Home
Reviews
 By Genre
 All Reviews
About
 Contact
 About Me
Login

Hi there visitor! Welcome to my Web Site :-)

I'm glad that you are visiting my web site <http://www.mywebsite.com>

Feel free to have a **look around**.

You can **log in** here

Select a theme
DarkGrey ▼

SiteMapPath

- Shows you where you are in the site's structure.

| PROPERTY | DESCRIPTION |
|--------------------------------------|---|
| <code>PathDirection</code> | Supports two values: <code>RootToCurrent</code> and <code>CurrentToRoot</code> . The first setting shows the root element on the left, intermediate levels in the middle, and the current page at the right of the path. The <code>CurrentToRoot</code> setting is the exact opposite where the current page is shown at the left of the breadcrumb path. |
| <code>PathSeparator</code> | Defines the symbol or text to show between the different elements of the path. The default is the greater than symbol (<code>></code>) but you can change it to something like the pipe character (<code> </code>). |
| <code>RenderCurrentNodeAsLink</code> | Determines whether the last element of the path (the current page) is rendered as a text link or as plain text. The default is <code>False</code> , which is usually fine, because you are already on the page that element is representing, so there's no real need for a link. |
| <code>ShowToolTips</code> | Determines whether the control displays tooltips (retrieved from the <code>description</code> attribute of the <code>siteMapNode</code> elements in the <code>web.sitemap</code> file) when the user hovers over the elements in the path. The default is <code>True</code> , which means the tooltips are shown by default. |

Example - SiteMapPath Control

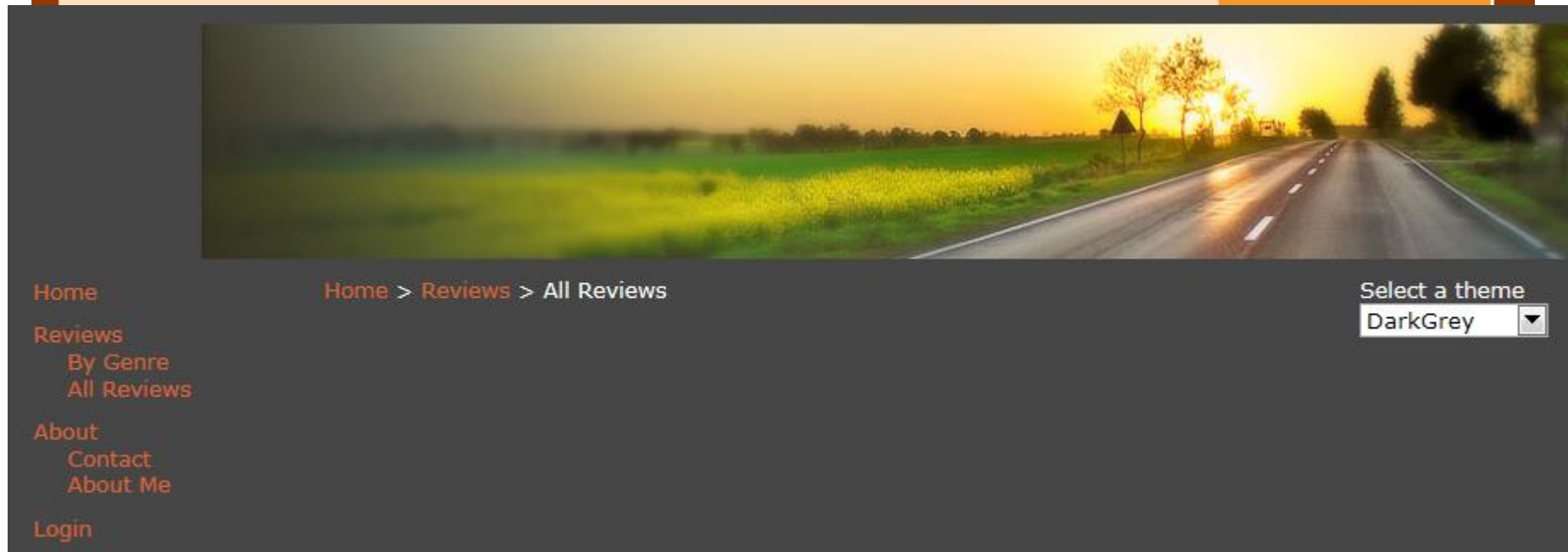
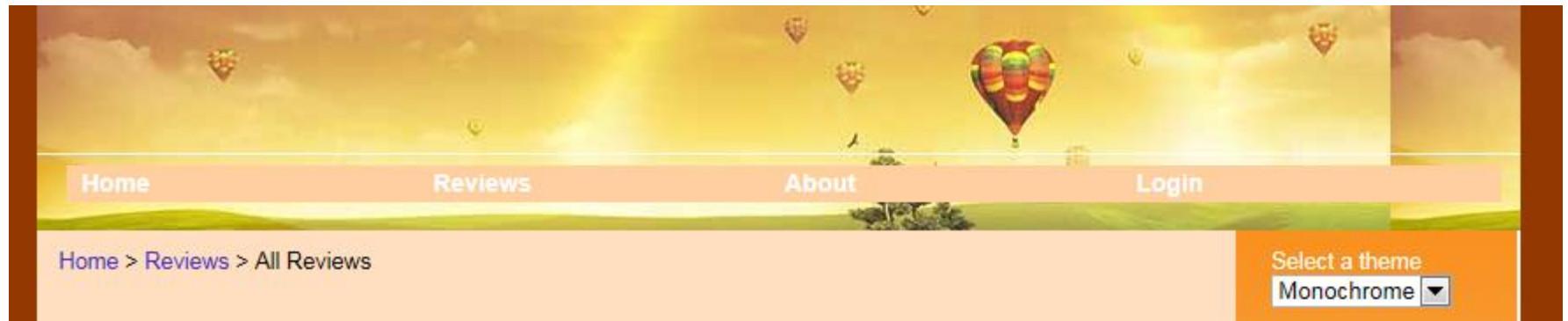
- Open the master page → **Frontend.master**
- Drag a **SiteMapPath** in **MainContent**

```
<div id="MainContent">  
    <asp:SiteMapPath ID="SiteMapPath1" runat="server">  
    </asp:SiteMapPath>  
    <asp:ContentPlaceholder id="cpMainContent" runat="server">  
  
    </asp:ContentPlaceholder>  
</div>
```

```
<div id="MainContent">  
    <asp:SiteMapPath ID="SiteMapPath1" runat="server">  
    </asp:SiteMapPath> <br/> <br/> ← Add two line breaks  
    <asp:ContentPlaceholder id="cpMainContent" runat="server">  
  
    </asp:ContentPlaceholder>  
</div>
```

Example - SiteMapPath Control

View in browser



Programmatically Redirecting

- `Response.Redirect`
 - “302” HTTP status code
 - Page has moved **temporarily**
- `Response.RedirectPermanent`
 - “301” HTTP status code
 - Page has moved **permanently**

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.RedirectPermanent("Default.aspx");
}
```


Query String

- Pass values from one page to another.
- Name/Value pairs
 - Separated from another by &



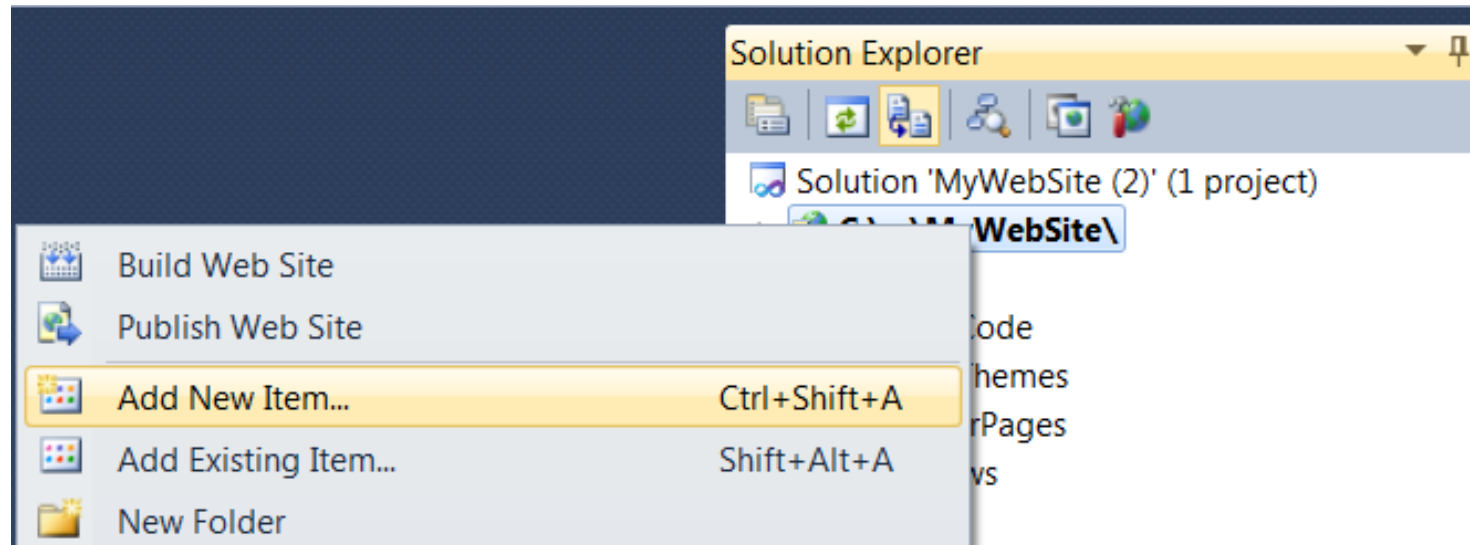
The diagram shows the URL `http://mysite.com/login.aspx?name=Alan&surname=Turing` in green text. A blue arrow points from the ampersand (&) in the query string up to the text "Separated from another by &". Two red brackets are positioned below the query string: one under "name=Alan" and another under "surname=Turing", indicating the two separate name/value pairs.

`http://mysite.com/login.aspx?name=Alan&surname=Turing`

Example

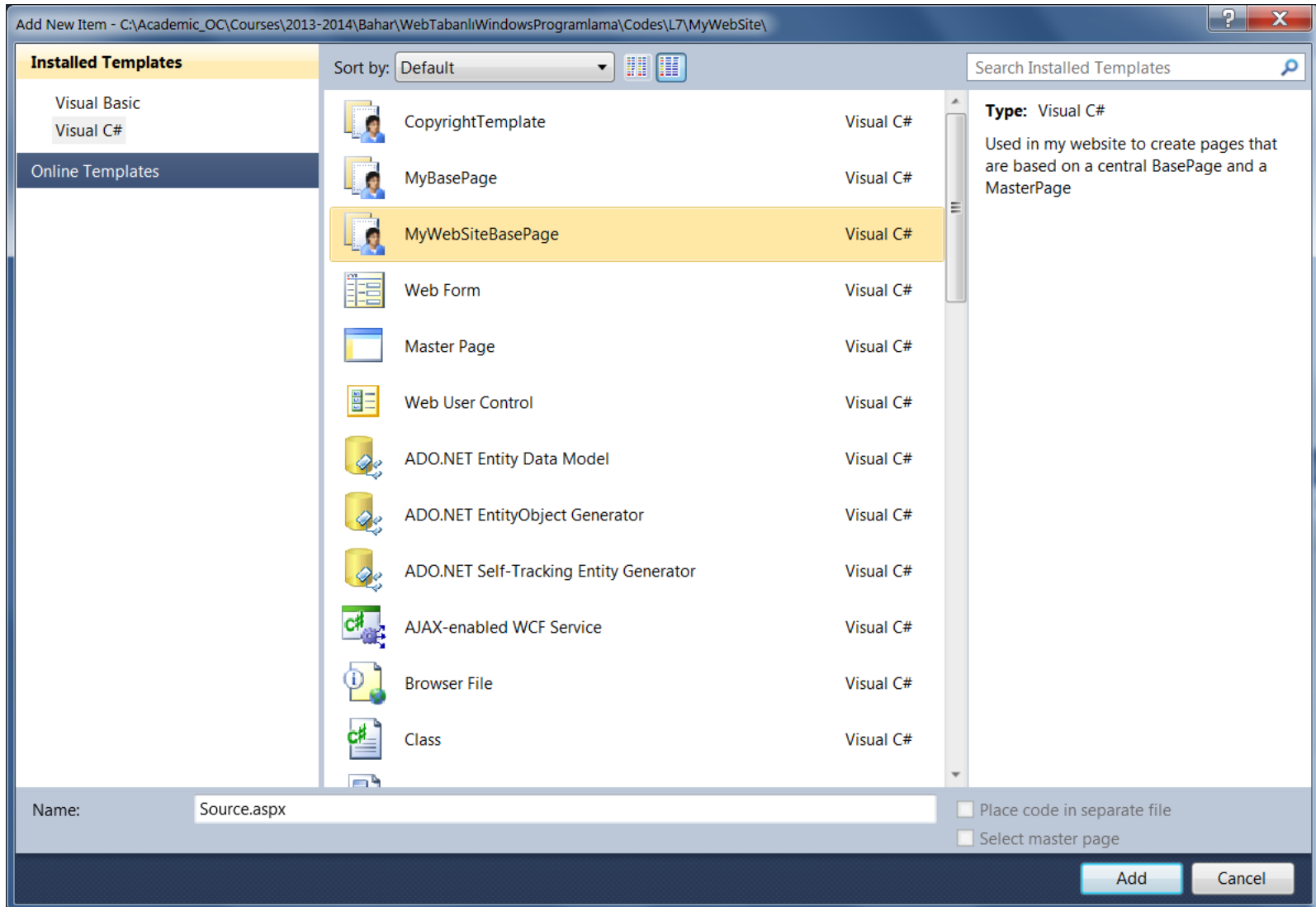
Redirecting the User to Another Page

- Add New Item



Example

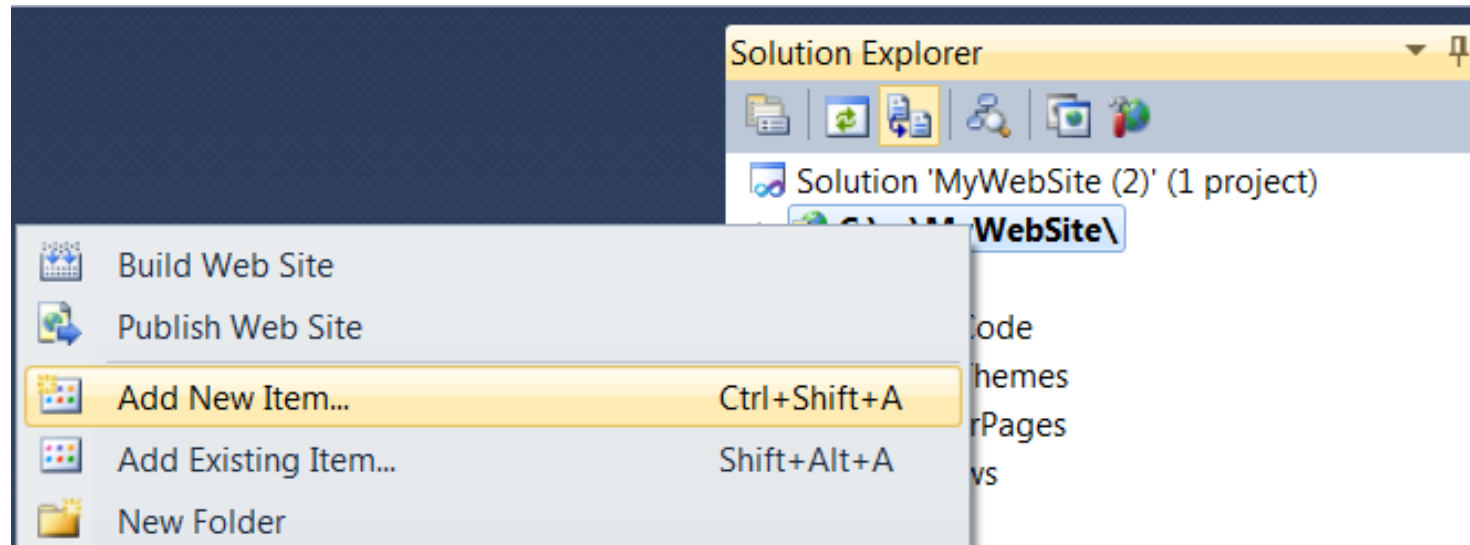
Redirecting the User to Another Page



Example

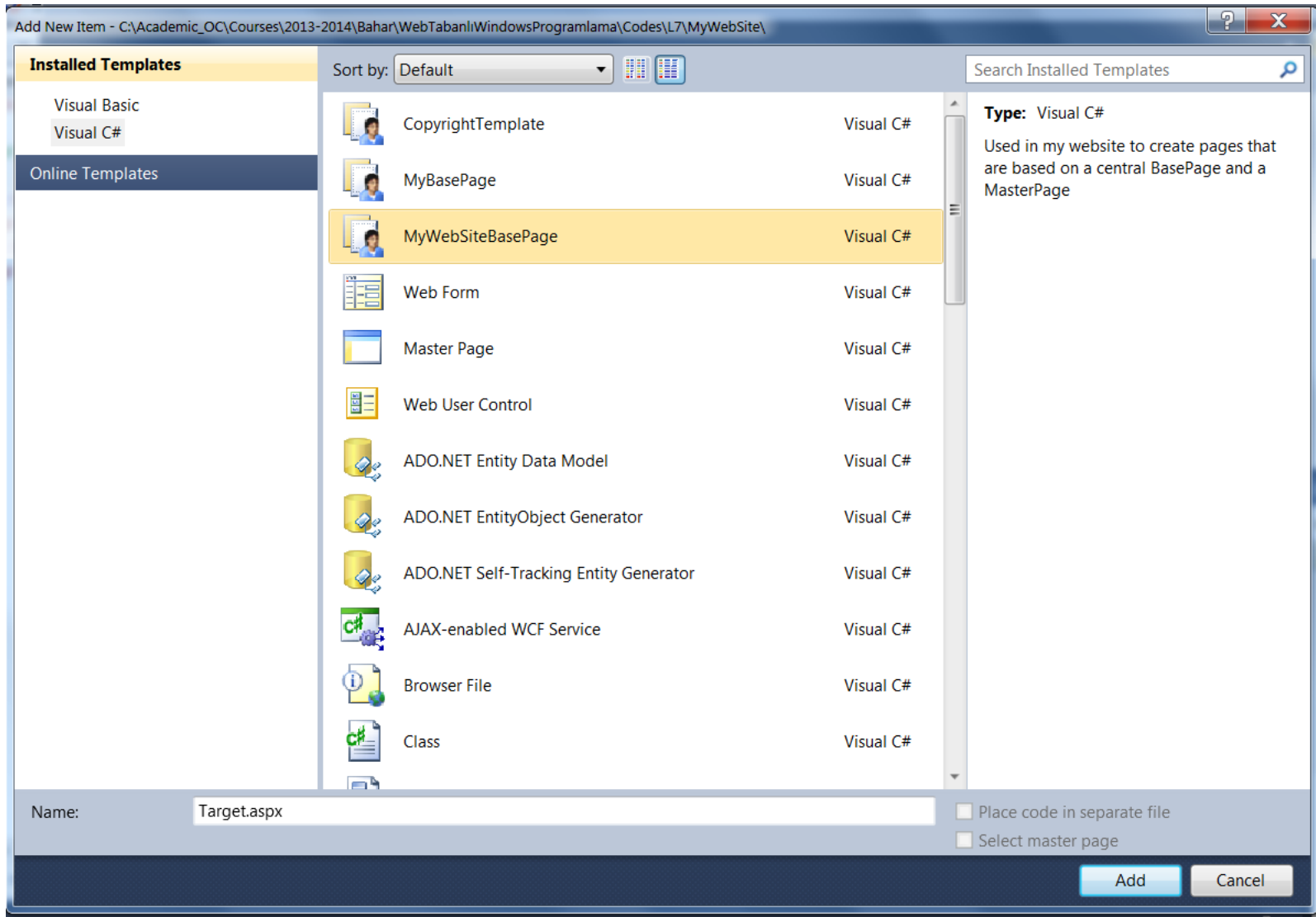
Redirecting the User to Another Page

- Add New Item



Example

Redirecting the User to Another Page



Example

Redirecting the User to Another Page

- Set their **Title** to **Source** and **Target**, respectively.
- Open **Source.aspx** in Design View.
- Double-click somewhere in the grey, read-only area of the page outside the **ContentPlaceHolder** to set up a **Page_Load** handler.

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Redirect("Target.aspx ?name=Jane");
}
```

Example

Redirecting the User to Another Page

- Open **Target.aspx** in Design View.
- Add a **Label** control



Example

Redirecting the User to Another Page

- Double-click the grey area:

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Text=Request.QueryString["name"];
}
```


Example

Redirecting the User to Another Page

View `Source.aspx` in browser



The address bar → `http://localhost:63979/Target.aspx?name=Jane`

Server-Side Redirect

- Send out a different page without modifying the client's address bar.
- Enables you to **hide details of page names and query strings**.
 - Cleaner URLs from a user's point of view.
- Speed up your site.

Server-Side Redirect

- **Server** object
 - Get information about the server name, IP address, and so on.
 - **Server.Transfer**
 - **Redirect** to other pages **within** your site.
 - **Cannot use** to send the user to pages on different domains.

Example

Server-Side Redirecting

- Open `Source.aspx.cs`

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Redirect("Target.aspx?name=Jane");
}
```



```
protected void Page_Load(object sender, EventArgs e)
{
    Server.Transfer("Target.aspx?name=Jane");
}
```

Example

Server-Side Redirecting

View `Source.aspx` in browser



The address bar → `http://localhost:63979/Source.aspx`