

**Visual Training**

**Of**

**Microsoft Visual Studio2015**

**Web based Applications using ASP.Net with MS Visual C-  
Sharp**



**Visual Training**

**Of**

**Microsoft Visual Studio2015**



# **Web based Applications using ASP.Net with MS Visual C-Sharp**



**Chapter No.1**

**Overview of ASP.Net**

**and**

**Visual Studio 2015**

**DAY1**



**In today's session:**

- ✓ **Opening MS Visual Studio 2015**
- ✓ **Creating First Web Site**
- ✓ **Differentiation of .aspx & .aspx.cs file**
- ✓ **Brief Overview of the Different Menu Options available**
- ✓ **Overview of Solution Explorer, Property Windows, Tool box etc**
- ✓ **Basic HTML**
- ✓ **Sample HTML Tags**
- ✓ **Writing some sample HTML Tags**
- ✓ **Installing IIS**
- ✓ **Running the Test web Site**

## 1.1 Overview of Visual Studio 2015

**Microsoft Visual Studio** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silver light. It can produce both native code and managed code.

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010[9]). Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) was supported in the past.

## 1.2 Steps for Creating a New Web Site in VS 2015

After you successfully installed the Visual Studio 2015, open Visual Studio. When you open Visual Studio 2015, you will see the following window

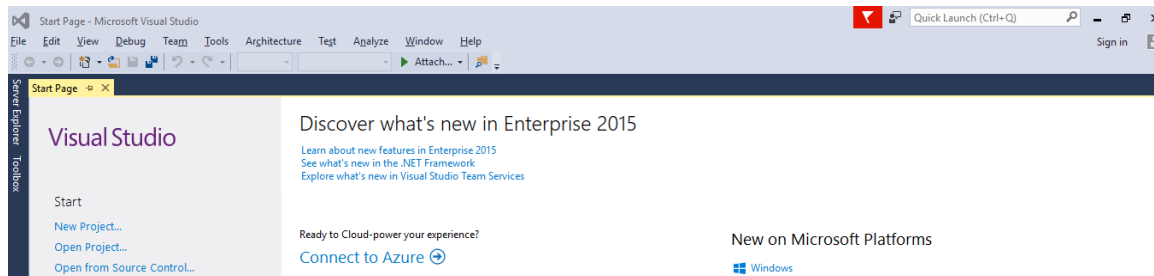


Figure 2.1

Next step is to Create a web site in Visual Studio. Choose

File → New → Web Site as shown in the next figure

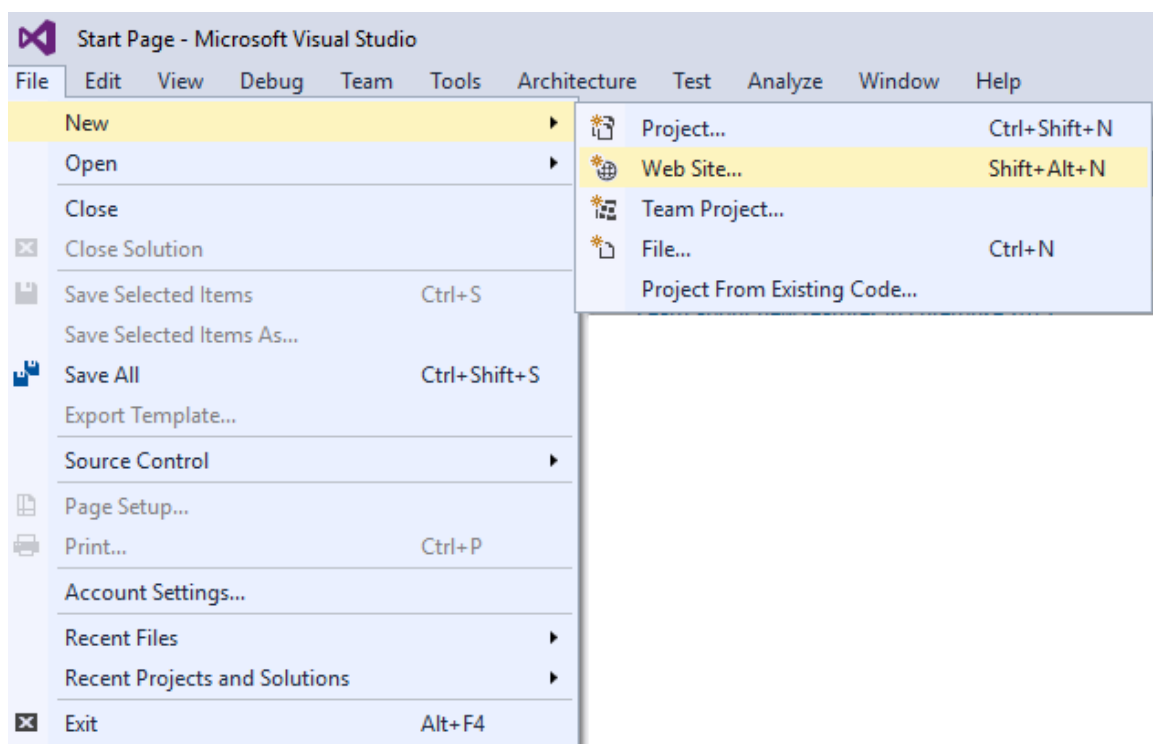


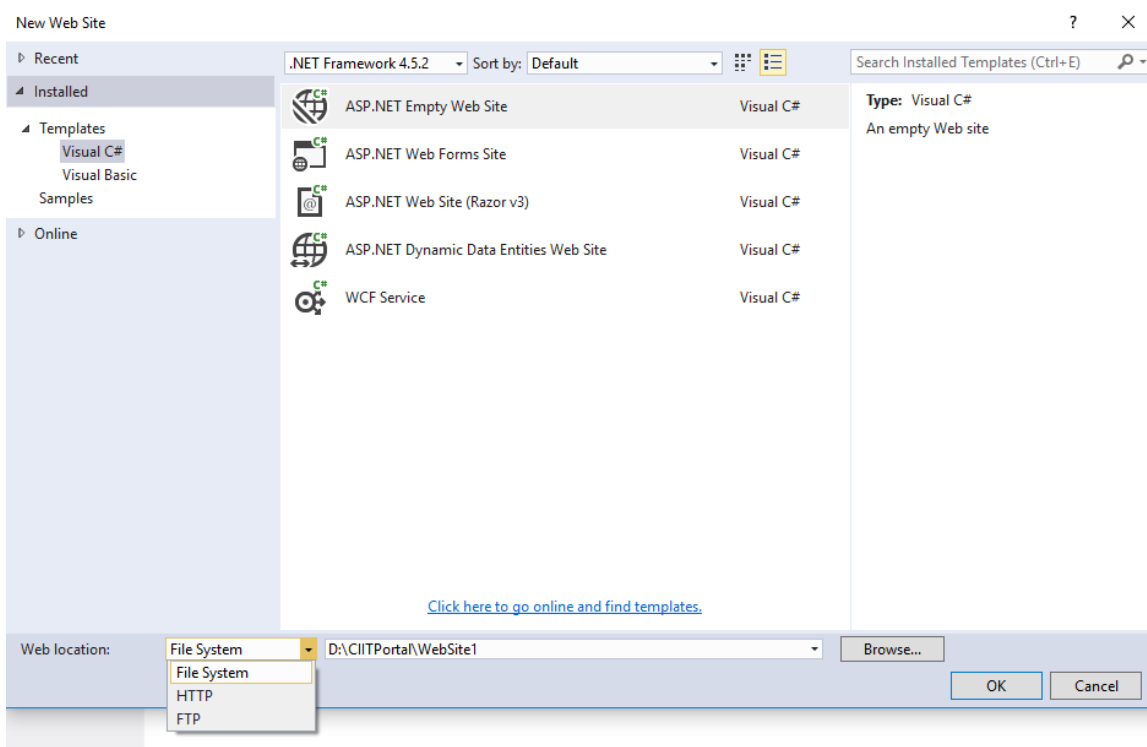
Figure 2.2



When you choose New Web Site, then Visual Studio allow developers to create the web site of different types. These different options are discussed in Table 2.1.

Type of Web Site	Description
ASP.NET Empty Web Site	This template creates an ASP.NET website that includes a Web.config file but no other files. Use this project template when you do not require the functionality built into the standard.
ASP.NET Web Site (Razor)	Use this project template to create a website that uses ASP.NET Web Pages with the Razor syntax. For more information,
WCF Service	Use this project template to create a Windows Communication Foundation (WCF) service. You can extend the service so that a website, web application, Silverlight application, Ajax application, or other client can call it.
ASP.NET Dynamic Data Entities Web Site	Use this project template to create a Dynamic Data website that uses the ADO.NET Entity Framework. This type of Dynamic Data web application can target any relational database.
ASP.NET Web Forms Site	<p>Use this project template to create a website that includes the following functionality. You can choose not to use any of these features when they are not required for your application.</p> <ul style="list-style-type: none"> <li>• A master page.</li> <li>• A cascading style sheet.</li> <li>• Login security that uses the ASP.NET membership system.</li> <li>• Ajax scripting that uses jQuery.</li> <li>• Navigation using a menu control.</li> </ul> <p>By default, the ASP.NET Web Site project template includes the following:</p> <ul style="list-style-type: none"> <li>• Folders to contain membership pages, client script files, and cascading style sheet files.</li> <li>• A data folder (App_Data), which is granted permissions that allow ASP.NET to read and write to it at run time.</li> <li>• A master page (.master file).</li> <li>• Web pages named Default.aspx and About.aspx. In the project template, these pages are content pages that use default master page.</li> <li>• A global application class (Global.asax file).</li> <li>• A Web.config file</li> </ul>

For this tutorial, we choose the **ASP.Net Empty Web Site.** Here, developer can Select .Net Framework, the Language (VB.Net/C# or other) and Name the Web site /Directory and set the Path of the Web Site) as shown. These options are shown in the next figure



Rename the Web Site as **LearningASPNETFirstDay** and then Press OK Button.... You will see the screen shown here. By default, there is no web page is added to the web site, only web site is created with a web.config file.

### **web.config**

web.config file lets you customize the way your site or a specific directory on your site behaves. For example, if you place a web.config file in your root directory, it will affect your entire site (www.coolexample.com). If you place it in a /content directory, it will only affect that directory (www.coolexample.com/content). However, in order for a web.config file to register in a specific directory, there must be a web.config file in the root directory.

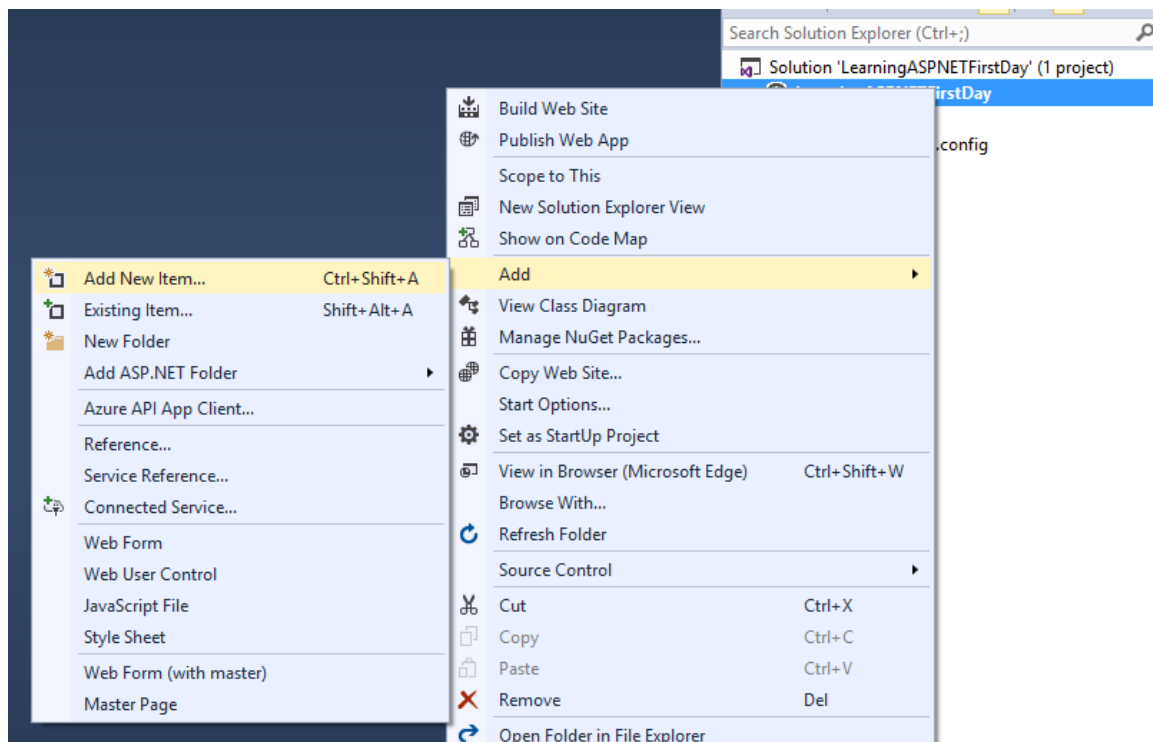
web.config files work on our Windows servers.

Using a web.config file, you can control:

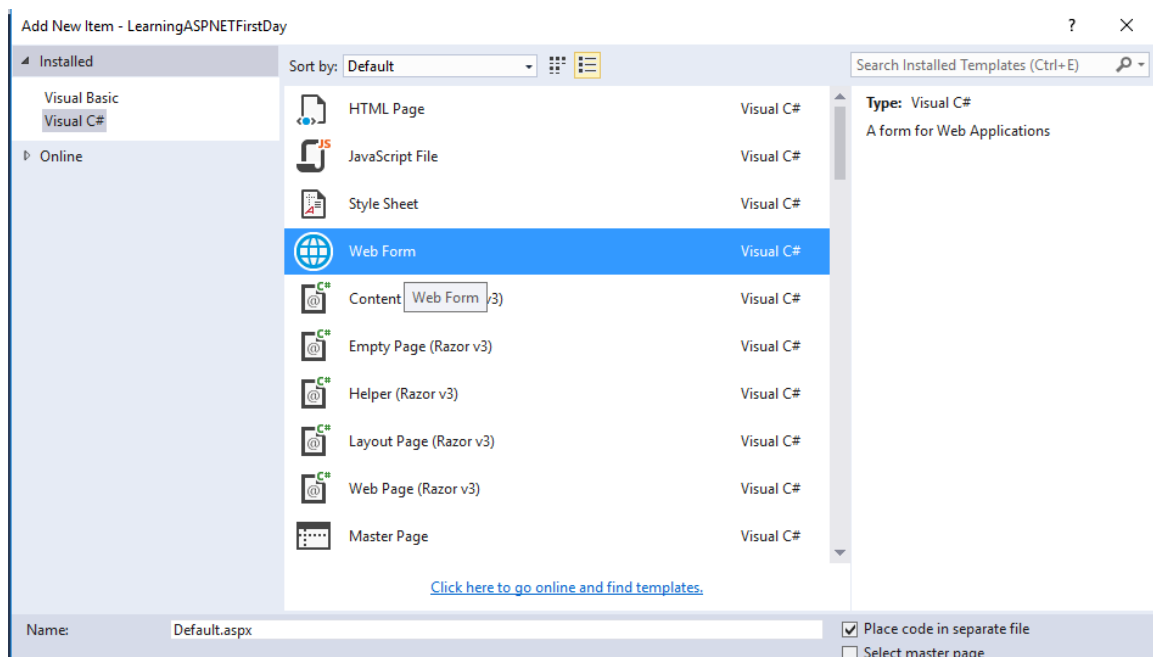
- Database connection strings.
- Error behavior.
- Security.

web.config files are XML documents. ".config" is not an extension like .html or .txt. For more information on how to set up web.config files,

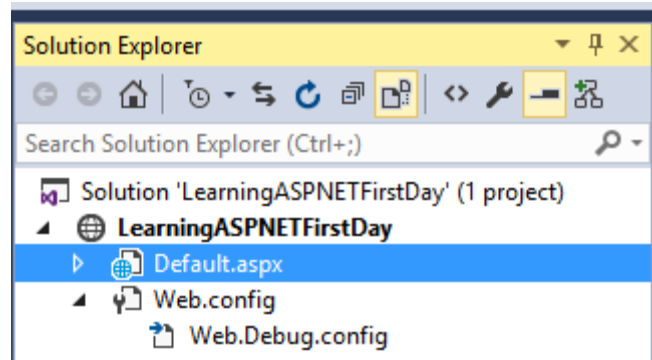
Next, add a web form to the newly created web site by Right Click on Web Site Name **LearningASPNETFirstDay**, Choose Add ➔ Add New Item as shown



From the Add New Item Dialog Window, Choose Web Form as shown



We can rename the web form. For today's tutorial, we keep it as Default.aspx. Press the Add button on the Add New Item Dialogue Window. The newly added web form is added under the web site **LearningASPNETFirstDay**, this can be seen in Solution Explorer, as shown



Each ASP.Net Web Form has two portions

<b>Default.aspx</b>	<ul style="list-style-type: none"> <li>• .aspx Page hold all of the client side code mean HTML/JavaScript Code</li> </ul>
<b>Default.aspx.cs</b>	<ul style="list-style-type: none"> <li>• .aspx.cs or just .cs page, also called the code behind, holds the server side code</li> </ul>

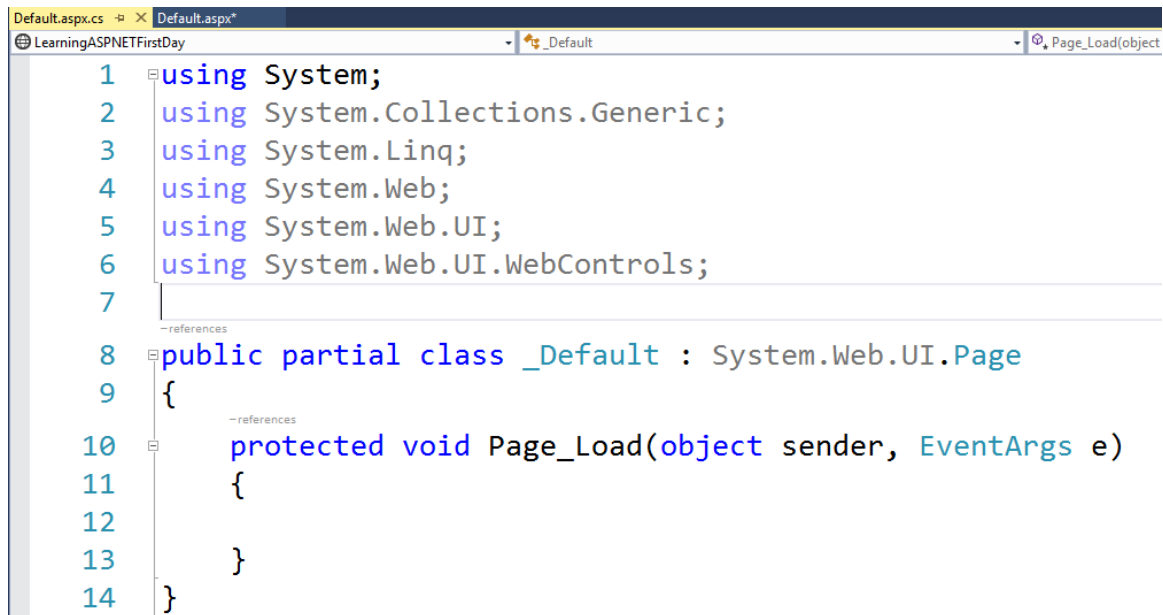
.aspx file displays the HTML code. Click on the .aspx file in Solution Explorer to view its contents

```

Default.aspx*  -> X
1  <%@ Page Language="C#" AutoEventWireup="true"
2      CodeFile="Default.aspx.cs" Inherits="_Default" %>
3
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head runat="server">
6      <title></title>
7  </head>
8  <body>
9      <form id="form1" runat="server">
10     <div>
11
12     </div>
13     </form>
14 </body>
15 </html>

```

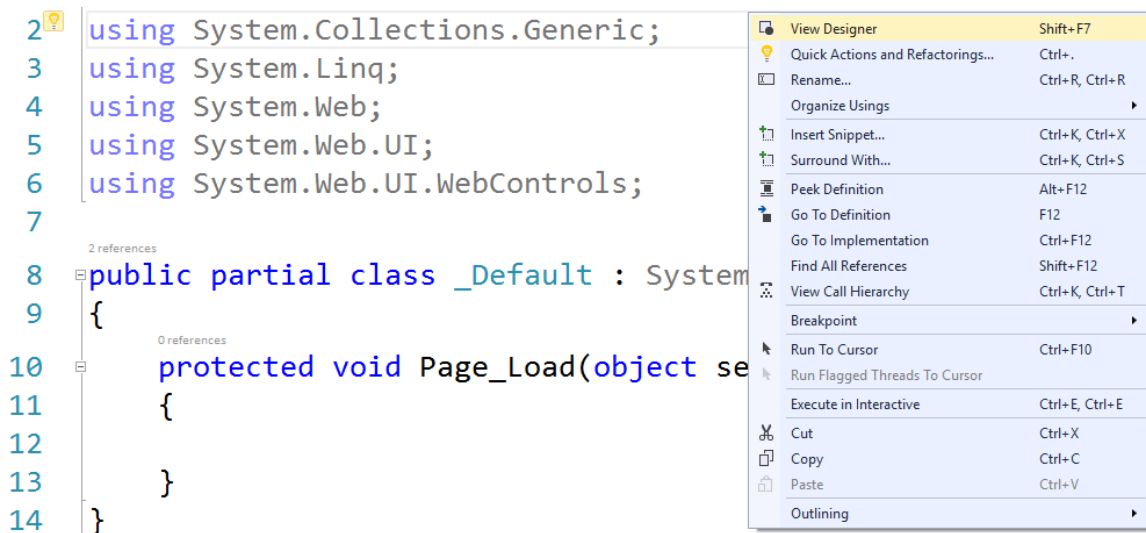
Also, Click on .aspx.cs file in Solution Explorer to open the code behind file. The contents of .aspx.cs file are shown here



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7
8 public partial class _Default : System.Web.UI.Page
9 {
10     protected void Page_Load(object sender, EventArgs e)
11     {
12     }
13 }
14 }
```

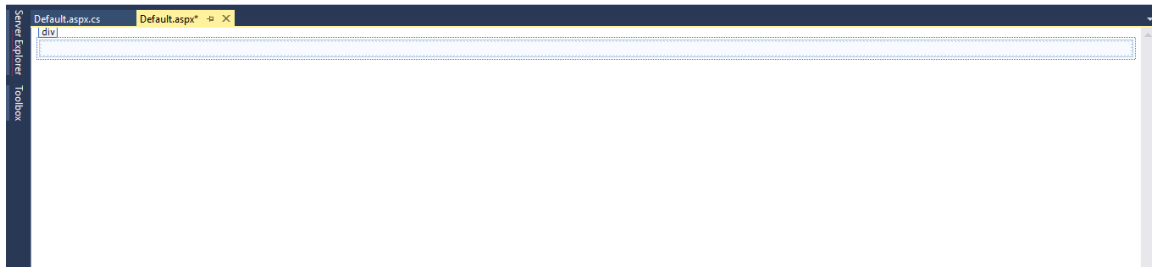
Every code behind file (.cs file) has the method name **Page Load()** which is fired every time user Refresh the web form and when the page is loaded first time to display the contents to the users. This method is shown in the above figure.

You can navigate from the code behind to design view by right click in code behind file and choose the View Designer, as shown in the next figure.

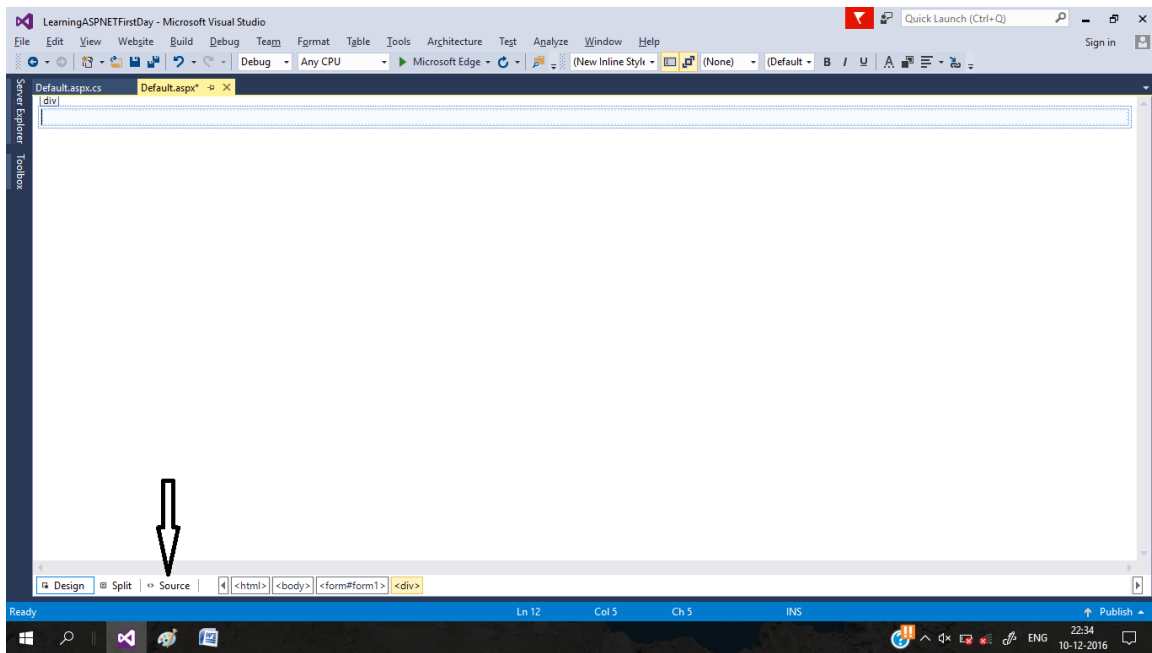


```
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7
8 public partial class _Default : System
9 {
10     protected void Page_Load(object se
11     {
12     }
13 }
14 }
```

This will open the screen as shown,



This is the screen where you can drag & drop the controls directly.... No write HTML code by yourself, you can move to the **Source View** by choosing the Source Tab from the left most corners with Source option (as shown by the Black arrow in next figure)



This will open:



```
1 <%@ Page Language="C#" AutoEventWireup="true"
2     CodeFile="Default.aspx.cs" Inherits="_Default" %>
3
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head runat="server">
6     <title></title>
7 </head>
8 <body>
9     <form id="form1" runat="server">
10    <div>
11
12    </div>
13    </form>
14 </body>
15 </html>
```

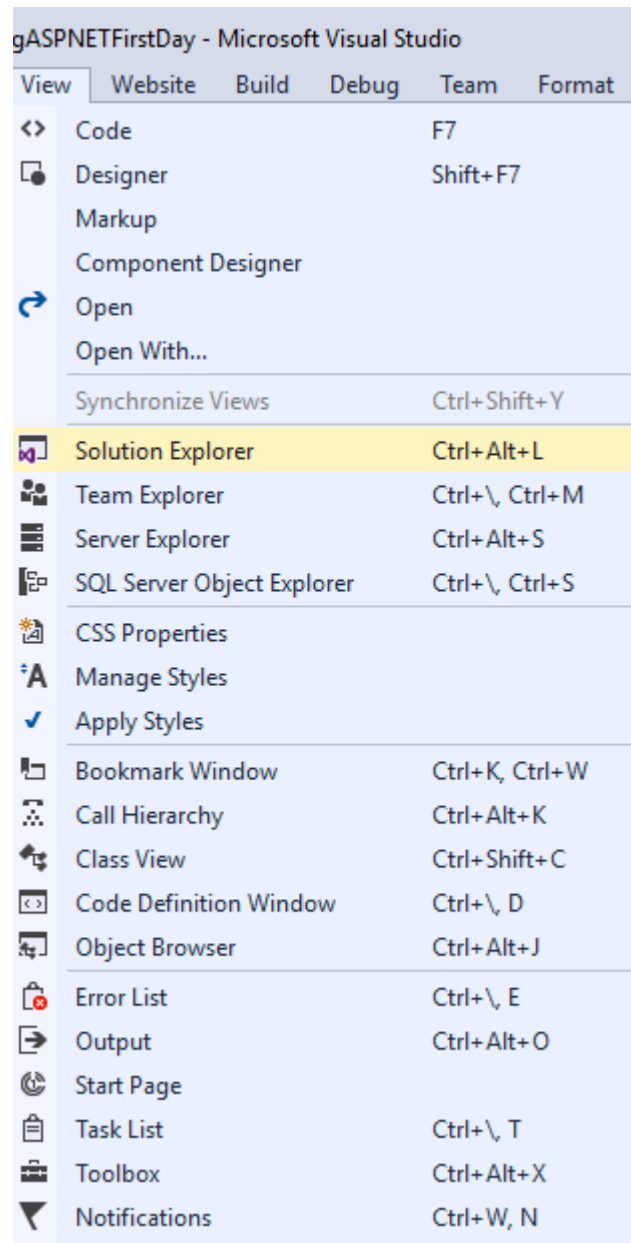
Similarly use its near Tab **Design** to move back to the design view...

### 1.3 Controls, Windows and toolbars in Visual Studio

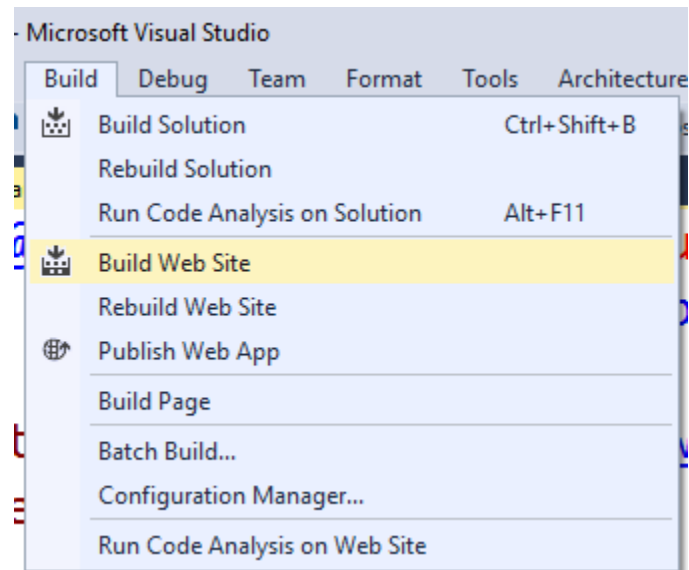
To open the ToolBox, Property Windows, or solution Explorer use the View Menu....

Here all the options are available..... As shown:



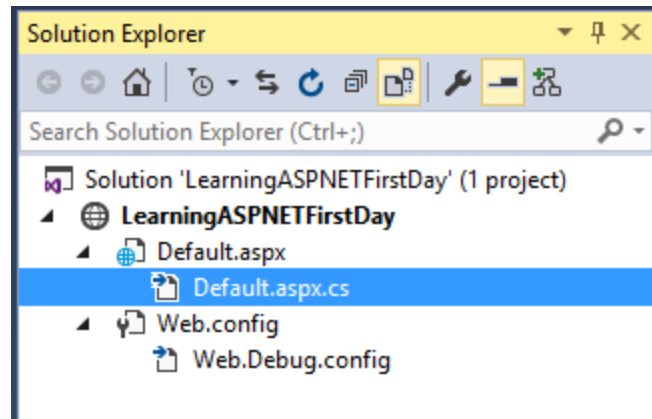


**To build (Compile)/Rebuild** full web site, Page use the **Build** option from Menu



## **Solution Explorer:**

The Solution Explorer provides the detail listing of all the Contents of the Web site, including the Style Sheet, Images, and Controls etc... It is shown on the right side of the Screen as shown:

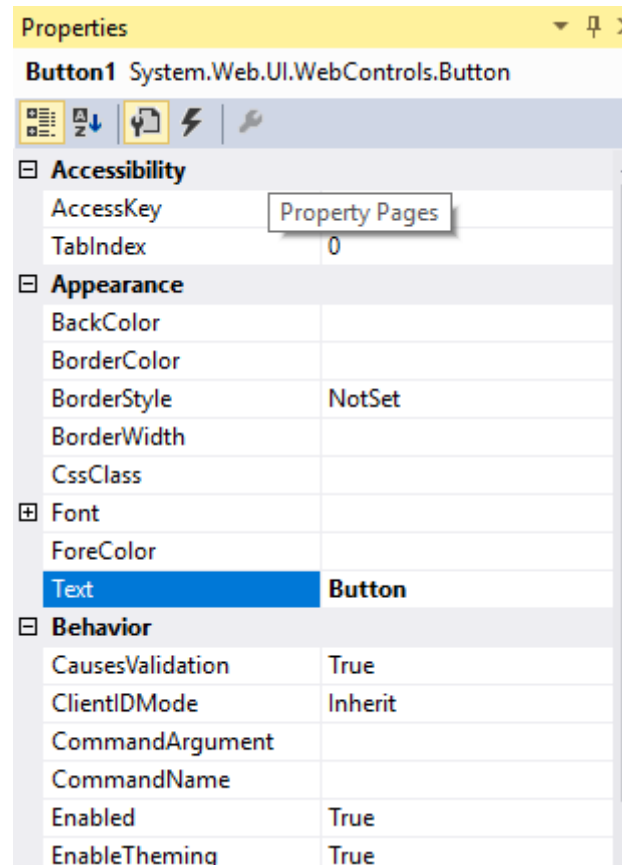


Solution Explorer can be displayed using

1. ALT + CTRL + L
2. View ➔ Solution Explorer

## Property Window:

Each control has some properties, which are listed in the property window, this window is under the Solution Explorer and is shown here:

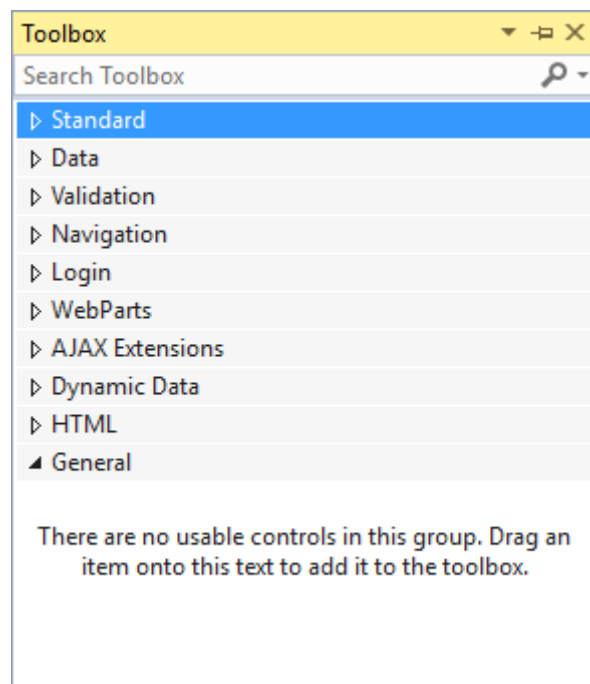


Property Window can be displayed using

1. Pressing F4
2. View ➔ Property Windows
3. Right Click on the Control ➔ Property

## **ToolBox:**

The Toolbox has the listing of controls that you can include into your web site. These controls are categorized and be collapsed. The Tool Box is on the left side of the screen and is shown here:



ToolBox Window can be displayed using

1. ALT + CTRL + X
2. View ➔ ToolBox

## 1.4 Naming Conventions for Different Web Controls

Naming Convention of the various controls in the ToolBox, others control you can adjust like these names:

Control Name	Userdefine Names (Examples)
TextBox	txtLoginID
Lable	lblLoginID
Button	btnAddLoginId
DropDownList	ddCountry
Calander	calDateofBirth
CheckBox	chkHobbies
Dataset	dsUsers
RadioButton	rdoStatus
GridView	gvUsers
Web Form	frmSystemUsers

### ***Some more guidelines:***

To Proceed Further You must have some details knowledge of the C# coding:

- ⇒ Each statement must be terminated by semicolon
- ⇒ Use good comments in your project code
- ⇒ Use braces with Decision Making/Repetition statement if each of these require the two statement to be executed
- ⇒ Use clear and good naming for the variables
- ⇒ Design good colorful interface of the form
- ⇒ Get aware of each component of the C-sharp

## 1.5 Understanding of ASP.NET web Page

First line,

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

Two important attributes are:

**Language** : tells the Language to be used for the code behind C# or VB

**CodeFile** : tells about the Code behind file name....

Then, we have a HTML Tag, Head Tag, Title Tag, Body Tag, Form Tag etc.

In ASP.Net, everything is written in Tags. To move forward you must have sound knowledge of HTML. **HTML** stands for Hyper Text Markup Language

It is used to create the web pages. And every command is written as Tag in HTML. Tag is basically pre defined in HTML, like

<B>

B is the Tag text and < & > tells the start and end of the tag.



## 1.6 Some well known tags

S #	Tag	Definition
1	<B>TEXT</B>	To bold the Text
2	<I>TEXT</I>	To Italic the Text
3	<TABLE></TABLE>	To create a table
4	<TR><TR>	To create a row in the table
5	<TD>TEXT</TD>	To add a column in a row of a table
6	<SUP>TEXT</SUP>	To make text Super subscript
7	<SUB>TEXT</SUB>	To make text subscript
8	<U></U>	To underline the Text
9		To display an image on the web form
10	<a href= <a href="http://www.google.com">http://www.google.com</a> >Google</a>	To open a Web Form within the web site or open another web site

**Note:** HTML is not case sensitive, means <B> or <b> are same

These tags will be shown in today's lecture. For a details description of each tag visit the site

<http://www.w3schools.com>

and check the HTML link on its left side.

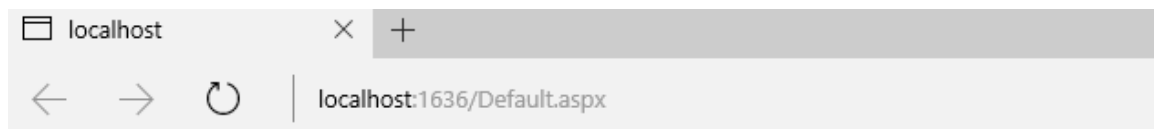
## 1.7 Running the web site...

In the Default.aspx form, add a message

**WelCome To ASP.Net Web Page**

```
1 <%@ Page Language="C#" AutoEventWireup="true"
2     CodeFile="Default.aspx.cs" Inherits="_Default" %>
3
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head runat="server">
6     <title></title>
7 </head>
8 <body>
9     <form id="form1" runat="server">
10    <div>
11        WelCome To ASP.Net Web Page
12    </div>
13    </form>
14 </body>
15 </html>
16
```

To display the output in the Browser, Click the Green small arrow or press F5



**WelCome To ASP.Net Web Page**

Next, add few more tags to the Web form and run the application, as shown

```
1 <%@ Page Language="C#" AutoEventWireup="true"
2     CodeFile="Default.aspx.cs" Inherits="_Default" %>
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head runat="server">
5     <title></title>
6 </head>
7 <body>
8     <form id="form1" runat="server">
9         <div>
10             <b>This is Bold Tag Example</b> <br />
11             <i>This is Italic Tag Example</i> <br />
12             <u>This is Underline Tag Example</u> <br />
13         </div>
14     </form>
15 </body>
16 </html>
```

Save and Refresh the web Form, output is



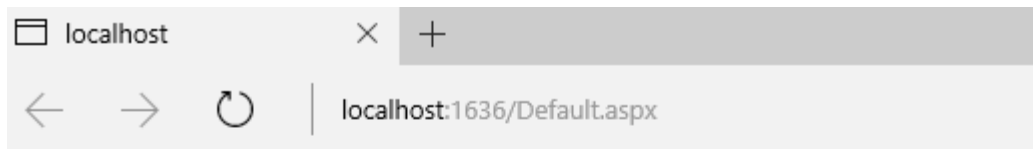
## 1.8 Adding Table Tag

<Table> is used to create table design in ASP.Net Web Form. In <Table> tag, <TR> tag is used to add a ROW in table and <TD> tag is used to add a COLUMN in the <TR> tag.

For example, the following code a TABLE with three ROWs and Three Columns in each table ROW.

```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <h3> Table Tag demo</h3>
        <table border="2">
            <tr>
                <td>Name</td>
                <td>Class</td>
                <td>Semester</td>
            </tr>
            <tr>
                <td>Muhammad Abdullah</td>
                <td>MCS</td>
                <td>2nd</td>
            </tr>
            <tr>
                <td>Ayesha Khan</td>
                <td>BSIT</td>
                <td>7th</td>
            </tr>
        </table> </form>
    </body>
</html>
```

The output is shown here



## Table Tag Demo

Name	Class	Semester
Muhammad Abdullah	MCS	2nd
Ayesha Khan	BSIT	7th

## 1.9 Using Some ASP.NET Web Controls

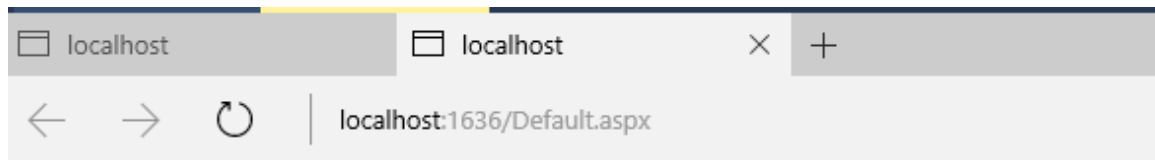
Design the form as shown here by drag and drop

- Button Control
- Label
- TextBox
- CheckBox
- Radio Button

After adding these Controls, open the **Property Window** for each control and define the property individually for each of these controls. The ASP.Net code is shown here

```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
    <form id="form1" runat="server">
        <h3>ASP.NET Web Controls Demo</h3>
        <asp:Label ID="lblFullName" runat="server" Text="Full Name"></asp:Label> :
        <asp:TextBox ID="txtFullName" runat="server" ></asp:TextBox> <br /> <br />
        <asp:Label ID="lblGender" runat="server" Text="Gender"></asp:Label> :
        <asp:RadioButton ID="rdbMale" runat="server" Text="Male"></asp:RadioButton>
        <asp:RadioButton ID="rdbFemale" runat="server" Text="Female"></asp:RadioButton> <br
    />
        Hobbies:
        <asp:CheckBox ID="chkCricket" runat="server" Text="Cricket"></asp:CheckBox>
        <asp:CheckBox ID="chkReading" runat="server" Text="Reading"></asp:CheckBox>
        <asp:CheckBox ID="chkCooking" runat="server" Text="Cooking"></asp:CheckBox><br />
        <asp:Button ID="btnShowData" runat="server" Text="Show Data"></asp:Button>
    </form>
</body>
</html>
```

The output is



## ASP.NET Web Controls Demo

Full Name :

Gender : ☐ Male ☐ Female

Hobbies: ☐ Cricket ☐ Reading ☐ Cooking

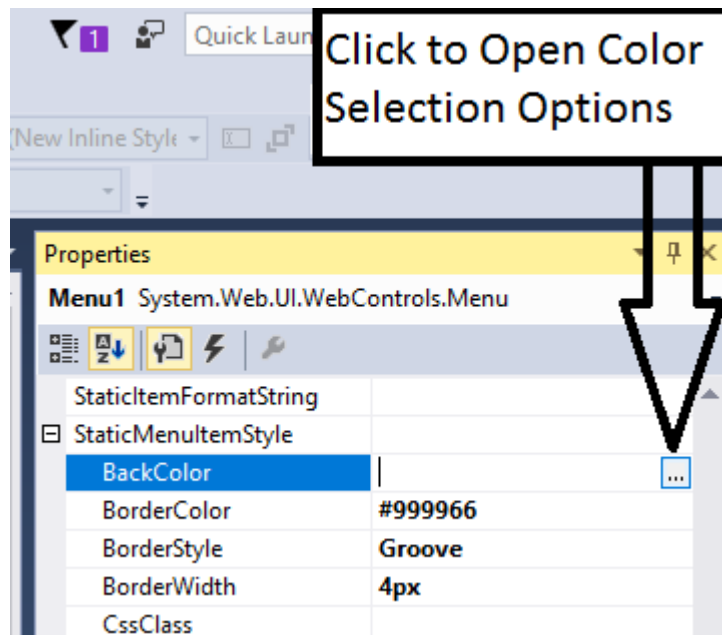
Show Data

### 1.10 Important and Common Properties of ASP.NET Web Controls

The most important and common properties of these controls are

Property Name	Description
<b>ID</b>	It is the name of the control, which is referred in C# code behind file
<b>Text</b>	This property is used to display or attach the Textual value with the control
<b>BackColor</b>	This property is used to change the back color of the control
<b>BorderColor</b>	This property is used to change the Border color of the control
<b>BorderStyle</b>	This property is used to define the Border Style of the selected control. This can be dotted, solid or dashed style
<b>Enabled</b>	This property defines whether the control will be Enabled or disabled at run time. If this property has the value TRUE then the control will be shown at run time but it will be only visible. Its value cannot be changed and no event associated to this control is fired
<b>Visible</b>	This property is used to define whether the control is visible on the Form or hidden.

When you Choose the **BackColor** Property of any control, you will see the following



**Congratulations! You have successfully Completed the Today's  
Activities**

**These ASP.Net Tutorials are prepared only for the purpose of learning ASP.Net with C#.**

**Anyone can copy, print or reuse it in any format**

Saif Ur Rehman Saifi

Assistant Professor, UIIT

PMAS Arid Agriculture University, Rawalpindi

If you found this tutorial helpful or you want to give us any valuable suggestions contact at

[www.ASPUIIT.blogspot.com](http://www.ASPUIIT.blogspot.com)