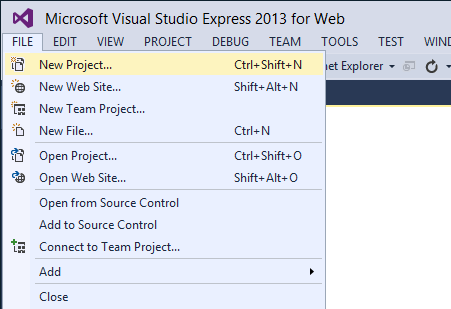
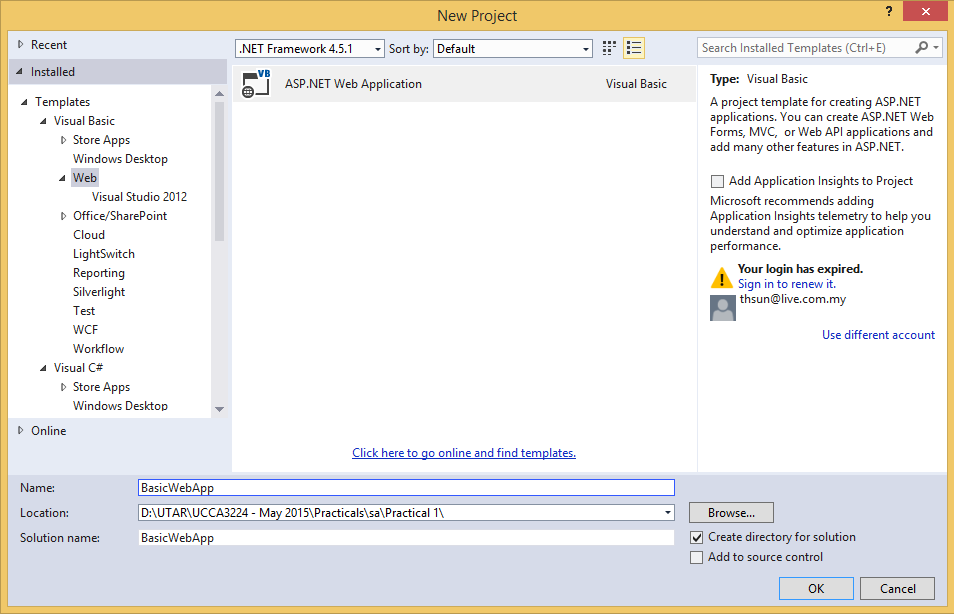
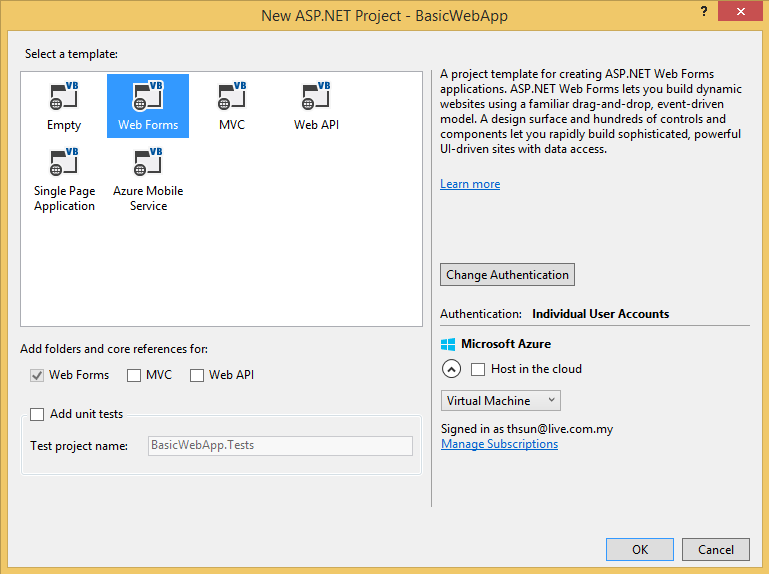
# Practical 1: Familiarization with the Tools Used To Create XHTML Documents

Q1) Creating a Web application project.

1. Open Microsoft Visual Studio.
2. On the **File** menu, select **New Project**.  
   
3. The **New Project** dialog box appears.
4. Select the **Templates** -> **Visual C#** -> **Web** templates group on the left.
5. Choose the **ASP.NET Web Application** template in the center column.
6. Name your project ***BasicWebApp*** and click the **OK** button.



1. Next, select the **Web Forms** template and click the **OK** button to create the project.



Visual Studio creates a new project that includes prebuilt functionality based on the Web Forms template. It not only provides you with a *Home.aspx* page, an *About.aspx* page, a *Contact.aspx* page, but also includes membership functionality that registers users and saves their credentials so that they can log in to your website. When a new page is created, by default Visual Studio displays the page in **Source** view, where you can see the page's HTML elements. The following illustration shows what you would see in **Source** view if you created a new Web page named *FirstWebPage.aspx*.  


1. Open the file Default.aspx by double-clicking it and remove all the code blocks inside the <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent"> block that are after the <div class="jumbotron"> code block. Replace it with the following bolded code:

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<div class="jumbotron">

<h1>ASP.NET</h1>

<p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.</p>

<p><a href="http://www.asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>

</div>

<div class="row">

<div class="col-md-4">

**<h2>Hello World</h2>**

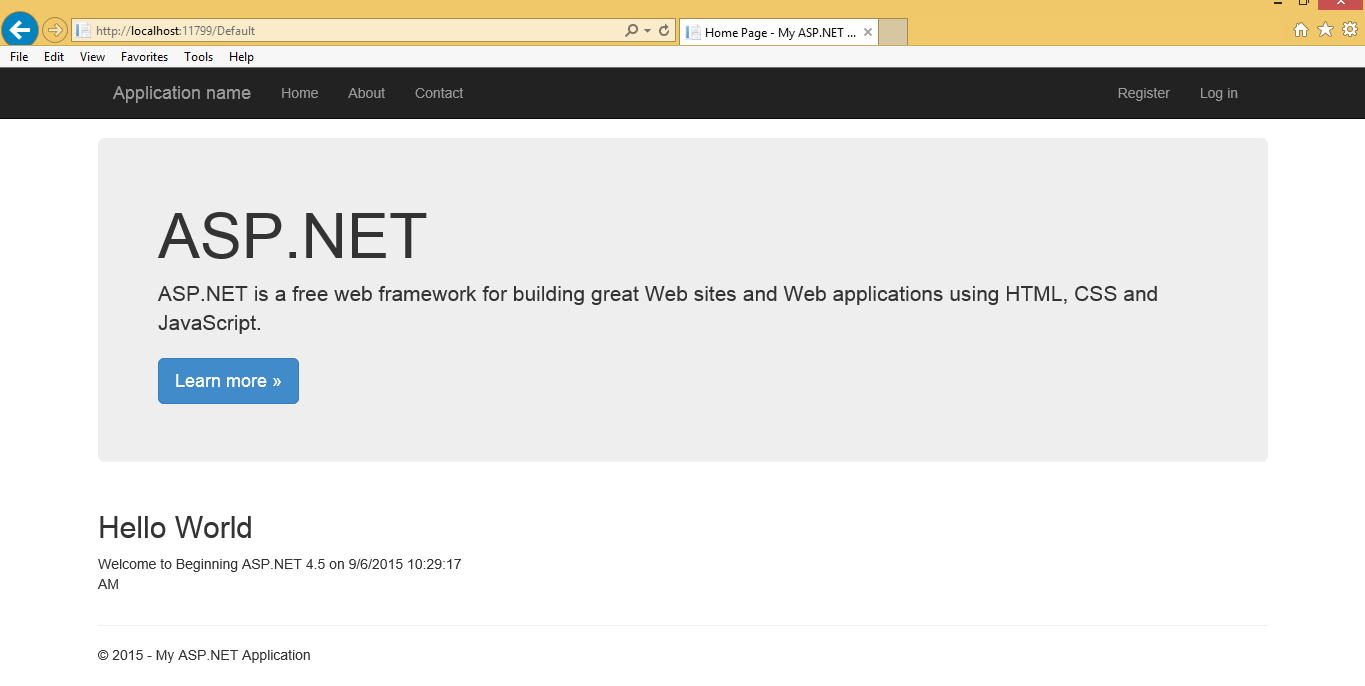
**<p>Welcome to Beginning ASP.NET 4.5 on <%: DateTime.Now.ToString() %></p>**

</div>

</div>

</asp:Content>

1. Press Ctrl+F5 to open the page in your default web browser. You see a page similar to the one shown below.



If you already have some experience with Visual Studio you may be used to pressing F5 instead. If you use that option, the site is opened in debug mode and you may get a dialog asking if you want to enable debugging (which you can safely do). If you see an information bar warning about intranet settings in Internet Explorer, click Turn on Intranet Settings.

If you don’t see the date and time in the page, or if you get an error, look again at the code in the welcome message. It starts with an angle bracket (<) followed by a percentage symbol and a colon. It closes with a single percentage sign and another angle bracket (>). Also, make sure you typed in the code exactly as shown here, including capitalization.

1. Notice how a small icon representing *IIS Express* has appeared in the tray bar of Windows, visible in Figure below.



If you see a different icon, right-click your site in VS and choose Use IIS Express. If you don’t see the icon in the tray, click the arrow near the other icons in the Windows tray and click the Customize option. Then set IIS Express System Tray to Show Icon and Notifications. The icon belongs to the built-in web server called IIS Express. This web server has been started by VS automatically to serve the request for your page. You learn more about how the web server processes your page later in this chapter.

That’s it. You just created your very first ASP.NET Web application with Visual Studio.

Q2) Adding Web Forms with Code to Your Site

1. In the Solution Explorer, right-click your Web application and choose Add Í New Folder. Name the folder **Demos** and press Enter.
2. Right-click the Demos folder and choose Add Í Add New Item. In the dialog box that appears, choose your preferred programming language on the left, click the Web Form template, and name the file **CodeBehind.aspx**. The page should open in Markup View so you can see the HTML for the page.
3. At the bottom of the Document Window, click the Design button to switch the page from Markup View into Design View. The page you see has a white background with a small, dashed rectangle at the top of it. The dashed rectangle represents the <div> element you saw in Markup View.
4. From the Toolbox, drag a Label control from the Standard category and drop it in the dashed area of the page. Remember, you can open the Toolbox with the shortcut Ctrl+Alt+X if it isn’t open yet. In Design View, your screen should now look like Figure 2-1.



Figure 2-1

1. Double-click somewhere in the white area *below* the dashed line of the <div> element. VS switches from Design View into the Code Behind of the file and adds code that fires when the page loads in the browser:

Protected Sub Page\_Load(sender As Object, e As EventArgs) Handles Me.Load

End Sub

Although this odd syntax may look a little scary at this point, don’t worry about it too much now. In most cases, VS adds it for you automatically, as you just saw. The code you’re going to place between the lines that start with Protected Sub and End Sub in Visual Basic will be run when the page is requested in the browser.

1. Place your cursor in the open line in the code that VS created and add the bolded line of code that assigns today’s date and time to the label, which will eventually show up in the browser:

Protected Sub Page\_Load(sender As Object, e As EventArgs) Handles Me.Load

**Label1.Text = "Hello World; the time is now " & DateTime.Now.ToString()**

End Sub

Note that as soon as you type the L for Label1, you get a list with options to choose from. This is part of Visual Studio’s *IntelliSense*, a great tool that helps you rapidly write code. Instead of typing the whole word Label1, you simply type the letter L or the letters La and then you pick the appropriate item from the list, visible in Figure 2-2.



Figure 2-2

To complete the selected word, you can press Enter or Tab or even the period. In the latter case, you immediately get another list that enables you to pick the word Text simply by typing the first few letters, completing the word by pressing the Tab or Enter key. This feature is a real productivity tool because you can write code with a minimum of keystrokes. IntelliSense is available in many other file types as well, including ASPX, HTML, CSS, JavaScript, and XML. In many cases, the list with options pops up automatically if you begin typing. If it doesn’t, press Ctrl+Spacebar to invoke it. If the list covers some of your code in the code window, press and hold the Ctrl key to make the window transparent.

1. Right-click the CodeBehind.aspx page in the Solution Explorer and choose View in Browser (Internet Explorer). Depending on the default browser you’ve configured for your computer, the browser name in the parentheses may be different. I’ll simply refer to this menu item as View in Browser from now on.
2. Click Yes if you get a dialog box that asks if you want to save the changes, and then the page will appear in the browser, similar to the browser window you see in Figure 2-3.



Figure 2-3

If you don’t see the message with the date and time appear or you get an error on the page in the browser, make sure you saved the changes to all open pages. To save all pages at once, press Ctrl+Shift+S or click the Save All button on the toolbar (the one with the multiple floppy disk symbols). Additionally, make sure you typed the code for the right language. When you created this new page, you chose a programming language that applies to the entire page. You can’t mix languages on a single page, so if you started with a Visual C# page, make sure you entered the C# code snippet in step 6.

1. Setting up a page with inline code is very similar. Start by adding a new Web Form to the Demos folder. Call it **CodeInline.aspx.** Remove<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="CodeBehind.aspx.vb" Inherits="BasicWebApp.CodeBehind" %> and replace it with **<%@ Page Language="vb"%>**.
2. Just as you did in steps 3, 4, and 5, switch the page into Design View, drag a label inside the <div> element, and double-click the page somewhere outside the <div> that now contains the label. Instead of opening a Code Behind file, VS now switches your page into Markup View and adds the Page\_Load code directly in the page.
3. On the empty line in the code block that VS inserted, type the bolded line you see in step 6 of this exercise. You should end up with the following code at the top of your .aspx file:

<script runat="server">

Protected Sub Page\_Load(sender As Object, e As EventArgs)

**Label1.Text = "Hello World; the time is now " & DateTime.Now.ToString()**

End Sub

</script>

1. Right-click the page in the Solution Explorer and choose View in Browser. Alternatively, press Ctrl+F5 to open the page in your browser. You should see a page similar to the one you got in step 7.

Q3) Create a web page that will create a Product object, get its HTML representation, and then display it in the web page. The output is as shown in Figure 3-1.



Figure 3-1

You should have the following files:-

Product.vb: This file contains the code for the Product class. This file should be placed in the App\_Code subfolder, which allows ASP.NET to compile it automatically.

Garbage.jpg: This is the image that the Product class will use. [You could obtain this file from WBLE]. Place this file in the Images subfolder.

Default.aspx: This file contains the web page code that uses the Product class.

The details of the Product class shown in Figure 3-2 and Figure 3-3.

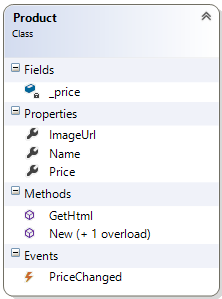


Figure 3-2

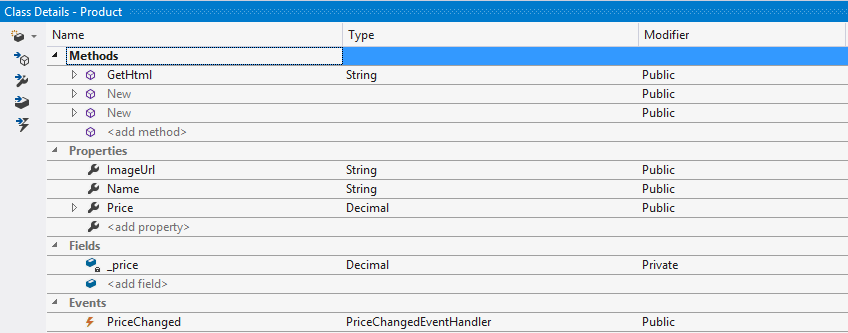


Figure 3-3

1. Write the code for the Product class.
2. Write the markup for Default.aspx.
3. Compile and browse Default.aspx.