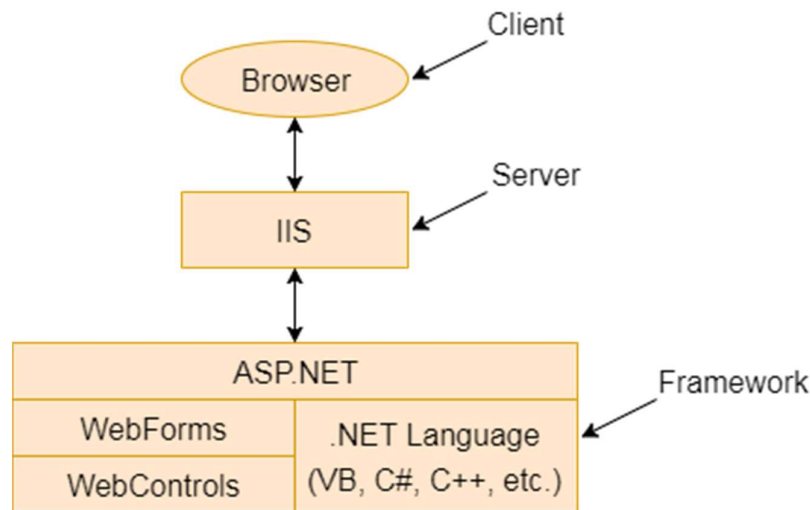


# ASP.NET Web Forms

Web Forms are web pages built on the ASP.NET Technology. It executes on the server and generates output to the browser. It is compatible to any browser to any language supported by .NET common language runtime. It is flexible and allows us to create and add custom controls.

We can use Visual Studio to create ASP.NET Web Forms. It is an IDE (Integrated Development Environment) that allows us to drag and drop server controls to the web forms. It also allows us to set properties, events and methods for the controls. To write business logic, we can choose any .NET language like: Visual Basic or Visual C#.

Web Forms are made up of two components: the visual portion (the ASPX file), and the code behind the form, which resides in a separate class file.



**Fig:** This diagram shows the components of the ASP.NET

The main purpose of Web Forms is to overcome the limitations of ASP and separate view from the application logic.

**ASP.NET provides various controls like:** server controls and HTML controls for the Web Forms. We have tables all these controls below.

---

## Server Controls

The following table contains the server-side controls for the Web Forms.

Control Name	Applicable Events	Description
Label	None	It is used to display text on the HTML page.
TextBox	TextChanged	It is used to create a text input in the form.
Button	Click, Command	It is used to create a button.
LinkButton	Click, Command	It is used to create a button that looks similar to the hyperlink.
ImageButton	Click	It is used to create an imagesButton. Here, an image works as a Button.

Hyperlink	None	It is used to create a hyperlink control that responds to a click event.
DropDownList	SelectedIndexChanged	It is used to create a dropdown list control.
ListBox	SelectedIndexCnhaged	It is used to create a ListBox control like the HTML control.
DataGrid	CancelCommand, EditCommand, DeleteCommand, ItemCommand, SelectedIndexChanged, PageIndexChanged, SortCommand, UpdateCommand, ItemCreated, ItemDataBound	It used to create a frid that is used to show data. We can also perform paging, sorting, and formatting very easily with this control.
DataList	CancelCommand, EditCommand, DeleteCommand, ItemCommand, SelectedIndexChanged, UpdateCommand, ItemCreated, ItemDataBound	It is used to create datalist that is non-tabular and used to show data.
Repeater	ItemCommand, ItemCreated, ItemDataBound	It allows us to create a non-tabular type of format for data. You can bind the data to template items, which are like bits of HTML put together in a specific repeating format.
CheckBox	CheckChanged	It is used to create checkbox.
CheckBoxList	SelectedIndexChanged	It is used to create a group of check boxes that all work together.
RadioButton	CheckChanged	It is used to create radio button.
RadioButtonList	SelectedIndexChanged	It is used to create a group of radio button controls that all work together.
Image	None	It is used to show image within the page.
Panel	None	It is used to create a panel that works as a container.
Placeholder	None	It is used to set placeholder for the control.
Calendar	SelectionChanged, VisibleMonthChanged, DayRender	It is used to create a calendar. We can set the default date, move forward and backward etc.
AdRotator	AdCreated	It allows us to specify a list of ads to display. Each time the user re-displays the page.
Table	None	It is used to create table.
XML	None	It is used to display XML documents within the HTML.
Literal	None	It is like a label in that it displays a literal, but allows us to create new literals at runtime and place them into this control.

## HTML Controls

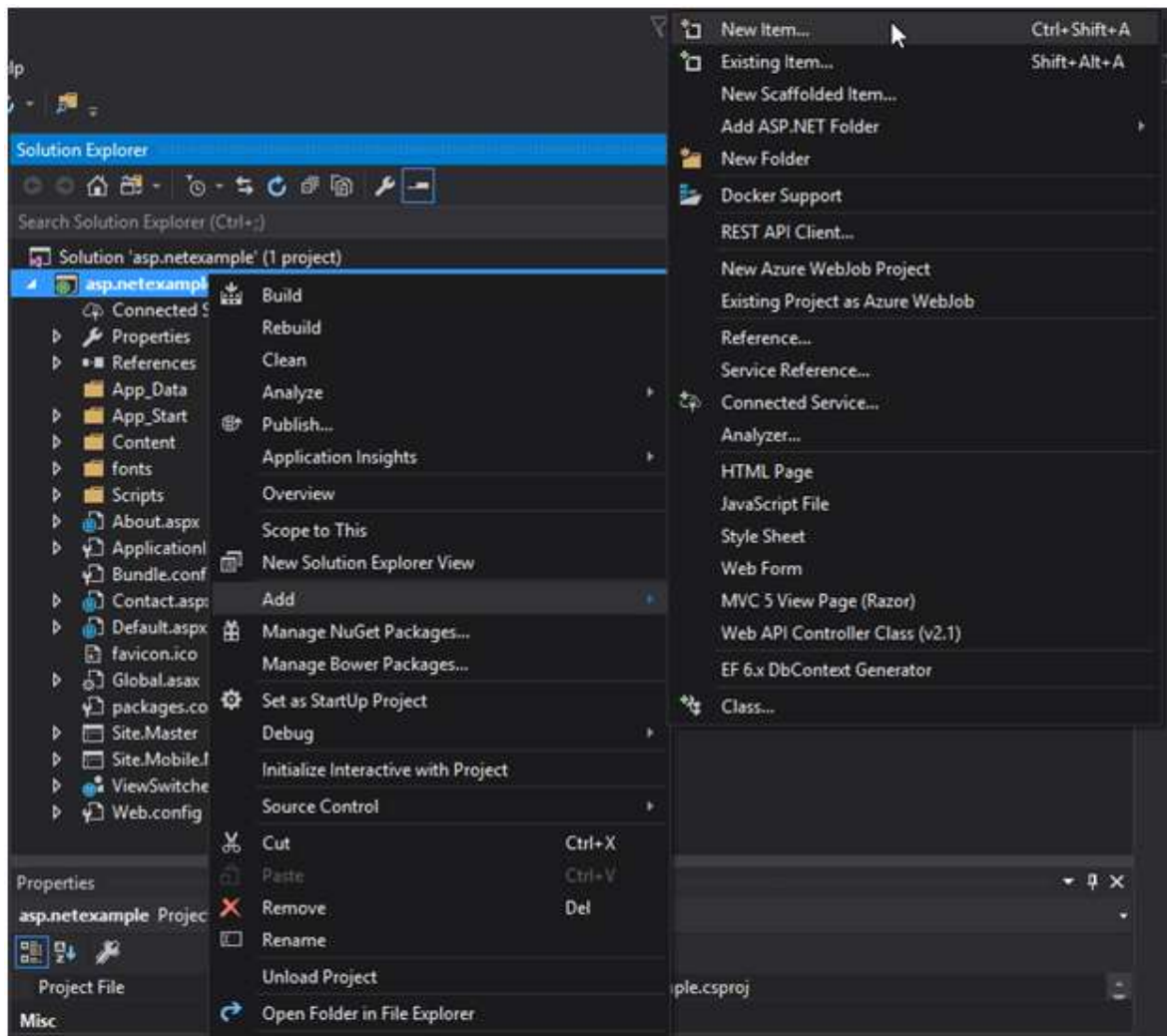
These controls render by the browser. We can also make HTML controls as server control. we will discuss about this in further our tutorial.

Controls Name	Description
Button	It is used to create HTML button.

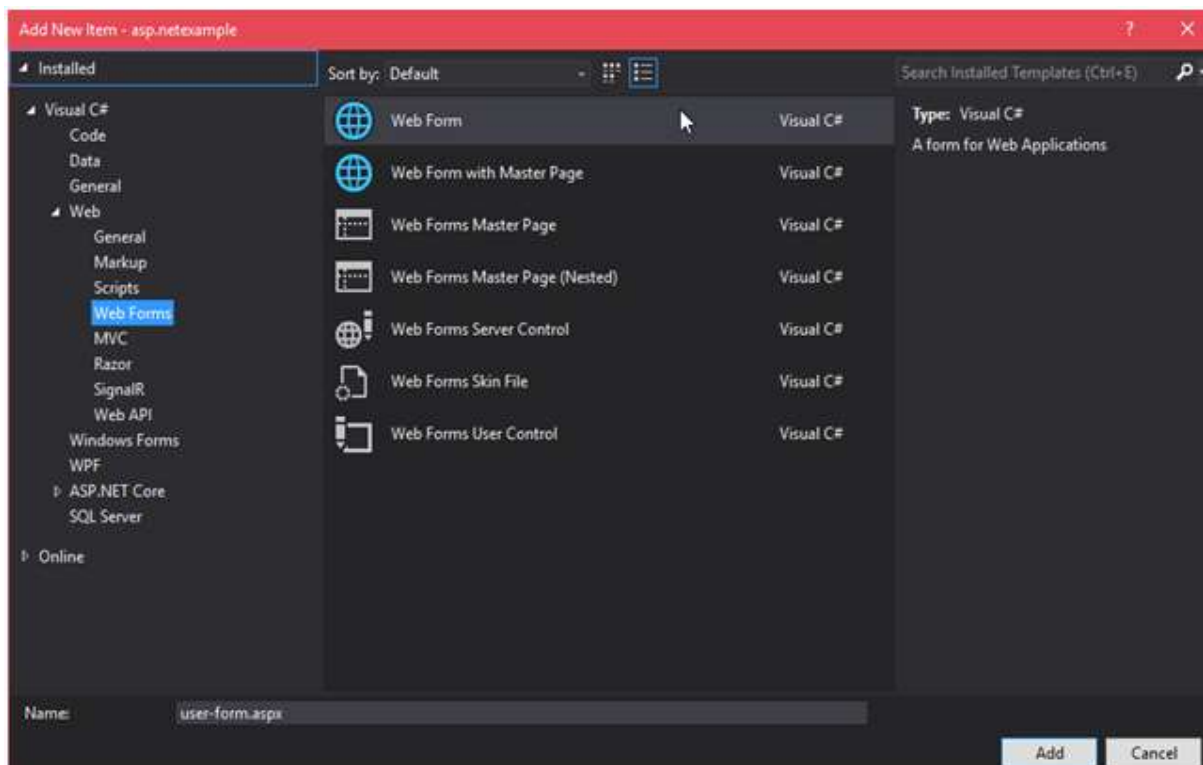
Reset Button	Resets all other HTML form elements on a form to a default value
Submit Button	Automatically POSTs the form data to the specified page listed in the Action attribute in the FORM tag
Text Field	Gives the user an input area on an HTML form
Text Area	Used for multi-line input on an HTML form
File Field	Places a text field and a Browse button on a form and allows the user to select a file name from their local machine when the Browse button is clicked
Password Field	An input area on an HTML form, although any characters typed into this field are displayed as asterisks
CheckBox	Gives the user a check box that they can select or clear
Radio Button	Used two or more to a form, and allows the user to choose one of the controls
Table	Allows you to present information in a tabular format
Image	Displays an image on an HTML form
ListBox	Displays a list of items to the user. You can set the size from two or more to specify how many items you wish show. If there are more items than will fit within this limit, a scroll bar is automatically added to this control.
Dropdown	Displays a list of items to the user, but only one item at a time will appear. The user can click a down arrow from the side of this control and a list of items will be displayed.
Horizontal Rule	Displays a horizontal line across the HTML page

## Create a New Web Form

Here, we are using the project that we created in last topic. To add a new web form in our existing project, first select project then right click and add new item.

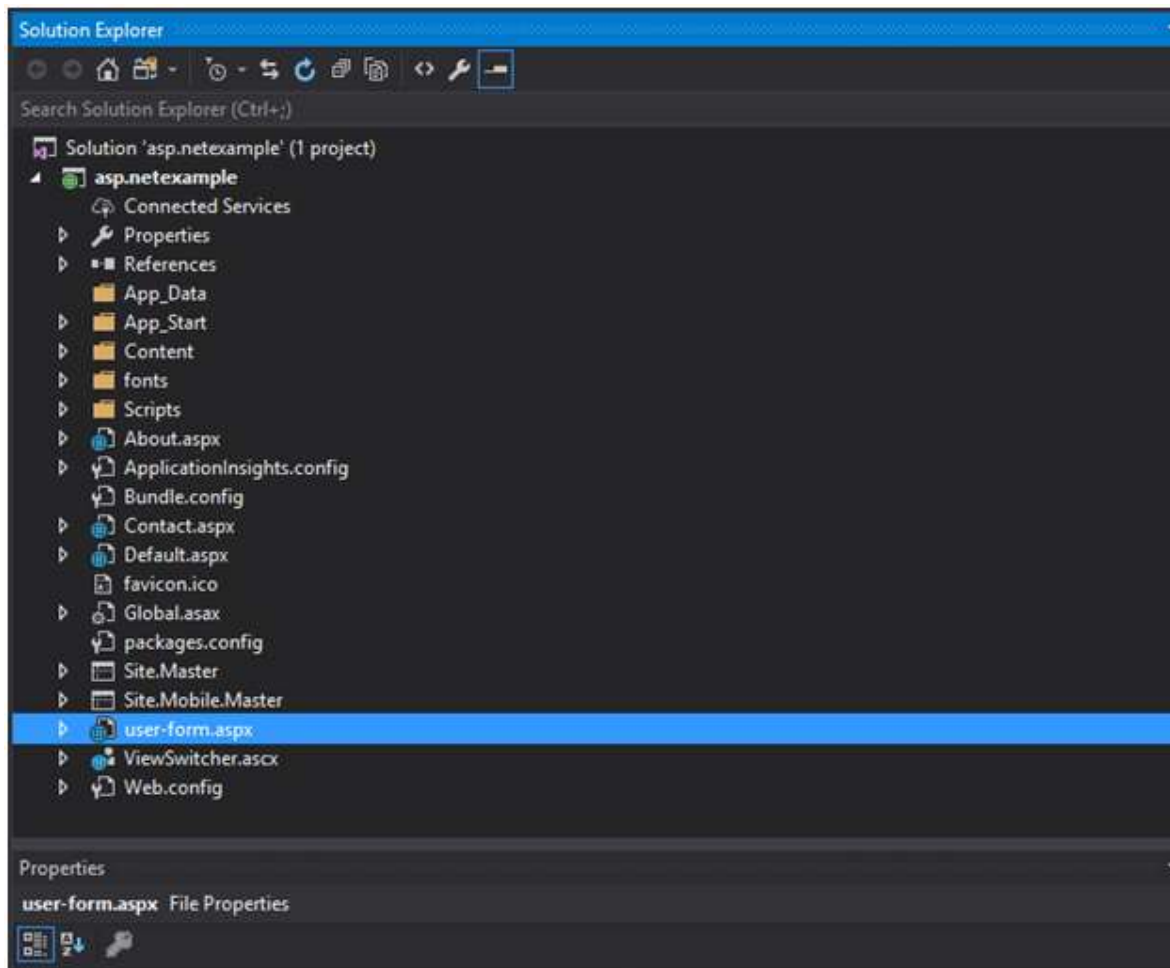


Select web forms option in left corner and then select web form and hit add button.

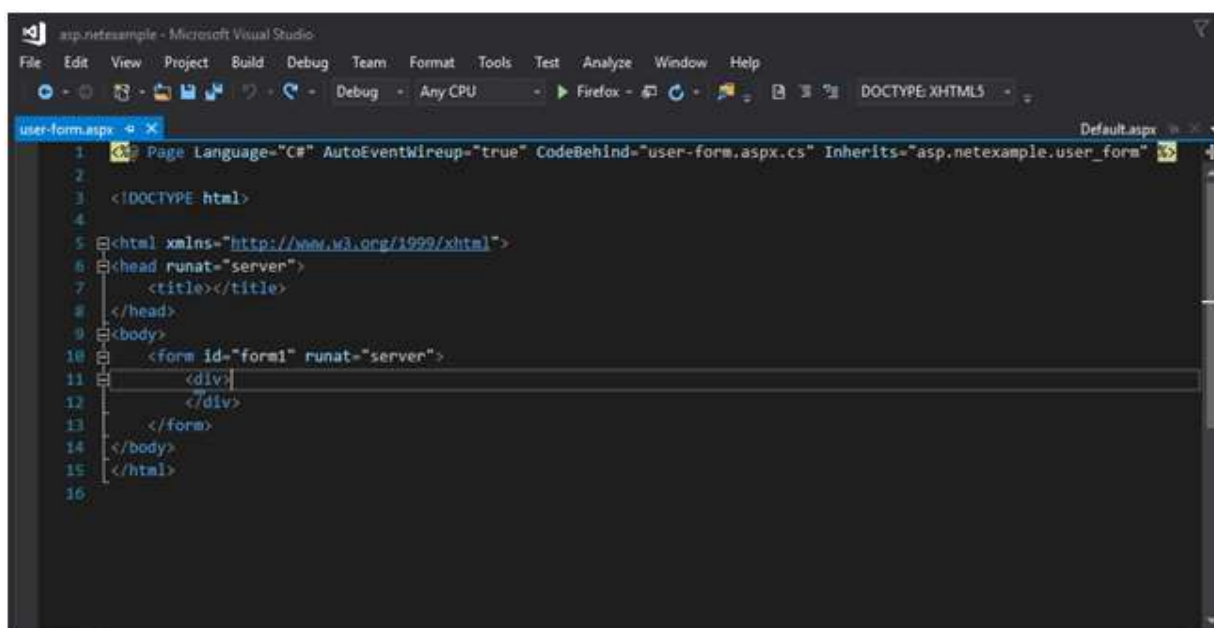


Now click on the add button and this form will add to our project.

After adding form, we can see that this is now in our project as we have shown in the below image.



Double click on this form and this will show some auto generated code like this:



// user-form.aspx

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="user-form.aspx.cs"
2. Inherits="asp.netexample.user\_form" %>
3. <!DOCTYPE html>
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. </div>
12. </form>
13. </body>
14. </html>

If we run this file on the browser, it does not show any output. So, let's print some message by this form.

The modified code is as below.

#### **// user-form.aspx**

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="user-form.aspx.cs"
2. Inherits="asp.netexample.user\_form" %>
3. <!DOCTYPE html>
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. <h2>Welcome to the Web Forms!</h2>
12. </div>
13. </form>
14. </body>
15. </html>

After running it on the browser it yields the following output.



Apart from this message, we can do lots and add controls to this page as well. We will add controls to form in our next chapters.

## ASP.NET Web Forms Server Controls

ASP.NET provides web forms controls that are used to create HTML components. These controls are categories as server and client based. The following table contains the server controls for the web forms.

Control Name	Applicable Events	Description
Label	None	It is used to display text on the HTML page.
TextBox	TextChanged	It is used to create a text input in the form.
Button	Click, Command	It is used to create a button.
LinkButton	Click, Command	It is used to create a button that looks similar to the hyperlink.
ImageButton	Click	It is used to create an imagesButton. Here, an image works as a Button.
Hyperlink	None	It is used to create a hyperlink control that responds to a click event.
DropDownList	SelectedIndexChanged	It is used to create a dropdown list control.
ListBox	SelectedIndexCnhaged	It is used to create a ListBox control like the HTML control.
DataGrid	CancelCommand, EditCommand, DeleteCommand, ItemCommand, SelectedIndexChanged, PageIndexChanged, SortCommand, UpdateCommand, ItemCreated, ItemDataBound	It used to create a frid that is used to show data. We can also perform paging, sorting, and formatting very easily with this control.
DataList	CancelCommand, EditCommand, DeleteCommand, ItemCommand, SelectedIndexChanged, UpdateCommand, ItemCreated, ItemDataBound	It is used to create datalist that is non-tabular and used to show data.
CheckBox	CheckChanged	It is used to create checkbox.
CheckBoxList	SelectedIndexChanged	It is used to create a group of check boxes that all work together.

RadioButton	CheckChanged	It is used to create radio button.
RadioButtonList	SelectedIndexChanged	It is used to create a group of radio button controls that all work together.
Image	None	It is used to show image within the page.
Panel	None	It is used to create a panel that works as a container.
Placeholder	None	It is used to set placeholder for the control.
Calendar	SelectionChanged, VisibleMonthChanged, DayRender	It is used to create a calendar. We can set the default date, move forward and backward etc.
AdRotator	AdCreated	It allows us to specify a list of ads to display. Each time the user re-displays the page.
Table	None	It is used to create table.
XML	None	It is used to display XML documents within the HTML.
Literal	None	It is like a label in that it displays a literal, but allows us to create new literals at runtime and place them into this control.

## ASP.NET Web Forms Label

This control is used to display textual information on the web forms. It is mainly used to create caption for the other controls like: textbox.

To create **label** either we can write code or use the drag and drop facility of visual studio 2017.

This is server side control, asp provides own tag to create label. The example is given below.

1. `<asp:LabelID="Label1" runat="server" Text="Label" ></asp:Label>`

This control has its own properties that are tabled below.

Property	Description
AccessKey	It is used to set keyboard shortcut for the label.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the label.
BorderColor	It is used to set border color of the label.
BorderWidth	It is used to set width of border of the label.
Font	It is used to set font for the label text.
ForeColor	It is used to set color of the label text.
Text	It is used to set text to be shown for the label.
ToolTip	It displays the text when mouse is over the label.
Visible	To set visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control.

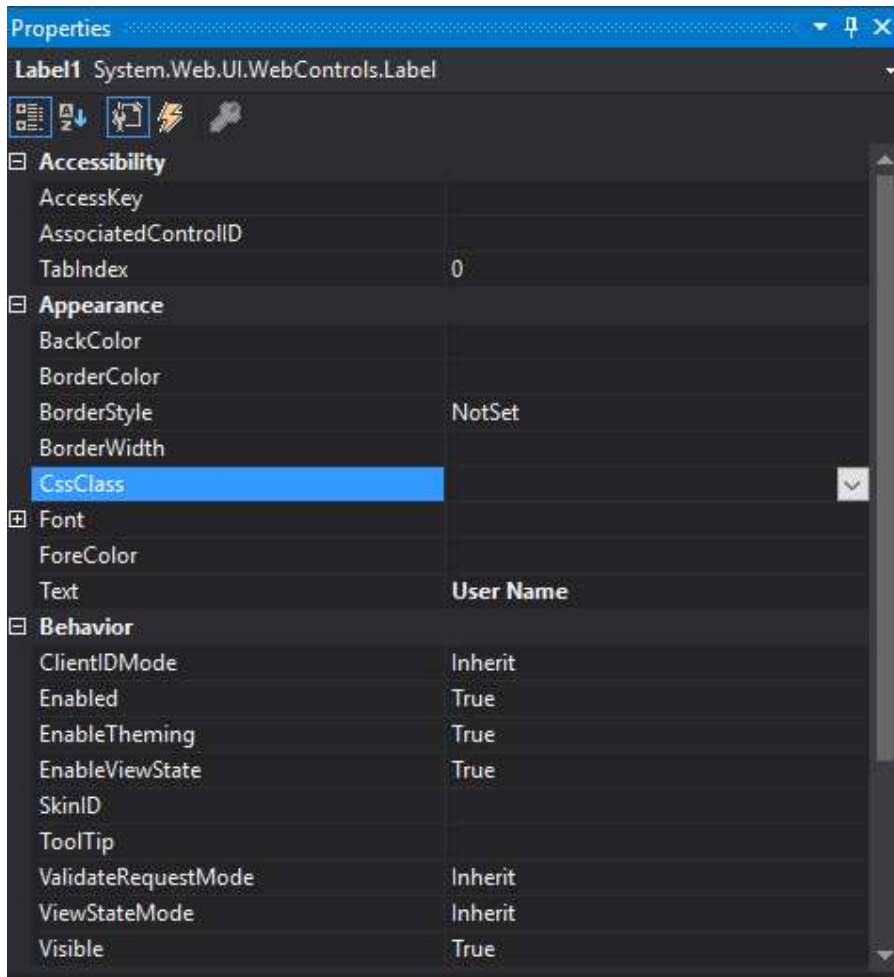
## Example



## // WebControls.aspx

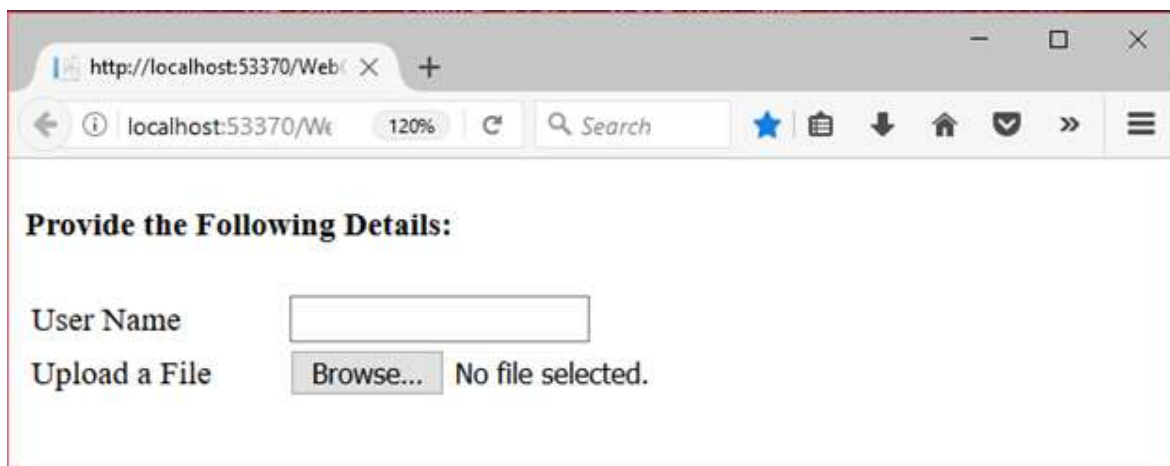
```
1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebControls.aspx.cs"
2. Inherits="WebFormsControlls.WebControls" %>
3. <!DOCTYPE html>
4. <html xmlns="http://www.w3.org/1999/xhtml">
5. <head runat="server">
6.     <title></title>
7.     <style type="text/css">
8.         .auto-style1 {
9.             width: 100%;
10.        }
11.        .auto-style2 {
12.            margin-left: 0px;
13.        }
14.        .auto-style3 {
15.            width: 121px;
16.        }
17.    </style>
18. </head>
19. <body>
20.     <form id="form1" runat="server">
21.         <div>
22.             <h4>Provide the Following Details:</h4>
23.             <table class="auto-style1">
24.                 <tr>
25.                     <td class="auto-style3">
26.                         <asp:Label ID="Label1" runat="server" Text="User Name"></asp:Label></td>
27.                     <td>
28.                         <asp:TextBox ID="TextBox1" runat="server" CssClass="auto-
style2"></asp:TextBox></td>
29.                 </tr>
30.                 <tr>
31.                     <td class="auto-style3">
32.                         <asp:Label ID="Label2" runat="server" Text="Upload a File"></asp:Label></td>
33.                     <td>
34.                         <asp:FileUpload ID="FileUpload1" runat="server" /></td>
35.                 </tr>
36.             </table>
37.         </div>
38.     </form>
39. </body>
40. </html>
```

This is a property window of label control.



Output:

Here, we have used label control with two different controls. It produces the following output.



## ASP.NET Web Forms TextBox

This is an input control which is used to take user input. To create **TextBox** either we can write code or use the drag and drop facility of visual studio IDE.

This is server side control, asp provides own tag to create it. The example is given below.

1. `<asp:TextBoxID="TextBox1" runat="server" ></asp:TextBox>`

Server renders it as the HTML control and produces the following code to the browser.

1. `<input name="TextBox1" id="TextBox1" type="text">`

This control has its own properties that are tabled below.

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
BorderWidth	It is used to set width of border of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control.
MaxLength	It is used to set maximum number of characters that can be entered.
ReadOnly	It is used to make control readonly.

## Example

// **WebControls.aspx**

1. `<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebControls.aspx.cs"`
2. `Inherits="WebFormsControls.WebControls" %>`
3. `<!DOCTYPE html>`
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. `<asp:Label ID="labelId" runat="server">User Name</asp:Label>`
12. `<asp:TextBox ID="UserName" runat="server" ToolTip="Enter User Name"></asp:TextBox>`
13. `</div>`
14. `<p>`
15. `<asp:Button ID="SubmitButton" runat="server" Text="Submit" OnClick="SubmitButton_Click" />`
16. `</p>`
17. `<br />`
18. `</form>`
19. `<asp:Label ID="userInput" runat="server"></asp:Label>`
20. `</body>`
21. `</html>`

[Code Behind](#)

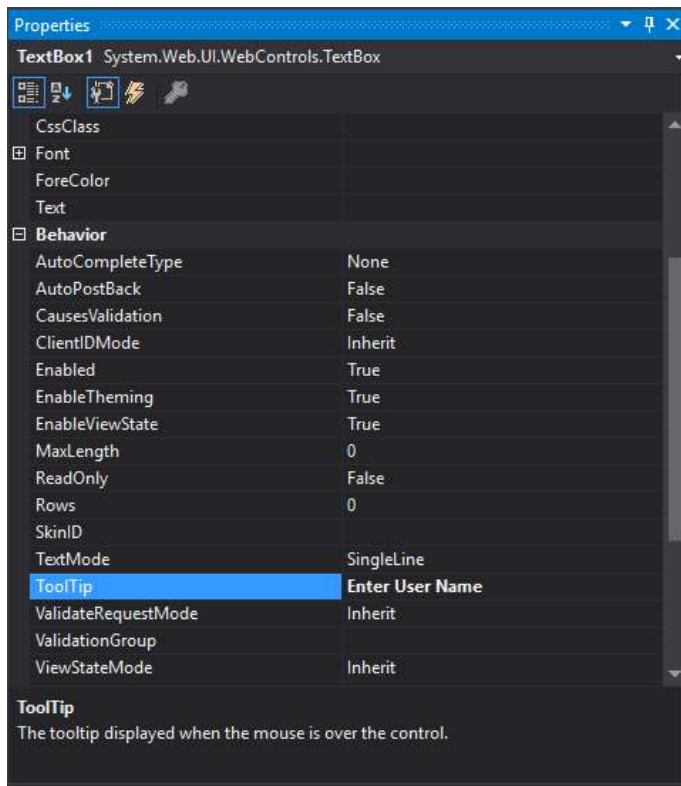
// **WebControls.aspx.cs**

```

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. namespace WebFormsControls
8. {
9.     public partial class WebControls : System.Web.UI.Page
10.    {
11.        protected void SubmitButton_Click(object sender, EventArgs e)
12.        {
13.            userInput.Text = UserName.Text;
14.        }
15.    }
16. }

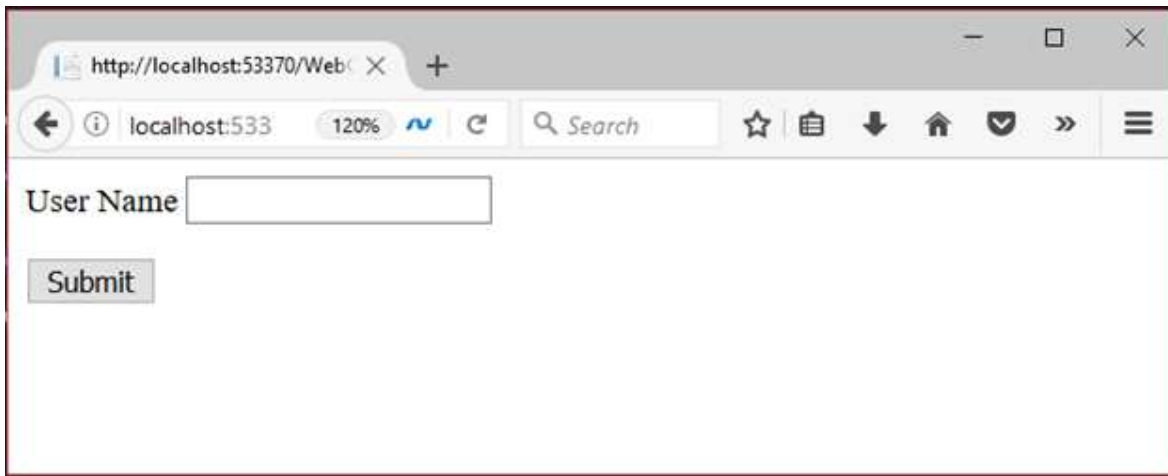
```

This is a property window of the TextBox control.

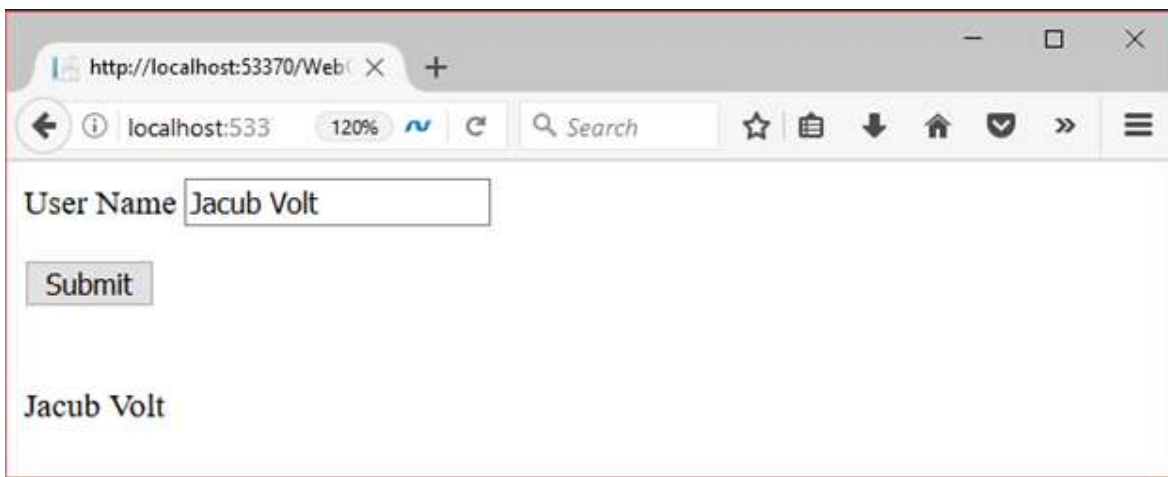


Output:

It produces the following output.



It displays user input, when user submits the input to the server. The following screen shot taking and showing user input.



## ASP.NET Web Forms Button

This control is used to perform events. It is also used to submit client request to the server. To create **Button** either we can write code or use the drag and drop facility of visual studio IDE.

This is a server side control and asp provides own tag to create it. The example is given below.

1. `<asp:ButtonID="Button1" runat="server" Text="Submit" BorderStyle="Solid" ToolTip="Submit"/>`

Server renders it as the HTML control and produces the following code to the browser.

1. `<input name="Button1" value="Submit" id="Button1" title="Submit" style="border-style:Solid;" type="submit">`

This control has its own properties that are tabled below.

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the control.

BorderColor	It is used to set border color of the control.
BorderWidth	It is used to set width of border of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control.

## Example

### // WebControls.aspx

```

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebControls.aspx.cs"
2. Inherits="WebFormsControls.WebControls" %>
3. <!DOCTYPE html>
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.             <asp:Button ID="Button1" runat="server" Text="Click here" OnClick="Button1_Click" />
12.         </div>
13.     </form>
14.     <br />
15.     <asp:Label ID="Label1" runat="server"></asp:Label>
16. </body>
17. </html>

```

### Code Behind

### // WebControls.aspx.cs

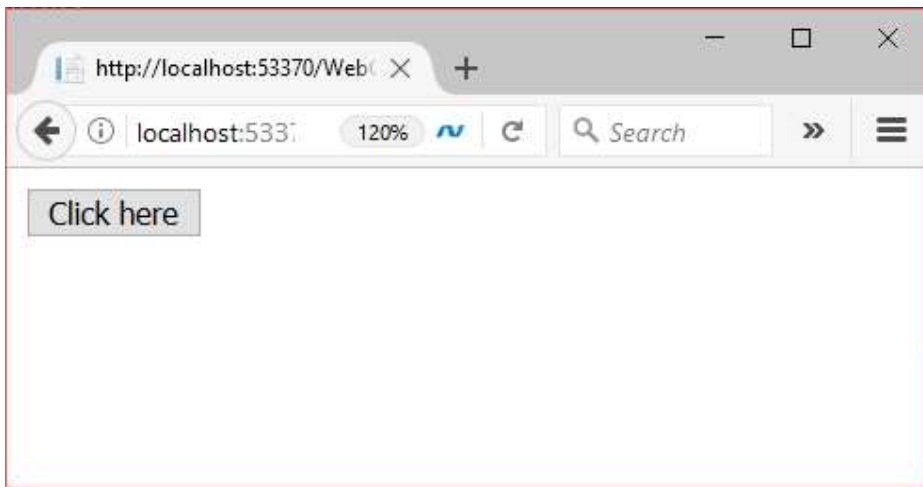
```

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. namespace WebFormsControls
8. {
9.     public partial class WebControls : System.Web.UI.Page
10.    {
11.        protected void Button1_Click(object sender, EventArgs e)
12.        {
13.            Label1.Text = "You Clicked the Button.";
14.        }
15.    }
16. }

```

Output:

It produces the following output.



This button displays a message when clicked, as shown below.

