Dynamic Web Pages Using ASP.NET

1. Introduction:

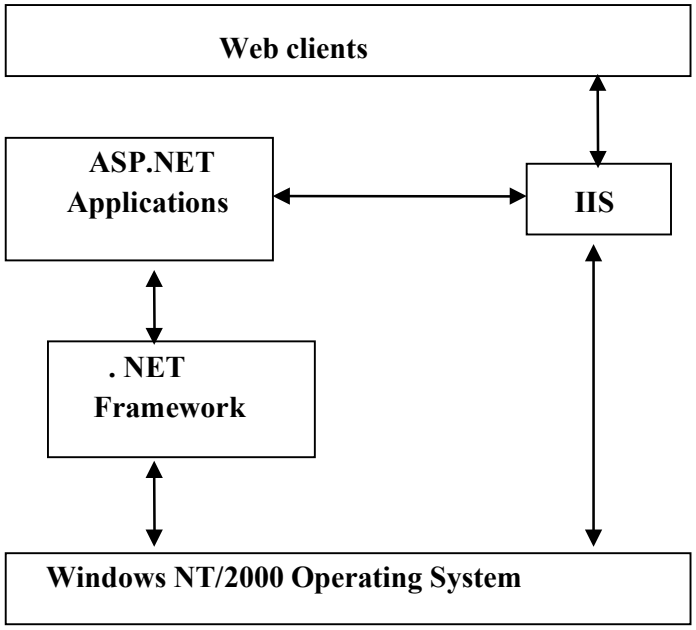
ASP.NET is a programming framework built on the .NET. It is a powerful tool for creating dynamic and interactive web pages. ASP.NET is an environment for building and delivering Web-server-based applications. It offers features for Web application development, deployment, and maintenance. ASP.NET takes the dynamic Web application to the next level, offering Windows-like functionality and ease of use in the most complex applications.

ASP.NET is the latest version of ASP(Active Server Pages). ASP is a server-side web technology for building dynamic, interactive, and database-driven websites. It is a product of Microsoft and one of the most popular technologies for developing web applications and websites.

2. ASP.Net Features:

- The ASP.NET page framework is a programming framework that runs on a web server to produce and manage ASP.NET web form pages dynamically
- These pages are extensions of standard HTML forms. They render dynamic, interactive and database-driven content
- The ASP.NET page framework creates an abstraction of the traditional client-server web interaction so that the user can program an application using traditional methods and tools that support rapid application development and object-oriented programming
- Web form pages can expose HTML elements as server-side objects with properties, methods, and events
- The ASP.NET page framework and web forms pages also support ASP.NET server controls that encapsulate the common UI(User Interface) functionality, easy-to-use and reusable controls

3. Architecture of ASP.NET:



- All web clients communicate with ASP.NET applications through IIS (Internet Information Services)
- IIS deciphers and optionally authenticates the request.
- If allow Anonymous option is turned on, no authentication takes place
- IIS finds the requested resource and if the client is authorized, returns the appropriate resource
- An ASP.NET application can also use the low-level security features of the .NET framework

4. Understanding ASP.NET Controls:

The ASP.NET Framework contains over 90 controls. These controls can be divided into eight groups:

- **Standard Controls**— The standard controls enable you to render standard form elements such as buttons, input fields, and labels.
- **Validation Controls** -The validation controls enable you to validate form data before you submit the data to the server. For example, you can use a RequiredFieldValidator control to check whether a user entered a value for a required input field.
- **Rich Controls**—The rich controls enable you to render things such as calendars, file upload buttons, rotating banner advertisements, and multi-step wizards.
- **Data Controls**—The data controls enable you to work with data such as database data. For example, you can use these controls to submit new records to a database table or display a list of database records.
- **Navigation Controls**—The navigation controls enable you to display standard navigation elements such as menus, treeviews, and bread crumb trails.
- **Login Controls**—The login controls enable you to display login, change password, and registration forms.
- **HTML Controls**— The HTML controls enable you to convert any HTML Tag into a Server-side control.

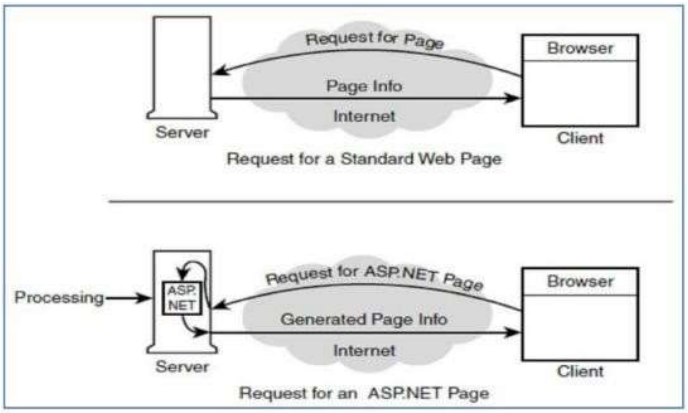
5. Understanding ASP.Net Applications:

- ASP.NET is a server-side scripting technology that enables scripts (embedded in web pages) to be executed by an Internet server.
- ASP.NET is a Microsoft Technology
- ASP stands for Active Server Pages
- ASP.NET is a program that runs inside IIS
- IIS (Internet Information Services) is Microsoft's Internet server
- IIS comes as a free component with Windows servers
- IIS is also a part of Windows 2000 and XP Professional
  ASP.NET File contains HTML, XML, and scripts and has the file extension ".aspx"

5.1 How does it work?

Step 1: When a browser requests an HTML file, the server returns the file.

Step 2: When a browser requests an ASP.NET file, IIS passes the request to the ASP.NET engine on the server.
Step 3: The ASP.NET engine reads the file, line by line, and executes the scripts in the file.
Step 4: ASP.NET file is returned to the browser as plain HTML

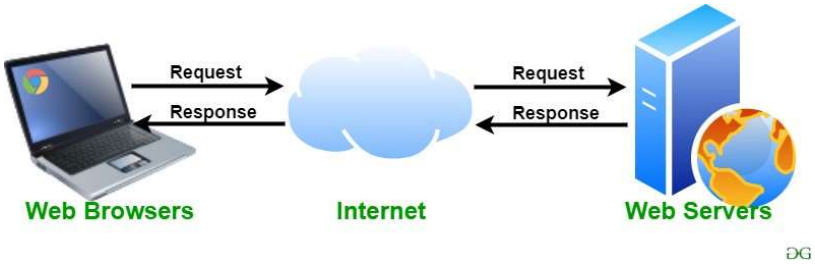


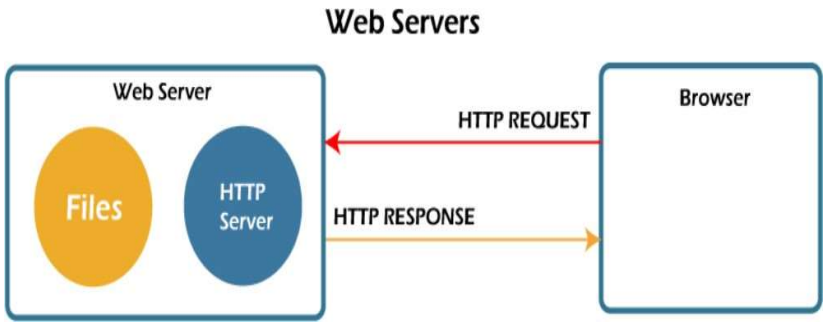
5.2 Different Parts of ASP.Net Applications:
The parts of an ASP.NET Web application include:
Web Forms, or .aspx pages: Web Forms and .aspx pages provide the UI for the Web application.
Code-behind pages: Code-behind pages are associated with Web Forms and contain the server-side code for the Web Form.
Configuration files: Configuration files are XML files that define the default settings for the Web application and the Web server. Every Web application has one Web. Config configuration file. In addition, each Web server has one machine.config file.
Global.asax file: Global.asax files contain the needed code for responding to application-level events that are raised by ASP.NET.
XML Web service links: XML Web service links allow the Web application to send and receive data from an XML Web service.
Database connectivity: Database connectivity allows the Web application to transfer data to and from database sources.
Caching: Caching allows the Web application to return Web Forms and data more quickly after the first request.

6. Web Servers:

- A web server is a dedicated computer responsible for running websites sitting out on those computers somewhere on the Internet.
- They are specialized programs that circulate web pages as summoned by the user.
- The primary objective of any web server is to collect, process and provide web pages to the users.
- Web server is a program that processes the network requests of the users and serves them with files that create web pages.
- This exchange takes place using Hypertext Transfer Protocol (HTTP).
- Basically, web servers are computers used to store HTTP files that makes a website and when a client requests a certain website, it delivers the requested website to the client.
- The web server can be any software or hardware but is usually software running on a computer.

- One web server can handle multiple users at any given time.
- A web server is never disconnected from the internet.





There are many web servers available in the market both free and paid. Some of them are described below:

a. Apache HTTP server:

- It is the most popular web server and about 60 percent of the world's web server machines run this web server.
- The Apache HTTP web server was developed by the Apache Software Foundation.
- The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows.
- It is an open-source software which means that we can access and make changes to its code and mold it according to our preference.
- The Apache Web Server can be installed and operated easily on almost all operating systems like Linux, MacOS, Windows, etc.

b. Microsoft Internet Information Services (IIS):

- IIS (Internet Information Services) is a high-performing web server developed by Microsoft.
- Internet Information Services (IIS) for Windows® Server is a flexible, secure, and manageable Web server for hosting anything on the web.
- It is strongly united with the operating system and is therefore relatively easier to administer.
- It is developed by Microsoft, and it has a good customer support system which is easier to access if we encounter any issue with the server.
- It has all the features of the Apache HTTP Server except that it is not an open-source software and therefore its code is inaccessible which means that we cannot make changes in the code to suit our needs.

c. Lighttpd:

- Lighttpd is pronounced as 'Lightly'.
- secure, fast, compliant, and very flexible web server that has been optimized for high-performance environments.
- It currently runs about 0.1 percent of the world's websites.
- Lighttpd has a small CPU load and is therefore comparatively easier to run.
- It has a low memory footprint and hence in comparison to the other web servers, requires less memory space to run which is always an advantage.

d. Jigsaw Server

- Jigsaw has been written in the Java language and it can run CGI (common gateway interference) scripts as well as PHP programs.
- It is not a full-fledged server and was developed as an experimental server to demonstrate the new web protocols.
- It is an open-source software which means that we can access its code and add changes to it according to our needs and then upload our own module (the changed code).

e. Sun Java System:

- Sun Java System Web Server 6.1 is a multi-process, multi-threaded, secure web server built on open standards. It provides high performance, reliability, scalability, and manageability for enterprises of any size.
- The Sun Java System supports various languages, scripts, and technologies required for Web 2.0 such as Python, PHP, etc.
- It is not an open-source software and therefore its code is inaccessible which means that we cannot make changes in the code to suit our needs.

6.1 Installation of IIS:

Steps to follow while installing IIS server in windows operating system.

1. On the **Start** page, click the **Control Panel** tile.
2. In **Control Panel**, click **Programs**, and then click **Turn Windows features on or off**.
3. In the **Windows Features** dialog box, click **Internet Information Services** to install the default features.
4. Expand the **Application Development Features** node and click **ASP.NET 4.5** to add the features that support ASP.NET. (If you installed **.NET 3.5**, select **ASP.NET 3.5** also.)

The following additional features are automatically selected:

- **.NET Extensibility 4.5**
- **ISAPI Extensions**
- **ISAPI Filters**
- **.NET Extensibility 3.5** (If **ASP.NET 3.5** was selected)

5. Click **OK** to close the **Windows Features** dialog box.
6. To verify that IIS installed successfully, type the following into a web browser:

http://localhost

The default IIS Welcome page is displayed.

7. Web forms:

Web Forms are Webpages that serve as the user interface for a Web application. An ASP.NET Web application comprises one or more Web Forms. A Web Form is a dynamic page that can access server resources.

An ASP.NET Web Form, conversely, can also run server-side code to access a database, to generate additional Web Forms, or take advantage of built-in security on the server. Web Form does not rely on client-side scripting, it is not dependent on the client‘s browser type or operating system. This independence allows you to develop a single Web Form that can be viewed on practically any device that has Internet access and a Web browser.

7.1 The Purpose of Web Forms:

Web Forms and ASP.NET were created to overcome some of the limitations of ASP.

These new strengths include:

- Separations of HTML interface from application logic.
- A rich set of server-side controls that can detect the browser and send out appropriate markup language such as HTML.

- Less code to write due to the data binding capabilities of the new server-side.NET controls
- Event-based programming model that is familiar to Microsoft Visual Basic programmers
- Compiled code and support for multiple languages, as opposed to ASP which was interpretedas Microsoft Visual Basic Scripting (VBScript) or Microsoft JScript
- Allows third parties to create controls that provide additional functionality

On the surface, Web Forms seem just like a workspace where you draw controls. In reality, they can do a whole lot more. But normally you will just place any of the various controls onto the WebForm to create your UI. The controls you use determine which properties, events, and methods youwill get for each control. There are two types of controls that you can use to create your user interface: HTML controls and Web Form controls.

7.2. web form controls:

A control is an object that can be drawn onto the Web Form to enable or enhance user interactionwith the application. Examples of these controls include Text Boxes, Buttons, Labels, Radio Buttons, etc. All these Web server controls are based on the **_System.Web. UI.Control** class.

The ASP.NET Framework contains over 90 controls. There are two ways to work with the controls—in the client browser (HTML client controls) and in the server (HTML server controls and Web-Server controls). In the client, to handle these controls, you must use a scripting language, such as JavaScript. You can make these controls run at the server and so handle them with some server scripting language like Visual Basic or C#. So HTML controls can be categorized in client-side controls and server-side controls; whereas Web-Controls are always server-side controls. Let‘s see what we mean here by server controls and client controls:

7.3 server controls:

Server controls have the capability to be processed and manipulated by a server machine rather than aclient machine. Server controls provide .NET native server-side programming support.

Server controls are tags that are understood by the server. The types of server controls are

- HTML server controls (HTML tags)
- Web controls (ASP.NET tags)

HTML Server Controls:

These are HTML tags understood by the server. By default, ASP.NET treats them as text To make these tags programmable, add **runat=“server”** to the HTML tags. It indicates that the element should be treated as a server control. All HTML server controls must be within the form tag with the runat=“server” attribute.

- This attribute indicates that the form should be processed at the server
- The enclosed controls can be accessed by server scripts

Web Controls:

These are special ASP.NET tags that are understood by the server. These are also created by the server and require **runat=“server”** attribute.

Syntax to create web form controls

<asp:control_name id=“name” runat=“server”/>

E.g. Button, Checkbox, List box, radio button

Types of Web Controls

There are four types of web controls. They are

- Web server controls
- Validation web controls
- Data controls
- Rich web controls

Web server Controls

These controls are also created on the server and they require runat=”server” attribute to work. The syntax for creating a web server control is

<asp:control_name id=“name” runat=“server”/>

Ex: Button, Checkbox, List box, radio button

<asp:Label id="myDateLabel" runat="server" />

Validation Web Controls:

These are used for form validation. They do not correspond to any HTML tag. They includes:

- RequiredFieldValidator
- CompareValidator
- RangeValidator
- RegularExpressionValidator
- CustomValidator
- ValidationSummary

Data Controls

These controls are used to display data from a data source such as a table. They include

- DataGrid
- DataList
- DataRepeater

Rich Web Controls

These are customized controls that provide high-level functionality

- Calendar
- AdRotator

Program Example for Server Control:

Program name: ClickTime.aspx

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

<script runat="server">
protected void Button1_Click(object Source, EventArgs e)
{
DateTime currDate = new DateTime();
```

```
currDate = DateTime.Now;
myDateLabel.Text = currDate.ToString();
}
</script>
<html>
<head>
<title>Date and Time</title>
</head>
<body>
<h3 align="center"> The time on server is:
</h3>
<form runat="server">
<center>
<asp:Label id="myDateLabel" runat="server" />
<br /><br />
<asp:Button id="Button1" runat="server" Text="Update" onclick="Button1_Click"
/>
</center>
</form>
</body>
</html>
```

When you run this program and click the button; this program will always show you the time of the machine where the website is hosted (i.e. server machine); because all the processing is being done on the server machine by the C#. But note that if you are testing your website on your machine on some web server software then you will only see the time of your machine; because at that timeyour machine is acting as a server. Also note that here we used two server-controls: ASP button and ASP label; they will be processed at the server machine by C#; because they are server controls.

7.4 Client controls:

Standard HTML controls are client controls. Client controls are the same HTML 4.01 controls thathave been part of the HTML language for a very long time. Use these when their native client-side functionality serves your purpose (lowest overhead with these controls).

When you add an HTML control to a Web form, by default, it's an HTML client control and only available for scripting in the Web page itself. To make it available to script code in the browser, you must give it an ID value, which you do with the (id) property in the Properties window; you can then refer to the control by ID in your script code.

Let‘s build an example with these types of client controls, but remember these controls can‘t be controlled with the help of c# or vb; because these are server-scripting languages. To control thesetypesof client controls we can use JavaScript, which runs of the client browser; as shown in the following example:

Program example for client control:

Program name: ClickTime.aspx

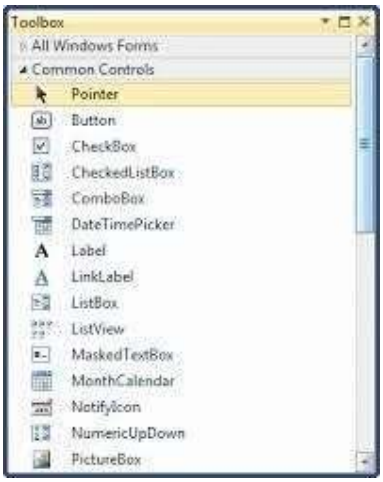
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<script type="text/javascript">function show_alert()
```

```
{
var now = new Date();
var hour = now.getHours();
var minute = now.getMinutes(); var second = now.getSeconds();
var timeString = hour + ':' + minute + ':' + second;
document.getElementById("lb1").innerText=timeString;
}
</script>
<html>
<head>
<title>Web Technology</title>
</head>
<body>
<label id="lb1"></label>
<input id="Button1" type="button" value="Show time of client machine"
onclick="show_alert()" />
</body>
</html>
```

When you run this program and click the button; this program will always show you the time of the machine where the browser is running (i.e. client machine); because all the processing is being done on the client machine by the Java script. But note that here we used two client-controls: HTML button and HTML label; they will be processed at the client machine by Java script; because they are client controls. You do even don ‘t have to have an internet connection for this event listener action mechanism. Because the action is transferred with an HTML file first user opens the page. To see what ‘s actually happing view the source (Menu View -> Source) of the webpage while running it in the browser. In one line, client controls will be fully processed on the client machine.

7.5 Adding controls to a web form:

The Toolbox window is used to add controls to a form (if the Toolbox is not displayed, you can open it byselecting View_Toolbox).



Tool Box

You add controls from the toolbox in the Design view of the Web Forms page (the ASPX

file). The toolbox categorizes the different types of controls in separate tabs, such as Web Forms, HTML, Components, and Data. You can use the HTML tab to add HTML controls and use the Web Formstab to add the ASP.NET server controls to Web Forms. However, to make the HTML controls available for coding at the server end, these controls need to be converted to server controls.

You can also add a Web control to a page by using the ASP.NET code. You can access the ASP.NET code in the HTML view of the page (ASPX file). The actual syntax depends on the type of control that you want to add. The syntax used to add an HTML TextBox control is given as follows:

<input id="Text1" Type=text runat="server">

You can add ASP.NET server controls by using an Extensible Markup Language (XML) tag referenced as asp. When you add an ASP.NET TextBox control, the syntax is:

<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>

Note: when you use the toolbox to add Web controls in the Design view, the corresponding ASP.NETsyntax is automatically generated.

Adding Controls at Run-time:

You can also programmatically add a control at run time. The following code to demonstrate creates a textbox using programmatically.

```
TextBox txt = new TextBox();
txt.Text = “”;
txt.ID = "que1";
pnlQuestions.Controls.Add(txt);
```

Every control has specific properties and methods. You can set control properties to modify the appearance or behavior of controls. For example, you can set the font, color, and size of the control.You can use control methods to perform a specific task, such as moving control. You can setcontrol properties at design times by using the Properties window or at run time by using the code.Every control has a property called ID that is used for the unique identification of the control. Youcan set the property of a control at run time by using the following syntax:

ControlID.PropertyName=Value

In this syntax:

- **ControlID** represents the ID property of the control.
- **PropertyName** represents the control property.
- **Value** represents the value assigned to PropertyName, which is a control's property.

7.6 Common Properties of Control:

These properties are common to all the web controls. They can be specified in the control‘s tag(**<asp: ControlName>)** and in ASP.NET code using **ControlName.PropertyName**

1. *AccessKey:*

AccessKey allows for specifying a control‘s keyboard accelerator key using a singleletter whilepressing ALT. Access keys are not supported by some browsers.

2. *BackColor:*

The BackColor property specifies the color behind the control. This property acceptsstandard HTMLcoloridentifiers, like—blue‖ or an —yellow‖, or an RGBvalue, like —#CC00CC‖.

3. *BorderWidth:*

This property lets you set the size of the control‘s border in pixels. This property maynot

work insome browsers.

4. *BorderStyle:*

This property allows you to configure the control‘s border style. The possible valuesare: Dashed,Dotted, Double, Groove, Insert, None, NotSet, OutSet, Ridge and Solid.

5. *CssClass:*

This property specifies the Cascading Style Sheet (CSS) class to assign to the control.

6. *Enabled:*

Setting _Enabled‘ to true enables the control. Setting it to false disables it. Enableddenotes acontrol‘s active state on a form.

7. *Font:*

This property gets the control‘s font information.

8. *ForeColor:*

This property sets the color of a control‘s text. This property may not work in somebrowsers.

9. *Height:*

The Height property simply sets a control‘s height in pixels. This property may notwork in somebrowsers.

10. *TabIndex:*

The TabIndex for a control sets its order of focus as you press the tab button. Forexample, if theTabIndex is set to 5, the control will be the sixth control to get focuswhen you press the tab key.

11. *ToolTip:*

Define the text in the Tooltip with this property. A ToolTip appears when the userhovers themouse pointer over a control. This property doesn‘t work in somebrowsers.

12. *Width:*

The Width property simply sets the width of a control in pixels. It may not work insome browsers.

13. *Visible:*

Sets if the control will be visible on the form or not.

8. Running a Web Application:

To run the asp.net applications we have several methods.
Method 1: In VB.Net IDE, go to Toolbar, next select Start Debugging Icon.
Method 2: In VB.Net IDE, go to Menu bat, next select Debug menu and then select Start debugging submenu.
Method3: Click function button F5

9. Creating Multiform Web Project:

To create a multiform web project, we suppose to follow several steps:
Step 1: Open the ASP.Net application and then go to Solution Explorer
Step 2: Right-click on the project and click on Add -> new Item.
Step 3: Add new Item window gets open. In installed templates click on web and then select web form from the right panel.
Step 4: Specify the name for the new web form and then Click on Add button.

10. STANDARDCONTROLS:

Introduction

The System.Web.UI.Control class is the base class for all web server controls. This class is directly derived from the Object class that resides in the System namespace. The hierarchy of the Control class is

System.Object System.Web.UI.Control

All the web server controls exist within the System.Web.UI.WebControl's namespace and can be used on any web form. The hierarchy of WebControl class is
System.Object System.Web.UI.Control
System.Web.UI.WebControls.WebControl

The public properties of WebControl class are:

AccessKey, Attributes, BackColor, BorderColor, BorderStyle, BorderWidth, CssClass, Enabled, Font, EnableTheming, ForeColor, Height, IsEnabled, SkinID, Style, TabIndex, ToolTip, Width

10.1 TextBox Control:

The TextBox web server control is used to accept input. It allows the user to enter text in the control. It exists within System.Web. UI.WebControl namespace. When we add a TextBox control in a web form, the following HTML code is added to the web form
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

Property	Description
AutoPostBack	A Boolean value that specifies whether the control is automatically posted back to the server when the
CausesValidation	Specifies if a page is validated when a Postback
Columns	The width of the textbox
MaxLength	The maximum number of characters allowed in the
ReadOnly	Specifies whether or not the text in the text box can
Rows	The height of the textbox (only used if
Runat	Specifies that the control is a server control. Must be set to
Text	The contents of the textbox

TextMode	Specifies the behavior mode of a TextBox control (SingleLine,MultiLine or Password)
OnTextChanged	The name of the function to be executed when the text in the

10.2 Button Control

The Button control is used to display a push button. The push button may be a submit button or a command button. By default, this control is a submit button.
A submit button does not have a command name and it posts the Web page back to the server when it is clicked. It is possible to write an event handler to control the actions performed when the submit button is clicked.
A command button has a command name and allows you to create multiple Button controls on a page. It is possible to write an event handler to control the actions performed when the command button is clicked.

Property	Description
CausesValidation	Specifies if a page is validated when a button is clicked
CommandArgument	Specifies additional information about the command to perform
CommandName	Specifies the command associated with the Command event
OnClientClick	Specifies the name of the function to be executed whena button is clicked
PostBackUrl	Specifies the URL of the page to post to from the
Runat	Specifies that the control is a server control. Must beset to "server"
Text	Specifies the text on a button

10.3 Label Control

The Label control is used to display text on a page that a user cannot edit. The text is programmable i.e. lets you apply styles to its content

Property	Description
AssociatedControlID	Obtains or sets the identifier for a server control with which aLabel control is associated.
Text	The text to display in the label

10.4 Image Control

The Image control is used to display an image. It exists within System.Web.UI.WebControl namespace.

Property	Description
AlternateText	An alternate text for the image
DescriptionUrl	The location to a detailed description for the image
ImageAlign	Specifies the alignment of the image

ImageUrl	The URL of the image to display for the link

10.5 ImageButton Control

The ImageButton control is used to display a clickable image. This control allows the user to display images that can handle click events.

Property	Description
CausesValidation	Specifies if a page is validated when an ImageButton
CommandArgument	Additional information about the command to perform
CommandName	The command associated with the Command
GenerateEmptyAlternateText	Specifies whether or not the control creates an empty
OnClientClick	The name of the function to be executed when the
PostBackUrl	The URL of the page to post to from the current page

10.6 DropDownList:

The DropDownList control is used to create a drop-down list. The dropdown list displays data as a drop-down list from which you can make single selection. The user **cannot** select multiple items from the list because the list closes automatically once the selection is made. Each selectable item in a DropDownList control is defined by a ListItem element. This control supports data binding.

Property	Description
SelectedIndex	The index of a selected item
OnSelectedIndexChanged	The name of the function to be executed when the index of the selected item has
BorderColor	Sets a border color to the dropdown list
BorderStyle	Sets border style to the dropdown list
BorderWidth	Sets border style to the dropdown list

10.7 CheckBox Control

The CheckBox control is used to create a check box that the user can select by clicking. They display a check mark that allow a user to toggle between true and false.

Property	Description
CausesValidation	Specifies if a page is validated when a Button control is
Checked	Specifies whether the check box is checked or not
Runat	Specifies that the control is a server control. Must be set to
Text	The text next to the check box
TextAlign	On which side of the check box the text should appear (right
OnCheckedChanged	The name of the function to be executed when the Checked property

10.8 CheckBoxList Control

The CheckBoxList control is used to create a number of check boxes simultaneously. Each selectable item in a CheckBoxList control is defined by a ListItem element. This control is useful when we need to bind the data from data source to check box.

Property	Description
CellPadding	The amount of pixels between the border and the contents of the table cell
CellSpacing	The amount of pixels between table cells
RepeatColumns	The number of columns to use when displaying the check box
RepeatDirection	Specifies whether the check box group should be repeated
RepeatLayout	The layout of the check box group
Runat	Specifies that the control is a server control. Must be set to
TextAlign	On which side of the check box the text should appear

10.9 The RadioButton Control

Radio buttons provide a set of choices or options that you can select. Radio buttonstraditionally display themselves as a label with a dot to the left of it, which can beeither selected or not. Radiobuttons are controls that are generally used in groups.

When one button is selected, any others grouped with it are generally unselected. Assuch, they arehelpful when the user has a limited number of choices to make. Theyare also handy when you wantall the choices to be displayed—for example, whenselecting gender, male or female, or selecting marital status, single or married. Tocreate a radio button, you use the RadioButton class.

CheckBox also support following properties in addition to common propertiesdiscussed above:

Property	Description
Text	Represents the caption of the RadioButton control.
Checked	Enables you to get or set whether the RadioButton control is checked.
GroupName	It is used a group a set of radion buttons so only one of them can be selected at a time.

The RadioButton control supports the following method:

- **Focus**—Enables you to set the initial form focus to the RadionButton control.

Finally, the RadioButton control supports the following event:

- **CheckedChanged**—Raised on the server when the RadioButton is checked or unchecked.

10.10 AdRotator Control

ASP.NET supports banner advertisements. These advertisements are image files that a user clicks to make the browser navigate to the advertisement site. The AdRotator control is used to display advertisement banners in a web page. This control displays a sequence of images that can be pre-defined or random.

Property	Description
AdvertisementFile	Specifies the path to the XML file that contains ad
AlternateTextField	Specifies a data field to be used instead of the Alt text for
ImageUrlField	Specifies a data field to be used instead of the ImageURL attribute
KeywordFilter	Specifies a filter to limit ads after categories
NavigateUrlField	Specifies a data field to be used instead of the NavigateUrl
Runat	Specifies that the control is a server control. Must be set to "server"
Target	Specifies where to open the URL

10.11 Calender Control

The Calendar control is used to display a calendar in the browser. This control displays a one-month calendar that allows the user to select dates and move to the next and previous months.

Property	Description
Caption	The caption of the calendar
CaptionAlign	The alignment of the caption text
CellPadding	The space, in pixels, between the cell walls and contents
CellSpacing	The space, in pixels, between cells
DayHeaderStyle	The style for displaying the names of the days
DayNameFormat	The format for displaying the names of the days
DayStyle	The style for displaying days
FirstDayOfWeek	What should be the first day of week
NextMonthText	The text displayed for the next month link
NextPrevFormat	The format of the next and previous month links
NextPrevStyle	The style for displaying next and previous month links
OtherMonthDayStyle	The style for displaying days that are not in the current
PrevMonthText	The text displayed for the previous month link
SelectedDate	The selected date
SelectedDates	The selected dates
SelectedDayStyle	The style for selected days
SelectionMode	How a user is allowed to select dates
SelectMonthText	The text displayed for the month selection link
SelectorStyle	The style for the month and weeks selection links
SelectWeekText	The text displayed for the week selection link

ShowDayHeader	A Boolean value that specifies whether the days of the week

ShowNextPrevMonth	A Boolean value that specifies whether the next and previous
ShowTitle	A Boolean value that specifies whether the title of the calendar
TodayDayStyle	The style for today's date
TodaysDate	Today's date
VisibleDate	The date that specifies the month that is currently visible in the
WeekendDayStyle	The style for weekends

11. Form Validation:

ASP.NET helps us to build websites with web-forms. We use web-forms to display information and to gather information from a user. Validation is a set of rules that you apply to the data youcollect to make sure that the data is correct and in range according to your need.
So any application on your computer has the risk of bad inputted data, and one of the things that can be done to mitigate the risk is to validate that input, stopping the user from entering bad input in the first place.
We can divide validation in two categories: Client side validation and Server side validation; on the base of where the validation activity will be performed.

11.1 Client-Side Validation

Client-side validation involves check form values directly on the browser via JavaScript before passing the information on to the server. If the user is working with a browser that supports Dynamic HTML (DHTML), such as Internet Explorer 4.0 or higher, validation controls can perform validation using client script (i.e. javascript, vbscript etc).
In this example, add a textbox with ID —phone‖ and a button. As we know, phone number shouldconsist of only numbers; we can validate this thing on client side by some javascript function like shown in following program:

Program 25: ValScript.aspx

```
<%@ Page Language="C#" AutoEventWireup="True" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script type="text/javascript">
function validateMyForm(form)
{
if (parseInt(form.phone.value) != form.phone.value)
{
alert('Please enter a phone number, numbers only');form.phone.focus();
form.phone.select(); return false;
}
}
```

```
</script>
<head>
<title></title>
</head>
<body>
<form id="Form1" runat="server" onsubmit="return validateMyForm(this);">
<asp:TextBox ID="phone" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Button" />
</form>
</body>
</html>
```

As shown in figure this program will show a text box and a button. When youenter some text in textbox and click the button, the client side script will validate thetext for numbers and display a msgbox if rules are violated. Furthermore the form isnot submitted to the server if the rules are violated. All this validation is done withoutthe help of server, because all the needed things to
validate a control i.e. the controland the script are now on client machine.

Advantages

- We can provide instant feedback to the user. For example, if a user neglects to enter a valuein a required form field, you can instantly display an error message without requiring a roundtrip back to the server.
- There will be comparatively less load on the server as the computing of validation will beshifted to the client machine.

Disadvantages

- Browser Compatibility issue- First of all, not all browsers support client side scripting. Soobviously we cannot use this approach everywhere.
- Security Settings- Some folks turn off scripting because they see it as a security risk.

Forthese folks client side validation will simply not work.

- Easily Thwarted- client side validation is the more insecure form of validation. When a page is generated in an end user‘s browser, this end user can look at the code of the page quiteeasily (simply by right-clicking his mouse in the browser and selecting view-code). When hedoes this, in addition to seeing the HTML code for the page, he can also see all the JavaScriptthat is associated with the page. If you are validating your form client-side, it doesn‘t take much for the crafty hacker to repost a form (containing the values he wants in it).

11.2 Server-side Validation

After the user enters data into a Web form, clicks the Submit button, and sends theform data to theserver as a request, you can perform server-side validation on thedata. If you do this, it is called server-side validation because it occurs on the server.If the data is incorrect or not valid, you can send back a response stating this.

The bad thing about server-side validation is that it requires trips back and forth tothe server. Thistakes a lot of resources and makes for a slower-paced form for theuser. Nothing is more annoyingto a user who is on a dial-up connection than clickingthe Submit button on the form and then waiting for 20 seconds to find out that theydidn't enter their password correctly.

Advantages

- Server side validation means that the validation checks are performed on the server; so more secure as they cannot be bypassed easily.
- Browser-independent: As the processing is done on server, there is not client side coding that are dependent on browsers.
- More secure- No validation coding will be sent to browser, so no one can see the codingand post it again to bypass the validation security.

Disadvantages

- It requires trips back and forth to the server. This takes a lot of resources and makes for a slower-paced form for the user. Nothing is more annoying to a user who is on a dial- up connection than clicking the Submit button on the form and then waiting for 20 seconds to find out that they didn't enter their password correctly.
- The processing for all the clients to validate entries is done on server. So the load on the server will be comparatively more.

11.3 VALIDATION CONTROLS

Introduction

Validation controls are controls used to validate the data entered in an input control such as textbox on a web page. When the user enters invalid data into an associated control, the validation control displays an error message on the screen. The data entered by the user is validated each time it is entered, and the error message disappears only when the data is found valid. Validation controls save time and enhance the efficiency of the applications by validating the data before they are sent to the server for processing.

Here, we deal with the following validation controls.

- RequiredFieldValidator
- RangeValidator
- RegularExpressionValidator
- CompareValidator

- CustomValidator
- ValidationSummary

The BaseValidator Class:

The System.Web.UI.WebControls.BaseValidator class is the base class for all the validation controls. It provides the basic implementation needed for all the controls. The inheritance hierarchy for System.Web.UI.WebControls.BaseValidator class is as follows:

System.Object
System.Web.UI.WebControl
System.Web.UI.WebControls.WebControl System.Web.UI.WebControls.Label
System.Web.UI.WebControls.BaseValidator

Properties of BaseValidator Class

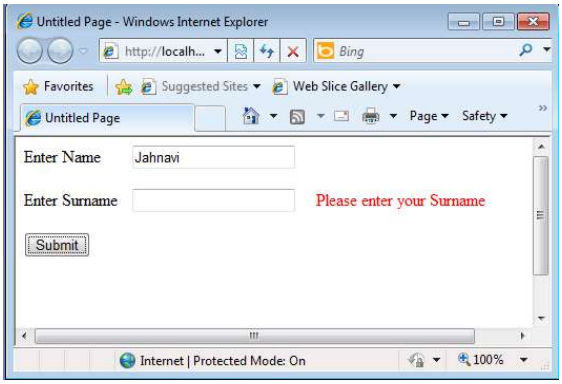
Property	Description
ControlToValidate	The id of the control to validate
Display	The display behavior for the validation control. Legal values are: • None (the control is not displayed. Used to show the error message only in the ValidationSummary control) • Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation. • Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the
EnableClientScript	A Boolean value that specifies whether client-side validation is
Enabled	A Boolean value that specifies whether the validation control is
ErrorMessage	The text to display in the ValidationSummary control when validation fails. This text will also be displayed in the validation control if the Text property is not set
IsValid	A Boolean value that indicates whether the control specified by
Text	The message to display when validation fails

The RequiredFieldValidator Control:

The RequiredFieldValidator control ensures that the user has entered the required data in the input control, such as text box control to which it is bound. With this control, the validation fails if the input value does not change from its initial value. By default, the initial value is an empty string ("").The InitialValue property does not set the default value for the input control. It indicates the value that you do **not** want the user to enter in the input control.

Property	Description
InitialValue	Specifies the starting value of the input control. Default value is ""

Output:



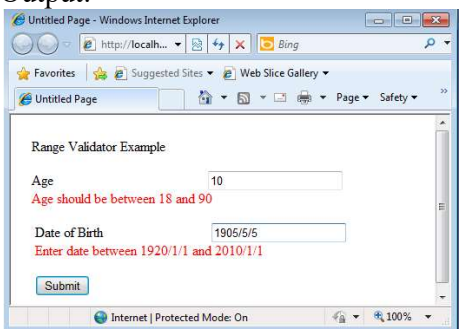
The RangeValidator Control

The RangeValidator Control checks whether or not the value of an input control lies within a specified range of values. This control is used to check that the user enters an input value that falls between two values. It is possible to check ranges within numbers, dates, and characters. The validation will not fail if the input control is empty. Use the RequiredFieldValidator control to make the field required. The following are the properties of RangeValidator control:

- ControlToValidate – Contains the input control to validate
- Minimum Value – Holds the minimum value of the valid range
- Maximum Value – Holds the maximum value of the valid range

Property	Description
MaximumValue	Specifies the maximum value of the input control
MinimumValue	Specifies the minimum value of the input control

Output:



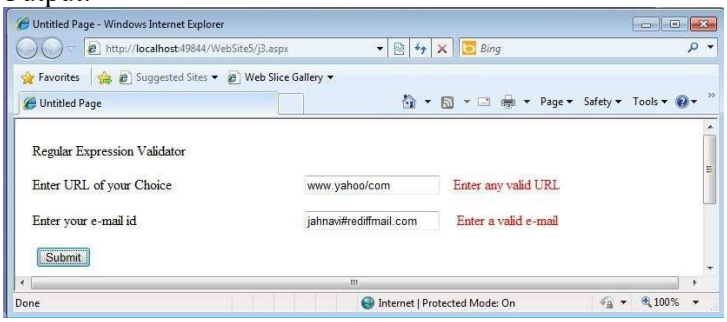
The RegularExpressionValidator Control

The RegularExpressionValidator control is used to ensure that an input value matches a specified pattern. It checks whether the text in the input control matches the format

specified by the regular expressions like email-id, date format. This control exists within the Sysem.Web.UI.WebControls namespace

Property	Description
ValidationExpression	Specifies the expression used to validate input control. The expression validation syntax is different on the client than on the server. JScript is used on the client. On the server, the language you

Output:

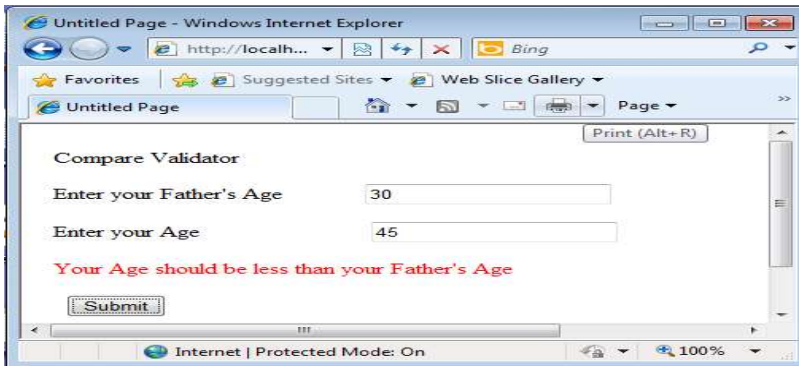


The CompareValidator Control

The CompareValidator control is used to compare the value entered by a user into one input control with the value entered into another or with an already existing value. This control exists within the Sysem.Web.UI.WebControls namespace

Property	Description
ControlToCompare	Obtains or sets the input control to compare with the input control
Operator	Sets the comparison operator like equal, notequal,greaterthan, lessthan
ValueToCompare	Sets a constant value to compare with the value entered by the user in the input control being validated

Output:



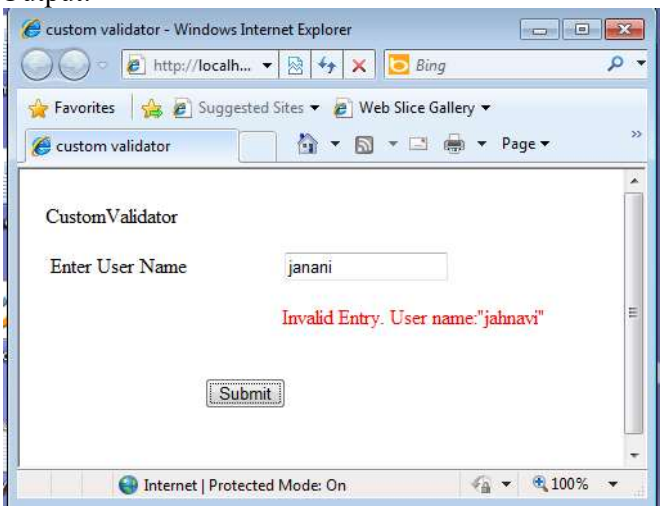
The CustomValidator Control

The CustomValidator control allows you to write a method to handle the validation of the

value entered. It checks the input given matches with a given condition or not. It basically validates the input provided by the user against user-defined validations. This control exists within the Sysem.Web.UI.WebControls namespace

Property	Description
ClientValidationFunction	Specifies the name of the client-side validation script function to be executed.
ValidateEmptyText	Sets a boolen value indicating whether empty text should be validated or not

Output:

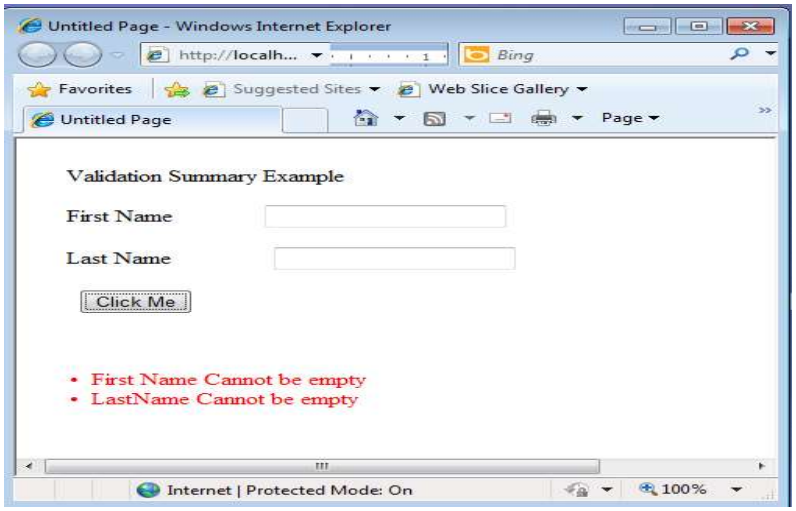


The ValidationSummary Control

The ValidationSummary control is used to display a summary of all validation errors occurred in a Web page at one place. The error message displayed in this control is specified by the ErrorMessage property of each validation control. If the ErrorMessage property of the validation control is not set, no error message is displayed for that validation control.This control exists within the Sysem.Web.UI.WebControls namespace.

Property	Description
DisplayMode	How to display the summary. Legal values are: • BulletList • List • SingleParagraph
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not
ForeColor	The fore color of the control
HeaderText	A header in the ValidationSummary control
ShowMessageBox	A Boolean value that specifies whether the summary should be displayed in a message box or not
ShowSummary	A Boolean value that specifies whether the ValidationSummary

Output:


Untitled Page - Windows Internet Explorer
http://localh...
Bing
Favorites
Suggested Sites
Web Slice Gallery
Untitled Page
Page
Validation Summary Example
First Name
Last Name
Click Me
First Name Cannot be empty
LastName Cannot be empty
Internet | Protected Mode: On
100%