



School of Information Technology

---

**Practical 03: Master Page & Navigation**

**OBJECTIVES:**

By the end of this Practical students should be able to:

- Create a server based web application with consistent layout for multiple web pages using Masterpage.

## Introduction

### What is Master Page

ASP.NET supports Master Pages, in which you create a common base master file that provides a consistent layout for multiple pages in your application. A single master page defines the look and feel and standard behaviour that you want for all of the pages (or a group of pages) in your application. In ASP.NET, this template is called a Master Page.

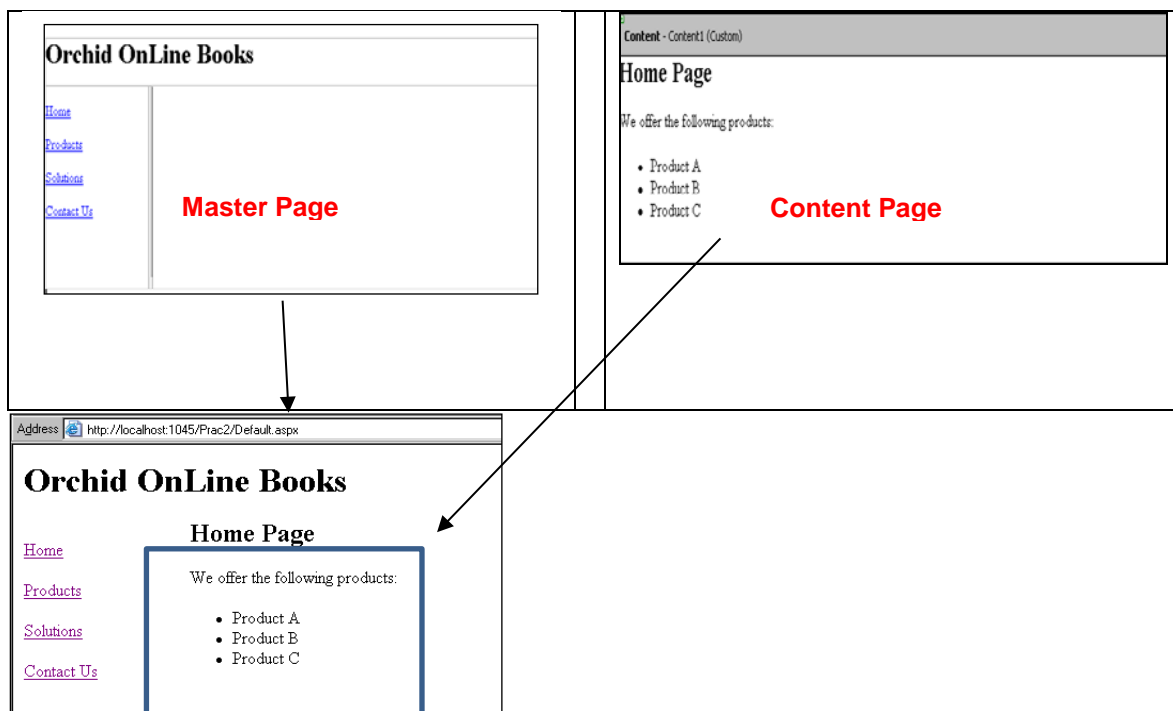
You may refer to MSDN for more information on Master Page:

[http://msdn.microsoft.com/en-us/library/wtxbf3hh\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/wtxbf3hh(v=vs.100).aspx)

A professional looking web site will typically contain features such as a common header, menu system for the entire site, and a common footer providing copyright information. These elements are present on every page and should have a consistent layout, and look and feel. Instead of repeating the codes necessary to maintain this consistency across all pages of a website, it is better if we can use a template that all pages can reference to. This way, any changes made to the website design need to be made on the template only.

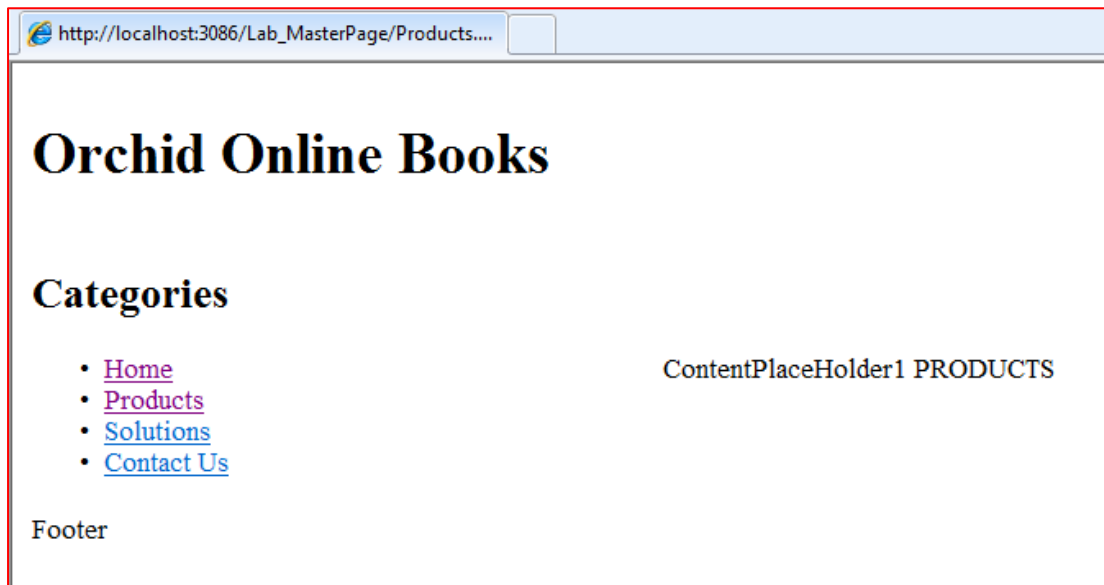
A Master Page defines the layout used by all pages of the web site. Regions that are suppose to contain the contents of the different web pages are indicated by ContentPlaceHolders controls.

Each Web Form (.aspx pages) reference to a Master Page. This reference is coded into the *MasterPageFile* property in the Page Directive. Each Web Form has one or more *ASP.NET Content* controls that correspond to each of the *ContentPlaceHolders* controls in the Master Page. This is where contents such as text, HTML and other ASP.NET controls are added. When the Web Form is requested, ASP.NET will map the contents of the *Content* controls into the respective *ContentPlaceholder* controls of the Web Page to create a complete web page and send that to the client as a complete web page.



## Exercise 1

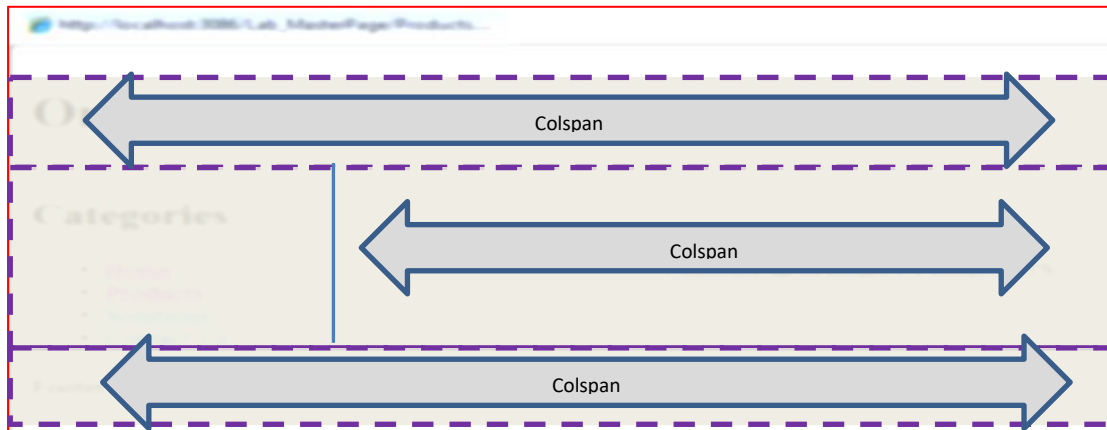
Create 4 Web pages and attached them with a MasterPage.



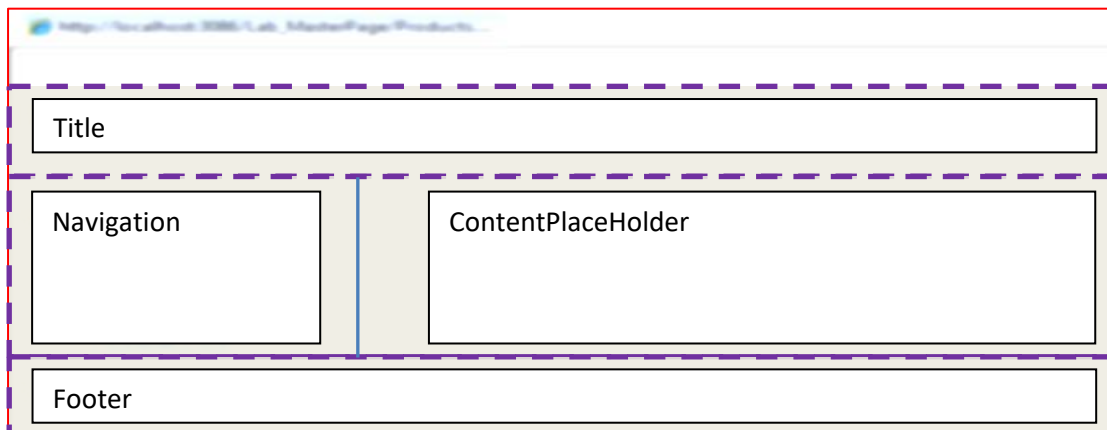
HTML Table will be used for designing the layout of the masterpage.



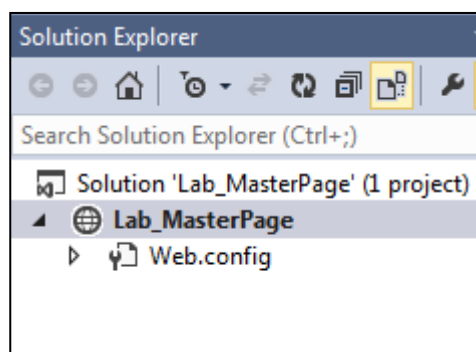
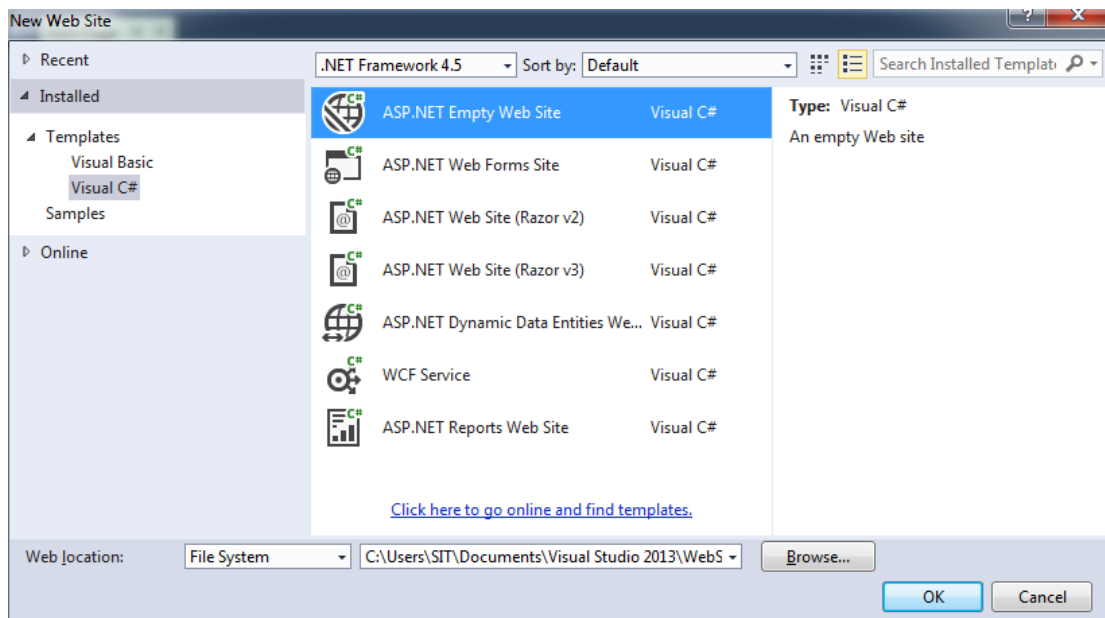
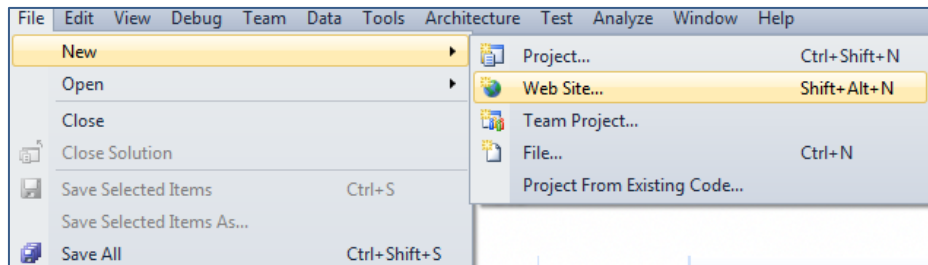
Apply 'Column Span' to combine some of the columns.



Final outcome.

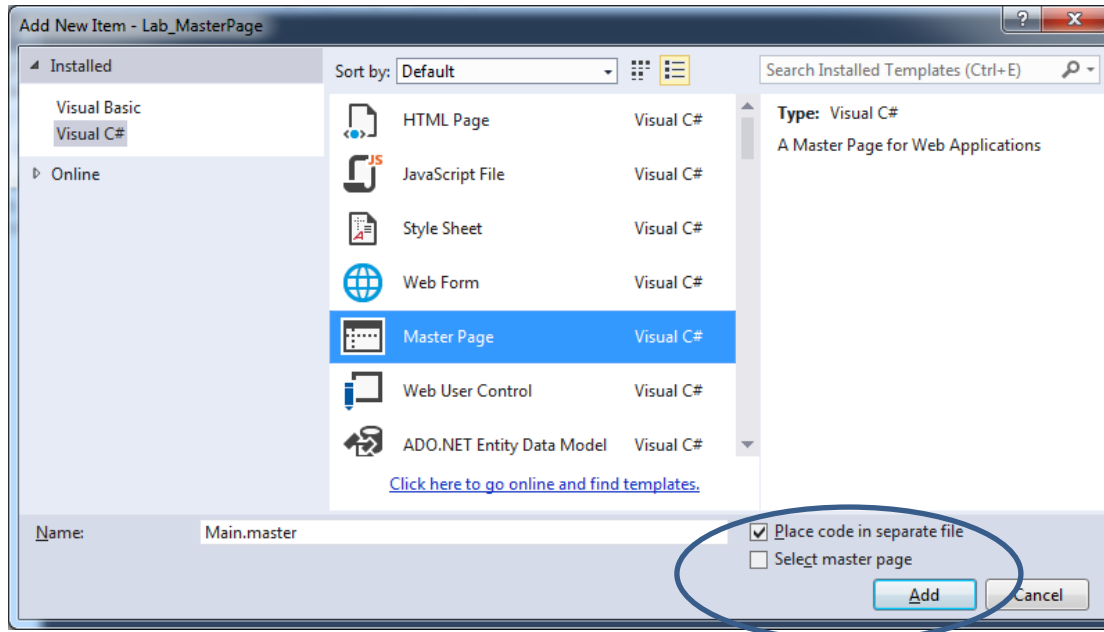


1. Run Visual Studio 2015.
2. Create a new **ASP.NET EMPTY Web Site** and name it **Lab\_MasterPage**.
3. From menu : Select New → "Web Site" → "ASP.NET Empty Web Site". Choose Visual C# under the "installed Templates".



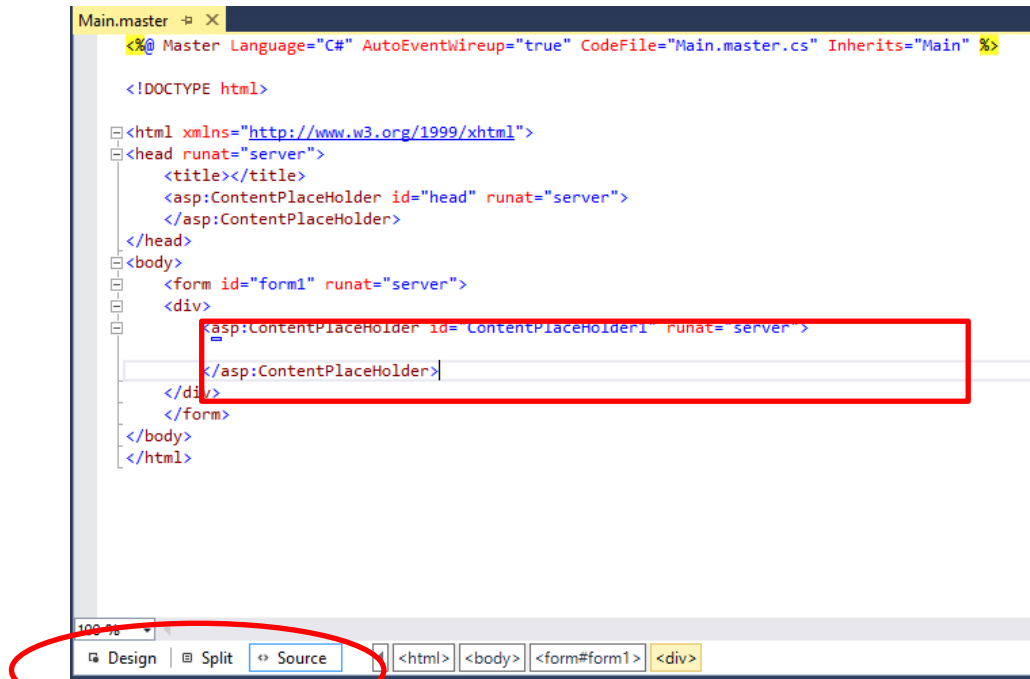
If the operation is successful, a project folder will be created together with a standard set of files. At this point, there are not files for us to work on. We will need to create a webform and a masterpage to get started.

4. Let's create our very own MasterPage and web pages. In ASP.NET each webpage is equivalent to a "Web Form". To create a new Web page, we need to create a new "Web Form" in ASP.NET.
5. Add a Master Page to your existing project. In Menu, Choose Website → Add New Item → MasterPage (Check that you have chosen "Visual C#" as the main language)
  - Name it *Main.master*.
  - Check the "Place code in separate file". This means to enable code behind.



6. Examine Main.master in Source View by double-clicking on "Main.master" from the Solution Explorer. Take note that there are 3 different views (Design, Split and Source)

By default, Main.Master creates a ContentPlaceHolder for you to display any of your Web Form (your Web Form must use Main.Master in the first place). If this is not the place you intend to display your web form you may need to reposition it.

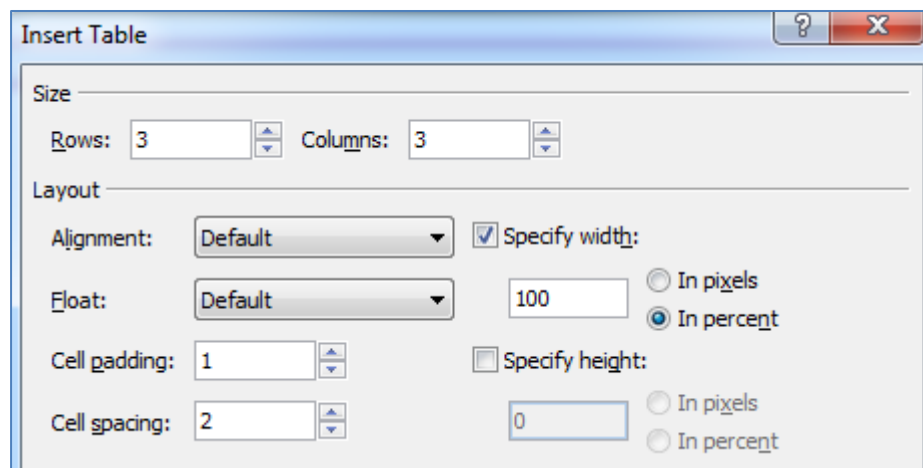
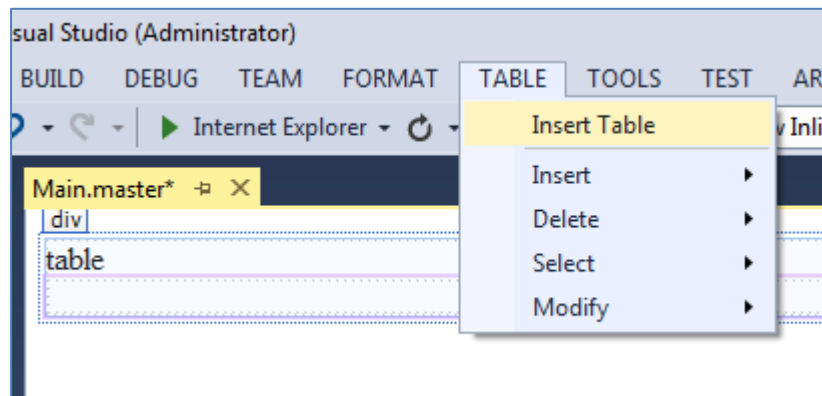


7. In the same Source view, place your mouse cursor immediately after the <div> tag and create a new empty line by hitting on the "Enter" button of your keyboard. Type the word "table" as shown in the figure below.

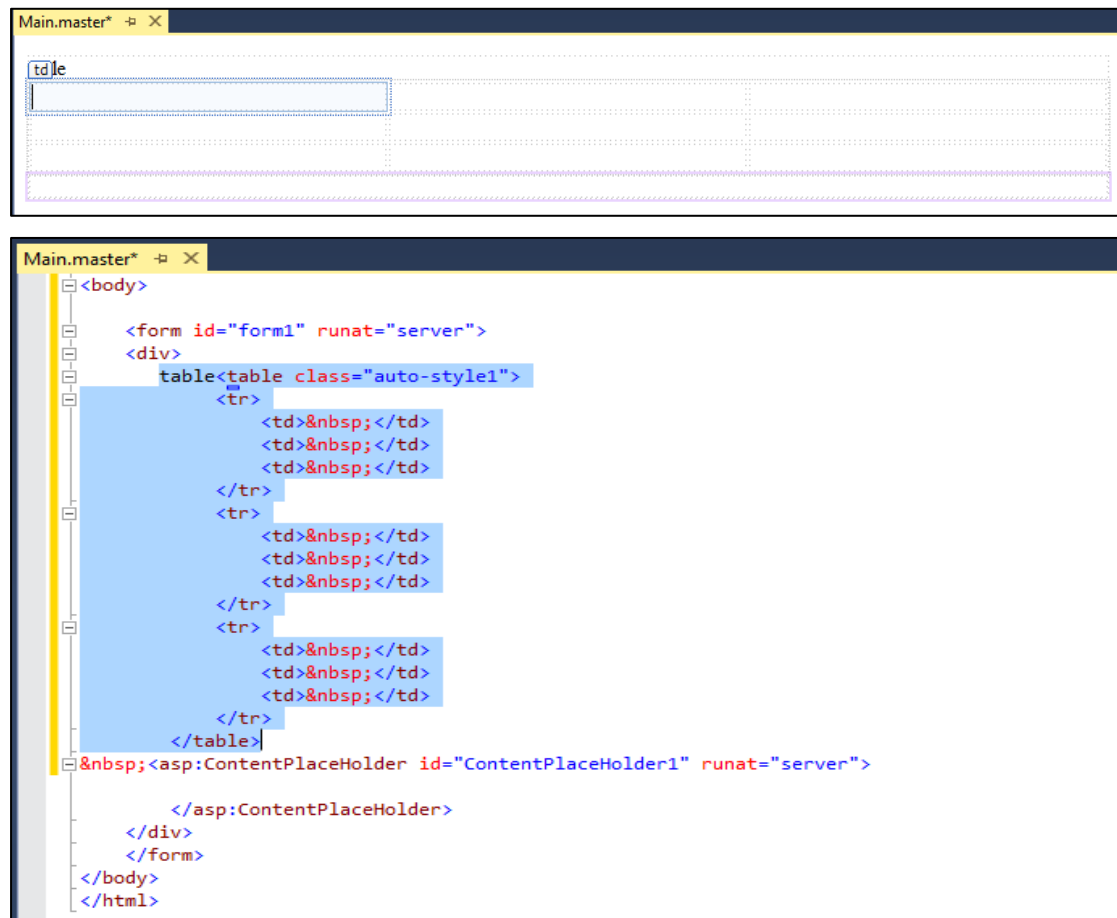


8. Switch to Design view and insert a Table into the area. Create 3 rows and 3 columns.

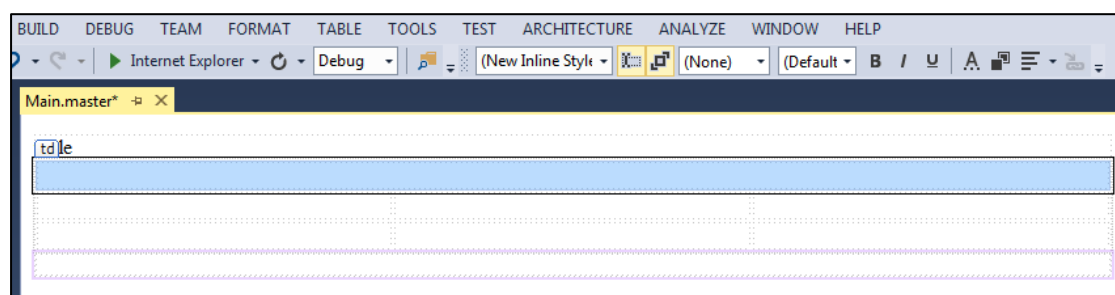
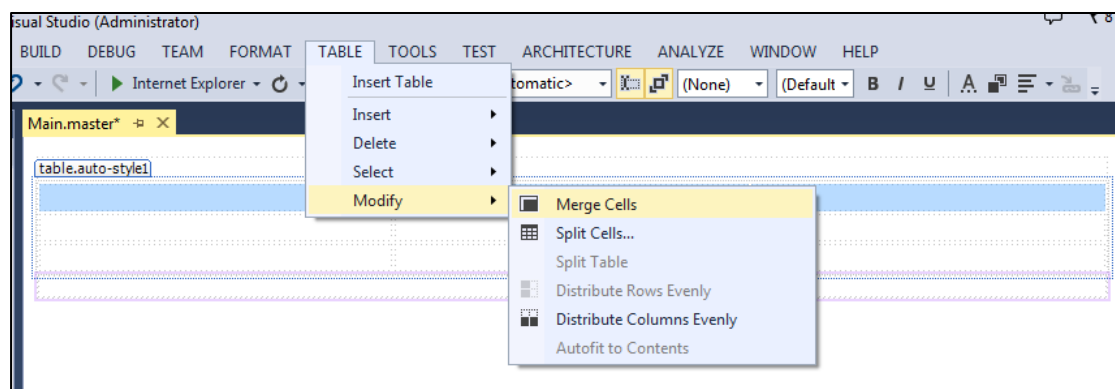
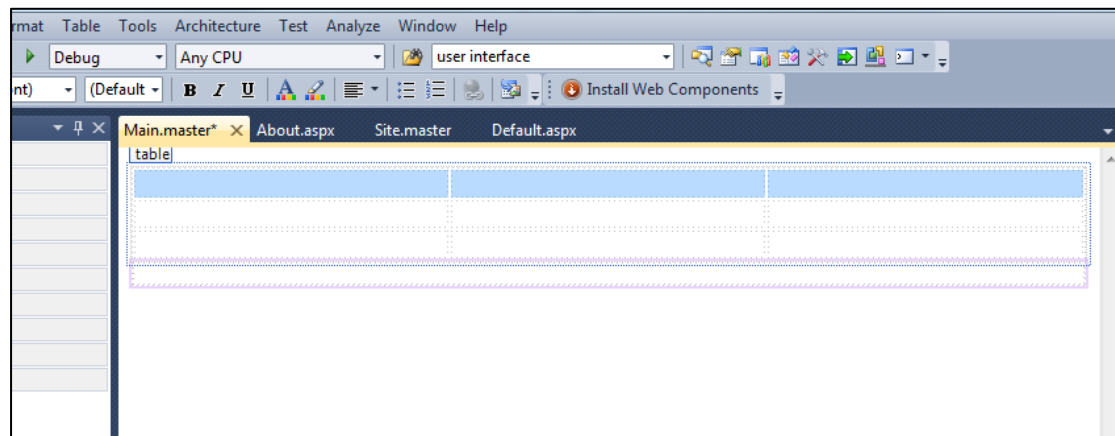




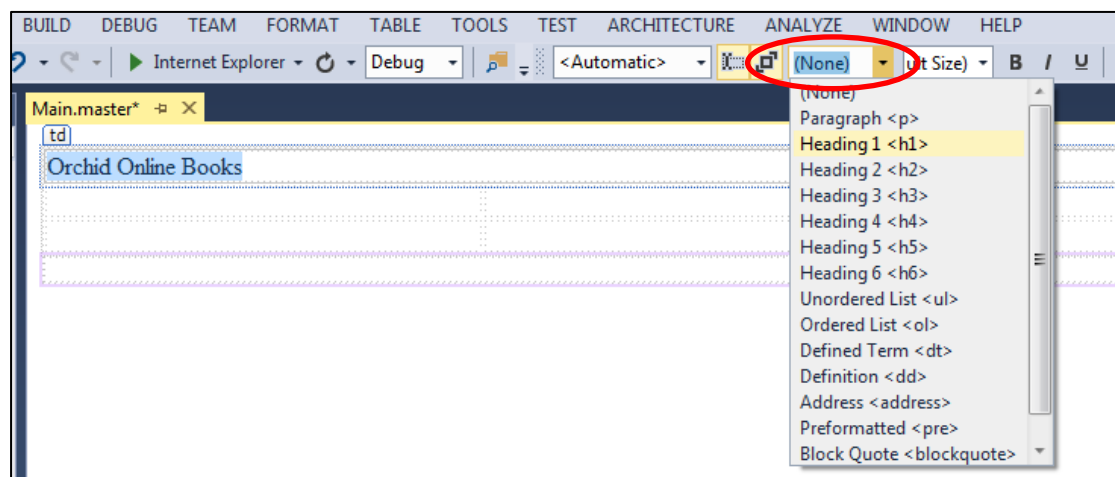
9. The moment the table is dropped, VS will generate a HTML table onto the space. A series of HTML Table codes will be generated. Switch back to “Source view” to examine the codes.



10. Take note that the tables should be created outside the <asp:ContentPlaceholder ...>
11. Switch to "Design View". Select the 1<sup>st</sup> row of the table. Merge all the cells in from the 1<sup>st</sup> row. In Menu, select Table → Modify → Merge Cells



12. Enter the text into the merge cells "Orchid Online Books". Set it to Heading 1 <h1>. At the same time remove the word "Table" just above the HTMLTable.



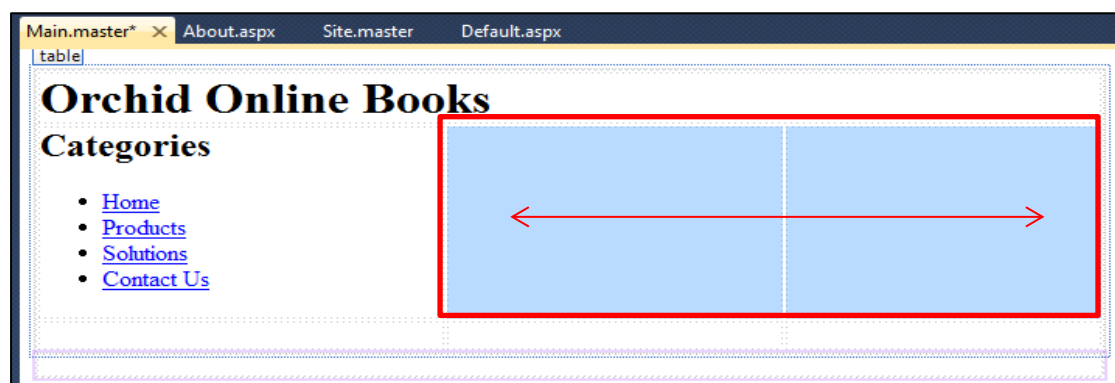
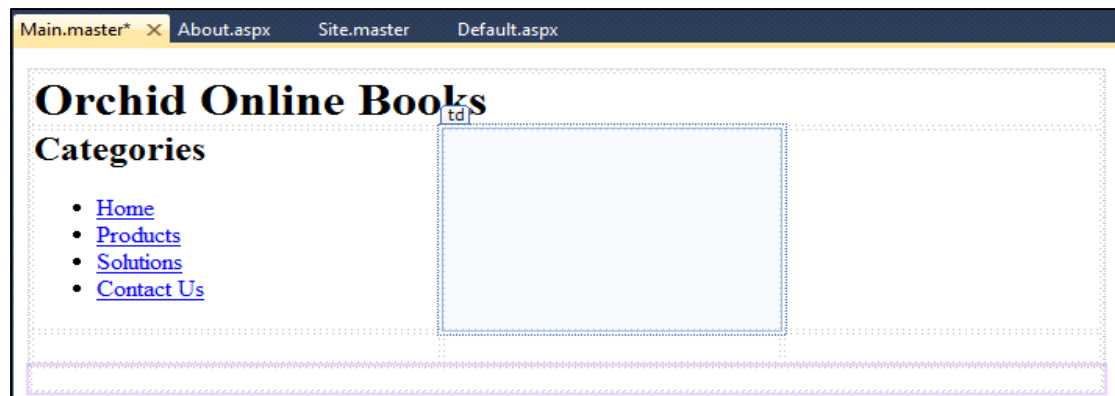
13. Switch to Source View.

14. Enter the HTML codes into the space shown.

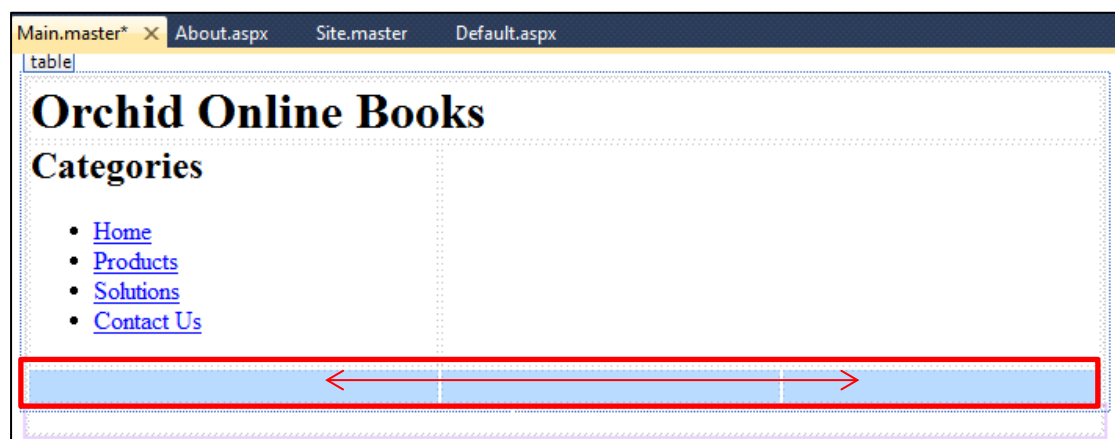


```
<ul>
<h2>Categories</h2>
<ul><li><a href="Default.aspx">Home</a></li>
<li><a href="Products.aspx">Products</a></li>
<li><a href="Solutions.aspx">Solutions</a></li>
<li><a href="ContactUs.aspx">Contact Us</a></li>
</ul>
```

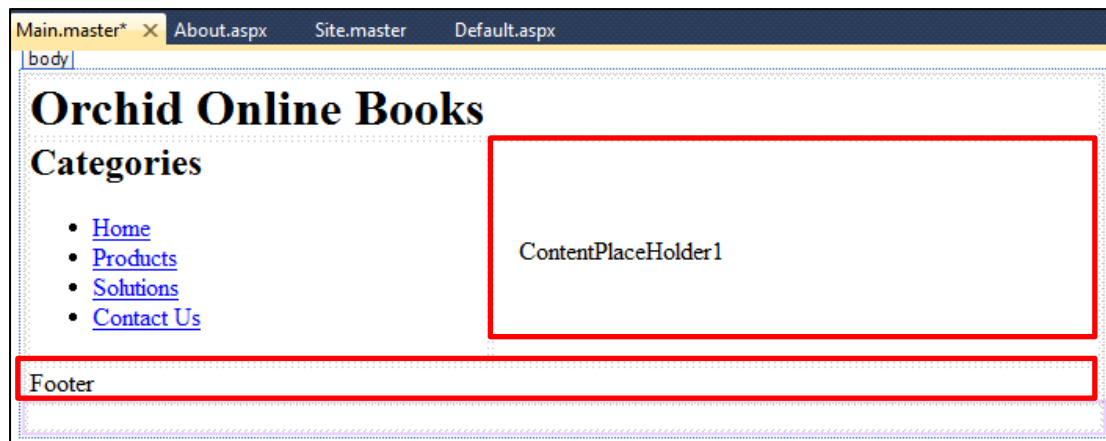
15. Switch to Design view to check the output. Shift the columns if necessary.



16. Merge the Cells.



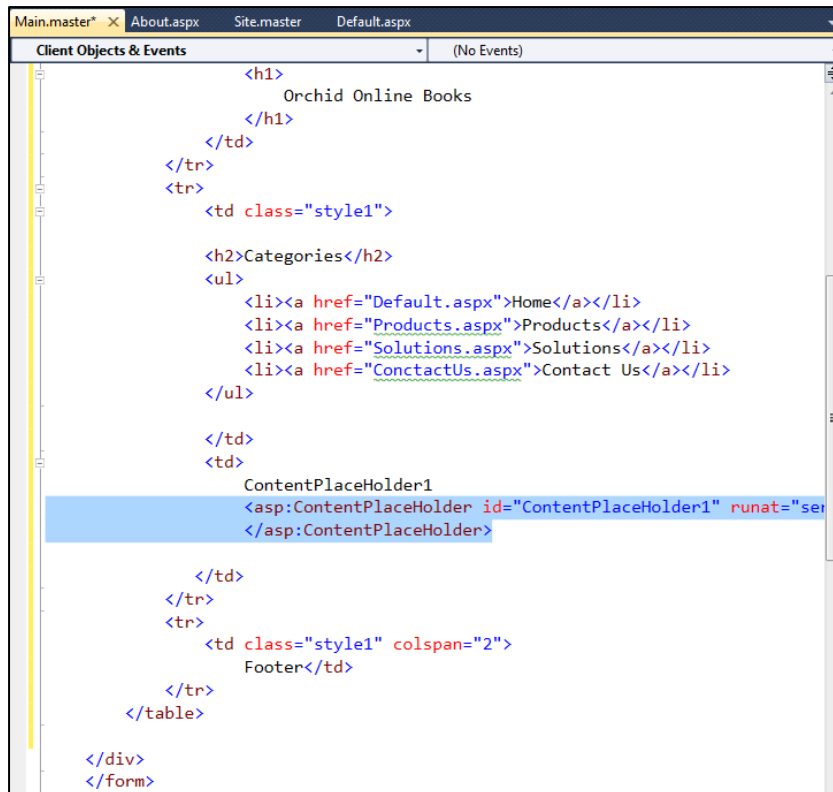
17. Type the text into the merged cells.



18. Switch to Source View. Move `<asp:ContentPlaceHolder id="...">` to the end of the text "ContentPlaceHolder1".



19. Final source code should look like this :



```

<h1>
    Orchid Online Books
</h1>
</td>
</tr>
<tr>
<td class="style1">

    <h2>Categories</h2>
    <ul>
        <li><a href="Default.aspx">Home</a></li>
        <li><a href="Products.aspx">Products</a></li>
        <li><a href="Solutions.aspx">Solutions</a></li>
        <li><a href="ConctactUs.aspx">Contact Us</a></li>
    </ul>

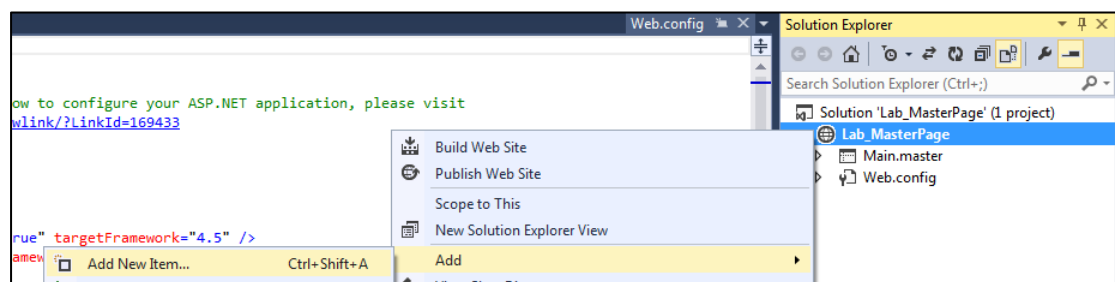
    </td>
    <td>
        ContentPlaceholder1
        <asp:ContentPlaceholder id="ContentPlaceholder1" runat="server">
        </asp:ContentPlaceholder>
    </td>
</tr>
<tr>
<td class="style1" colspan="2">
    Footer</td>
</tr>
</table>

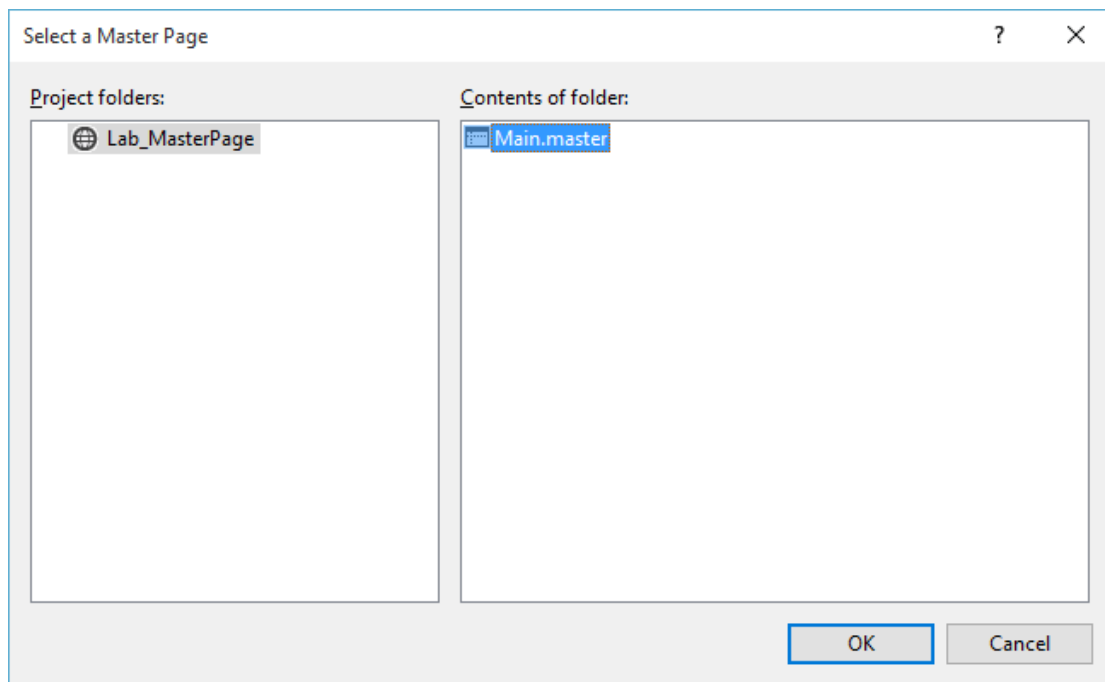
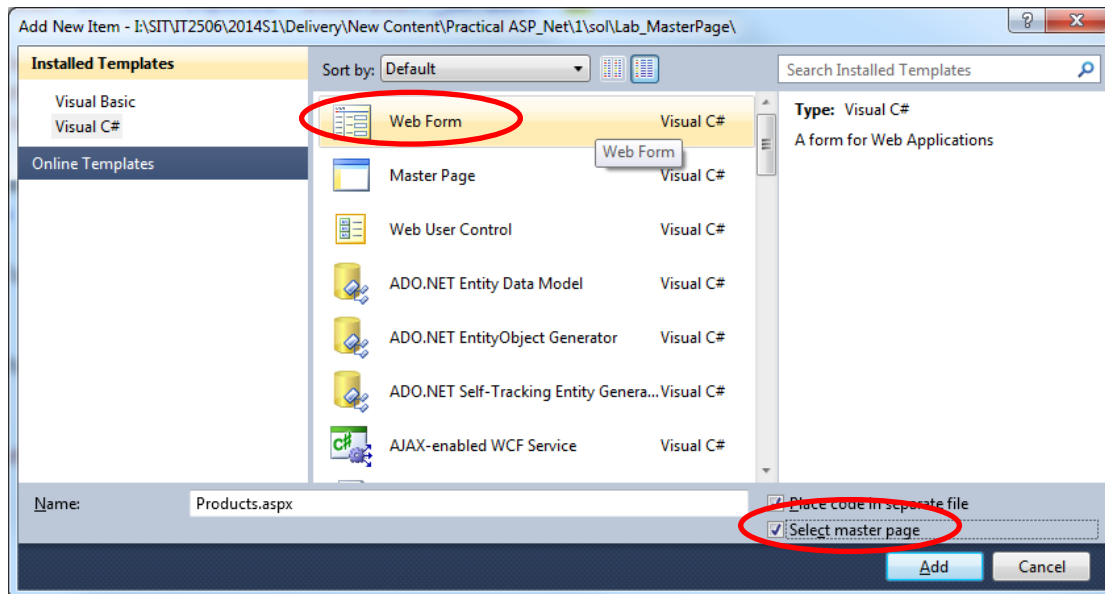
</div>
</form>

```

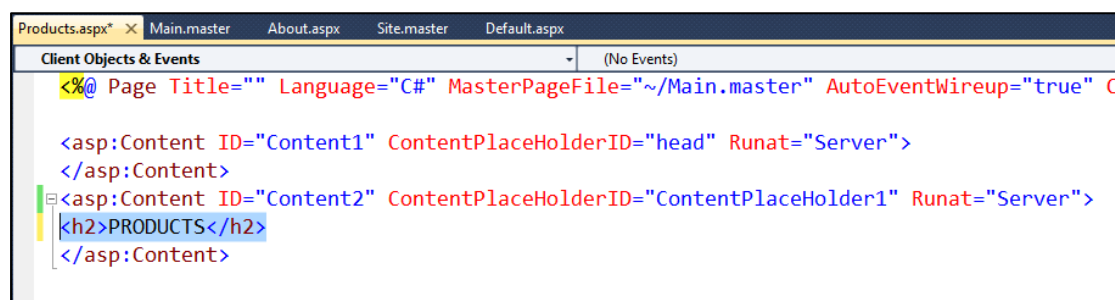
20. Create the 4 Web Forms and choose Main.Master as the MasterPage.

21. Right-click "Project Name" select "Add new item". Choose "Web Form" and choose "Select master page". Alternatively, you may choose from the Menu "Web site" → Add New Item → Web Form.



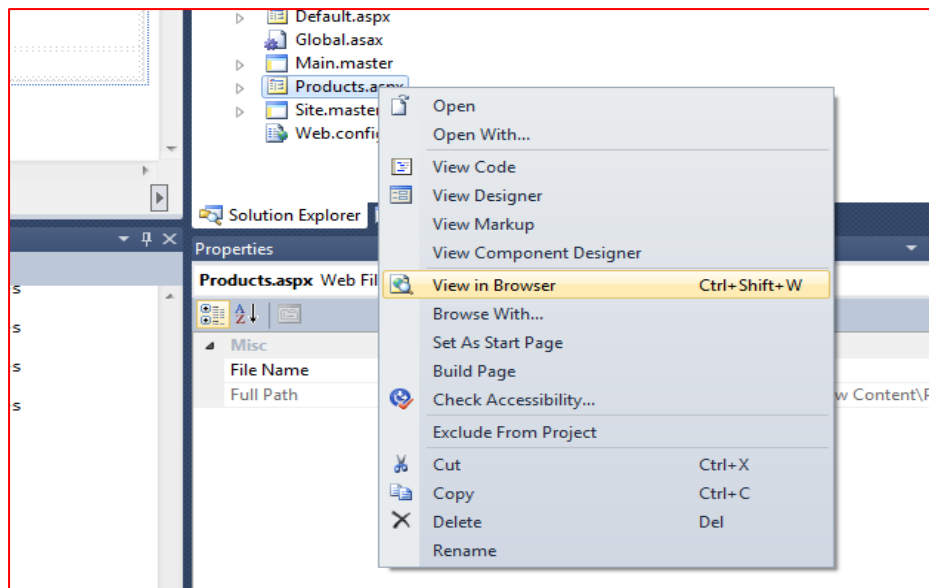


22. Type the text PRODUCTS as shown in Products.aspx. The content in the "ContentPlaceHolder1" will appear in the masterpage. Test it out.

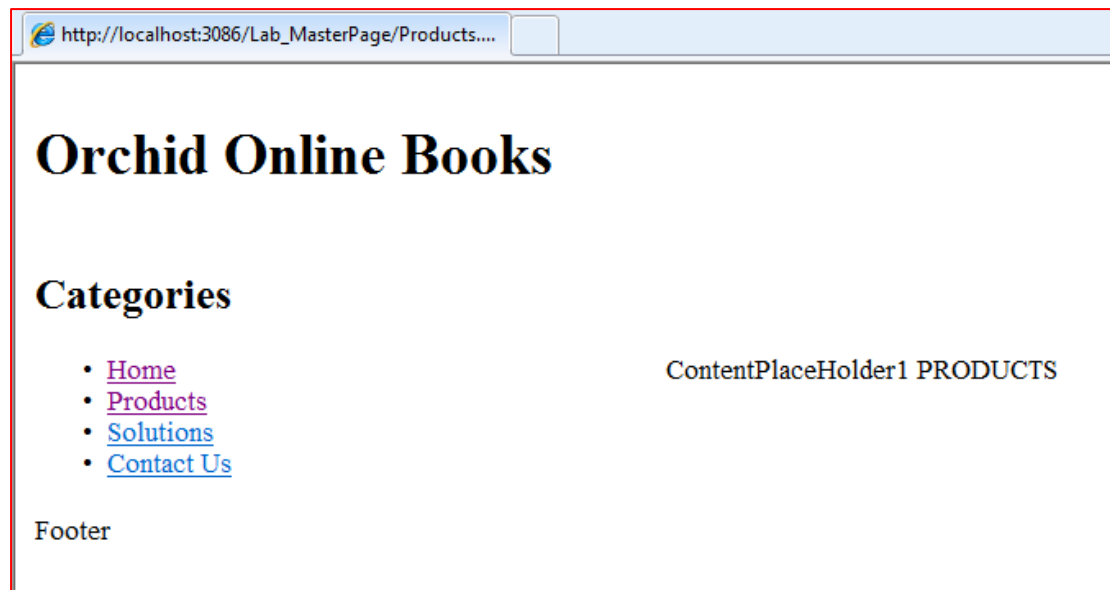




23. Test run your Products.aspx. Right-click “Products.aspx” → select “View in Browser”.



24. Output of the result.



25. Continue to create the 3 other Web Forms : **Solutions.aspx** and **ContactUs.aspx** and **Default.aspx**.

26. Set Default.aspx as the default start page for the entire website. Right-click “Default.aspx” and choose “Set as start page”.

27. Since Default.aspx is the default **Start page**, you can hit F5 to test run your application. (To change any Web Form to be the start page, right-click the Form name and select “Set as Start page”.)

28. You are now ready to create own very own MasterPage. Your MasterPage may not necessary be similar to our example. Be creative!

End of Practical