Darshan
UNIVERSITY
योग: कर्मसु कौशलम्

ASP.NET

(2302CS521)

Subject Material (English)
Diploma 5th Semester
(Computer)

Prof. Akash N Siddhpura
Department of Computer Engineering

Darshan UNIVERSITY
योगः कर्मसु कौशलम्

TABLE OF CONTENT

Darshan UNIVERSITY
योगः कर्मसु कौशलम्

LIST OF FIGURES

Unit-2 ASP.NET Introduction and Controls

2.1 Features of ASP.NET

- **Simplified Programming Model** - ASP.Net is a technology which can be implemented using any .Net language such as C#, VB, F# etc.
- **Side by Side Execution** - Side by side execution is the ability to run multiple versions of an application or component on the same computer.
- **Server-Side Controls** - ASP.Net technology has introduced rich powerful web server controls which are capable to identify the clients' browser and can render the HTML tags accordingly.
- **Performance** - Architecture of the ASP.Net is designed in order to increase the performance of the application.
- **Powerful Database Support** - ASP.Net supports very large amount of database class and it also provide the easy deployment of the connectivity with the application.
- **Web Services** - With the help of ASP.Net we can also develop the web services, which are useful for the communication of protocols, it can be also used by other application, it is an application component.
- **Simplified Form Validation** - In other language we have to write so may line of codes to validate a control, while in ASP.NET it becomes Easy and Simple to deploy validation.
- **Rich Text box Support** - ASP.Net supports a very high range of tools which can be easily implemented
- **Simplified Deployment** - With the help of the CLR, we can develop our applications in any language (which are accepted by the .Net Framework), so developer don't have to worry about programming language. Visual Studio provides the functionality of drag of controls, so because of that deployment becomes fast and easy. Developer can easily configure IIS (Internet Information Service) also

2.2 Types of files in ASP.NET

Type of File	Description
.aspx	An ASP.Net Web Forms page that can contain Web controls and presentation and business logic.
.sln	A solution file for a visual studio project.
.master	A master page that defines the layout for other Web pages in the application.
.mdb, .ldb	An Access database file.
.mdf	A SQL Server Express database file.
.dll	A compiled class library file (assembly). In a Web site project, instead of placing compiled assemblies in the bin subdirectory, you can put source code for classes in the App_Code subdirectory.

.asax	The Global.asax file, also known as the ASP.Net application file, is an optional file that contains code for responding to application-level events raised by ASP.Net
.cs, .vb	Source code files that define code that can be shared between pages, such as code for custom classes, business logic, HTTP modules and HTTP handlers.
.config	A configuration file contains XML elements that represent settings for ASP.Net features.
.skin	A skin file that contains property settings to apply to Web controls for consistent formatting.

2.3 Web form round trip

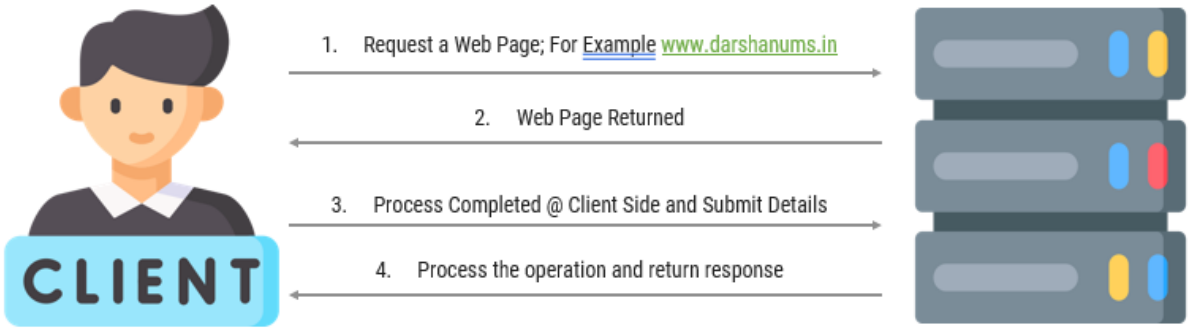

Fig.1. Web form round trip

- The browser presents the user with a form, and the user interacts with the form, causing the form to post back to the server.
- However, since all processing that interacts with server components must occur on the server.
- Means for each action that requires processing, the form must be posted to server, processed, and returned to the browser. This sequence of events is known as round trip.
- **What is IsPostBack**
    - Postback is actually sending all the information from client to web server, then web server process all those contents and returns back to the client.
    - Most of the time ASP control will cause a post back (e. g. buttonclick).
    - but some don't unless you tell them to do In certain events (Listbox Index Changed, RadioButton Checked etc..) in an ASP.NET page upon which a PostBack might be needed.
    - IsPostBack is a property of the page that tells whether or not the page is on its initial load or if a user has performed a button click on your web page that has caused the page to post back to itself.
    - The value of the Page.IsPostBack property will be set to true when the page is executing after a postback, and otherwise it will set to false.
    - Is Postback is normally used on page _load event to detect that;

- if web page is getting generated due to postback requested by a control on the page
- if wep page is getting loaded for the first time

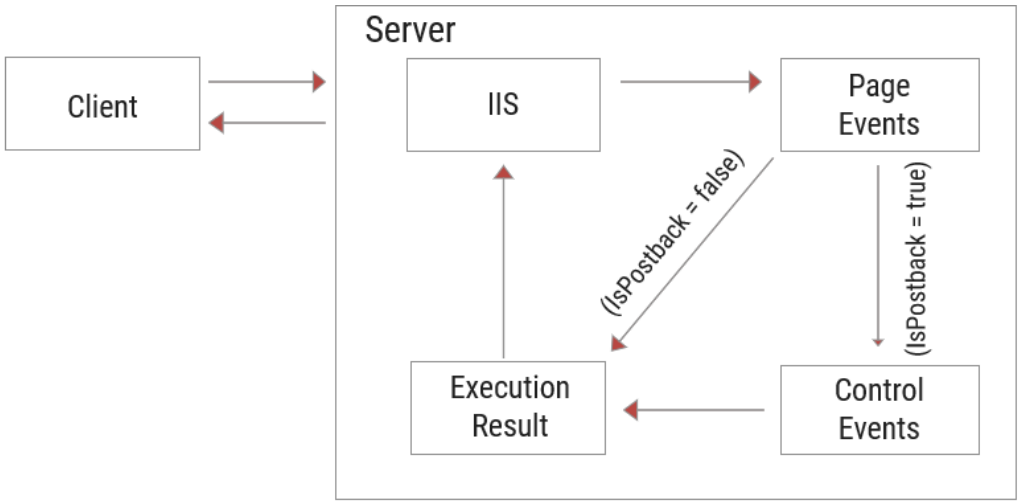

Fig. 2.web page cycle to understand Postback

2.4 Stages in web form

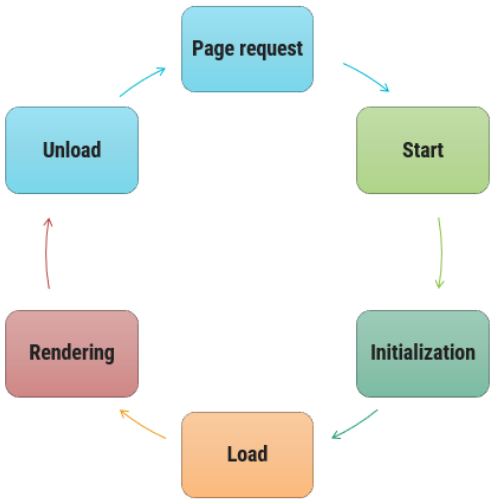

Fig. 3. Stages in web form

- When a user request for ASP.Net page, the page gone through many stages that is known as Life Cycle.
- **Page Request**
  - It checked that requested page's request is NEW or OLD.
  - If NEW then it compiles and executes that page.
  - Else It's OLD, cached copy will be returned
- **Start**
  - Request and Response properties are set and using IsPostBack property new or old request can be identified.
- **Initialization**

- In this Phase, controls are available on the page. Every control has Unique ID property.

- **Load**
  - During the load phase, the request is post back and control properties are loaded with information.
- **Rendering**
  - The view state of the controls on the page is saved before rendering on the server. The page calls the Render method for every control.
- **Unload**
  - Request and response properties of the page are unloaded and cleaning performed if required.

2.5 ASP.NET Objects

- Request Object
  - The request object retrieves the values that the client browser passed to the server during an HTTP request.
  - Properties

Property Name	Description
Browser	Gets or sets information about the requesting client's browser capabilities.
QueryString	Gets the collection of HTTP query string variables.
Url	Gets information about the URL of the current request.

  - Methods

Method Name	Description
SaveAs	Saves an HTTP request to disk.
ToString	Returns a String that represents the current object.

- Response Object
  - The response object is going to use for sending output to the client.
  - Properies

Property Name	Description
Expires	Gets or sets the number of minutes before a page cached on a browser expires.
Status	Sets the status line that is returned to the client.

Cookies	Specifies cookie values. Using this collection, you can set cookie values.

- Methods

Method Name	Description
Write	Writes a variable or text to the current HTTP output as a string.
Redirect(String)	Redirects a request to a new URL and specifies the new URL.

- Server Object
  - It Describes the methods and properties of the object that provides methods for various server tasks.
  - Properties

Property Name	Description
MachineName	Returns Name of the server computer.
ScriptTimeOut	Gets or Sets the request time-out value in seconds.

  - Methods

Method Name	Description
Transfer	Transfers execution to another web page in the current application.
Equals (Object)	Checks whether the specified Object is equal to the current Object or not.

- Application Object
  - It Describes the methods, properties, and collections of the object that stores information related to the entire Web application, including variables and objects that exist for the lifetime of the application.
  - Properties

Property Name	Description
Contents.Remove	Deletes an item from the collection.
Contents.RemoveAll	Deletes all items from the collection.

  - Methods

Method Name	Description
Lock	Prevents other clients from modifying Application object properties.
UnLock	Allows other clients to modify Application object properties.

- Session Object

- It Describes the methods, properties, and collections of the object that stores information related to the user's session, including variables and objects that exist for the lifetime of the session.
- Properties

Property Name	Description
Timeout	The time-out period for the session state for this application, in minutes.
CodePage	The code page specifies to the server how to encode characters for different languages.
SessionID	Returns the session identification for this user.

- Methods

Method Name	Description
Abandon	Destroys a Session object and releases its resources.

2.6 Standard Controls

- **Label Control**
  - The Label Web server control provides a way to display text programmatically control in an ASP.NET Web page.
  - This Control is used to display information on web page.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely label control in a web page.
BackColor	Gets or Sets the background colour of label control.
Font	Gets or Sets the appearance of the label control.
ForeColor	Gets or Sets the font color of the label control.
Text	Gets or Sets the text of the label control.
ToolTip	This property helps to identifies uniquely label control in a web page, it displays the purpose of the label control when user keeps mouse on the control.
Width	Gets or Sets the width of the label control.
BorderStyle	Gets or Sets the style of border around the label control.
BorderColor	Gets or Sets the border colour of the border around the label control.

  - Example

```
<asp:Label ID="lblName" runat="server" Font-Bold="True" Text="Name:" BackColor="White" ForeColor="Green" Font-Size="Large" ToolTip="Enter your Name:" BorderColor="Red" BorderStyle="Dashed"></asp:Label>
```

- **Textbox control**
  - The Textbox control is used to create a text box where the user can input text.
  - The Textbox Web server control provides a way for users to type information into an ASP.Net Web page, including text, numbers, and dates.
  - This control is used to take input from user into string format by default.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely textbox control in a web page.
BackColor	Gets or Sets the background color of textbox control.
Font	Gets or Sets the appearance of the textbox control.
ForeColor	Gets or Sets the font color of the textbox control.
Text	Gets or Sets the text of the textbox control.
ToolTip	This property helps to identifies uniquely textbox control in a web page, it displays the purpose of the textbox control when user keeps mouse on the control.
Width	Gets or Sets the width of the textbox control.
BorderStyle	Gets or Sets the style of border around the textbox control.
BorderColor	Gets or Sets the border color of the border around the textbox control.
TabIndex	Gets or Sets the tab order of the textbox control in a web page.
TextMode	Gets or Sets the behavior mode of the textbox control.
Wrap	Gets or Sets the behavior of the textbox control that the text should be wrap or not.

  - Example

```
<asp:TextBox ID="txtName" runat="server" TextMode="SingleLine"
Wrap="true"></asp:TextBox>dasdas
```

- **Button Control**
  - Displays a push button.
  - Buttons in an ASP.Net Web page allow users to send a command.
  - Buttons submit the page to the server and cause it to be processed along with any pending events.
  - ASP.Net provides three types of button control
    - **Button**: It displays text within a rectangular area
    - **Link Button**: It displays text that looks like a hyperlink
    - **Image Button**: It displays an image

- **Button Properties**

Property Name	Description
ID	This property helps to identifies uniquely button control in a web page.
BackColor	Gets or Sets the background color of button control.
Font	Gets or Sets the appearance of the button control.
ForeColor	Gets or Sets the font color of the button control.
Text	Gets or Sets the text of the button control.
ToolTip	This property helps to identifies uniquely button control in a web page, it displays the purpose of the button control when user keeps mouse on the control.
Width	Gets or Sets the width of the button control.
BorderStyle	Gets or Sets the style of border around the button control.
BorderColor	Gets or Sets the border color of the border around the button control.
TabIndex	Gets or Sets the tab order of the button control in a web page.
PostBackUrl	Gets or Sets the value of the URL to post to when the button is clicked.
ValidationGroup	Gets or Sets the name of the group for button control, that it has to verify the group of controls causes a postback.

- Example

```
<asp:Button ID="btnClick" runat="server" Text="Submit" />
```

- **Link Button**
  - The LinkButton control is used to create a hyperlink-style button on the Web page.
  - This control looks like a Hyperlink control but almost has the functionality of the Button control.
  - Properties

Property Name	Description
CausesValidation	Gets or Sets the value of validation when this link button will be clicked.
ValidationGroup	Gets of Sets the value of group of controls for which the LinkButton control causes validation when it posts back to the server.
PostBackUrl	Gets or Sets the value of the URL to post to when the link button is clicked.
OnClientClick	Gets or Sets the value for attaching a client side (JavaScript) method that will fire when this link button will be clicked.

- Example

  <asp:LinkButton ID="lnk" runat="server" Text="Click me" PostBackUrl="AboutUs.aspx"></asp:LinkButton>

- **Image Button**
  - Its like an ASP Button control, the only difference is, you have the ability to place your own image as a button.
  - You use an image Button when you want your button to look different than the plain rectangular button.
  - Properties

Property Name	Description
ImageUrl	Gets or Sets the url for the control.
ImageAlign	Gets or Sets the alignment of the image for the control.
AlternateText	Gets or Sets the alternate text displayed when the image cannot be shown.

  - Example

    <asp:ImageButton ID="img" runat="server" AlternateText="image" ImageUrl="~/Image/smiley.jpg" PostBackUrl="AboutUs.aspx" Height="56px" Width="75px" />

- **Hyperlink control**
  - The Hyperlink control is used to create a hyperlink.
  - This control provides easy navigation between various Pages.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely hyperlink control in a web page.
BackColor	Gets or Sets the background color of hyperlink control.
Font	Gets or Sets the appearance of the hyperlink control.
ForeColor	Gets or Sets the font color of the hyperlink control.
Text	Gets or Sets the text of the hyperlink control.
ToolTip	This property helps to identifies uniquely hyperlink control in a web page, it displays the purpose of the hyperlink control when user keeps mouse on the control.
Width	Gets or Sets the width of the hyperlink control.
BorderStyle	Gets or Sets the style of border around the hyperlink control.
BorderColor	Gets or Sets the border color of the border around the hyperlink control.

TabIndex	Gets or Sets the tab order of the hyperlink control in a web page.
NavigateUrl	It navigate to webpage when the user clicks on the hyperlink.

- Example

```
<asp:HyperLink ID="hldu" runat="server" Text="Darshan University"
ToolTip="Darshan University"
NavigateUrl="https://www.darshan.ac.in"></asp:HyperLink>
```

- **Image Control**
  - The Image control is used to display an image on the web page.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely image control in a web page.
ToolTip	Displays the purpose of the image control when user keeps mouse on the control.
Width	Gets or Sets the width of the image control.
Height	Gets or Sets the height of the image control
BorderStyle	Gets or Sets the style of border around the image control.
BorderColor	Gets or Sets the border color of the border around the image control.
TabIndex	Gets or Sets the tab order of the image control in a web page.
AlternateText	Gets or Sets the alternate text displayed when the image cannot be shown.
ImageUrl	Gets or Sets the url of the image which we have to shown in to the image control.
ImageAlign	Gets or Sets the alignment of the image in the image control.

  - Example

```
<asp:Image ID="img" runat="server" AlternateText="Darshan University" Height="70"
Width="70" ImageUrl="~/Images/Darshan-University-Icon.png" BorderColor="Green"
BorderStyle="Solid" ImageAlign="Middle" />
```

- **Checkbox Control**
  - This control is used to select multiple values from the list.
  - Display a checkbox that allows the user to select value, if checkbox is selected then its value will become TRUE otherwise its value will become FALSE.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely checkbox control in a web page.
BackColor	Gets or Sets the background color of checkbox control.

Font	Gets or Sets the appearance of the checkbox control.
ForeColor	Gets or Sets the font color of the checkbox control.
Text	Gets or Sets the text of the checkbox control.
ToolTip	This property helps to identifies uniquely checkbox control in a web page, it displays the purpose of the checkbox control when user keeps mouse on the control.
Width	Gets or Sets the width of the checkbox control.
TabIndex	Gets or Sets the tab order of the checkbox control in a web page.
Checked	Gets or Sets the checked state of the checkbox control.
Visible	Gets or Sets the visibility of the checkbox control on the web page
AutoPostBack	Sets the checkbox that, it automatically posts back to server when it is checked or unchecked.

- Example

```
<asp:CheckBox ID="chkCricket" runat="server" Text="Cricket" Checked="true" />
<asp:CheckBox ID="chkFootBall" runat="server" Text="Football" Checked="false" />
<asp:CheckBox ID="chkBasketBall" runat="server" Text="Basketball" Checked="false" />
```

- **Radiobutton control**
  - The RadioButton control is used to display a radio button.
  - RadioButton present a group of options from which the user can select just one option.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely radiobutton control in a web page.
BackColor	Gets or Sets the background color of radiobutton control.
Font	Gets or Sets the appearance of the radiobutton control.
ForeColor	Gets or Sets the font color of the radiobutton control.
Text	Gets or Sets the text of the radiobutton control.
ToolTip	This property helps to identifies uniquely radiobutton control in a web page, it displays the purpose of the radiobutton control when user keeps mouse on the control.
Width	Gets or Sets the width of the radiobutton control.
TabIndex	Gets or Sets the tab order of the radiobutton control in a web page.
Checked	Gets or Sets the checked state of the radiobutton control.
Visible	Gets or Sets the visibility of the radiobutton control on the web page

AutoPostBack	Sets the radiobutton that, it automatically posts back to server when it is checked or unchecked.
GroupName	Gets or Sets the name of the group to which radiobutton control belongs

- Example

```
<asp:RadioButton ID="rbMale" runat="server" Text="Male" GroupName="Gender" Checked="true" />
<asp:RadioButton ID="rbFemale" runat="server" Text="Female" GroupName="Gender" />
```

- **Hidden field control**
  - The Hidden Field control is used to store a value that needs to be persisted across posts to the server.
  - It is rendered as an <input type= "hidden"/> element.
  - Normally view state, session state, and cookies are used to maintain the state of a Web Forms page.
  - However, if these methods are disabled or are not available, then you can use the Hidden Field control to store state values.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely hidden field control in a web page.
Visible	Gets or Sets the visibility of the hidden field control.
Value	Gets or Sets the value of the hidden field control.

  - Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblName" runat="server" Text="Name”></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtName" runat="server" ></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>
                    <asp:Button ID="btnClick" runat="server" Text="Submit" OnClick="btnClick_Click" />
    </td>
```

```
        <td>
          <asp:Label ID="lblInfo" runat="server"> </asp:Label>
          <asp:HiddenField ID="hdn" runat="server"/>
        </td>
      </tr>
    </table>
```

Home.aspx.cs

```
protected void btnClick_Click(object sender, EventArgs e)
{
   hdn.Value = txtName.Text;
   lblInfo.Text = hdn.Value;
}
```

- **Literal Control**
  - The Literal Control is similar to the Label Control as they both are used to display static text on a web page.
  - Programmers cannot apply a style to a Literal Control as well as there is no properties like BackColor, ForeColor, BorderColor, BorderStyle as Label control.
  - But Literal Control is light weight Control compare to Label Control.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely literal control in a web page.
Visible	Gets or Sets the visibility of the literal control.
Text	Gets or Sets the text of the literal control.

  - Example

```
<asp:Literal ID="ltrlinfo" runat="server" ></asp:Literal>
```

- **File Upload Control**
  - It is a controller which is used to upload file
  - It creates a browse button on the form that pop up a window to select the file from the local machine.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely file upload control in a web page.
BackColor	Gets or Sets the background color of file upload control.
Font	Gets or Sets the appearance of the file upload control
ForeColor	Gets or Sets the font color of the file upload control.

ToolTip	Displays the purpose of the file upload control when user keeps mouse on the control.
Width	Gets or Sets the width of the file upload control.
BorderStyle	Gets or Sets the style of border around the file upload control.
BorderColor	Gets or Sets the border color of the border around the file upload control.
TabIndex	Gets or Sets the tab order of the file upload control in a web page.
AllowMultiple	Gets or Sets the value for whether to enable multiple files to upload or not.

- Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblFile" runat="server" Font-Bold="True" Text="Select a File
Here:"></asp:Label>
    </td>
    <td>
      <asp:FileUpload ID="fuFile" runat="server" AllowMultiple="True" />
    </td>
  </tr>
  <tr>
    <td>
      <asp:Button ID="btnClick" runat="server" Text="Submit"
OnClick="btnClick_Click1" />
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="lblFileDetails" runat="server"></asp:Label>
    </td>
  </tr>
</table>
```

Home.aspx.cs

```
protected void btnClick_Click1(object sender, EventArgs e)
{
  StringBuilder sb = new StringBuilder();
  if(fuFile.HasFile)
  {
    try
    {
      fuFile.SaveAs("D:\\uploaded\\" + fuFile.FileName);
```

```
            sb.AppendFormat("<br/> Save As: ", +fuFile.PostedFile.FileName);
            sb.AppendFormat("<br/> File type: " +fuFile.PostedFile.ContentType);
            sb.AppendFormat("<br/> File length: " +fuFile.PostedFile.ContentLength);
            lblFileDetails.Text = sb.ToString();
          }
          catch(Exception ex)
          {
            sb.Append("<br/> Error <br/>");
            sb.AppendFormat("Unable to save file <br/> {0}", ex.Message);
            lblFileDetails.Text = sb.ToString();
          }
        }
        else
        {
          lblFileDetails.Text = sb.ToString();
        }
      }
```

- **Calender Control**
  - Calendar control provides functionalities like display one month at a time, select a day, a week or a month, select a range of days etc.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely calendar control in a web page.
BackColor	Gets or Sets the background color of calendar control.
Font	Gets or Sets the appearance of the calendar control
ForeColor	Gets or Sets the font color of the calendar control.
ToolTip	Displays the purpose of the calendar control when user keeps mouse on the control.
BorderStyle	Gets or Sets the style of border around the calendar control.
BorderColor	Gets or Sets the border color of the border around the calendar control.
Caption	Gets or sets the caption for the calendar control
DayHeaderStyle	Gets the style properties for the section that displays the day of the week.
FirstDayOfWeek	Gets or sets the day of week to display in the first column.
SelectedDate	Gets or sets the selected date.

  - Example

```
<asp:Calendar ID="calender" runat="server" FirstDayOfWeek="Monday" OnSelectionChanged="calender_SelectionChanged" Caption="Select Date" SelectedDate="01/01/2011 20:00:00" > </asp:Calendar>
```

2.7 ListBox Controls

- **Dropdown List**
  - Dropdown list provides a list of items with only one item visible at a time and user can select single item at a time.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely dropdown list control in a web page.
BackColor	Gets or Sets the background color of dropdown list control.
Font	Gets or Sets the appearance of the dropdown list control
ForeColor	Gets or Sets the font color of the dropdown list control.
ToolTip	Displays the purpose of the dropdown list control when user keeps mouse on the control.
Width	Gets or Sets the width of the dropdown list control.
TabIndex	Gets or Sets the tab order of the dropdown list control in a web page.
Items	It displays list of items in the dropdown list.
DataTextField	Gets or Sets field in the data source which provides the item text.
DataValueField	Gets or Sets field in the data source which provides the item value.

  - Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:DropDownList ID="ddlstSports" runat="server" AutoPostBack="True“
OnSelectedIndexChanged="ddlstSports_SelectedIndexChanged">

        <asp:ListItem Text="Cricket" Value="CRIC"></asp:ListItem>
        <asp:ListItem Text="Football" Value="FB"></asp:ListItem>
        <asp:ListItem Text="Basketball" Value="BB"></asp:ListItem>
        <asp:ListItem Text="Kabbadi" Value="KABBADI"></asp:ListItem>

      </asp:DropDownList>
    </td>
    <td> <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" /> </td>
  </tr>
  <tr> <td> <asp:Label ID="lblinfo" runat="server"></asp:Label> </td> </tr>
</table>
```

Home.aspx.cs

```
protected void ddlstSports_SelectedIndexChanged(object sender, EventArgs e)
{
   foreach (ListItem list in ddlstSports.Items)
   {
      if (list.Selected)
      {
         lblinfo.Text = list.Value;
      }
   }
}
```

- **List Box**
  - The ListBox Control provides us a user interface that will display the List of the items.
  - From there, the users can select one or more items from the List.
  - These lists can be loaded either by code or by the ListItem objects.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely listbox control in a web page.
BackColor	Gets or Sets the background color of listbox control.
Font	Gets or Sets the appearance of the listbox control
ForeColor	Gets or Sets the font color of the listbox control.
ToolTip	Displays the purpose of the listbox control when user keeps mouse on the control.
Width	Gets or Sets the width of the listbox control.
TabIndex	Gets or Sets the tab order of the listbox control in a web page.
Items	It displays list of items in the listbox.
Rows	Gets or Sets the number of rows in listbox control.
DataTextField	Gets or Sets field in the data source which provides the item text.
DataValueField	Gets or Sets field in the data source which provides the item value.
SelectionMode	Gets or Sets selection mode for the listbox control.

  - Example

Home.aspx

```
<table>
   <tr>
      <td>
         <asp:ListBox ID="lstSports" runat="server" SelectionMode="Multiple" >
```

```
                <asp:ListItem Text="Cricket" Value="CRIC"></asp:ListItem>
                <asp:ListItem Text="Football" Value="FB"></asp:ListItem>
                <asp:ListItem Text="Basketball" Value="BB"></asp:ListItem>
                <asp:ListItem Text="Kabbadi" Value="KABBADI"></asp:ListItem>

            </asp:ListBox>
        </td>
        <td>  <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />  </td>
    </tr>
    <tr>  <td>  <asp:Label ID="lblinfo" runat="server"></asp:Label> </td> </tr>
</table>
```

Home.aspx.cs

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    foreach (ListItem list in lstSports.Items)
    {
        if (list.Selected)
        {
            lblinfo.Text += list.Value + "<br>";
        }
    }
}
```

- **Checkbox List**
  - Provides a checkbox with each data and user can select more than one item.
  - A check box list presents a list of independent options.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely checkbox list control in a web page.
BackColor	Gets or Sets the background color of checkbox list control.
Font	Gets or Sets the appearance of the checkbox list control
ForeColor	Gets or Sets the font color of the checkbox list control.
ToolTip	Displays the purpose of the checkbox list control when user keeps mouse on the control.
Width	Gets or Sets the width of the checkbox list control.
TabIndex	Gets or Sets the tab order of the checkbox list control in a web page.
Items	It displays list of items in the checkbox list.

RepeatDirection	It specifies the direction in which the controls to be repeated. The values available are Horizontal and Vertical.

- Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:CheckBoxList ID="chklstSports" runat="server"
SelectionMode="Multiple" >

            <asp:ListItem Text="Cricket" Value="CRIC"></asp:ListItem>
            <asp:ListItem Text="Football" Value="FB"></asp:ListItem>
            <asp:ListItem Text="Basketball" Value="BB"></asp:ListItem>
            <asp:ListItem Text="Kabbadi" Value="KABBADI"></asp:ListItem>

      </asp:CheckBoxList>      </td>
    <td>  <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />  </td>
  </tr>
  <tr>  <td>  <asp:Label ID="lblinfo" runat="server"></asp:Label> </td> </tr>
</table>
```

Home.aspx.cs

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
  foreach (ListItem list in chklstSports.Items)
  {
    if (list.Selected)
    {
      lblinfo.Text += list.Value + "<br>";
    }
  }
}
```

- **Radiobutton List**
  - The RadioButton List control works like the Checkbox List control but it enables a user to select only one list item at a time.
  - The RadioButton List control displays a list of radio buttons that can be arranged either horizontally or vertically.
  - Properties

Property Name	Description
ID	This property helps to identifies uniquely radiobutton list control in a web page.

BackColor	Gets or Sets the background color of radiobutton list control.
Font	Gets or Sets the appearance of the radiobutton list control
ForeColor	Gets or Sets the font color of the radiobutton list control.
ToolTip	Displays the purpose of the radiobutton list control when user keeps mouse on the control.
Width	Gets or Sets the width of the radiobutton list control.
TabIndex	Gets or Sets the tab order of the radiobutton list control in a web page.
Items	It displays list of items in the radiobutton list.
RepeatDirection	It specifies the direction in which the controls to be repeated. The values available are Horizontal and Vertical

- Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:RadioButtonList ID="rblstgender" runat="server">

        <asp:ListItem Text="Male" Value="M"></asp:ListItem>
        <asp:ListItem Text="Female" Value="F"></asp:ListItem>
        <asp:ListItem Text="Others" Value="O"></asp:ListItem>

      </asp:RadioButtonList>
    </td>
    <td>
      <asp:RadioButtonList ID="rblstbranch"
runat="server“OnSelectedIndexChanged="rblstbranch_SelectedIndexChanged">

        <asp:ListItem Text="Computer" Value="CE"></asp:ListItem>
        <asp:ListItem Text="Information & Technology"
Value="IT"></asp:ListItem>
        <asp:ListItem Text="CI/ME/EE" Value="Core Branch"></asp:ListItem>

      </asp:RadioButtonList>
    </td>
  </tr>
  <tr> <td> <asp:Button ID="btnSubmit" runat="server" Text="Submit" /></td> </tr>
  <tr> <td> <asp:Label ID="lblinfo" runat="server"></asp:Label> </td> </tr>
</table>
```

Home.aspx.cs

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
```

```
        lblinfo.Text = "";
        foreach (ListItem list in rblstgender.Items)
        {
            if (list.Selected)
            {
                lblinfo.Text += list.Value + "<br>";
            }
        }
        foreach (ListItem list in rblstbranch.Items)
        {
            if (list.Selected)
            {
                lblinfo.Text += list.Value + "<br>";
            }
        }
    }
```

2.8 Validation Controls

- Validation controls are used to validate the values that are entered into controls of a web page.
- Validation controls perform client-side validation, server-side validation, or both.
- Validation controls offer the following advantages;
    - You can associate one or more validation controls with each control that you want to validate.
    - You can specify programmatically whether validation should occur, which is useful if you want to provide a cancel button so that the user can exit without having to fill valid data in all of the fields.
- The validation controls automatically detect whether validation should be performed on the client side or the server side.
- Common Properties of Validation Controls

Property Name	Description
ID	It identifier for the validation control on the web page.
Runat	With the help of it, Validation controls will execute on Server.
ControlToValidate	It indicates, that validation control has to be attached with which server-side control or to which control validation control has to validate.
Enabled	It enables or disables the validator control.
ErrorMessage	It indicates error string.
Display	The display behavior for the validation control. 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 input passes validation.

- **Required Filed Validator**
  - It provides validation for emptiness of any control.
  - This Validator is applied to such controls where input is necessary or mandatory.
  - Example

    Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblName" runat="server" Text="Enter Name:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
      <asp:RequiredFieldValidator ID="requiredvatxtname" runat="server"
ControlToValidate="txtName" ErrorMessage="Please Enter Name"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Button ID="btnClick" runat="server" Text="Submit" />
    </td>
  </tr>
</table>
```

- **Range Validator**
  - The Range Validator control verifies that the input value falls within a pre-determined range or not.
  - It provides validation for range of an input by setting minimum value and maximum value.
  - This Validator is used when input is taken between certain type of values

- Properties

Property Name	Description
Type	It defines the type of the data. The values are: Currency, Date, Double, Integer, and String
MinimumValue	It specifies the minimum value of the range.
MaximumValue	It specifies the maximum value of the range.

- Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblAge" runat="server" Text="Enter Age:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
       <asp:RangeValidator ID="rangeValidatorAge" Display="Dynamic"
runat="server" ControlToValidate="txtAge" ErrorMessage="Please Enter Age Between
18 to 40" ForeColor="Red" MaximumValue="40"
MinimumValue="18"></asp:RangeValidator>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Button ID="btnClick" runat="server" Text="Submit"
OnClick="btnClick_Click" />
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="lblInfo" runat="server" Font-Bold="true"></asp:Label>
    </td>
  </tr>
</table>
```

Home.aspx.cs

```
protected void btnClick_Click(object sender, EventArgs e)
{
   lblInfo.Text = "Entered Age is: " + txtAge.Text;
}
```

- **Compare Validator**
  - The Compare Validator control compares a value of one control with a fixed value or a value in another control.

- Properties

Property Name	Description
Type	It specifies the data type.
ControlToCompare	It specifies the value of the input control to compare with.
ValueToCompare	It specifies the constant value to compare with.
Operator	It specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, and DataTypeCheck.

- Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblEmail" runat="server" Text="Enter Email ID:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>
    <asp:Label ID="lblEmail2" runat="server" Text="Confirm your Email ID:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtEmail2" runat="server"></asp:TextBox>
  <asp:CompareValidator ID="cmprValidatorEmail" Display="Dynamic" runat="server" ControlToValidate="txtEmail2" ControlToCompare="txtEmail" ErrorMessage="Please Enter Same Email ID" ForeColor="Red"></asp:CompareValidator>
    </td>
  </tr>
  <tr>
    <td>
   <asp:Button ID="btnClick" runat="server" Text="Submit" OnClick="btnClick_Click" />
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="lblInfo" runat="server" Font-Bold="true"></asp:Label>
    </td>
  </tr>
</table>
```

Home.aspx.cs

```
protected void btnClick_Click(object sender, EventArgs e)
{
   lblInfo.Text = "Entered Email id: " + txtEmail.Text;
}
```

- **Regular Expression Validator**
  - The Regular Expression Validator allows validating the input text by matching against a pattern of a regular expression.
  - The regular expression is set in the ValidationExpression property.
  - Property

Property Name	Description
ValidationExpression	Specifies the expression used to validate input control.

  - Example

Home.aspx

```
<table>
   <tr>
      <td>
         <asp:Label ID="lblMobile" runat="server" Text="Enter Mobile
No.:"></asp:Label>
      </td>
      <td>
         <asp:TextBox ID="txtMobile" runat="server"></asp:TextBox>
         <asp:RegularExpressionValidator ID="regularValidatorMobile"
Display="Dynamic" runat="server" ControlToValidate="txtMobile"
ErrorMessage="Please Enter Proper Mobile No." ForeColor="Red"
ValidationExpression="[0-9]{10}"> </asp:RegularExpressionValidator>
      </td> </tr>
    <tr>  <td>
         <asp:Button ID="btnClick" runat="server" Text="Submit"
OnClick="btnClick_Click" />
      </td>  </tr>
   <tr>
      <td>
         <asp:Label ID="lblInfo" runat="server" Font-Bold="true"></asp:Label>
      </td>  </tr> </table>
```

Home.aspx.cs

```
protected void btnClick_Click(object sender, EventArgs e)
{
   lblInfo.Text = "Entered Mobile No.: " + txtMobile.Text;
}
```

- **Custom Validator**
  - It doesn't come with a predefined way of working; programmer write the code for validating data.
  - The control allows you to validate both client side and server side, where the server-side approach is probably the most powerful.
  - In this example, we will simply check the length of the string in the Textbox.
  - Example

Home.aspx

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblPinCode" runat="server" Text="Enter Pincode:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtpincode" runat="server"></asp:TextBox>
      <asp:CustomValidator ID="cstPincodeValidator" Display="Dynamic"
runat="server" ControlToValidate="txtpincode" ErrorMessage="Please Enter Valid
Pincode" ForeColor="Red" OnServerValidate="cstPincodeValidator_ServerValidate" >
</asp:CustomValidator>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Button ID="btnClick" runat="server" Text="Submit"
OnClick="btnClick_Click" />
    </td> </tr>
  <tr>  <td>
      <asp:Label ID="lblInfo" runat="server" Font-Bold="true"></asp:Label>
    </td>  </tr> </table>
```

Home.aspx.cs

```
protected void cstPincodeValidator_ServerValidate(object source,
ServerValidateEventArgs args)
{
  if (args.Value.Length == 6)
  {
    lblInfo.Text = "Entered Pincode: " + txtpincode.Text;
    args.IsValid = true;
  }
  else
  {
    args.IsValid = false
  }
}
```

- **Validation Summary**
  - The Validation Summary control is used to display a summary of all validation errors occurred in a Web page.
  - 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.
  - Properties

Property Name	Description
DisplayMode	How to display the summary. Legal values are: BulletList, List, SingleParagraph.
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 control should be displayed or hidden.

  - Example

```
<table>
  <tr>
    <td>
      <asp:Label ID="lblName" runat="server" Text="Enter Name:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
      <asp:RequiredFieldValidator ID="requiredvatxtname" runat="server" ControlToValidate="txtName" ErrorMessage="Please Enter Name" ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="lblAge" runat="server" Text="Enter Age:"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
      <asp:RequiredFieldValidator ID="requiredValidatorAge" Display="Dynamic" runat="server" ControlToValidate="txtAge" ErrorMessage="Please Enter Age" ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
  </tr>
```

```
    <tr>
      <td>
                    <asp:Button ID="btnClick" runat="server" Text="Submit"
OnClick="btnClick_Click" />
      </td>
    </tr>
    <tr>
      <td>
        <asp:Label ID="lblInfo" runat="server" Font-Bold="true"></asp:Label>
    <asp:ValidationSummary ID="vsummary" runat="server" DisplayMode="BulletList"
ShowMessageBox="false" ShowSummary="true" />
      </td>
    </tr>
</table>
```

2.9 Data bound Controls

- To display data in various forms and to perform different different database activities such as add, edit, update and delete operations on data, need data bound controls.
- The controls makes the data more organized and presents the data in an efficient way for the users.
- **Gridview Control**
  - The GridView control is used to display the values of a data source in a table.
  - Each column represents a field where each row represents a record. It can also display empty data.
  - The GridView control provides many built-in capabilities that allow the user to sort, update, delete, select and page through items in the control.
  - Features of GridView control
    - Item as row
    - Built-in sorting capability
    - Built-in select, edit and delete capabilities
    - Built-in paging capability
    - Built-in row selection capability
    - Control over Alternate item, Header, Footer, Colors, font, borders, and so on
- **Listview**
  - The ListView control resembles the GridView control. The only difference between them is that the ListView control displays data using user-defined templates instead of row fields.
  - Creating your own templates gives you more flexibility in controlling how the data is displayed.

- It enables you to bind to data items that are returned from a data source and display them.
- Features of ListView control
  - Binding to data source controls Customizable appearance through user-defined templates and styles.
  - Built-in sorting and grouping capabilities
  - Built-in insert, edit and delete capabilities
  - Support for paging capabilities using a DataPager control.
  - Built-in item selection capabilities
  - Fast performance as compared to GridView

- **Repeater**
  - Repeater is a basic container control that allows you to create custom lists from any data available to the page.
  - It renders a read-only template; in other words, it supports only the ItemTemplate to define custom binding. The repeater control is a data bind control, also known as container control.
  - It has no built-in layout or styles, so you must explicitly declare all layout, formatting and style tags within the control templates.
  - Features of repeater control
    - List format
    - Item as row
    - Paging, Sorting and Grouping requires custom code writing
    - no built-in selection capabilities
    - no built-in support for edit, insert and delete capabilities
    - no built-in support for paging, sorting and grouping capabilities
    - no built-in layout or styles, need to declare all layout, formatting and style tags explicitly

- **Datalist**
  - DataList is the next step up from a Repeater; DataList allows you to repeat columns horizontally or vertically.
  - The DataList control renders data as a table and enables you to display data records in various layouts, such as ordering them in columns or rows.
  - You can configure the DataList control to enable users to edit or delete a record in the table.
  - The DataList control works like the Repeater control, used to display the data in a repeating structure, such as a table.

- It displays data in a format that you can define using a template and styles.

- **Detailsview**
  - The DetailsView control uses a table-based layout where each field of the data record is displayed as a row in the control.
  - DetailsView control displays one row from a data source at a time by rendering an HTML table.
  - It shows the details for the row in a separate space. We can customize the appearance of the DetailsView control using its style properties.
  - Alternatively, we can also use CSS to provide styles to control.
  - Features of Detailsview control
    - Tabular rendering
    - Supports column layout, by default two columns at a time
    - Optional support for paging and navigation.
    - Built-in support for edit, insert and delete capabilities

- **Formview**
  - The FormView control renders a single data item at a time from a data source, even if its data source exposes a multiple records data item from a data source.
  - A FormView is a data bound control used to insert, display, edit, update and delete data in ASP.NET that renders a single record at a time.
  - FormView control does not specify a pre-defined layout for displaying a record. It allows for a more flexible layout when displaying a single record.
  - The template contains the formatting, controls and binding expressions used to layout the form.
  - When using templates, we can place any control such as a dropdown list, checkbox and we can even place tables and rich controls like a GridView and so on.
  - Features of Formview control
    - Built-in support for paging and grouping
    - Built-in support for insert, edit and delete capabilities