Introduction to ASP.NET

Today's Lecture Outline

- What is ASP.NET
- ASP.NET Web Forms Model
- Components of .NET Framework
- Setting the Environment
- ASP.NET: An Example
- Event Handling in ASP.NET
- Application and Session Events

Today's Lecture Outline…

- **Page and Control Events**
- **Event Handling using Controls**
- **Common Control Events**
- **PostBack and NonPostBack Events**
- **Controls with Default Events**

1. What Is ASP.NET

- **ASP.**
- ASP stands for **A**ctive **S**erver **P**ages
- ASP is a program that runs inside **IIS**
- IIS stands for **I**nternet **I**nformation **S**ervice
- **.NET**
- A programming infrastructure created by Microsoft for building, deploying, and running applications and services that use .**NET**technologies, such as desktop applications and Web services.

1. What Is ASP.NET

- **What is an ASP File?**
- **An ASP file is just** the same as an HTML file
- **An ASP file can contain** text, HTML, XML, and scripts
- **Scripts in an ASP** file are executed on the server
- **An ASP file** has the file extension ".asp"

1. What Is ASP.NET

- **ASP.NET is a web application** framework developed and marketed by **Microsoft to allow programmers to build dynamic web sites**. It allows you to use a **full featured programming** language such as C# or VB.NET to build web applications easily.
- **The ASP.NET application codes** can be written in any of the following languages:
- C#
- Visual Basic.Net
- Jscript
- J#

1. What Is ASP.NET

- **ASP.NET is a web development platform**, which provides
  - **A programming model**
  - **A comprehensive software infrastructure**
  - **Various services required to build up robust web applications for PC as well as mobile devices.**
- **ASP.NET works on top of the HTTP protocol**, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.
- **ASP.NET is a part of Microsoft .Net platform**.
- **ASP.NET applications are compiled codes**, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

1. What Is ASP.NET…

- **ASP.NET application codes can be written in any of the following languages:**
  - C#
  - Visual Basic.Net
  - Jscript
  - J#
- **ASP.NET is used to produce interactive**, data-driven web applications over the internet.
- **It consists of a large number of controls** such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

2. ASP.NET Web Forms Model

- **ASP.NET web forms extend** the event-driven model of interaction to the web applications.
  - The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.
- **All client side user activities are forwarded** to the server for stateful processing.
    - The server processes the output of the client actions and triggers the reactions.
- **HTTP is a stateless protocol.**
  - ASP.NET framework helps in storing the information regarding the state of the application, which consists of:
    - Page state
    - Session state

2. ASP.NET Web Forms Model…

- The page state is the state of the client, i.e., the content of various input fields in the web form.
- The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state (Just as we discussed in PHP Lectures).

2. ASP.NET Web Forms Model…

- **Example:**
  - **Let us take an example of a shopping cart:**
    - **User adds** items to a shopping cart.
    - **Items are selected** from a page, say the items page
    - **The total collected items and price** are shown on a different page, say the cart page.
    - **Only HTTP cannot keep track** of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

2. ASP.NET Web Forms Model…

- **ASP.NET runtime carries** the page state to and from the server across page requests while **generating ASP.NET runtime codes**, and incorporates the state of the server side components in hidden fields.
- **This way, the server becomes aware** of the overall application state and operates in a two-tiered connected way.

3. Components of .NET Framework

- **Common Language Runtime (CLR)**
  - **Performs Memory Management**
  - Exception Handling
  - **Debugging**
  - Security Checking
  - **Thread Execution**
  - Code Execution
  - **Code Safety**
  - Verification
  - **Compilation.**
  - The code that is directly managed by the CLR is called the managed code.
  - **When the managed code is compiled**, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just-In-Time (JIT) compiler compiles the IL code into native code, which is CPU specific.

3. Components of .NET Framework…

- **.NET Framework Class Library**
  - **It contains a huge library of reusable types**, classes, interfaces, structures, and enumerated values, which are collectively called types.
- **ASP.NET & ASP.NET AJAX**
  - **ASP.NET is the web development model and AJAX** is an extension of ASP.NET for developing and implementing AJAX functionality.
  - **ASP.NET AJAX contains the components** that allow the developer to update data on a website without a complete reload of the page.

3. Components of .NET Framework…

- **ADO.NET**
  - **It is the technology used for working with data** and databases. It provides access to data sources like SQL server, OLE DB, XML etc.
  - **The ADO.NET allows connection to data** sources for retrieving, manipulating, and updating data.
- **LINQ**
  - **It imparts data querying capabilities to .Net languages** using a syntax which is similar to the tradition query language SQL.

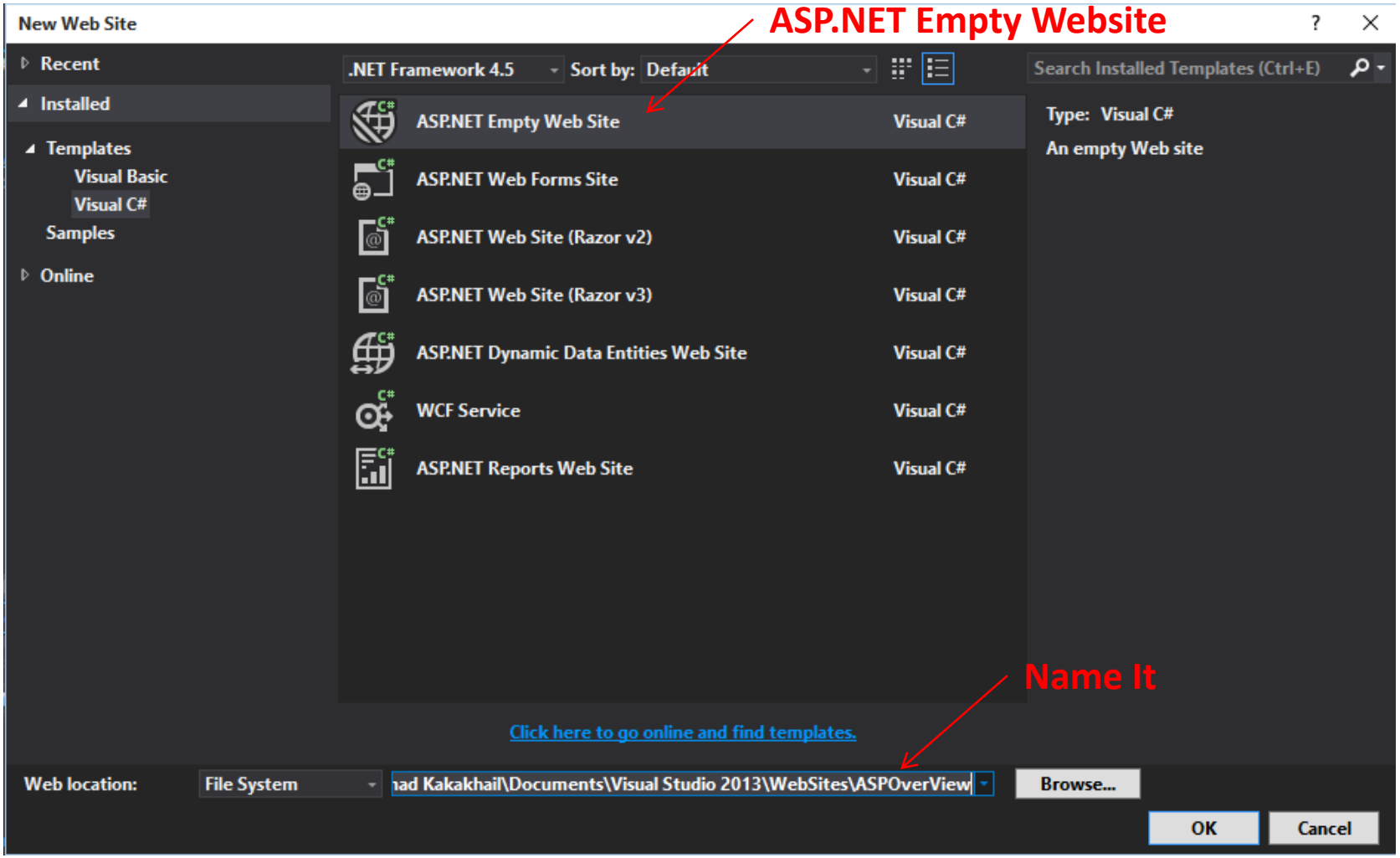
4. Setting The Environment

- **Visual Studio IDE**
  - **Can be download from microsoft.com**
  - **You can get Express Edition because it is free**
  - **It allows you to write code, compile it, debug it and run it in the browser**
  - **It has variety of templates for multiple languages which supports .NET framework**

4. Setting The Environment…

- **How to Work With ASP.NET:**
  - **Open Visual Studio**
  - **From the File Menu, Click on New then Click on Website**
  - **Select ASP.NET Empty Website**
  - **Name It**
  - **The Website Will Be Created**

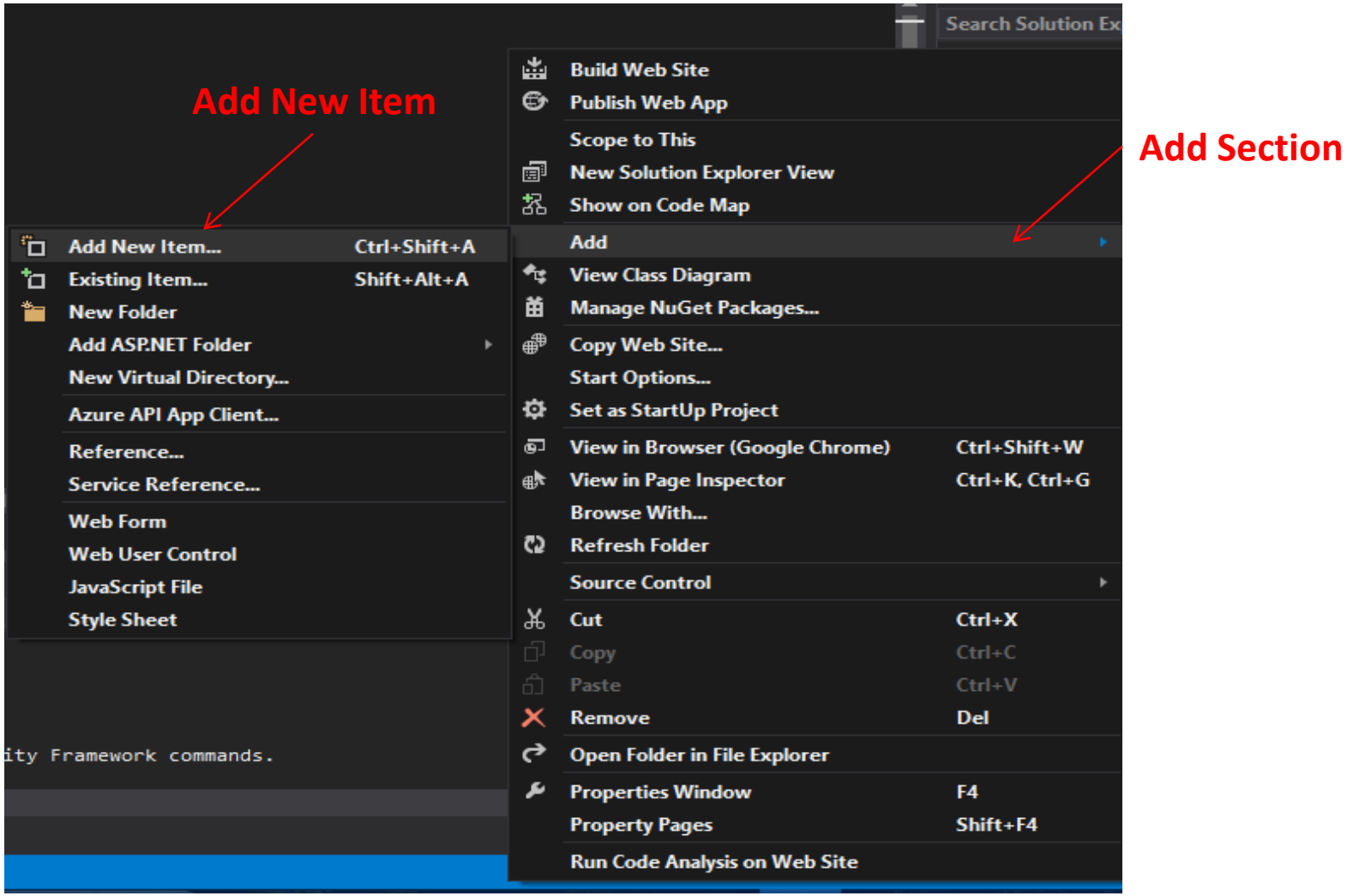
4. Setting The Environment…



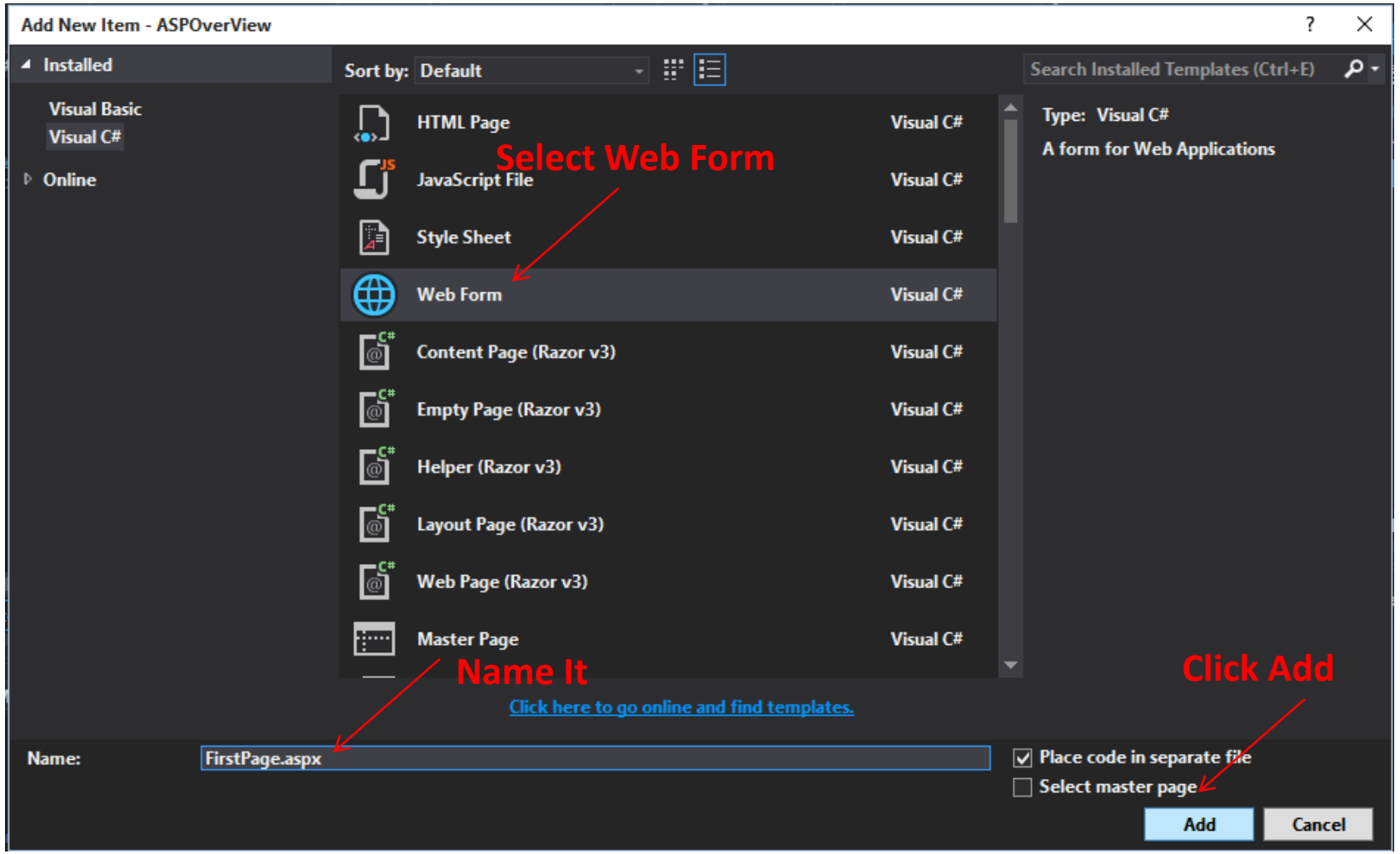
4. Setting The Environment…

- **Now**
  - **Right click on ASPOverView in Solution Explorer**
  - **Expand Add Section**
  - **Select Add New Item**
  - **Select Web Form**
  - **Name It**
  - **Click Add**

4. Setting The Environment…



4. Setting The Environment…



5. ASP.NET: an Example

- **Now, Let's Add Some Code in our first ASP.NET Web Site**
- **You can see the Web Form As an .aspx file in the Solution Explorer**
- **Click on FirstPage.aspx and you can see the HTML code.**
- **You can also see that there is a small arrow with FirstPage.aspx in Solution Explorer**
  - **Expand it, and you will see FirstPage.aspx.cs file**
  - **That's the file whre Code Behind is written**
- **.aspx file contains HTML and Scripting Code while .aspx.cs contains C# code**

5. ASP.NET: an Example…

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>First Page </title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox1" runat="server" Style="width: 224px">
            </asp:TextBox>
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" Text="Show Result" OnClick="Button1_Click" Width="98px" />
            <hr />
            <h3>Results: </h3>
            <span runat="server" id="changed_text" />
        </div>
    </form>
</body>
</html>
```

Page Directives

5. ASP.NET: an Example…

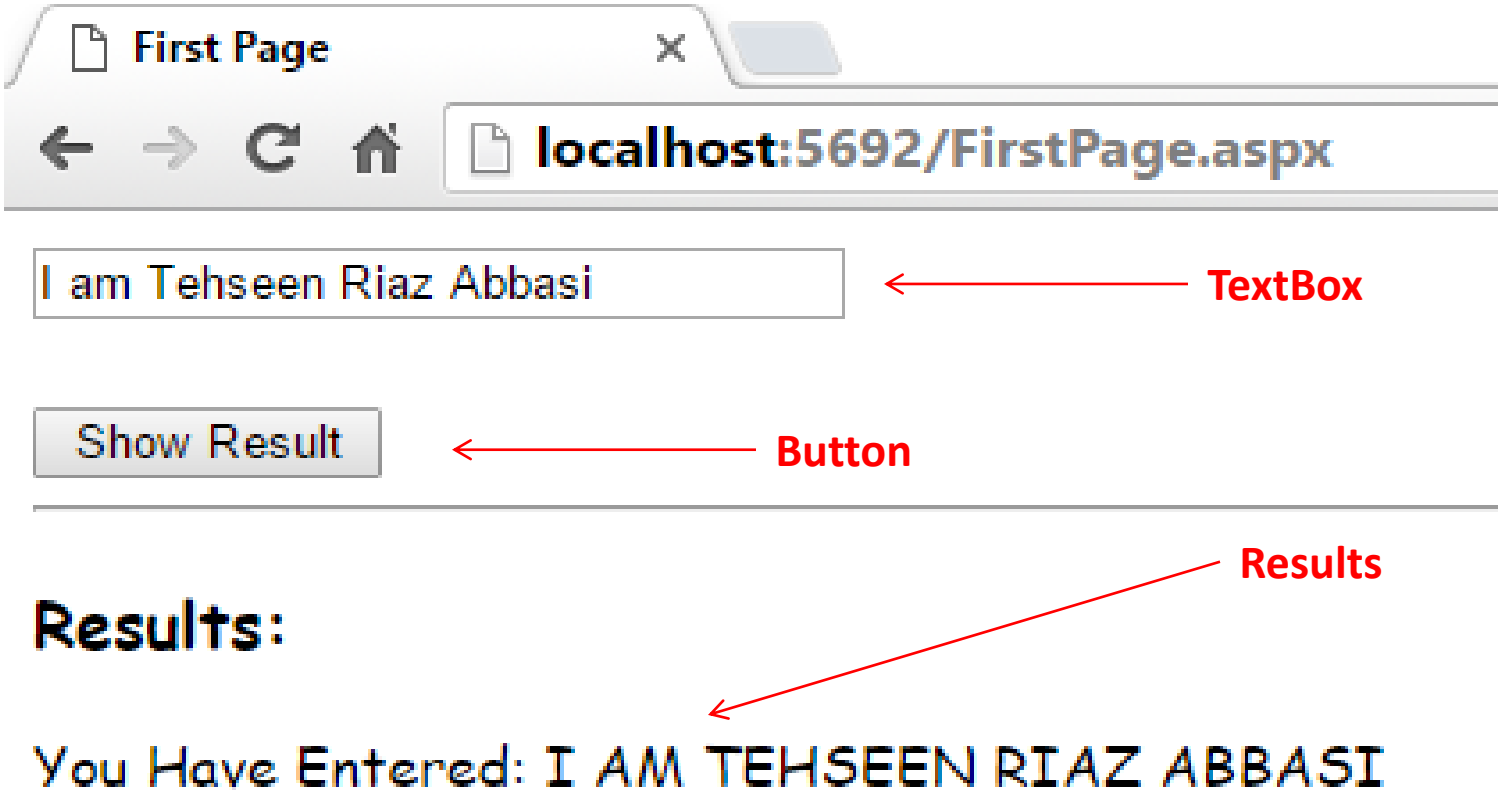
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

0 references
public partial class FirstPage : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    0 references
    protected void Button1_Click(object sender, EventArgs e)
    {
        string buf = TextBox1.Text;
        changed_text.InnerHtml = "You Have Entered: " + buf.ToUpper();
    }
}
```

Button Click Event

5. ASP.NET: an Example…



6. Event Handling in ASP.NET

- **An event is an action or occurrence** such as a mouse click, a key press, mouse movements, or any system-generated notification.
- **A process communicates through events**.
  - For example, interrupts are system-generated events. When events occur, the application should be able to respond to it and manage it.
- **Events in ASP.NET** are raised at the client machine and handled at the server machine.
  - For example, a user clicks a button displayed in the browser. **A Click event is raised**. The browser handles this client-side event by posting it to the server.

6. Event Handling in ASP.NET…

- **The server has a subroutine describing what to do when the event is raised**; it is called the event-handler.
  - **When the event message is transmitted to the server, it checks whether the Click event has an associated event handler. If it has, the event handler is executed**.
- ASP.NET event handlers **generally take two parameters and return void**. The first parameter **represents the object raising** the event and the second **parameter is event argument.**
  - private void EventName (object sender, EventArgs e);

7. Application and Session Events

- **Application_Start:**
  - It is raised when the application/website is **started**
- **Application_End:**
  - It is raised when the application/website is **stopped.**
- **Session_Start:**
  - It is raised when a **user first requests a page** from the application.
- **Session_End:**
  - It is raised when the **session ends.**

8. Page and Control Events

- **DataBinding**
  - It is raised when a control binds to a data source
- **Disposed**
  - It is raised when the page or the control is released
- **Error**
  - It is a page event, occurs when an unhandled exception is thrown
- **Init**
  - It is raised when the page or the control is initialized

8. Page and Control Events…

- **Load**
  - It is raised when a page or control is loaded
- **PreRender**
  - It is raised when the page or the control is to be rendered
- **Unload**
  - It is a raised when a page or control is unloaded from memory

9. Event Handling Using Controls

- **All ASP.NET controls** are implemented as classes, and they have events which are fired when a user performs a certain action on them.
  - **For example, when a user clicks a button the 'Click' event is generated.**
- For handling events, there are **in-built attributes and event handlers.**
- **Event handler** is coded to respond to an event and take appropriate action on it.

9. Event Handling Using Controls…

- **The ASP tag for a button control:**
  - <asp:Button ID="btnCancel" runat="server" Text="Cancel" />
- **The event handler for click event:**
  - protected void btnCancel_Click(object sender, EventArgs e)
    {
    // Handles Event
    }

10. Common Control Events

- **Click**
  - **Attribute:** OnClick
  - **Associated Controls:** Button, ImageButton, LinkButton, ImageMap
- **Command**
  - **Attribute:** OnCommand
  - **Associated Controls:** Button, ImageButton, LinkButton
- **TextChanged**
  - **Attribute:** OnTextChanged
  - **Associated Controls:** TextBox

10. Common Control Events…

- SelectedIndexChanged
  - **Attribute:** OnSelectedIndexChanged
  - **Associated Controls:** DropDownList, ListBox, RadioButton, CheckBoxList
- CheckedChanged
  - **Attribute:** OnCheckedChanged
  - **Associated Controls:** RadioButton, CheckBox

11. PostBack Events & Non-PostBack Events

- **Some events cause the form to be posted back to the server immediately,** these are called the postback events.
  - **For example**, the click event such as Button.Click.
- **Some events are not posted back to the server immediately**, these are called non-postback events.
  - For example, the change events or selection events such as TextBox.TextChanged or CheckBox.CheckedChanged.
- **The nonpostback events** could be made to post back immediately by setting their AutoPostBack property to true.

12. Controls with Default Events

- **AdRotator**
  - **Default Event:** Ad Created
- **BulletedList**
  - **Default Event**: Click
- **Button**
  - **Default Event:** Click
- **Calendar**
  - **Default Event:** SelectionChanged
- **CheckBox**
  - **Default Event:** CheckedChanged

12. Controls with Default Events…

- **CheckBoxList**
  - **Default Event:** SelectedIndexChanged
- **DataGrid**
  - **Default Event:** SelectedIndexChanged
- **DataList**
  - **Default Event:** SelectedIndexChanged
- **DropDownList**
  - **Default Event:** SelectedIndexChanged
- **HyperLink**
  - **Default Event:** Click

12. Controls with Default Events…

- **ImageButton**
  - **Default Event:** Click
- **ImageMap**
  - **Default Event:** Click
- **LinkButton**
  - **Default Event:** Click
- **ListBox**
  - **Default Event:** SelectedIndexChanged
- **Menu**
  - **Default Event:** MenuItemClick

12. Controls with Default Events…

- **RadioButton**
  - **Default Event:** CheckedChanged
- **RadioButtonList**
  - **Default Event:** SelectedIndexChanged

Summary of Today's Lecture

- What is ASP.NET
- ASP.NET Web Forms Model
- Components of .NET Framework
- Setting the Environment
- ASP.NET: An Example
- Event Handling in ASP.NET
- Application and Session Events

Summary of Today's Lecture…

- Page and Control Events
- Event Handling using Controls
- Common Control Events
- PostBack and NonPostBack Events
- Controls with Default Events

Good Link:
http://www.tutorialspoint.com/asp.net/asp.net_quick_guide.htm

THANK YOU