...t a Web site project, if you make changes to the page ...at would affect the generated class whether by adding controls or modifying the code, the compiled class code is invalidated and a new class is generated.

Code-Behind Pages

- Code-behind pages are by default present in Web application projects and are optional in Web site projects.
- In the code-behind model, the markup of page and server-side elements, including control declarations, are in an .aspx file and our page code is in a separate code file.
- Partial class included in code file i.e partial keyword used in class declaration indicates that it includes only some of the total code that create the full class for the page.
- In the partial class, we add the code that our application needs for the page. This typically consists of event handlers, but can include any methods or properties that we need.
- The inheritance model for code-behind pages is difficult to implement than that for single-file pages.

Syllabus Topic : Global.asax File

1.4.6 Global.asax File

Q. What is global.asax file ? (1 Mark)

- This is an optional file which is known as ASP.NET application file.
- If we does not define this file then ASP.NET page framework assumes that application or session event handlers are not defined.
- This file permits writing event handlers that handles the global events.

Syllabus Topic : Web.config

1.4.7 Web.config

Q. Write note on web.config (4 Marks)

- ASP.NET applications use configuration system that enables defining configuration settings for the web server or a web site.
- These setting can be done at time when ASP.NET application is delivered to client.
- Configuration setting can be added or repeated at any time with less impact on application and web server functions.
- These setting are stored in XML/K-based file.
- It is simple to make changes in configuration of application because format of this file is ASCII text file.
- ASP.NET uses two files for configuration setting of application which are machine.config and web.config.

Review Questions

Q. 1 What is .NET framework architecture ? (4 Marks)
(Refer Section 1.1)

Q. 2 Explain data types in C#. (4 Marks)
(Refer Section 1.2.2)

Q. 3 What is constant ? Explain types of constant in c#. **(Refer Section 1.2.3)** (4 Marks)

Q. 4 What is keyword and list some keyword in C# ? (2 Marks)
(Refer Section 1.2.5)

Q. 5 Write note on types of function call. (4 Marks)
(Refer Section 1.2.9(B))

Q. 6 Write note on namespaces. (4 Marks)
(Refer Section 1.2.12(A))

Q. 7 Write note on assemblies. (4 Marks)
(Refer Section 1.2.12(C))

Q. 8 What is mean by casting object ? Explain types of casting object. (4 Marks)
(Refer Section 1.2.15)

Q. 9 What is mean by code-behind class? How to write code behind class? (4 Marks)
(Refer Section 1.3.4)

Q. 10 What is mean by event ? (2 Marks)
(Refer Section 1.3.5(A))

Q. 11 What is mean by view state and how to create view state? **(Refer Section 1.4.1)** (4 Marks)

Q. 12 Write note on HtmlContainerControl class? (4 Marks)
(Refer Section 1.4.3)

Q. 13 Write note on HtmlInput class. (4 Marks)
(Refer Section 1.4.4)

UNIT II

CHAPTER 2

Web Controls

Syllabus Topics

Web Controls : Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle.

State Management : ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State.

Validation : Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions.

Rich Controls : Calendar Control, AdRotator Control, MultiView Control

Themes and Master Pages : How Themes Work, Applying a Simple Theme, Handling Theme Conflicts, Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths.

Website Navigation : Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control.

Syllabus Topic : Web Control

2.1 Web Controls

Q. List all ASP.Net web controls. (2 Marks)

- Small building blocks of the Graphical User Interface are controls, which include buttons, check boxes, list boxes, text boxes, labels, and various other tools.
- Users can enter data, make selections and state priority of selection using these web controls.
- Structural jobs such as validation, data access, security, creating master pages, and data manipulation etc. can be done using these web controls.
- In ASP.NET,there are five types of web controls available which are :
  i) HTML controls
  ii) HTML server controls
  iii) ASP.NET server controls
  iv) User controls and custoASP.NET
  v) User controls and custom controls

Syllabus Topic : Web Control Classes

2.1.1 Web Control Classes

- ASP.NET web control classes are the basic control classes used in ASP.NET.
- These web control classes can be grouped into the following categories :

i) **Validation controls :** Web control classes of this type are used to validate user input and they work by using client-side script which is written in clientside scripting language - JavaScript.

ii) **Data source controls :** Web control classes of this type provides feature of data binding to different data sources.

iii) **Data view controls :** Web control classes of this type are various lists and tables which can bind the data from data sources for displaying the data.

v) **Personalization controls :** Web control classes of this type are used for personalization of a page according to the user priority based on user information.

v) **Login and security controls :** Web control classes of this type are controls which provide user authentication.

vi) **Master pages :** Web control classes of this type provides constant layout and interface throughout the application.

vii) **Navigation controls :** Web control classes of this type helps in navigation. For example, menus, tree view etc.

viii) **Rich controls :** Web control classes of this type implement various types of special features. For example, AdRotator, FileUpload, and Calendar control.

Syntax

```
<asp:controlType ID="ControlID" runat="server"
Property1=value1 [Property2=value2] />
```

- Visual studio provides following features while writing program in ASP.NET:

1) Visual studio provides Dragging and dropping facility for controls in design view.
2) Visual studio provides IntelliSense feature which show and auto-complete the properties.
3) The properties window is used to set the values of property directly.

Properties of the web Control classes

1) ASP.NET web control classes with a visual aspect are derived from the WebControl base class and can access all the properties, events, and methods of this class.
2) The WebControl class itself and some other web control classes that are not visually rendered are inherited from the System.Web.UI.Control class. For example, PlaceHolder control or XML control.
3) ASP.Net web control classes access all properties, events, and methods of the WebControl and System.Web.UI.Control class.
4) The Table 2.1.1 state the derived properties, common to all web control classes.

Table 2.1.1

Property	Description
Back Color	This property is used to set background color of web control.
AccessKey	This property is used to set focus on web control.
BorderWidth	This property is used to set Border width of web control.
Attributes	This property makes the collection of arbitrary attributes which do not correspond to properties on the web control.
BindingContainer	Set the control that contains this control's data binding.
BorderColor	This property is used to set Border color of web control.
ClientID	This property state Control ID for HTML markup.
Causes Validation	This property shows if web control causes validation.
Border Style	This property is used to set Border style of web controls.
Child Control Created	This property state that whether child controls created from web control or not.
Enabled	This property show that control is enabled or disabled i.e active or deactive.
Context	This property define HttpContext object which is associated with the server control.
ControlStyle	This property set style of the Web server control.
Css Class	This property state CSS class for our current program.
Font	This property set font of content in web controls.
DataKeysContainer	If the naming container implements IDataKeysContainer then this property is used to get a reference to the naming container.
ID	This property set Identifier for the control.
EnableTheming	This property state whether theming applies to the web control or not.

Property	Description
EnableViewState	This property state whether the view state of the control is maintained or not.
Events	This property is used to get a list of event handler delegates for the control.
Forecolor	This property is used to set Foreground color of web control.
HasAttributes	This property state whether the control has set of attributes or not.
HasChildViewState	This property state whether the current child controls of web controls have any saved view-state settings or not.
Height	This property is used to set Height of web control in pixels or %.
IsEnabled	This property is used to get a value which state whether the control is enabled or not.
Page	This property is used to create Page containing the control.
Parent	This property is used to set Parent for web control.
SkinID	This property is used get and set skin to apply to the control.
Style	This property is used to get a collection of text attributes that will be rendered as a style attribute on the outer tag of the Web server control.
TagName	This property is used to get the name of the control tag.
UniqueID	This property is used to set Unique identifier to web control
ViewState	This property is used to get a dictionary of state information that saves and restores the view state of a web control across multiple requests for the same page.
Visible	This property states whether a server control is visible.
Width	This property is used to get and set the width of the Web server control.

Methods of the web Control classes

Table 2.1.2 provides the some methods of the web control classes.

Table 2.1.2

Method	Description
AddAttributesToRender	To add HTML attributes and styles in our web page that need to be rendered to the specified HtmlTextWriterTag, AddAttributesToRender() is used.
SaveViewState	This method is used save any state that was modified after the TrackViewState method was invoked.
Focus	This method is used to Sets input focus to a control.
ClearChildState	This method is used to deletes the view-state and control-state details of child controls of web control.
ClearChildViewState	This method is used to delete the view-state information for all the child controls of web controls.
CreateChildControls	This method is used in creating child controls.
DataBind	This method is used to binds a data source to the web control and all its child controls.
DataBind(Boolean)	This method is used to bind a data source to the server control and all its child controls with an option to raise the DataBinding event.
Dispose	This method is used to enable a web control to perform final clean up before it is released from memory.
EnsureChildControls	This method is used to check whether the web control has child controls. If there is no child control exists then creates it.

Method	Description
EnsureID	This method is used to create identifier for controls that do not have an identifier.
Equals(Object)	This method state whether the specified object is equal to the current object or not.
Finalize	Before reclaimed the object by garbage collection, this method allows an object to attempt to free resources and perform other cleanup operations.
FindControl(String)	This method performs searching the current naming container for a server control with the determined id parameter.
GetType	This method is used to get the type of the current instance.
HasControls	This method checks whether the server control contains any child controls or not.
HasEvents	This method states whether events are registered for the control or any child controls.
OnInit	This method is used to raises the Init event.
OnLoad	This method is used to raises the Load event.
OnPreRender	This method is used to raise the PreRender event.
OnUnload	This method is used to raise the Unload event.
OpenFile	This method is used to open the file.
RemovedControl	This method is called after a child control is removed from the controls collection of the control object.
Render	This method is used to render the control to the specified HTML writer.
SaveControlState	This method is used to Saves any server control state changes that appear.
ToString	This method Returns string which represents the current object.
TrackViewState	This method is used to control track of changes to view state of control so that they can be stored in the object's view state property.

Syllabus Topic : WebControl Base Class

2.1.2 WebControl Base Class

- The Web Control class is Serves as the base class which creates the methods, properties and events for all web controls in the System.Web.UI.WebControls namespace.

Syntax

```
Public class WebControl : Control, IAttributeAccessor
```

Following is list of constructors in Web Control base class

Sr. No.	Name	Description
1.	WebControl()	This default constructor is used for initialization of a new instance of the Web Control class that represents a Span HTML tag.
2.	WebControl (HtmlTextWriterTag)	This parameterized constructor is used for initialization of new instance of the Web Control class by using the specific HTML tag.
3.	WebControl(String)	This parameterized constructor is used to initialize a new object of the Web Control class using the specific HTML tag.

Events

Sr. No.	Name	Description
1.	DataBinding()	This event occurs when data source is bind to server control.
2.	Init	This event is occurs when initialization of the server control is done, which is the first step in its lifecycle ASP.NET page

Sr. No.	Name	Description
3.	Load	This event occurs when web control is loaded in page object.
4.	PreRender	This event occurs after the page object has created all the controls that are needed in order to render the page. This page object raises prerender event on the page object and then repeat same for each child control. The prerender event of the page is followed by prerender event of individual Controls.
5.	Unload	After rendering of page, the real cleanup started and the page unload event is raised to clean the memory occupied by that application.
6.	Disposed	This event occurs when a web control is released from memory, which is the last stage of the server control lifecycle when an ASP.NET page is requested.

Syllabus Topic : List Controls

2.1.3 List Controls

Q. Write short note on : List Controls. (4 Marks)

- List controls in ASP.NET are special web controls that contains Drop-down list,List box,Radio button list,Check box list,Bulleted list etc. that may or may not bound to data source or programmatically fills with items.
- These controls allow a user to choose from one or more items from the list.
- These controls are derived from System.Web.UI - WebControls.ListControl class.
- All controls from list controls allow user to select options except Bulletedlist. Bulletedlist show static data.
- RadioButtonList or CheckBoxList provides multiple checkboxes or option buttons.
- DropDownList and ListBox shows predefined values or values which are added at runtime.
- Default event for all this controls is SelectedIndexChanged except for BulletedList.
- This event is fired automatically when autopostback property of control is set to true.

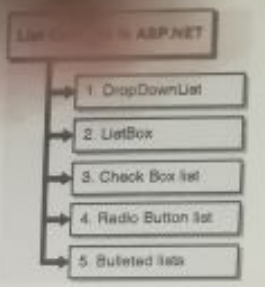


Fig. C2.1 : List Controls in ASP.NET

1. DropDownList

Syntax

```
<asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="True" OnSelectedIndexChanged
="DropDownList1_SelectedIndexChanged">
</asp:DropDownList>
```

- This control permit users to select item from predefined list.
- It does not remain hidden until user click on dropdown button like ListBox and it does not support for selecting multiple items at same time.

Following are some properties of drop-down Lists

Property	Description
Items	This property provides group of List Item objects that presents the items in the control. Object of type List Item Collection is returned by this property.
Selection Mode	This property state whether a list box permits single selection or multiple selection.
Rows	This property determines the number of items shown in the box. Scroll bar is added when necessary.
Selected Index	This property determines index of the currently selected item. The index of the first selected item is determined by this property if more than one items are selected. If no item is selected then value of this property is -1.

Property	Description
Selected Value	Returns the value of the currently selected item. If more than one items are selected, then returns the value of the first selected item. If no item is selected, the value of this property is an empty string ("").

2. ListBox

syntax

```
<asp:ListBox ID="ListBox1" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ListBox1_SelectedIndexChanged">
</asp:ListBox>
```

- ListBox web control can be used to show multiple items at a time which permit user to select one or more items from predefined list.

Following are some properties of ListBox

Property	Description
Items	This property provides group of List Item objects that presents the items in the control. Object of type List Item Collection is returned by this property.
Selection Mode	This property state whether a list box permits single selection or multiple selection.
Rows	This property determines the number of items shown in the box. Scroll bar is added when necessary.
Selected Index	This property determines index of the currently selected item. The index of the first selected item is determined by this property if more than one items are selected. If no item is selected then value of this property is -1.
Selected Value	Returns the value of the currently selected item. If more than one items are selected, then returns the value of the first selected item. If no item is selected, the value of this property is an empty string ("").

3. Check Box list

Syntax

```
<asp:CheckBoxList ID="CheckBoxList1" runat="server" AutoPostBack="True" OnSelectedIndexChanged="CheckBoxList1_SelectedIndexChanged">
</asp:CheckBoxList>
```

- It provides multiselection check box group that can be populated at design time as well as runtime.
- It includes Items collection with members corresponding to each item in list.
- To specify checked items, group can be iterated and selected property of each item in list can be tested.

Following are properties of check box lists

Property	Description
Repeat Layout	This property state the table tags or flow to use when formatting the list while it is run server. The default RepeatLayout Table.
Repeat Columns	This property state the column number to use when repeating the controls. Default value of RepeatColumns is 0.
Repeat Direction	This property states the direction in which the controls to be repeated. Available values for Repeat Direction are Horizontal and Vertical. Default value of RepeatDirection is Vertical.

4. Radio Button list

- A radio button list shows a list of mutually exclusive options.

syntax

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server" AutoPostBack="True" OnSelectedIndexChanged="RadioButtonList1_SelectedIndexChanged">
</asp:RadioButtonList>
```

- To allow user to select from small set of mutually exclusive predefined options, this control is used.
- These control permits to define any number of radio buttons with labels.

5. Bulleted lists

- It permit defining list of items either by creating static items at design time or by binding control to data source to manipulate at run time.

Syntax

```
<asp:BulletedList ID="BulletedList1" runat="server">
</asp:BulletedList>
```

- If we know at design time the types of items need to display, we can set group of control items to set of individual items in markup.
- If items to be shown are dynamic ,we can create group of items in code at run time.

Common properties of the bulleted list

Property	Description
BulletStyle	To state the style and looks of the bullet list and number of bullets, this property is used.
RepeatColumns	This property state the column number to use when repeating the controls. Default value of RepeatColumns is 0.
RepeatDirection	This property states the direction in which the controls to be repeated. Available values for RepeatDirection are Horizontal and Vertical. Default value of RepeatDirection is Vertical.

Syllabus Topic : Table Controls

2.1.4 Table Control

Q. Explain any four properties of TableControl class. (4 Marks)

- In the .NET Framework, the Table class is used to build an HTML table.
- Table class is included in System.Web.UI.Controls namespace.
- The Table control is used with the TableCell control and the TableRow control to create a table in .NET.
- We can create a Table control at run-time as well as at design-time using the Visual studio.

Following are the ASP.NET Table and Helper control classes

Control	Code	Description
Table	<asp:Table>	Table class used to create an HTML table with the help of TableRow and Table Cell class.
Table Row	<asp:Table Row>	TableRow class used to create a row in the table which can be useful for getting and setting cells value of rows using TableCell control.
Table Cell	<asp:Table Cell>	Table Cell class used to create cell in table.
Table Row Collection	<asp:Table Row Collection>	Table Row Collection class is used to maintains group of table rows by inserting and removing a row from it.
TableCellCollection	<asp:TableCell Collection>	Table Cell Collection class is used to maintain group of table cells by adding a cell to a row and removing a cell from row.
TableHeaderCollection	<asp:Table HeaderCell>	Table Header Collection class is used to create header cells of table.

Properties of Table class

Property	Description
Runat	This property state that the web control is a server control.For this purpose we have to set value of runat to "server".
Back Image Url	This property state URL to an image which is used as background to table.
Rows	This property state group of rows in the table.
Caption	This property is used to set title to table.
Caption Align	This property is used to set alignment of the caption text.
Horizontal Align	This property is used to set alignment of table as in the page.
Cell Padding	This property is used to state the space between the cell walls and contents in table.
Cell Spacing	This property is used to determines distance between cells of tables.
Grid Lines	This property is used to set gridline format in the table.

Program 2.1.1

Write a program to create HTML table using Table class in ASP.NET.

Solution :

Program to create HTML table using table class in ASP.NET

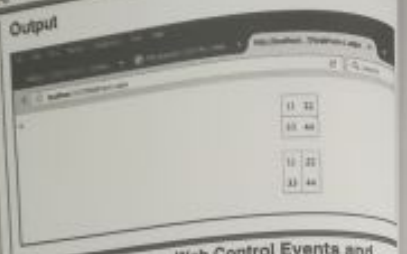
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="Form1" runat="server">
<asp:Table ID="Tbl" runat="server" CellPadding="5"
GridLines="Horizontal" HorizontalAlign="Center">

<asp:TableRow>

<asp:TableCell>11</asp:TableCell>
<asp:TableCell>22</asp:TableCell>
</asp:TableRow>
<asp:TableRow>
<asp:TableCell>33</asp:TableCell>
<asp:TableCell>44</asp:TableCell>
</asp:TableRow>
</asp:Table>
<br />
<asp:Table ID="Tbl" runat="server" CellPadding="5"
GridLines="Vertical" HorizontalAlign="Center">
<asp:TableRow>
<asp:TableCell>11</asp:TableCell>
<asp:TableCell>22</asp:TableCell>
</asp:TableRow>
<asp:TableRow>
<asp:TableCell>33</asp:TableCell>
<asp:TableCell>44</asp:TableCell>
</asp:TableRow>
</asp:Table>
</form>

</body>
</html>
```

Create table

Create table row

Create table cells which contains table data

Output



Syllabus Topic : Web Control Events and AutoPostBack

2.1.5 Web Control Events and AutoPostBack

Q. What are the types of web control events? What happen if AutoPostBack is set to "true"? (4 Marks

2.1.5(A) Web Control Events

- Events are occoured due to ASP.NET Web To server controls which work somewhat differently events in HTML pages or events in any W applications which runs at client side.
- The difference is at basic level because of separation of the event itself from where the event handled.
- In client-based applications, events are raised handled at the client side.
- In ASP.NET, events which are related with controls are occour due to client action but these are handled on the Web server by the ASP.NET pag
- For events occoured on the client side(browser ASP.NET Web Forms control event model needed the details of event be captured on the client side as message of event occurrence is transmitted to server using an HTTP post method.

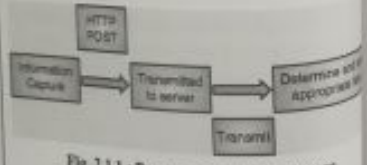


Fig. 2.1.1 : Process of event occourance

- The ASP.NET page must run the post metho determine which event is occurred and then cal correct method on the server to handle the that e
- Task of capturing, transmitting, and running the is handeled by ASP.NET

- ASP.NET web form supports many web controls like Check Box, Button, Radio Button List, Text Box etc.
- Each web control has related events.types of web control events are :

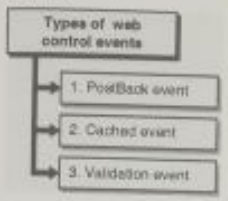


Fig. C2.2 : Types of web control events

➔ **1. PostBack event**

- In PostBack events, we send an ASP.NET page to the server for processing.
- PostBack event is occur when specific credentials of the page are to be cross checked against some sources i.e. verification of username and password using database which cannot done at client machine and thus these information have to be 'posted back' to the server.
- To capture a postback event using web control, the web control must implement the System.Web.UI.I PostBack Event Handler interface.
- Use of this interface permits web control to raise events on the server as response to postback from the client.
- The IPostBackEventHandler interface contains one method Raise Post Back Event() to handle Post Back event.

```
public interface IPostBackEventHandler
{
    void RaisePostBackEvent(string eventArgument);
}
```

➔ **2. Cached event**

- These events store data of page that gets processed when page is submitted to the server by using PostBack event.
- Cached event are placed at view state of page and executed at serverside

☛ **Example**

```
TextChanged event of TextBox
SelectedIndexChanged event of DropDownList
```

➔ **3. Validation event**

- Validation event is handled on the page before the page is posted back to server.
- The sequence of web control events on a Web form is as below.
  1. First validations Event is occurred before the page is send to the server by client for processing.
  2. Postback Event occurs that cause the page to be send to the server.
  3. Then,Page_Init and Page_Load events are handled by server.
  4. Cached events are occourd and handled.
  5. Lastly, the event that caused the postback is processed.

2.1.5(B) AutoPostBack

- AutoPostback and Postback are same.
- These properties are used to submit pages to web server.
- In AutoPostback ,web page is submitted to server, then Server processes the values and server sends response back to same page or redirect that response to different page.
- Every Web controls will have their own AutoPostBack property.
- If we create a web Page which contains one or more Web Controls that are configured to use AutoPostBack property then the ASP.NET adds a special JavaScipt function in HTML Page while running.
- Name of this javascript function is _doPostBack().
- When we Call this function, it sends data to web server for processing i.e postback.
- Two hidden input fields are added by ASP.NET in web form that are used to pass details back to the server.
- These details contains ID of the Control due to which event occurs and any additional details if required.

☛ **Syntax for hidden field**

```
<input type="hidden" name="__EVENTTARGET"
id="__EVENTTARGET" value="" />
```

- It is responsibility of _doPostBack() function to set these values with the appropriate information about the event and submit the form.

- The _doPostBack() function is shown below.

```
<script language="text/javascript">
function __doPostBack(event_Target, event_Argument)
{
    if (Form.onsubmit() || (Form.onsubmit() != false))
    {
        Form.__EVENTTARGET.value = event_Target;
        Form.__EVENTARGUMENT.value =
                                    event_Argument;
        Form.submit();
    }
}
</script>
```

- ASP.NET creates the _doPostBack() function automatically for web control on the web page which uses automatic postbacks.
- For any control if the property AutoPostBack is set to true, then it is possible to connect that control to the _doPostBack() function with the help of onclick and onchange attributes.
- These attributes decides the type of action which should be carried out by the browser response to the Client-Side javascript events- onclick and onchange.
- That means ASP.NET automatically changes a javascript event which raise at client side into a server-side ASP.Net event by using the _doPostBack() function.

Program 2.1.2

Write program in ASP.NET to show ice-cream flavour selected by user using Autopostback

Solution :

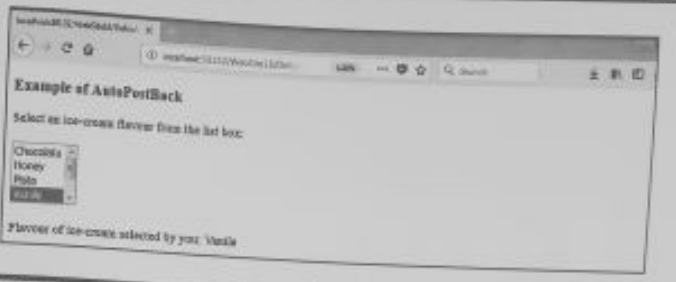
Program to show ice-cream flavour selected by user using Autopostback

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script runat="server">
    void Page_Load(Object sender, EventArgs e)
    {
        if (ListBox1.SelectedItem != null)
        {
            Label1.Text = "Flavour of ice-cream selected by
            you: " + ListBox1.SelectedItem.Value;
        }
        else
            Label1.Text = "";
    }

    </script>
</head>
<body>
    <form id="form1" runat="server">
    <h3> Example of AutoPostBack </h3>

    Select an ice-cream flavour from the list box: <br
    /><br />

    <asp:ListBox id="ListBox1"
    Rows="4"
    AutoPostBack="True"
    SelectionMode="Single"
    runat="server">

    <asp:ListItem>Chocolate</asp:ListItem>
    <asp:ListItem>Honey </asp:ListItem>
    <asp:ListItem>Pista</asp:ListItem>
    <asp:ListItem>Vanila</asp:ListItem>

    </asp:ListBox>

    <br /><br />

    <asp:Label id="Label1"
        runat="server">
    </form>
</body>
</html>
```

Show the flavour selected by user in textbox

Autopostback set to true for submitting web page to server, then Server processes the values and server sends response back to same page

Output



Syllabus Topic : Page Life Cycle

2.1.6 Page Life Cycle

Q. State the page life cycle with diagram. (4 Marks)

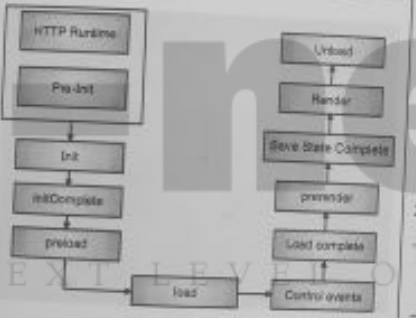


Fig. 2.1.2 : Page life cycle

- When a web page is requested, that request is send to web server and requested web page is loaded into the server memory which is then processed, and send to the browser as response. Then it is unloaded from the memory.
- At each stage, methods and events are available which can be overrided to fulfill the requirement of the application.
- These stages include initialization, instantiating controls, restoring and managing state, running event handler code, and running the page on server.

Steps of ASP.NET page life cycle

1. Pre-Init

- When page lifecycle starts, pre-init event is occoured before next Initialization stage begins.
- This event is used to perform following task :
  1. Check IsPostBack property to state wheather this is a first time page is being processed and determines whether the page is a postback.
  2. To create or recreate dynamic controls.
  3. Set master page or theme dynamically.
  4. To read or assign profile property values.

2. Init

- Init event occours after initialization of all controls is done. Unique id of each control is set and skin setting have been applied.
- Init event of individual control occurs before init event of page.
- This event can be use to read or initialize control properties.
- This event can be handled by overloading the OnInit method or by creating a Page_Init handler.

3. InitComplete

- It occurs when page initialization is done.
- The tracking of view state changes is turned on between Init and InitComplete events.
- View state tracking permit control to store any values that programmatically added to view state collection.
- If tracking of view state is turned off ,any values added to view state are lost during postback
- This event is usually used to make changes to view state that is to be stored during postback event.

4. Preload

- This event is occurred after page load its view state and repetitively load view state for all the controls on the page.
- Before the Page object raise this event, it loads view state for itself and all controls on that page, and then processes any postback data contained by the Request instance.

5. Load

- The Load event is first occours for page and then recursively for all child controls.
- To handle this event OnLoad() method on page object is called and repeat same for all the child controls on that page.
- Load event of page is followed by load event of individual controls.
- This event is useful for setting properties of controls on that page and to connect with database.

6. Control events

- When page is completely loaded and validated ASP.NET fires all the events which are occoured due to last postback.
- ASP.NET events has two types :
  1) **Immediate response event :** This event include clicking on submit button or some other button, image or link due to which post back occurs by calling _doPostBack() javascript function.
  2) **Change event :** This event include changing the selected value in web control i.e. checkbox,radio button or text in textbox. This event fired immediately for web controls if AutoPostBack property is set to true. Otherwise fire next time when page is postback.

7. LoadComlete

- This event is occured when the loading process is completed, event handlers of web controls are run and validation of page is done.
- It is occured at end of event handling stage.
- It is used for task that need all the web controls of page are loaded.

8. PreRender

- This event indicates the last action before output is rendered.
- During prerender stage, page and control instances are available so last-minute action such as storing additional information in view state is done.
- Page object fires PreRender event on page first and then on web controls on that page.
- This event is used to make final changes to contents of page or it's controls before page is rendered.

9. PreRenderComplete

- As the PreRender event is occured repetitively for all controls on the page, this event gurantee the completion of pre-rendering phase.
- This event is occour before DataBind() method for each control is called.

10. SaveStateComplete

- This event is fired when control state and view state information is saved.
- Any change to page or web controls on that page affect rendering of that page but changes are not retrived on next postback.

11. Render

- At the end of life cycle of page it is rendered to HTML.
- At this stage of life cycle, page call this method repetitively for all web controls.
- Each web server control invoke Render() method which create markup of control to be send to browser.

12. Unload

- The UnLoad phase is the last phase of the page life cycle.
- It fires the UnLoad event for all web controls repetitively first and then for the page .
- Final cleanup such as closing file stream,database connection etc. is done and all resources and references are free.
- This event can be handled by using the OnUnLoad() method.
- At this stage, page object is still available but final HTML is already rendered and cannot changed.

Syllabus Topic : State Management

2.2 State Management

Q. Why we need state management concept? List all state management techniques. (4 Marks)

- Web is Stateless that means a new web page object is re-created each time to serve request of client.
- HTTP is a stateless protocol that means it can not maintain the client information entered on page.

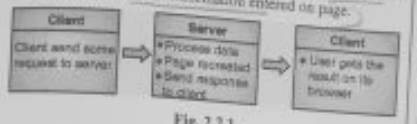


Fig. 2.2.1

- Page is recreated before it comes to clients and it happens for each and every request. So it is a big issue to maintain the state of the page and information for a web application.
- This issue can be solved by using State Management.
- ASP.NET Provides some features like View State, Cookies, Session, Application objects etc. to manage the state of page.

Syllabus Topic : View State

2.2.1 View State

Q. Write short note on view state. (4 Marks)

- View state is one of the most useful client side information management system.
- View state is the mechanism that permits state values of client to be stored during page postbacks.
- Because of the stateless behavior of web pages, values entered by user in web control is not maintained during postbacks.
- When we require values of page variable to be maintained during page postbacks, we can use View state to store those values.
- "EnableViewState" Property is used for both Page Level and Server Control Level to manage the view state.
- Values stored in hidden form fields are send to server to maintain state of client. When we view the page source in browser of a page that uses hidden form field to maintain View state, we see code like this.

```
<input type="hidden" name="viewstate" id="viewstate1"
value="123"/>
```

- This single hidden field contains all the view state values for all the page controls.
- ASP.NET pages provide the View State property as a built-in structure for automatically storing values between multiple requests for the same page.

Program 2.2.1

Write a program using viewstate to maintain and retrieve data entered by user.

Solution : Program using viewstate to maintain and retrieve data entered by user.

Web1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication8.WebForm1" %>
```

Create form with text box and two buttons

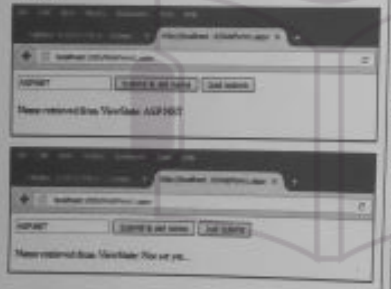
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <asp:TextBox runat="server" id="NameField1" />
    <asp:Button
    runat="server" id="SubmitForm" onclick="SubmitForm_Click" text="Submit & set name" />
    <asp:Button runat="server" id="RefreshPage" text="Just submit" />
    <br /><br />
        Name retrieved from ViewState: <asp:Label
    runat="server" id="NameLabel1" />
    </form>
</body>
</html>
```

Web1.aspx.cs (code behind)

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

namespace WebApplication8
{
Public partial class WebForm1 : System.Web.UI.Page
    {
Protected void Page_Load(object sender, EventArgs e)
```

```
{


if(ViewState["NameOfUser"] != null)
        NameLabel1.Text =
ViewState["NameOfUser"].ToString();
else
        NameLabel1.Text = "Not set yet...";
    }


protectedvoid SubmitForm_Click(object sender, EventArgs
e)
    {
ViewState["NameOfUser"] = NameField1.Text;
        NameLabel1.Text = NameField1.Text;
    }
  }
}
```

Output



Here enter string in the textbox and press the first button "Submit & set name". The name will be saved in the ViewState and set to the label which contains the entered string. Now press the second button. This button just posts back data to the server.

Limitation of view state

1. Viewstate can be used only with single page.
2. Because it stores information as hidden field, it can be seen in source code in browser, hence it is not secure way.

Syllabus Topic : Cross-Page Posting

2.2.2 Cross-Page Posting

Q. What is cross-page posting ? (4 Marks)

- When there is a need to transfer some data from one web page to another web page then usually we use session to maintain state.
- But the use of a session can not be good every time because the page becomes heavy due to session tracking.
- There is other way to maintain state of client and using which we can avoid use sessions, we can use a cross page postback. It simply transfers the data from one page to another.
- ASP.NET by default submit the forms to the same pages.
- Cross page posting submits the form to a different page.
- This is usually needed when we are generating a multipage form to group information from the user on every page, when moving from the source to the target page.

Program 2.2.2

Write a program using cross-page posting to maintain and retrieve data entered by user.

Solution :

Program using cross-page pasting to maintain and retrieve data entered by user

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication9.WebForm1" %>


 <html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table>
<tr>
<td><b>Enter UserName:</b></td>
```



```
<td><asp:TextBox ID="username"
runat="server"/></td>
</tr>
<tr>
<td><b>Enter city:</b></td>
<td><asp:TextBox ID="city" runat="server"/></td>
</tr>
<tr>
<td></td>
<td><asp:Button ID="btnPostback" Text="Postback"
runat="server"
PostBackUrl="~/WebForm2.aspx" /></td>
</tr>
</table>
</div>
</form>
</body>
</html>
```

Information entered in webForm1.aspx is send to WebForm2.aspx

WebForm2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication9.WebForm2" %>
 <html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<b><u>WebForm2.aspx Page</u>
</b><br /><br />
<label id="label1" runat="server" />
<br /><br />
<label id="label2" runat="server" />
</div>
</form>
</body>
</html>
```

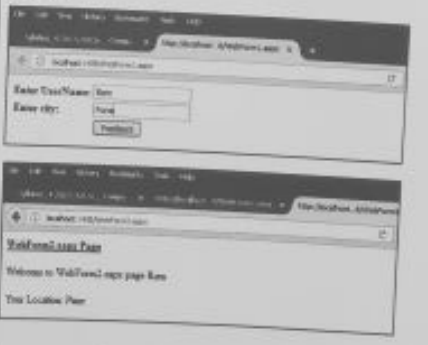
Label are used to show data enter by user at page WebForm1.aspx

Web2.aspx.cs

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


namespace WebApplication9
{
public partial class WebForm2 : System.Web.UI.Page
  {
protected void Page_Load(object sender, EventArgs e)
    {
if (PreviousPage != null &&
PreviousPage.IsCrossPagePostBack)
        {
TextBox name =
(TextBox)PreviousPage.FindControl("username");
            TextBox txtcity =
(TextBox)PreviousPage.FindControl("city");
            label1.InnerText = "Welcome to
WebForm2.aspx page " + name.Text;
            label2.InnerText = "Your Location: " +
txtcity.Text;
        }
else
        {
Response.Redirect("WebForm1.aspx");
        }
    }
  }
}
```

This label shows name and city data enter at web form 1.aspx

Output



Syllabus Topic : Query String

2.2.3 Query String

Q. What is query string ? (Ref Sec. 2.2.3) (2 Marks)

- Query string is the mechanism which used to send data from one web form to another web form using URL.
- Query string is made up of two parts; field and value and each pair is separated using ampersand (&).
- ?(Question Mark) is used at starting of a query string and it's value.
- There is a limitation to length of query string. So query strings cannot be useful to send very large data.
- Query strings are visible to the user, so it should not be used to send sensitive information such as username, Password without encryption.
- Request object of QueryString property is used to retrieve the query string.

Program 2.2.3

Write a program using query string to maintain and retrieve data entered by user.

Solution :

Program using query string to maintain and retrieve data entered by user

WebForm3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs"
Inherits="WebApplication9.WebForm3" %>

<!DOCTYPE html>
<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div><b>QueryString Example</b></div><br />
<table>
<tr>
<td> <b>Enter UserId :</b> </td>
<td> <asp:TextBox ID="userid" runat
="server"> </td>
</tr>
<tr>
<td> <b>Enter UserName :</b> </td>
<td> <asp:TextBox ID="username"
runat="server"></td>
</tr>
<tr>
<td></td>
<td> <asp:Button ID="buttonSend" Text="Send Values"
runat="server"
onclick="btnSend_Click"> </td>
</tr>
</table>
</div>
</form>
</body>
</html>
```

Enter username and password data

WebForm3.aspx.cs

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

namespace WebApplication9
{
public partial class WebForm3 : System.Web.UI.Page
    {
protected void Page_Load(object sender, EventArgs e)
        {

        }

protected void btnSend_Click(object sender, EventArgs e)
        {
Response.Redirect("WebForm4.aspx?UserId=" +
userid.Text +
"&UserName=" + username.Text);
        }
    }
}
```

Send data to next page WebForm4.aspx

WebForm4.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm4.aspx.cs"
Inherits="WebApplication9.WebForm4" %>

<!DOCTYPE html>
```



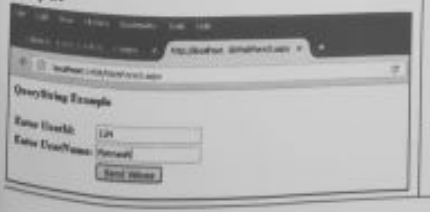
```
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div><b>QueryString parameter Values in
WebForm4.aspx Page</b></div><br />
<div><b>UserId:</b><asp:Label ID="labeluserid"
runat="server"/></div><br />
<div><b>UserName:</b><asp:Label
ID="labelusername" runat="server"/></div>
</form>
</body>
</html>
```

WebForm4.aspx.cs

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication9
{
public partial class WebForm4 : System.Web.UI.Page
    {
protected void Page_Load(object sender, EventArgs e)
        {
if (!IsPostBack)
            {
                labeluserid.Text =
Request.QueryString["userid"];
                labelusername.Text =
Request.QueryString["username"];
            }
        }
    }
```

Display userid and username entered by user

Output



Syllabus Topic : Cookies

2.2.4 Cookies

Q. Explain cookies in details. (4 Marks)

- Cookie is a small piece of information stored on the client machine which is used to identify a user.
- It is used to store private information of user such as Username, Password, Address, Contact no. etc on client machines.
- Cookies are used for authentication of a user, store private information of user and shopping cart contents, or anything else that can be accomplished through storing text data.
- Cookies can also be used for transformation of data from one page to another page.
- For example, if a user requests a page from a web site, then the web application sends not just a page,but also cookie which includes the date and time of the page. The browser also gets the cookie, which is stored in a folder on the hard disk of client machine.
- After that, if user requests a page from the site again by entering the URL of web page in the browser, the browser see in the local hard disk for a cookie associated with that URL. If the cookie exists for that URL, the browser sends the cookie to the site with the page request. The application can then state the date and time on which user lastly visit our web site. We can use the information in the cookie to display a message to the user or check an expiration date.
- Cookies are not associated with a Web page, it is associated with web site. So the server and browser will exchange information stored in cookie without limitation of what page the user wants from our site. As the user visits many web sites, each web site may send a cookie to the browser of user. All the cookies separately stored by the browser.
- To use cookie concept, we need to import namespace called Systen.Web.HttpCookie.

Type of Cookies

There are two types of cookie :

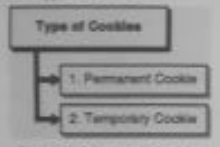


Fig. C2.3 : Type of Cookies

1. Permanent Cookie

- A cookie which does not have expiration time is called as Permanent Cookie.
- Permant cookie are stored as file on hard disk of client machine. These cookies are stored in "C:\Documents and Settings\username\Cookies" folder. Permanent cookie is created by setting "Expires" property.

2. TemporaryCookie

- A cookie which have expiration time is called as Temporary Cookie.
- Temporary Cookie is stored in RAM of client machine.
- It is transferred with request and response. Internally all cookies are temporary.
- It is easy to create a cookie in the ASP.NET with help of Response object of HTTPResponse class or HTTPCookie class.
- Each cookie have unique name so it can be identified later using that name.

Example

```
HttpCookie userInformation
= new HttpCookie("userInformation");
userInformation["username"] = "Ram";
userInformation["userColor"] = "White";
userInformation.Expires.Add(new TimeSpan(0, 1, 0));
Response.Cookies.Add(userInformation);
```

- It is easy to read cookie value form cookies by help of Request object of HTTPRequest class.

Example

```
string user_name = string.Empty;
string user_color = string.Empty;
user_name = Request.Cookies["username"].Value;
user_color = Request.Cookies["userColor"].Value;
```

Properties of cookies are:

1. **Domain** : Domain property is used to define association between cookies to domain.
2. **Secure** : We can set this property to true to permit creation of secure cookie.
3. **Value** : We can use this property to manipulate individual cookie.
4. **Values** : We can use this property to manipulate cookies with key/value pair.
5. **Expires** : This property is used to set expire date for the cookies.

Advantages of Cookie

1. Cookies contain text data only, hence user can easily read data is cookie .
2. We can store secure information of user on the client machine.
3. We can easily maintain cookies.
4. We can instantly acess the cookies.
5. Cookie can work transparently without user being aware that information needs to be stored.

Disadvantages of Cookie

1. If user deletes cookie information, we can not get that information back. There is no backup facility for cookie.
2. Each request of client will have associated cookie information.

Limitation of cookie

1. Cookies is small text file so it can store small amount of data approximately 4096 bytes of data.
2. It is not secure as it is stored on client side and are temporary.
3. In some cases browser does not support cookie.

- We can clear information in cookies in following way.
- We can delete cookie information from client machine from the cookie folder by setting Expires property.

```
userInformation.Expires = DateTime.Now.AddHours(1);
```

- Cookie information will deleted after the one hour of cookie creation.

Program 2.2.4

Write a program to use dropdownlist to show different color options such as pink ,red etc., maintain that selected value of color and set it as background color using cookies.

Solution :

A program to use dropdownlist to show different color options such as pink ,red etc., maintain that selected value of color and set it as background color using cookies

WebForm6.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm6.aspx.cs"
Inherits="WebApplication9.WebForm6" %>

<html>
<head runat="server">
<title></title>
</head>
<body runat="server" id="BodyTag">
<form id="form1" runat="server">

<asp:DropDownList runat="server" id="ColorSelector"
autopostback="true"
onselectedindexchanged="ColorSelector_IndexChanged">
<asp:ListItem value="White">Select
color...</asp:ListItem>
<asp:ListItem value="Pink">Pink</asp:ListItem>
<asp:ListItem value="Green">Green</asp:ListItem>
<asp:ListItem value="Yellow">Yellow</asp:ListItem>
</asp:DropDownList>
</form>
</body>

</html>
```

Create dropdown list

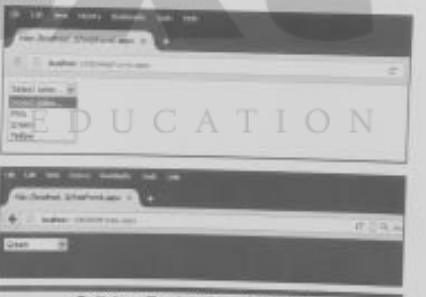
WebForm6.aspx.cs

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

namespace WebApplication9
{
public partial class WebForm6 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void ColorSelector_IndexChanged(object sender,
    EventArgs e)
    {
    BodyTag.Style["background-color"] =
    ColorSelector.SelectedValue;

    HttpCookie cookie = new HttpCookie("BackgroundColor");
        cookie.Value = ColorSelector.SelectedValue;
        cookie.Expires = DateTime.Now.AddHours(1);
    Response.SetCookie(cookie);
    }
  }
}
```

Set selectedcolor as background color

Create cookie using HttpCookie class and set value and expiration time of Cookie. Cookie will expire after one hour

Output



Syllabus Topic : Session State

2.2.5 Session State

Q. Explain session in detail. (4 Marks)

- A session is the timespan in which a user communicate with a Web application.
- ASP.NET session state permit us to store and retrieve values regarding user as the user visits different ASP.NET pages in a Web application.

- HTTP is a stateless protocol that means Web server treats each HTTP request for a page as a different request.
- The server does not maintain knowledge of variable values that were used during previous requests.
- ASP.NET session state identifies requests from the same browser during a limited time as a session, and provides a way to persist variable values for the duration of that session.
- By default, ASP.NET session state is enabled for all ASP.NET applications.
- ASP.NET manage session state by assigning unique Id to the user when the session is started.
- Sessions can be identified easily using unique identifier that can be read by using the SessionID property.
- When session state is enabled for web application, every request for web page in that application is analyzed for a SessionID value sent from the browser.
- If there is no SessionID value exists then ASP.NET starts a new session and the value of SessionID for that session is sent to the browser with the response.
- By default,cookie contains value of SessionID.
- We can also configure the settings of application to store SessionID values in the URL for a "cookieless" session.
- A session is supposed as active till requests are continually made through browser using same value of SessionID.
- If the time between two requests for a specific session cross the particular time-out value in minutes, the session is supposed to be expired.
- Requests made with an SessionID value which is expired can create new session.
- The SessionID is stored in an HTTP cookie and is send by client to the server on each request. The server can then read the SessionID from the cookie and restart the server session state.
- Config.web file which is ASP.NET XML configuration file is used to configure ASP.NET setting. Configuration files have two types: a machine configuration file and an application configuration file.
- Both machine configuration file and an application configuration file has name config.web. These two configuration files are same, except that the machine configuration file applies configuration settings to all applications but the application configuration files applies configuration settings to specific application.

Program 2.2.5

Write a program to accept username and password from user and maintain that data on next page using session state.

Solution :

A program to accept username and password from user and maintain that data on next page using session state.

webForm7.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm7.aspx.cs"
Inherits="WebApplication9.WebForm7" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
    User Name:-<asp:TextBox ID="txtusername"
runat="server"></asp:TextBox>
<br />
<br />
Password:-<asp:TextBox ID="txtpwd"
runat="server"></asp:TextBox>
<br />
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Submit" />
</div>
</form>
</body>
</html>
```

Take user name and password from user

WebForm7.aspx.cs

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

namespace WebApplication9
{
```



```
public partial class WebForm7 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
    {

    }

protected void Button1_Click(object sender, EventArgs e)
    {
        //textbox value is stored in Session
Session["UserName"] = txtusername.Text;
Session["Pwd"] = txtpwd.Text;

Response.Redirect("WebForm8.aspx");
    }
}
```

Conduct the session and preserve the username and password value

WebForm8.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm8.aspx.cs"
Inherits="WebApplication9.WebForm8" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
    User Name:-<asp:TextBox ID="txtUserName1"
runat="server"></asp:TextBox>
<br />
<br />
Password:-<asp:TextBox ID="txtpwd1" runat="server"
ontextchanged="txtpwd_TextChanged"></asp:TextBox>
<br />
</div>
</form>
</body>
</html>
```

WebForm8.aspx.cs

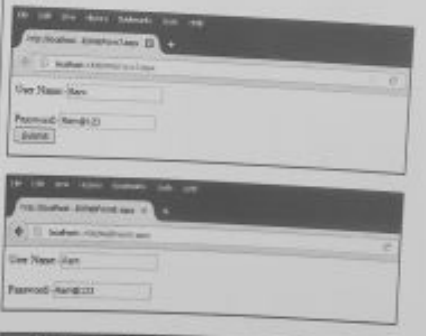
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication9
{
public partial class WebForm8 : System.Web.UI.Page
    {
protected void Page_Load(object sender, EventArgs e)
    {
        //Session value is assign on the text box
if (Session["UserName"] != null)
    {
        txtUserName1.Text =
Session["UserName"].ToString();
    }
if (Session["Pwd"] != null)
    {
        txtpwd1.Text = Session["Pwd"].ToString();
    }
}
protected void txtpwd_TextChanged(object sender,
EventArgs e)
    {
    }
}
```

Display value username preserve in session

Display value password preserve in session

Output



Syllabus Topic : Configuring Session State

2.2.6 Configuring Session State

Q. Which attributes are used to configure a session state? (4 Marks)

- Following is config.web file which is used to configure the session state settings for our ASP.NET application.

```
<configuration>
<sessionstate
mode="inproc"
cookieless="false"
timeout="25"
sqlconnectionstring="data source=127.0.0.1;user
id=<user_id>;password=<pwd>"
server="127.0.0.1"
port="6534" />
</configuration>
```

Following are attributes in <configuration> tag

i. **Mode** :There are three choices for mode: inproc, sqlserver, and stateserver.

ASP.NET supports two modes : Out of process and in process. There are also two options for out-of-process state management : memory based i.e. state server and SQL Server based i.e. sqlserver.

ii. **Cookieless** : This option has Boolean value true or false.

iii. **Timeout** : This option is used to set the amount of time of session for which session is valid. On each request the timeout value of session is set as the current time and the timeout value.

iv. **Sqlconnectionstring** : The sqlconnectionstring is used to identify the each database individually by assigning name to database.

v. **Server** : In the out-of-process mode stateserver, this attribute is used to assign name to the server that is running the ASP.NET program.

vi. **Port** : This port attribute is used to set port number of application.

Example

```
<Script runat=server>
Public void Session_Add(sender As Object, e As
EventArgs)
{
Session("SessionStateDemo") = text1.Value;
span1.InnerHtml = " data of session is updated ! <P>
our session contains: <font color=blue>" +
Session("SessionStateDemo ").ToString() + "</font>"
}
Public void CheckSession(sender As Object, e As
EventArgs)
{
If (Session("SessionStateDemo ") = Isnull)
{
span1.InnerHtml = "There is no information in session!";
}
Else
{
span1.InnerHtml = "our session contains:
<font color=green>" +
Session("SessionStateDemo ").ToString() + "</font>"
}
}
</Script>
<form runat=server ID="Form12">
<input id=text2
type=text runat=server NAME="text2">
<input type=submit runat=server
OnServerClick="Session_Add()" Value="Add_to_Session
State" ID="Submit2" NAME="Submit2">
<input type=submit runat=server
OnServerClick="CheckSession" Value="View Session
State" ID="Submit3" NAME="Submit3">
</form>
<h1 size=1>
<font size=6><span id=span1 runat=server/></font>
```

- This simple page raise two server-side events ,one is the Add and second for View buttons, and also set the session state to the value entered in the text box.

There are four configuration settings

i. in-process mode ii. out-of-process mode
iii. SQL Server mode iv. State Server Mode

i. In-process Mode

- ASP.NET session state can be used in a same way as of classic ASP session state. This is called In-process mode.
- That means session state is maintained in process and if the process is re-cycled then state of process is lost.
- The performance of session state will be much fast. For example the time it takes to read from and write to the session state dictionary will be less.
- In-process mode is the default mode for ASP.NET web page.
- When In-process mode is set, the only other session config.web settings used are cookieless and timeout.

ii. Out-of-process Mode

- **Out-of process** mode is included with .NET SDK as a Windows NT service.
- The Windows service is used by ASP.NET for out-of process session state management.

We need Out-of-Process Session States because :

1. Default sessions are stored in-process mode. So when the process is failed i.e server go out of order or recycled, the session state is lost. It is happened even if the browser has the Session Key stored with it.
2. In-process mode is not good for web farms. If the application is deployed to the Web Farms with many machines and each request can be served by different machines, then an in-process session state could not track the user.

ii. SQL server mode

- SQLServer mode is used to store details of session state in a SQL Server database.
- Using this mode, we can ensure that session state is maintained even when the Web application is restarted.
- For using SQLServer mode, we need confirmation about properly installation of ASP.NET session state database on SQL Server.
- To configure an ASP.NET application to make use of SQLServer mode,we need to do following changes in Web.config file of our application :
  - Set value of mode attribute of the sessionState to SQLServer.
  - Set the sqlConnectionString attribute to a connection string for our SQL Server database.

```
<configuration>
<system.web>
<sessionState mode="SQLServer"
sqlConnectionString="Integrated
Security=SSPI;data
source=SampleSqlServer;" />
</system.web>
</configuration>
```

iv. State Server Mode

- StateServer mode is used to store information about session state in a process.
- To use StateServer mode, we should sure about ASP.NET state service which is running on the server used for the session store.
- After installing ASP.NET and AP.NET framework, the ASP.NET state service is installed as a service.
- The ASP.Net state service is installed at the below location :

```
systemroot\Microsoft.NET\Framework\versionNumber\aspn
et_state.exe
```

- To use StateServer mode, we need to do following changes in Web.config file of our application :
  - Set the mode attribute of the sessionState element to StateServer.
  - Set the stateConnectionString attribute to tcpip=serverName:42424.
- Below example display changes in web.config file where session state is stored on a computer whose name is DemoStateServer :

```
<configuration>
<system.web>
<sessionState mode="StateServer"
stateConnectionString="tcpip=DemoStateServer:42424"
cookieless="false"
timeout="20" />
</system.web>
</configuration>
```

Syllabus Topic : Application State

2.2.7 Application State

Q. Explain the application state in detail ? (4 Marks)

- Application State is one of the state management mechanism.
- Application state is a data repository that is available to all classes in an ASP.NET application.
- Application State is stored in the memory of the the web server which is user specific.
- Retrieving application state from server is faster than storing and retrieving information in a database.
- Session state is maintained for a single user session, but Application State is applicable for all users and sessions.
- There is no a default expiration period for applications like cookie and session.
- When we close the working process then application object will be expired.
- The data is shared between different users by a HTTPApplicationState class and the data can be stored here in form of a key/value pair.
- This key/value pair data can also be accessed using the application property of the HTTPContext class.
- Application state is a useful for storing small amounts of data that does not change from one user to another user.
- HttpApplicationState class is used to store Application state.

Application State Life Cycle

Step 1 : When web server receives request send by browser , it first verify the extension to state whether or not it is ISAPI. (ISAPI extensions are true applications that run on IIS and have access to all of the functionality provided by IIS.)

Because this request can only be handled by the ISAPI extension. If the extension is different then the request is handled by the server itself.

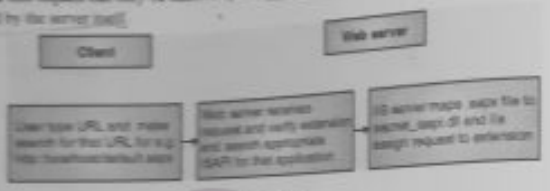


Fig. 2.1.2

Step 2 : After receiving the request, the application domain is created by application manager. In the application domain a instance of the class HostingEnvironment is created that facilates access to information about all application domains.

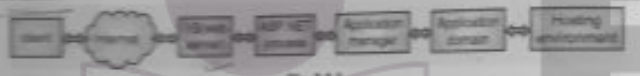


Fig. 2.1.3

Step 3 : After creating the application domain, ASP.NET initializes the basic objects as HTTPContext, HTTPRequest and HTTPResponse. HTTPContext holds objects to the specific application request. HTTPRequest contains all the information regarding the current request like cookies, browser information and so on and the HTTPResponse contains the response that is sent to the client.

Step 4 : Here all the basic objects are being initialized and the application is being started with the creation of the HTTPApplication class.

Step 5 : Then events are started by the HTTPApplication class.

Global.asax file

The Global.asax file is used for handling application events on methods. It resides at root level. Events are one of the following type in the Global application.

1. Events that will be raised on a certain condition.
2. Events that will be raised on every request.

The events of the Global.asax file are

1. **Application_Start()** : This method is used to call initially when first application domain is created.
2. **Session_Start()** : This method is called each time when a session is start.
3. **Application_BeginRequest()** : When an application has started, this method is executed for each user.
4. **Application_AuthenticateRequest()** : This method is used to checks whether the user is valid user or not.
5. **Application_Error()** : This event is raised in response to occurrence of unhandled exception.
6. **Session_End()** : When a user session is ended or all the data associated with that user is deleted then the Session_End() event is called.
7. **Application_End()** : This method is called before the application ends.
8. **Application_Disposed()** : This event is called after the application will be closed.

Program 2.2.6

Example of ASP.NET Application State.

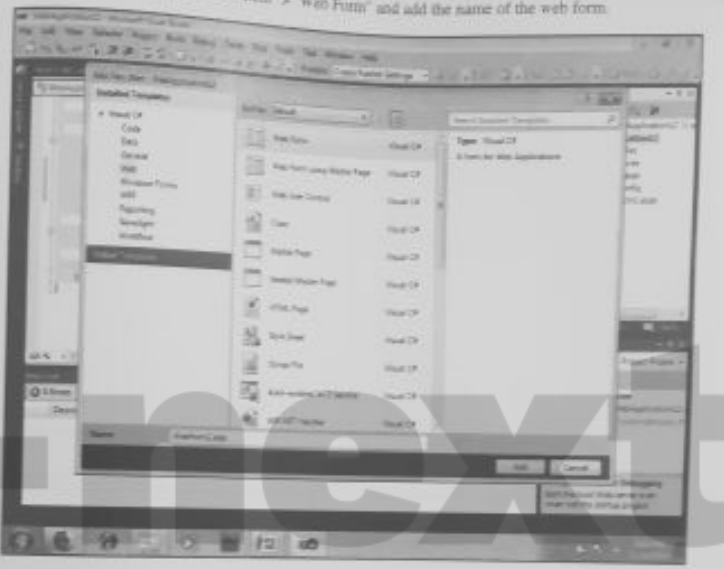
Solution : ASP.NET Application State

Step 1 : Open Visual Studio 2010.

Step 2 : click on "New Project" > "Web" > "ASP.NET Empty Web Application".

Step 3 : Now click on Solution Explorer.

Step 4 : Now right-click on "Add" > "New Item" > "Web Form" and add the name of the web form.



Step 5 : Create the Global.asax file. Again go to Solution Explorer and "Add" > "New Item" > "Global Application Class".code in Global.asax

Step 6 : For configure the session we need to use the web.config file as in the following :

```
<sessionState mode="InProc" timeout="20"
cookieless="true">
</sessionState>
```

Step 7 : For counting the number of online users we need to write following code in the Global.asax.cs file as in the following :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;
namespace WebApplication12
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender,
EventArgs e)
        {
            Application["user"] = 0;
        }
        protected void Session_Start(object sender, EventArgs e)
        {
            Application.Lock();
            Application["user"] = (int)Application["user"] + 1;
            Application.UnLock();
        }
```

This event is raised when application starts and it maintain server memory until worker process is restart

when session started application variable is increased by 1

```
protected void Application_BeginRequest(object
sender, EventArgs e)
{
}

protected void Application_AuthenticateRequest(object
sender, EventArgs e)
{
}

protected void Application_Error(object sender,
EventArgs e)
{
}

protected void Session_End(object sender, EventArgs e)
{
    Application.Lock();
    Application["user"] = (int)Application["user"] - 1;
    Application.UnLock();
}

protected void Application_End(object sender,
        EventArgs e)
{
}
```

when session ended, application variable is decrease by 1

Step 8 : Code behind the web form :

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication12
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      Response.Write("The number of online users :" +
Application["user"].ToString());
    }
  }
}
```

Output

Syllabus Topic : Validation and Validation Controls

2.3 Validation

Validation stands for checking data for as per the application requirements. ASP.NET provides various types of Validations controls to validate the data.

2.3.1 Validation Controls

Q. What is use of validation controls? List the different validation controls. (4 Marks)

- Validation controls in ASP.NET validate the user input data to assure that unauthenticated data should not get stored in database.
- Validation controls are used to :
  - i) Apply presentation logic.
  - ii) Verify data entered by user.
  - iii) Validate Data format, data type and data range of data entered by user

Validation control has two types

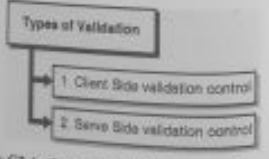


Fig C2.4 : Types of Validation Control

➜ 1. Client Side validation control

- Client side validation is suitable to users as they get the feedback immediately.
- The core advantage is that it stops the process of postback the page to server until the client validation is executed successfully.

- Client side validation is good but in it we have to be dependent on browser and scripting language support.
- For client script .NET uses JavaScript. WebUI Validation.js javascript file is used by .NET for client validation.

➜ 2. Serve Side validation control

According to developer point of view, serve side validation controls are suitable because server control will not fail and not dependent on browser and scripting language.

- We can use ASP.NET validation controls which will assure the client side as well as server validation.
- ASP.NET validation controls work at both the ends. First it will work on client validation and than on server validation.
- ASP.NET provides a set of validation controls which are user friendly and they provide powerful way to check for errors and if needed show error messages to the user.
- ASP.NET provides the following validation controls :
  1. Required Field Validator
  2. Range Validator
  3. Compare Validator
  4. Regular ExpressionValidator
  5. CustomValidator
  6. Validation Summary

Important points for validation controls

1. Control to Validate property is compulsory for all validation controls.
2. One validation control can be used to validate only one input control but multiple validation controls can be assigned to one input control.

Validation Properties

1. Validation of data in web form is done in response to user actions such click on submit button or entering data in textbox.
2. Server validation will only done when value of CauseValidation property is set to true.
3. When the value of the CausesValidation property is set to true ,we can also use the ValidationGroup property to state the name of the validation group for which the Button control causes validation.
4. Page has a Validate() method executes each validation control,if its value set to true.
5. Set the value of CauseValidation property to true for submit button as following :

```
<asp:Button ID="Button2" runat="server" Text="Submit"
CausesValidation=true />
```

Syllabus Topic : Server-Side Validation

2.3.2 Server-Side Validation

Q. Write note on server side validation. (4 Marks)

- This validation is executed after client-side validation is done, if we have client-side validation.
- Sever-side validation is compussary because a user or hacker can send the data through different channels also.
- Server-side validation is done after the user submits the data to server or data posts back to server.
- In the Server Side Validation, one of server side scripting languages such as ASP.Net, PHP etc is used to validate the data submitted by user to server.
- After finishing validation process on the Server Side, the feedback is sent to the client by server using a new dynamically created web page.
- It is better to validate user input on Server Side because we need to protect the data from unauthorized users or hackers.
- Server side validation is more Secure than client-side validation.

Syllabus Topic : Client-Side Validation

2.3.3 Client-Side Validation

Q. Explain client side validation in details. (4 Marks)

- In the Client Side Validation, we can give a better user experience by sending response quickly to browser.
- When we do Client Side Validation, all the user input data is validated in the browser of user.
- Client Side validation does not need a round trip to the server so it decrease the network traffic at server which will help the server to give better performance.
- Client side validation is done on the browser side using client-side scripting languages like JavaScript, VBScript etc.
- For example, if the user enter date in invalid format , our web application can show an error message instantly before the user move to the next field, so the user can correct every field before they submit the form.

- The Client Side Validation depends on the JavaScript Language. So if any user turn JavaScript off in setting of browser then the invalid data can easily bypass and submit to the server.
- So the Client Side Validation can not fully protect our web application from attacks on our server resources and databases.

Syllabus Topic : HTML5 Validation

2.3.4 HTML5 Validation

Q. List all HTML5 validation controls and explain any 2 HTML validation controls in details. **(4 Marks)**

Following are the HTML5 Form Validation controls in ASP.NET :

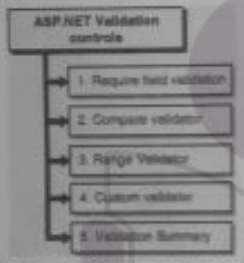


Fig. C2.5 : ASP.NET validation controls

1. Require field validation

- The Require field validation is used to verify user enter data in form filed such as textbox,checkbox , dropdownlist etc.
- Html5 attribute "required" is used to do Required Field validation.
- It is generally tied to a textbox to force input in it.

Syntax

```
<asp:RequiredFieldValidator ID="rfvemailid"
runat="server"ControlToValidate="emailid"
ErrorMessage="Please enter proper Emailid"
InitialValue=" Please enter proper Emailid ">
</asp:RequiredFieldValidator>
```

Program 2.3.1

Write a program to check whether name, email and age fields are filled by user or not. If not then display proper error message using Require field validation.

Solution :

A program to check whether name, email and age fields are filled by user or not. If not then display proper error message using Require field validation.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm6.aspx.cs"
Inherits="WebApplication7.WebForm6" %>
<html>
<head runat="server">
<title></title>
</head>
<body style = "font-family:Arial;font-size:10pt">
<form id="form1" runat="server">
Name: (Required)<br />
<asp:TextBox ID="txtName"
runat="server"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator1" CssClass
="Validators"
Display="Dynamic" ControlToValidate="txtName"
runat="server"
ErrorMessage="Name is
required."></asp:RequiredFieldValidator>
<br />
Email: (Required Email Address)<br />
<asp:TextBox ID="txtEmail"
runat="server"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator2"CssClass="Validators"
Display="Dynamic" ControlToValidate="txtEmail"
runat="server"
ErrorMessage="Email is
required."></asp:RequiredFieldValidator><br />
Age: (Required and Range 18 - 45)<br />
<asp:TextBox ID="txtAge"
runat="server"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator3" CssClass="Validators"
Display="Dynamic" ControlToValidate="txtAge"
runat="server"
```

Entering value in Name field is compulsory, if not enter then show message "Name is required".

User must enter value in Email field. If user does not enter value, error message shows as "Email is required"


```
ErrorMessage="Age is
required."></asp:RequiredFieldValidator>
<asp:RangeValidator ID="RangeValidator1"
CssClass="Validators" Display="Dynamic"
MinimumValue="18"MaximumValue="45"
Type="Integer" ControlToValidate="txtAge"
runat="server" ErrorMessage="Age Range 18 to
45"></asp:RangeValidator>
<br />
<br />
<asp:Button ID="Button1" runat="server" Text=
"Submit" />
</form>
</body>
</html>
```

User must enter value in age field and it must be in range 18-45 in age field.

Output

2. Compare Validator

- CompareValidator control compares value in one control with value in another control.
- For Example we can use Comapare validator to reconfirm the password provided in one textbox by user with value in another textbox.

Program 2.3.2

Write a program using compare validator to check value entered by user in password and confirm password field is same or not.

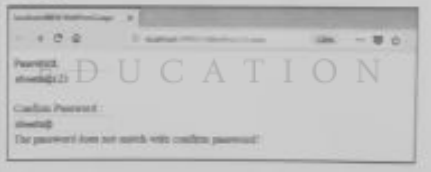
Solution :

A program using compare validator to check value entered by user in password and confirm password field is same or not

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs"
Inherits="WebApplication7.WebForm3" %>
<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
Password:<br />
<asp:TextBox runat="server" id="txtpwd" /><br />
<br />
Confirm Password :<br />
<asp:TextBox runat="server" id="txtconfirmpwd" />
<br />
<asp:CompareValidator runat="server" id="cmppwd"
controltovalidate="txtpwd"
controltocompare="txtconfirmpwd"
errormessage="The password does not match with
confirm password"
Type="String" />
<br />
</div>
</form>
</body>
</html>
```

Compare value in password field with confirm password field if values are not matched then display error message.

Output



3. Range Validator

- The RangeValidator control verifies that the input value should in between the range which is specified.

Range validator has three specific properties

Properties	Description
Type	This property state the type of the data. The available values for this property are:String,Integer, Currency, Date, Double.
MinimumValue	This property state lower limit of range.
MaximumValue	This property state upper limit of range.

Syntax

```
<asp:RangeValidator ID="rangevalidationclass"
runat="server" ControlToValidate="classroom"
  ErrorMessage="Enter your class (1 - 10)"
MaximumValue="10"
  MinimumValue="1" Type="Integer">

</asp:RangeValidator>
```

Program 2.3.3

Write a program to check age entered by user is between 18 and 100 using RangeValidator, if not then give appropriate error message.

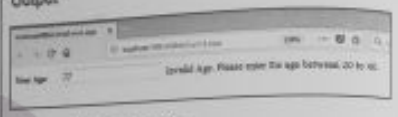
Solution :

A program to check age entered by user is between 18 and 100 using RangeValidator, if not then give appropriate error message.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication7.WebForm1" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

<asp:Label ID="Label2" runat="server"
Font-Bold="True" Font-Size="Small"
      Text="Your Age:"></asp:Label>

    

<asp:TextBox ID="TextBox1" runat="server"
Width="170px"></asp:TextBox>

<asp:RangeValidator ID="RangeValidator1"
runat="server"

      ControlToValidate="TextBox1"

      ErrorMessage="Invalid Age. Please enter the age
between 18 to 100."
      MaximumValue="100" MinimumValue="18"
Type="Integer"></asp:RangeValidator>

 <br />
</div>
</form>
</body>
</html>
```

Create textbox which accept age in range of 18 and 100

Output



4. Custom Validator

- For writing application specific custom validation routines for both the client side and the server side validations, CustomValidator control is used.
- The ClientValidationFunction property is used to perform client side validation.
- The client side validation routine should be written in a scripting language,like JavaScript or VBScript, which can be understood by the browser.
- ServerValidate control event handler is used to call the server side validation routine. The server side validation routine should be written in any .Net language, like C# or VB.Net.

Syntax

```
<asp:CustomValidator ID="CustomValidation1" runat
="server"
ClientValidationFunction="ms_func.ErrorMessage
="CustomValidator">

</asp:CustomValidator>
```

Program 2.3.4

Write a program to check Id enter by user is number or not using custom validator.

Solution :

Program to check id enter by user is number or not using custom validator

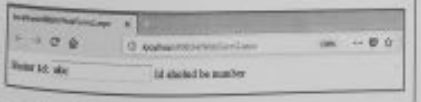
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication7.WebForm2" %>
```



```
<html>
<head id="Head1" runat="server">
<script language="javascript" type="text/javascript">
function check(sender, e) {
if (isNaN(e.Value))
      e.IsValid = false;
else

      e.IsValid = false;
  }
</script>

<title></title>
</head>

<body>
<form id="form1" runat="server">
  Enter Id:
<asp:TextBox ID="TextBox1" runat="server"
ontextchanged="TextBox1_TextChanged">
</asp:TextBox>
<asp:CustomValidator ID="CustomValidator1"
ControlToValidate="TextBox1"
    ErrorMessage="Id shoud be number "
ClientValidationFunction="check"
runat="server"></asp:CustomValidator>
<div>
</div>
</form>
</body>
</html>
```

Check input provided by user for id in textbox is number or not using check() javascript function

Output



5. Validation Summary

- Validation Summary control does not perform any validation.
- But Validation Summary control is used to displays a list of all validation errors on the Web page.
- Show Summary property is set as true by default.
- For displaying all validation errors Dispaly Mode property is used whose value can be SingleParagraph,BulletList and List.

Program 2.3.5

Write a program to show list of all the validation error messages using validation summary.

Solution :

A program to show list of all the validation error messages using validation summary.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm7.aspx.cs"
Inherits="WebApplication7.WebForm7" %>

<html>
<head runat="server">
<title></title>
</head>
<body>

<h3>ValidationSummary Sample</h3>

<form id="Form1" runat="server">

<table cellpadding="10">
<tr>
<td>
<table bgcolor="#eeeeee" cellpadding="10">

<tr>
<td colspan="3">
<b>enter Credit Card Information</b>
</td>
</tr>
<tr>
<td align="right">
Enter  Card Type:
</td>
<td>
<asp:RadioButtonList id="RadioButtonList1"
                RepeatLayout="Flow"
runat="server">

<asp:ListItem>MasterCard</asp:ListItem>
```

```
<asp:ListItem>Visa</asp:ListItem>

</asp:RadioButtonList>
</td>
<td align="middle" rowspan="1">
<asp:RequiredFieldValidator
id="RequiredFieldValidator1"
        ControlToValidate="RadioButtonList1"
        ErrorMessage="Card Type is invalid."
        Display="Static"
        InitialValue=""
        Width="100%"
        Text="*"
runat="server">
</td>
</tr>
<tr>
<td align="right">
        Card Number:
</td>
<td>
<asp:TextBox id="TextBox1"
runat="server" />
</td>
<td>
<asp:RequiredFieldValidator
id="RequiredFieldValidator2"
        ControlToValidate="TextBox1"
        ErrorMessage="Card Number is invalid."
        Display="Static"
        Width="100%"
        Text="*"
runat="server">
</td>
</tr>

<tr>
<td></td>
<td>
<asp:Button id="Button1"
        Text="Verify"
runat="server" />
```

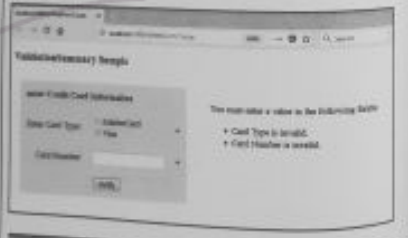
If card type is not selected from given options then error message is shown as "card type is invalid"

User must enter value in card number field.If user does not enter value error message is shown

```
</td>
<td></td>
</tr>
<tr>
</table>
</td>
<td valign="top">
<table cellpadding="20">
<tr>
<td>
<asp:ValidationSummary id="valSum"
        DisplayMode="BulletList"
        EnableClientScript="true"
HeaderText="You must enter a value in the following
fields:"
runat="server">
</td>
</tr>
</table>
</td>
</tr>
</table>

</form>

</body>
</html>
```

ValidationSummary control show list of validation error

Output



Syllabus Topic : Manual Validation

2.3.5 Manual Validation

- Client-side validation is triggered by default when submitting forms using buttons is done.

- Sometimes we may want to do client-side validation on our ASP page manually using custom JavaScript.
- We can do manual validation by calling Javascript validation functions provided by the ASP.NET framework.

Program 2.3.6

Write a program to demonstrate use of manual validation.

Solution :

Program to demonstrate use of manual validation.

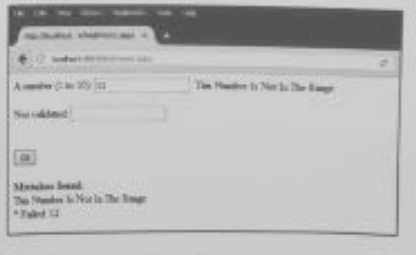
WebForm2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication10.WebForm2" %>

  <html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
    A number (1 to 10):
<asp:TextBox id="txtValidated"
runat="server"></asp:TextBox> 
<asp:RangeValidator id="RangeValidator"
runat="server"
        ErrorMessage="This Number Is Not In
        The Range"
        ControlToValidate="txtValidated"
        MaximumValue="10"
        MinimumValue="1"
        Type="Integer"

EnableClientScript="False"></asp:RangeValidator>
<br/>
<br />
  Not validated:
<asp:TextBox id="txtNotValidated"
runat="server"></asp:TextBox><br/>
<br />
<br />
<asp:Button id="cmdOK" runat="server" Text="OK"
OnClick="cmdOK_Click"
Width="36px"></asp:Button><br />
<br />
<asp:Label id="lblMessage" runat="server"
EnableViewState="False"></asp:Label>
</div>
</form>
</body>
</html>
```

WebForm2.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication10
{
public partial class WebForm2 : System.Web.UI.Page
  {
protected void Page_Load(object sender, EventArgs e)
    {
    }
protected void cmdOK_Click(object sender, EventArgs e)
    {
string errorMessage = "<b>Mistakes found:</b>
<br />";
bool pageIsValid = true;
foreach (BaseValidator ctrl in this.Validators)
    {
if (!ctrl.IsValid)
    {
pageIsValid = false;
errorMessage += ctrl.ErrorMessage + "<br />";
        TextBox ctrlInput =
(TextBox)this.FindControl(ctrl.ControlToValidate);
errorMessage += " * Failed: ";
errorMessage += ctrlInput.Text + "<br />";
    }
  }
if (!pageIsValid) lblMessage.Text = errorMessage;
  }
}
```

Output



Syllabus Topic : Validation with Regular Expression

2.3.6 Validation with Regular Expression

Q. Write short note on validation with regular expression. **(4 Marks)**

- The Regular Expression Validator permit us to validate the input text by matching against a pattern of a regular expression.
- The regular expression is used in the Validation Expression property of web control.
- Following are some escape sequence characters used in regular expressions :

Character Escapes	Description
\b	Used to find backspace.
\t	Used to find tab.
\r	Used to find a carriage return.
\v	Used to find vertical tab.
\f	Used to find form feed.
\n	Used to find new line.
\	Used to find Escape character.

- A class of characters could be state that can be matched, called the metacharacters.

Metacharacters	Description
.	Used to find any character except \n.
[abcd]	Used to find any character in the set.
[^abcd]	It is used excludes or remove any character in the set.
[2-7a-mA-M]	Used to find any character stated in the range.
\w	Used to find underscore and any alphanumeric character.
\W	Used to find any non-word character.
\s	Used to find whitespace characters such as space, tab,carriage return , new line etc.
\S	Used to find any non-whitespace character.
\d	Used to find any decimal character.
\D	Used to find any non-decimal character.

- Quantifiers could be used to states how many number of times a character could appear in string.

Quantifier	Description
*	It is used to show Zero or more matches.
+	It is used to show One or more matches.
?	It is used to show Zero or one matches.
{N}	It is used show N number of matches.
{N,}	It is used to show N or more number of matches.
{N,M}	It is used to show number of matches Between N and M.

Syntax

```
<asp:RegularExpressionValidator ID="string"
runat="server" ErrorMessage="string"
ValidationExpression="string" ValidationGroup="string">
</asp:RegularExpressionValidator>
```

Example :

1) Validation of Alphanumeric character with special Characters using RegularExpression

Minimum length is 7 charcters and Maximum length is 10 charcters allowed for input.

Characters allowed in input are a – z, A – Z, 0-9, @, &, #.

```
<asp:RegularExpressionValidator ID="RegularExpression1" runat="server"
ErrorMessage=" length of Password must be in range 7 to 10 charcters"
ControlToValidate=" Pass "
ValidationExpression=" ^ [a-zA-Z0-9@&#]{7,10}$" />
```

2) Validation of Alphanumeric character using RegularExpression

Minimum length is 7 charcters and Maximum length is 10 charcters allowed for input.

Characters allowed in input are a – z, A – Z, 0-9.

```
<asp:RegularExpressionValidator ID="RegExpression2"
runat="server"
ErrorMessage=" length of Password must be in range 7 to 10 charcters "
ControlToValidate=" Pwd1 "
ValidationExpression=" ^ [a-zA-Z0-9]{7,10}$" />
```


3) Validation of Alphabates using RegularExpression

Minimum length is 7 charcters and Maximum length is 10 charcters allowed for input.

Characters allowed in input are a – z, A – Z.

```
<asp:RegularExpressionValidator ID="RegExpression3"
runat="server"
ErrorMessage=" length of Password must be in range 7 to 10 charcters "
ControlToValidate="Pwd3"
ValidationExpression=" ^ [a-zA-Z]{7,10}$" />
```

4) Validation of numeric data using RegularExpression

Minimum length is 7 charcters and Maximum length is 10 charcters allowed for input.

Characters allowed in input are 0-9

```
<asp:RegularExpressionValidator ID="RegExpression4"
runat="server"
ErrorMessage=" length of Password must be in range 7 to 10 charcters "
ControlToValidate="Pwd4"
ValidationExpression=" ^ [0-9]{7,10}$" />
```

Program 2.3.7

Write a program to validate Email id entered by user using RegularExpression Validator.

Solution :

Program to validate Email id entered by user using RegularExpression Validator.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm5.aspx.cs"
Inherits="WebApplication7.WebForm5" %>
<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

Email: (Required and Valid Email Address)<br />
<asp:TextBox ID="txtEmail" runat="server"
Width="293px"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator2" CssClass="Validators"
Display="Dynamic" ControlToValidate="txtEmail"
runat="server"
ErrorMessage="Email is
required."></asp:RequiredFieldValidator>
<asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Invalid Email Address"
ControlToValidate="txtEmail" CssClass="Validators"
Display="Dynamic"
ValidationExpression="\w+([-+.']\w+)*@\w
+([-.]\w+)*\.\w+([-.]\w+)*">
</asp:RegularExpressionValidator>
<br />
<asp:Button ID="Button1" runat="server" Text
="Submit" />

</div>
</form>
</body>
</html>
```

Check Email id entered by user using ValidationExpression Property of RegularExpression validator

Output

Syllabus Topic : Rich Controls

2.4 Rich Controls

Q. List all rich controls and explain any one rich control in detail. **(4 Marks)**

- Rich controls provide object model that has complex HTML representation and also client side JavaScript.
- They provide object model that is separate from underlying HTML representation.
- Typical rich control is often programmed as single object but renders itself with complex sequence of HTML elements and may even use client-side javascript.
- These controls implement special features.

Some examples of rich controls

i) **AdRotator control** is used to display randomly selected advertisement manner, on web page.

ii) **FileUpload** is used to upload any file.

iii) **Calendar** control is used to display months, days, year.

Syntax

```
<asp:controlTypeID="ControlID"runat="server"Property1
=value1 [Property2=value2] />
```

Syllabus Topic : Calendar Control

2.4.1 Calendar Control

Q. Explain calender control in detail with example. (4 Marks)

- ASP.NET privides a Calendar control that is used to display a calendar on our Web page.
- This control shows a single month calendar that permits the user to select date and move to the next and previous months.
- The user can move between months and select date and even select range of days when multiple selections is turned on.
- Calendar control has number of properties, using which we can customize each part of this control. For example it allow to set foreground and background colors,font size ,date format and so on.
- Calendar control provide events that enable reacting such as when the user changes current month – VisibleMonthChanged event is occurred ,when user select date -*selection changed* event occurs and when calendar is about to render a day then *DayRender* event occurs
- The Calendar control is represented as :

```
<asp:Calendar ID="Calendar1" runat="server"
</asp:Calendar>
```

- By default, this control displays the name of the current month, day headings for the days of the weeks, days of the month and arrow characters for navigation to the previous or next month.

Calendar control provides following features

1. Postback occurs due to user interaction with calendar each time. This allows to give reaction to selection event immediately and re-render its interface,thereby showing new month or newly selected dates.
2. Calendar control does not use Autopostback property.
3. Look and feel of calendar control can be changed according to programmer.
4. Information from database can be showed in calendar.

Properties and Events of the Calendar Control

The calendar control has many properties and events through which we can customize look and feel of calender control.

Properties	Description
Caption	This property is used to get and set the title for the calendar control.
CaptionAlign	This property is used to get and set the alignment for the caption.
WeekendDayStyle	This property is used to get the style for the weekend dates on the Calendar control.
VisibleDate	This property is used to get and set date which is displayed to user.
CellPadding	This property is used to get and set the number of spaces between the data and the cell border.
CellSpacing	This property is used to get and set the the space between cells.
DayNameFormat	This property is used to get and set the format for day of the week.
FirstDayOfWeek	This property is used to get and set the day of week to display in the first column.
TodaysDate	This property is used to get and set the value of today's date.
NextPrevFormat	This property is used to get and set the format of the next and previous month.
OtherMonthDayStyle	This property is used to get the style for the days on the Calendar control which are not selected by user.
TitleFormat	This property is used to get and set the format for the title section.
SelectWeekText	This property is used to get and set the text showed for the week selection element in the selector column.
SelectedDate	This property is used to get the selected date by user.
ShowTitle	This property is used to get and set a value showing whether the title section is displayed or not.

Properties	Description
SelectedDates	This property is used to get a collection of DateTime objects representing the selected dates.
SelectionMode	This property is used to get and set selection mode that state whether the user can choice a single day, a week or an entire month.
SelectMonthText	This property is used to get and set the text for the selected month in the selector column.
TodayDayStyle	This property is used to get the style properties for today's date on the Calendar control.
SelectorStyle	Gets the style properties for the week and month selector column.

- The Calendar control has the following three important events that allow the developers to create the calendar control. They are :

Events	Description
SelectionChanged	This event is raised when day, week or an entire month is selected.
VisibleMonthChanged	This event is raised when user changes a month.
DayRender	This event is raised when each data cell of the calendar control is rendered.

Program 2.4.1

Write a program using calander control to display today's date and birthdate of user.

Solution :

Program using calander control to display today's date and birthdate of user.

WebForm8.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm8.aspx.cs"
Inherits="WebApplication7.WebForm8" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">

<div>
<h3> Your Birthday:</h3>
<asp:Calendar ID="Calendar1" runat="server"
SelectionMode="DayWeekMonth"
onselectionchanged="Calendar1_SelectionChanged">
</asp:Calendar>
</div>
<p>Todays date is:
<asp:Label ID="labeltday"
runat="server"></asp:Label>
</p>
<p>Your Birday is:
<asp:Label ID="labelbday"
runat="server"></asp:Label>
</p>
</form>
</body>
</html>
```

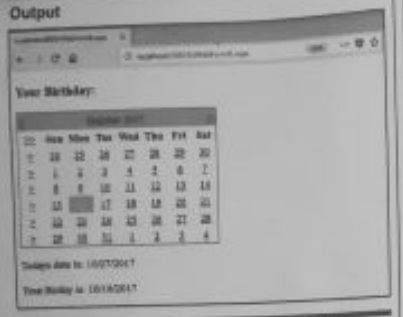
Show the calendar control from which user select his birthdate

webForm8.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication7
{
public partial class WebForm8 : System.Web.UI.Page
  {
protected void Page_Load(object sender, EventArgs e)
    {
    }
protected void Calendar1_SelectionChanged
(object sender, EventArgs e)
    {
        labelday.Text =
Calendar1.TodaysDate.ToShortDateString();
        labelbday.Text =
Calendar1.SelectedDate.ToShortDateString();
    }
  }
}
```

When user select his birthdate date, this method is called to show the today's date and birth date of user.

Output



Syllabus Topic : AdRotator Control

2.4.2 AdRotator Control

Q. Explain AdRotator control in detail. (2 Marks)

- AdRotator is a control in ASP.Net which is concerned with advertisements.
- It basically displays a sequence of advertisement images.
- This control uses an XML file to store the information of advertisement.
- The XML file must begin and end with an <Advertisements> tag.
- Inside the <Advertisements> tag there may be several <Ad> tags.
- XML file maintains list of advertisements and their associated attributes.
- This attributes include path of image to display, URL to link when control is clicked and alternate text to display when unable to display image, keyword and frequency of advertisement.

Program 2.4.2

Write a program in ASP.NET using AdRotator control to display advertisements from XML file.

Solution :

Program in ASP.NET using AdRotator control to display advertisements from XML file

XmlFile.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
 <Ad>
  <ImageUrl>~/Img/1.jpg</ImageUrl>

  <NavigateUrl>http://www.google.com</NavigateUrl>

    <AlternateText>
     AdRotator Control Demo
    </AlternateText>

    <Impressions>20</Impressions>
    <Keyword>Asp.Net</Keyword>

 </Ad>

</Advertisements>
```

Default3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default3.aspx.cs" Inherits="Default3" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>

    <asp:AdRotator ID="AdRotator1" runat="server"
AdvertisementFile ="~/XMLFile.xml" />
  </div>
</form>

</body>
</html>
```


Output

After clicking on image

Syllabus Topic : MultiView Control

2.4.3 MultiView Control

Q. Write short note on Multiview control. (4 Marks)

- MultiView and View controls permit us to separate the content of a page into various groups and show only one group at a time.
- Every View control is used to group controls of one group and all the View controls are grouped into a MultiView control.
- The MultiView control has task of showing one View control at a time.
- The View which is used to display the group of contents is called the active view.

Syntax of multiview control

```
<asp:MultiView ID= "MultiView1" runat= "server">
</asp:MultiView>
```

Syntax of view control

```
<asp:View ID= "View1" runat= "server">
</asp:View>
```

- We can not use View control only. It would show error if we do not use it with multiview.
- It is always used with a Multiview control as :

```
<asp:MultiView ID= "MultiView1" runat= "server">
<asp:View ID= "View1" runat= "server"></asp:View>
</asp:MultiView>
```

Properties of View and MultiView Controls

1. Control class is parent class for Both View and MultiView control classes and these classes derive all its properties, methods, and events.
2. The vital property of the View control is Visible property whose data type is Boolean i.e. true or false, which sets the visibility of a view.
3. The MultiView control has the following important properties :

Properties	Description
Views	View is Group of View controls in the MultiView control.
ActiveViewIndex	ActiveViewIndex is zero based index which is used to represent the active view. Value of index is -1 if view is not active.

4. The navigation of the MultiView control is performed using CommandName attribute of the button web control.
5. For example, button which has "NextView" as value of CommandName attribute related with the navigation of the multiview, it automatically navigates to the next view when the button is clicked.

The methods of the multiview control are

Methods	Description
SetActiveview	This method is used to set the active view
GetActiveview	This method is used to retrieve the active view

- Each time when view is changed, the page is send back to the server and many events are occoured. Some events are as follows :

Events	Description
ActiveViewChanged	This event is occoured when a view is changed

Events	Description
Activate	This event is fired by the active view
Deactivate	This event is fired by the inactive view

Program 2.4.3

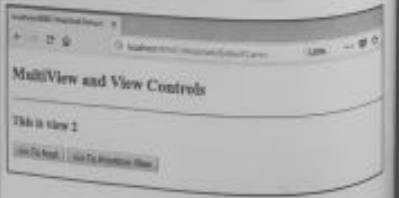
Write a program using multiview and view control to display three different views.

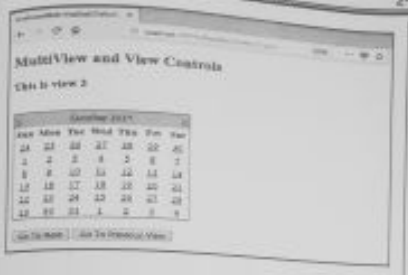
Solution :

Program using multiview and view control to display three different views.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<html>
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<h2>MultiView and View Controls</h2>

<br />

<asp:MultiView ID="MultiView1" runat="server"
ActiveViewIndex="2"
onactiveviewchanged="MultiView1_ActiveViewChanged">
<asp:View ID="View1" runat="server">
<h3>This is view 1</h3>
<br />
<asp:Button CommandName="NextView" ID="btnnext1"
runat="server"
        Text = "Go To Next"
onclick="btnnext1_Click" />
<asp:Button CommandArgument="View3"
CommandName="SwitchViewByID" ID="btnlast"
runat="server" Text ="Go To Last" />
</asp:View>

<asp:View ID="View2" runat="server">
<h3>This is view 2</h3>
<asp:Button CommandName="NextView" ID="btnnext2"
runat="server" Text = "Go To Next" />
<asp:Button CommandName="PrevView"
ID="btnprevious2" runat="server" Text = "Go To Previous
View" />
</asp:View>

<asp:View ID="View3" runat="server">
<h3>This is view 3</h3>
<br />
<asp:Calendar ID="Calender1" runat="server"
onselectionchanged="Calender1_SelectionChanged">
</asp:Calendar>
<br />
<asp:Button CommandArgument="0"
CommandName="SwitchViewByIndex" ID="btnfirst"
runat="server" Text = "Go To Next" />
<asp:Button CommandName="PrevView"
ID="btnprevious" runat="server"
        Text = "Go To Previous View" />
</asp:View>

</asp:MultiView>
</div>
</form>
</body>
</html>
```

Output





Syllabus Topic : Themes and Master Pages

2.5 Themes and Master Pages

Q. Explain themes and master pages in detail. **(4 Marks)**

Themes

- Themes are used to fix the look and feel of our website.
- It is a group of files. It can include skin files, CSS files and images.
- We can define themes in App_Themes folder. This folder contains one or more subfolders named Theme1, Theme2 etc. that define the actual themes.
- The theme property is applied to the life cycle of page to effectively overriding any customization for individual controls on our web page.

Master Pages

- Master page permits us to create a consistent look and feel for all the web pages in our web applications.
- Design of Master Page is common for all the pages.
- We will discuss Master Page in detail in section 2.5.4

Syllabus Topic : How Themes Work

2.5.1 How Themes Work ?

Q. How theme works? **(4 Marks)**

- There are 3 different options to apply themes to our website :

I. **Setting the theme at the page level :** The Theme attribute is added to the page directive of the page.

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2"
Theme="Theme1"%>
```

2. **Setting the theme at the web site level :** For setting the theme for the entire website, we can set the theme in the web.config file of the website. Open the web.config file and add theme attribute in <pages> element :

```
<pages theme="Theme1">
...
...
</pages>
```

3. **Setting the theme programmatically at runtime :** Here the theme is set at runtime through coding. It should be applied earlier in the page's life cycle i.e. PreInit event should be handled for setting the theme. The better option is to apply this to the Base page class of the site as every page in the site inherits from this class.

```
Page.Theme = "Theme1";
```

Uses of Themes

1. Themes can include CSS files, images and skins so we can change colors, fonts, positioning and images simply by applying the desired themes.
2. We can have many themes and we can change theme by setting a single attribute in the web.config file or an individual aspx page. Also we can change themes programmatically.
3. Themes allows us to provide the better usability(user friendliness) to our web site by giving users.

Syllabus Topic : Applying a Simple Theme

2.5.2 Applying a Simple Theme

Q. How to apply simple theme? **(4 Marks)**

Apply simple theme to application using following steps :

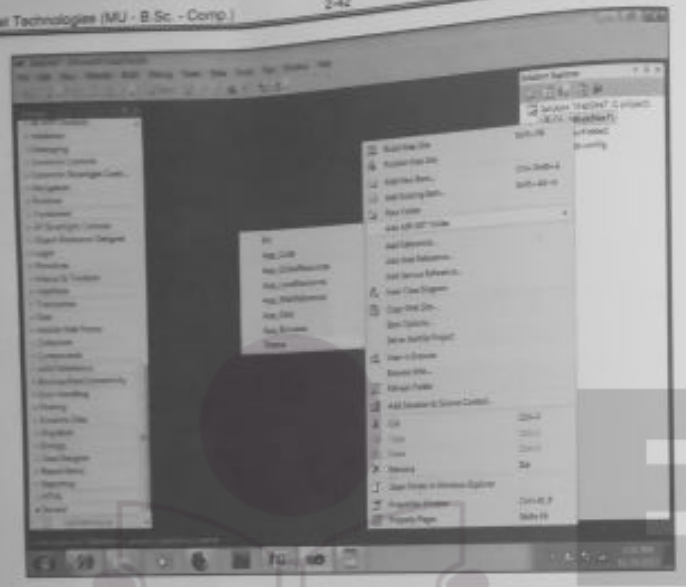
Step 1 : Open the Microsoft Visual Studio 2010.

Step 2 : Select File option,click on New and then select Web Site option. i.e File->New->Web site

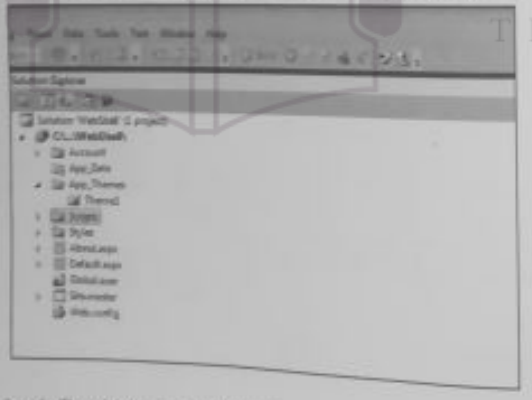
Step 3 : Select ASP.NET Web Site option and then click on Ok.

Step 4 : Then, Right click on the name of application in the Solution Explorer window and select Add ASP.NET Folder and then select submenu Theme from Add ASP.NET folder menu.

i.e. right click on name of application in Solution Explorer ->Add ASP.NET folder->Theme

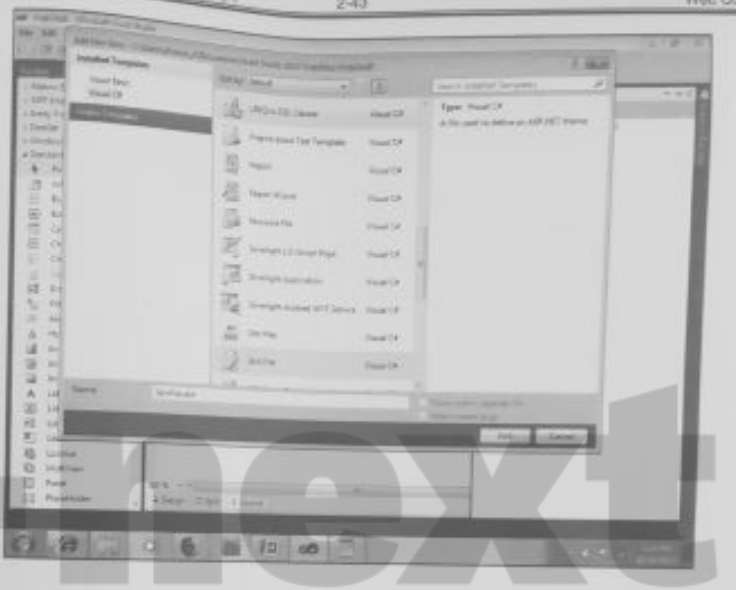


A sub folder named Theme1 is automatically created inside the APP_Themes folder in the Solution Explorer.

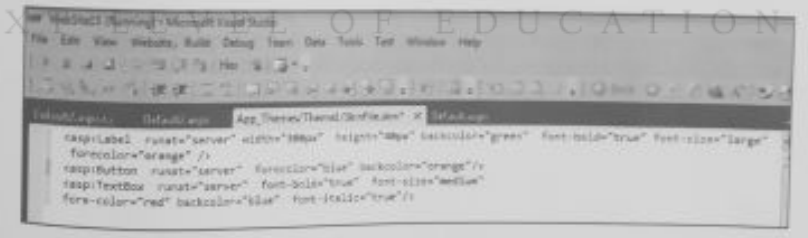


Step 5 : Right-click on the Theme1 folder and select the Add New Item option.i.e Theme1->Add New Item

Step 6 : Select the Skin file option and click on the Add button.



Now write the following code in the skin file.



```
<asp:Label runat="server" width="300px" height="50px" backcolor="green" font-bold="true" font-size="large"
forecolor="orange" />
<asp:Button runat="server" forecolor="blue" backcolor="orange"/>
<asp:TextBox runat="server" font-bold="true" font-size="medium"
fore-color="red" backcolor="blue" font-italic="true"/>
```

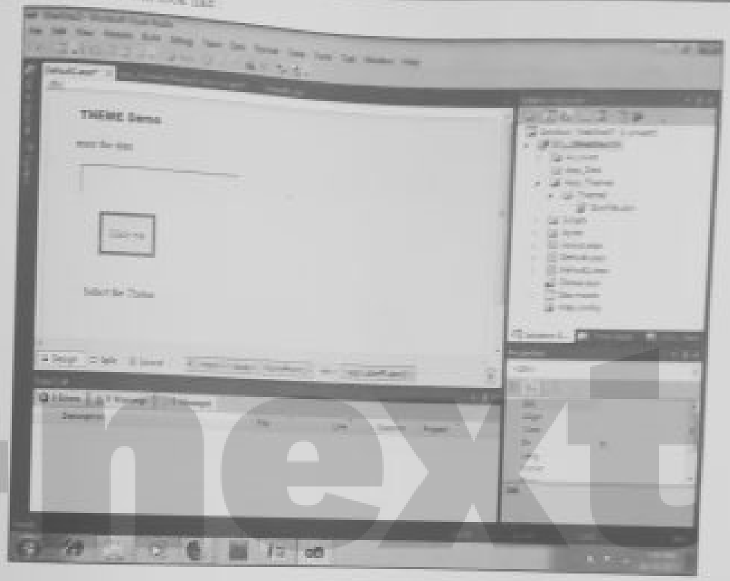
Step 7 : Write the following code for Default2.aspx page.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2"
Theme="Theme1" %>

<html>
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
```

Set a Theme to page as Theme1

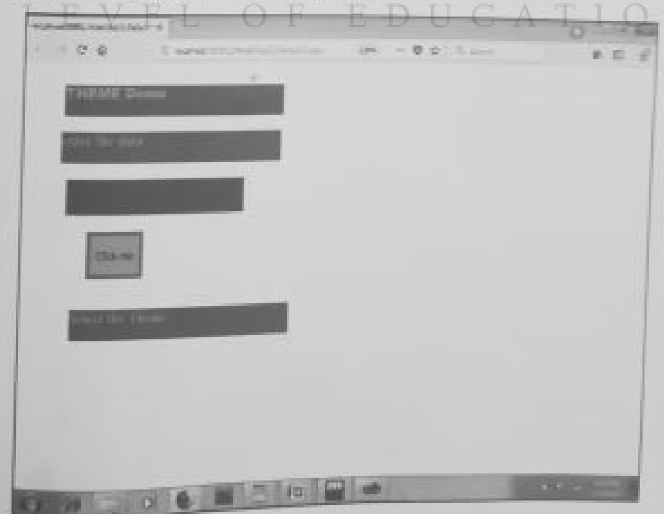
```
<div>     
<br />
           
<asp:Label ID="Label2" runat="server" Font-Bold="True"
        Font-Names="Arial black" Font-Underline="True"
        Text="THEME Demo"></asp:Label>
<br />
<br />
         
<asp:Label ID="Label1" runat="server" Text="enter the data">
</asp:Label>
  <br />
<br />
           
<asp:TextBox ID="TextBox1" runat="server" Height="[illegible]" Width="257px"></asp:TextBox>
<br />
<br />
            
     
<asp:Button ID="Button1" runat="server" Text="Click me"
        BorderColor="#990033" BorderStyle="Solid" BorderWidth="3px" Height="60px" />
<br />
<br />
<br />
              
<asp:Label ID="Label3" runat="server" Text="Select the Theme"></asp:Label>
<br />
<br />

<br />

<br />
<br />
<br />
</div>

</form>
</body>
</html>
```


The design window will look like



Step 8 : Now write the following code for the Default2.aspx.cs file.

Output

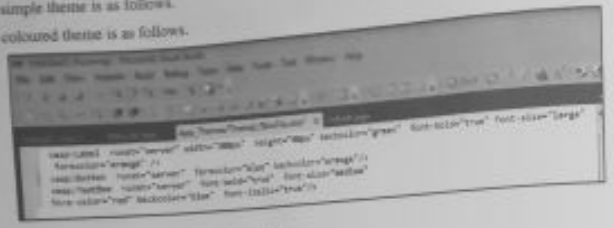


Now we will create an application for applying the theme at run time.

Step 9 : Press Ctrl+F5 keys to run the application.

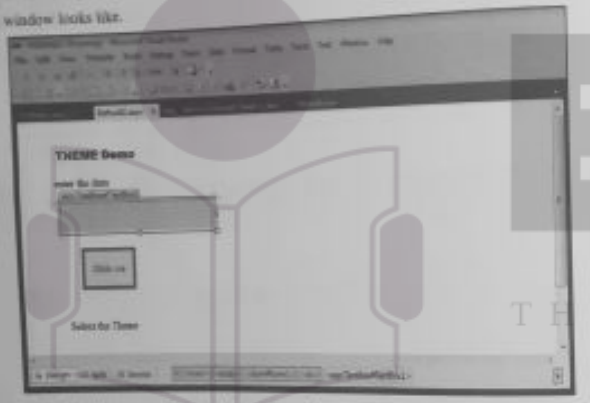
Code for simple theme is as follows.

Code for coloured theme is as follows.



Step 10 : Add the following code to the Default.aspx file.

The design window looks like.



Step 11 : Now write the following code for the default.aspx.cs file.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
    void Page_PreInit(object sender, EventArgs e)
    {
        // Get the theme name from a QueryString variable
        string ThemeName;
        ThemeName = Request.QueryString["Theme"];
        if (ThemeName != null)
```



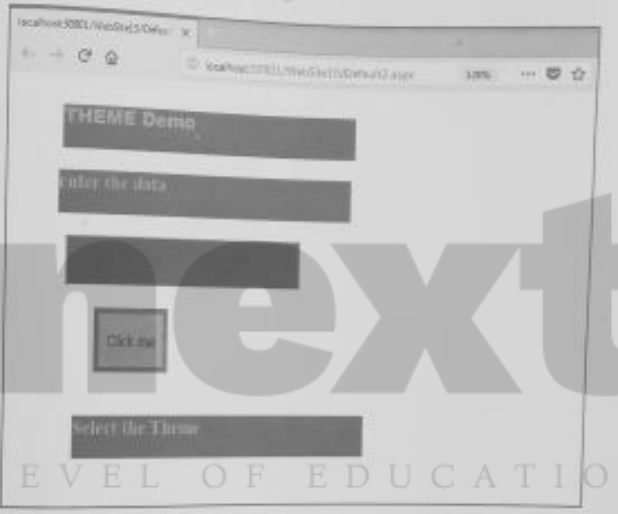
```
    {
        Page.Theme = ThemeName;
    }
}
```

Set theme to page

Step 12 : Press F5 keys to run the application.

Output

After selecting the Coloured theme the output as follows.



Syllabus Topic : Handling Theme Conflicts

2.5.3 Handling Theme Conflicts

Q. How to handle theme conflict? (2 Marks)

- When properties confuse between web controls and theme, properties can give priority to theme.For this purpose just use the StyleSheetTheme attribute at the place of the Theme attribute.

```
<%@ Page Language="C#" AutoEventWireup="true" ...
StyleSheetTheme="FunkyTheme" %>
```

Syllabus Topic : Simple Master Page and Content Page and Connecting Master Pages and Content Page

2.5.4 Simple Master Page and Content Page and Connecting Master pages and Content Page

Q. What is concept of master pages and content pages and how to connect them? (4 Marks)

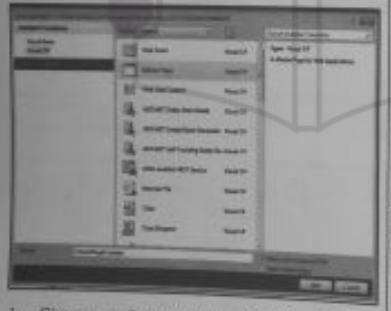
- Master page permits us to create a consistent look and feel for all the web pages in our web applications.
- Design of Master Page is common for all the pages.
- Master page contains of two parts, one is the master page itself and another is one or more content pages.

- A master page is an ASP.NET file with the extension .master.
- ContentPlaceholder control defines a region on the master page at which content pages can plug in the page specific content.
- The @Master directive is used to define a page as master page.
- The ContentPlaceHolder web control is available only for masterpages.
- Content pages are ASP.NET pages which uses master pages.
- Master page provide us a way to create common set of User Interface elements that are needed on number of pages in our website.
- ContentPage are ASP.NET web page that will use master page to have the common UI elements rendering.
- ContentPlaceHolder is a control that should be added on the MasterPage which will reserve the place for the content pages to run their contents.
- ContentControl is control which will be included on content pages to show these pages that the contents inside this control will be run where the ContentPlaceHolder of master page is located.

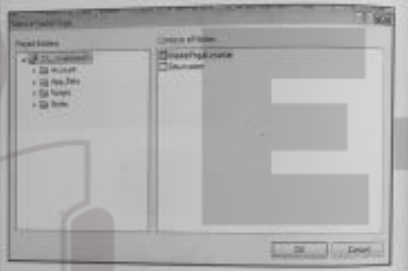
Following are the stpes to create Creating a MasterPage

Step 1 : Go to "Add New Item".

Step 2 : Select the MasterPage.



1. Give name to that master page as MasterPage1.Master.
2. Now add a menu bar on the master page at top of the page. This Menu bar will be common to all the pages because that menu bar is present in Master page.
3. Now we add some content pages like Home.aspx, default.aspx, about.aspx, Contact.aspx etc.
4. When we add these content pages in our web site, we should remember to select the option of "Use master Page" and select the master page.



- When we look at the MasterPage1, we will see that master page has a ContentPlaceHolder control. All the code that is common for the content pages is outside the ContentPlaceHolder control (in our case, a simple menubar).

Program 2.5.1

Write a program to create a master page which contain menu bar that include menus such as Home,About us,Gallary and so on and create some content pages under the master page.

Solution :

Program to create a master page which contain menu bar that include menus such as Home,About us,Gallary and so on and create some content pages under the master page.

MasterPage1.aspx(master page)

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage1.master.cs"
Inherits="MasterPage1" %>
```



```
<html>
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<form id="form1" runat="server">
<div align="center">
<h1>MyFirst WebSite</h1>
<asp:Menu ID="Menu1" runat="server"
BackColor="#B5C7DE" DynamicHorizontalOffset="2"
Font-Names="Verdana" Font-Size="0.8em"
ForeColor="#284E98" Orientation="Horizontal"
StaticSubMenuIndent="10px">
<StaticMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" />
<DynamicHoverStyle BackColor="#284E98"
ForeColor="White" />
<DynamicMenuStyle BackColor="#B5C7DE" />
<StaticSelectedStyle BackColor="#507CD1" />
<DynamicSelectedStyle BackColor="#507CD1" />
<DynamicMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" />
<Items>
<asp:MenuItem Text="HOME" Value="home"
NavigateUrl="~/Default2.aspx">
</asp:MenuItem>
<asp:MenuItem Text="ABOUT" Value="about"
NavigateUrl="~/about1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="CONTACTNO" Value="contact"
NavigateUrl="~/contact1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="GALLARY" Value="contact"
NavigateUrl="~/gallary.aspx">
</asp:MenuItem>
</Items>
<StaticHoverStyle BackColor="#284E98"
ForeColor="White" />
</asp:Menu>
```

Add menubar with Home,About,Contact No etc menus

```
<asp:contentplaceholder id="ContentPlaceHolder1"
runat="server">
</asp:contentplaceholder>

</div>

</form>
</body>
</html>
```

Program 2.5.2

Write a program to adding the ContentPages to our website.

Solution :

Program to adding the ContentPages to our website

Default2.aspx(content page)

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPage1.master"
AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="head" Runat="Server">
</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<h2>This is a the about page.</h2>
</asp:Content>
```

about1.aspx(content page)

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPage1.master"
AutoEventWireup="true" CodeFile="about1.aspx.cs"
Inherits="about1" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="head" Runat="Server">
</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<h2>This is a the Home page.</h2>
</asp:Content>
```

contact1.aspx(content page)

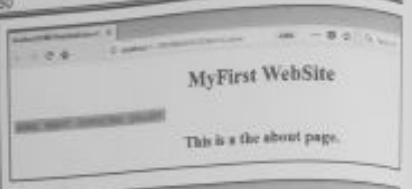
```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPage1.master"
AutoEventWireup="true" CodeFile="contact1.aspx.cs"
Inherits="contact1" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="head" Runat="Server">
</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<h2>This is a the  CONTACT  page.</h2>
</asp:Content>
```

gallary.aspx(content page)

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPage1.master"
AutoEventWireup="true" CodeFile="contact1.aspx.cs"
Inherits="contact1" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="head" Runat="Server">
</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<h2>This is a the  Gallary  page.</h2>
</asp:Content>
```

Once we have all the master pages and content pages ready, we can test our website.

Output



So the common set of controls and behavior has now been added to all the content pages. We can have any number of controls and functionality on the master pages.

Syllabus Topic : Master Page with Multiple Content Regions

2.5.5 Master Page with Multiple Content Regions

Q. Explain with example : Master page with multiple content regions. (4 Marks)

- Masterpage1.master is master page which contain two ContentPlaceHolders by default. It may contains more contentplaceholders .

Program 2.5.3

Write a program of Master Page with multiple contents

Solution :

Program of Master Page with multiple contents

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage1.master.cs"
Inherits="MasterPage1" %>

<html>
<head runat="server">
<title></title>
```

First ContentPlaceHolder


```
</asp:ContentPlaceHolder>
</head>
<body>
<form id="form1" runat="server">
<div align="center">
<h1>MyFirst  WebSite</h1>
<asp:ContentPlaceHolder id="body" runat="server">
<asp:Menu ID="Menu1" runat="server"
BackColor="#B5C7DE" DynamicHorizontalOffset="2"

    Font-Names="Verdana" Font-Size="10pt"

    ForeColor="#284E98" Orientation="Horizontal"

    StaticSubMenuIndent="10px">
<StaticMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" />
<DynamicHoverStyle BackColor="#284E98"
ForeColor="White" />
<DynamicMenuStyle BackColor="#B5C7DE" />
<StaticSelectedStyle BackColor="#507CD1" />
<DynamicSelectedStyle BackColor="#507CD1" />
<DynamicMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" />
<Items>
<asp:MenuItem Text="HOME" Value="home"
NavigateUrl="~/Default2.aspx">
</asp:MenuItem>
<asp:MenuItem Text="ABOUT" Value="about"
NavigateUrl="~/about1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="CONTACTNO" Value="contact"
NavigateUrl="~/contact1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="GALLARY" Value="contact"
NavigateUrl="~/gallary.aspx">
</asp:MenuItem>
</Items>
<StaticHoverStyle BackColor="#284E98"
ForeColor="White" />
</asp:Menu>
</asp:ContentPlaceHolder>
```

Second ContentPlaceHolder

```
<asp:contentplaceholder id="ContentPlaceHolder1"
runat="server">
</asp:contentplaceholder>

</div>
</form>
<asp:ContentPlaceHolder id="footer" runat="server">
</asp:ContentPlaceHolder>
</body>
</html>
```

Third ContentPlaceHolder

Syllabus Topic : Master Pages and Relative Paths

2.5.6 Master Pages and Relative Paths

Q. Explain master pages and relative paths in detail. (4 Marks)

- Relative path is used in master page to specify a link to our client javascript file or a Cascading Style Sheet (CSS) file or images.
- A path on a web page is called as relative path if the location of the resource it points to is relative to the web page's location in the website's folder structure.
- Any path that does not starts with forward slash (/) or a protocol such as http:// is relative because it is resolved by the browser based on the location of the web page that is written in that path.
- For example, our website has an ~/Images/ folder with ,Img1.gif. The master page file MasterPage1.master has an <img> element in the footerContent part with the following HTML code :

```
<div id="footerContent">
<img src="~/Images/Img1.gif"alt="Beautiful Image" />
</div>
```

- The ~/Admin/Default.aspx content page is sent to the HTML for the footerContentarea :

Html

```
<div id="footerContent">
<img src="img2.gif" alt="nic image" />
</div>
```

As the value of <img> element's src attribute is a relative path, the browser try to look for Images folder relative to the web page folder location. In other words, the browser is looking for the image file in Admin/Images/img.gif.

Replacing Relative URLs with Absolute path

- Absolute path is opposite of a relative path which is one that contains a forward slash (/) at start or a protocol such as http://.
- An absolute path state the place of a resource from fixed point.
- The same absolute path is valid in any other web page, regardless of the place of web page in the folder structure of website.
- ASP.NET offers a method for generating a valid relative path at runtime.

Using ~ and ResolveClient path

- In another way of generating absolute URL, ASP.NET permits coder to use the tilde (~) to state the root of the web application.
- For example, earlier we use path ~/Admin/Default.aspx in the text to refer to the Default.aspx page in the Admin folder.
- The tilde (~) state that the Admin folder is a subfolder of the root of web application.

Using ~ in the Declarative Markup

- Several ASP.NET Web controls include path -related properties such as the HyperLink control has a NavigateUrl property and the Image control has an ImageUrl property etc.
- When executed, these controls send their values of path -related properties to ResolveClientUrl. Consequently, if these properties contain a ~ to indicate the root of the web application, the path will be changed to make valid relative path.
- Remember that ASP.NET server controls can only work on tilde (~) in their path-related properties.
- ~ appears in static HTML code, such as

```
<img src="~/Images/Img2.gif" />
```

The ~ send by ASP.NET engine to the browser with the rest of the HTML content.

- The browser suppose that the ~ is part of the relative path.
- For example, if the browser receives the HTML code

```
<img src="~/Images/Img2.gif" />
```

It oppose that there is a subfolder named ~ with a subfolder Images that include the image file Img2.gif

- To fix the image code in MasterPage1.master, replace the existing <img>tag with an ASP.NET Image Web control.
- Set the ID of Image web control as Img2 and value of its ImageUrl property to ~/Images/Img2.gif, and its AlternateText property to "Beautiful Image!"

aspx

```
<div id="Contentatfooter">
<asp:Image ID="Img2" runat="server"
ImageUrl="~/Images/Img2.gif"
AlternateText="Beautiful Image!" />
</div>
```

- After making this change to the master page, now again open the~/Admin/Default.aspx web page again.
- This time the Img2.gif image file appears in the page.
- When the Image Web control is run, it uses the ResolveClientUrl method to resolve its ImageUrl property value.
- In ~/Admin/Default.aspx the ImageUrl is converted into an appropriate relative path, as the following snippet of HTML source shows :

html

```
<div id="footerContent">
<img
id="Img2" src="../Images/Img2.gif" alt="Image
by ASP.NET!" />
</div>
```

Syllabus Topic : Website Navigation

2.6 Website Navigation

- ASP.NET provides various site-navigation features which gives a consistent way for visitors to navigate the site comfortably.
- These features are : Site Maps, URL Mapping and Routing, SiteMapPath.

Syllabus Topic : Site Maps

2.6.1 Site Maps

Q. How to create sitemap? (4 Marks)

- To use ASP.NET site navigation, we must state the design of the site so that the site navigation API and the site navigation controls can interpret the design of site correctly.
- The site navigation system use an XML file that contains the site navigation hierarchy by default.

- We can also made setting in the site navigation system to use another data sources.
- Google, Yahoo and Microsoft create standard sitemap which provide an easy way to owners of web site to send information to search engines about pages on their site that are available for crawling.
- Web crawlers find pages from links within the site and from other sites.
- Sitemap attach this data to permits web crawlers that helps the sitemaps to pick up all URLs in the sitemap and learn about those URLs using the associated metadata(data about data).

The Web.sitemap File

- The easy way to generate a site map file is to create an XML file with name Web.sitemap that arrange the pages in the sitemap.
- The site map is provided automatically by the default site-map provider for ASP.NET.
- The Web.sitemap file must be stored at application root directory.
- The below code state how the site map might look for a simple site. The url attribute contain value that can start with the "~/" which indicates the application root directory.

```
<siteMap>
<siteMapNode title="Home" description="Home"
url="~/home.aspx">
<siteMapNode title="Products" description="products"
url="~/Products.aspx">
<siteMapNode title="Hardware" description="Hardware
choices"
url="~/Hardware.aspx" />
<siteMapNode title="Software" description="Software "
url="~/Software.aspx" />
</siteMapNode>
<siteMapNode title="Hardware" description="Hardware
that we can provide"
url="~/Hardware.aspx">
<siteMapNode title=" Personal Training"
description="Training classes"
url="~/Training/classes.aspx" />
<siteMapNode title="Carrier Consulting" description=
" Carrier Consulting services"
url="~/CarrierConsulting.aspx" />
<siteMapNode title="ProductGallary" description=
" ProductGallary "
url="~/ ProductGallary.aspx" />
<siteMapNode title=" Feedback" description="Feedback"
url="~/Feedback.aspx" />
</siteMapNode>
</siteMapNode>
</siteMap>
```

- We add a siteMapNode element in Web.sitemap file for each page in our Web site.
- Then hierarchy of nested siteMapNode elements is generated.
- In the above example, the pages of Hardware and Software are child elements of the Products element.
- The title attribute provide the text that is generally used as linked text.
- The description attribute is used to represent both documentation and tool tip in the SiteMapPath control.

Valid Site Maps

- A valid site-map file include only one siteMapNode element that is placed immediately under the sitemap element.
- Any number of child siteMapNode elements can be included in the first-level siteMapNode element.
- A valid sitemap file should not contains duplicate URLs.
- This restriction may be only for default site-map provider for ASP.NET.

Configuring Multiple Site Maps

- We can use many site-map files or many site-map providers to state the navigation structure of our whole Web site.
- For example, our root Web.sitemap file contains link for its child site-map file by using reference to it's child site-map file in a siteMapNode element by using the below code :

```
<siteMapNode siteMapFile="FirstSiteMap.sitemap" />
```

Syllabus Topic : URL Mapping and Routing

2.6.2 URL Mapping and Routing

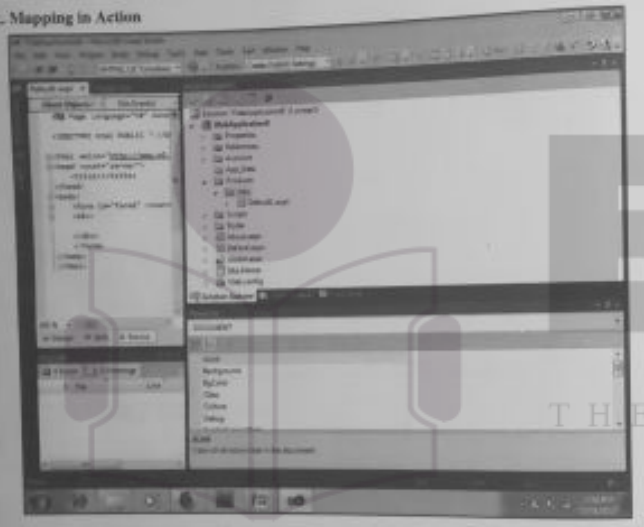
2.6.2(a) URL Mapping

Q. What is use of tilde (~) in URL mapping? (2 Marks)

- Ideally we create our Web application accurately in our first attempt. Pages should be created in the proper folder and stay there.

- For that matter, end users would not care about the URLs of the pages in our Web applications, so we could put pages anywhere without worrying about it.
- In the real world, things are not that much simple.
- We may decide after released an application for which we actually want to identify the folder structure.
- Users may like to see URLs like http://www.PhoenixGlobe.com/Default1.aspx instead of http://www.PhoenixGlobe.com/Products/new/ Default1.aspx.
- The urlMappings element resend URLs to new place. This element is part of an ASP.NET web.config file. It is simply included to Web.config to perform this complex task.

URL Mapping in Action



- By default, the Default1 .aspx page is accessible at ~/Products/new/Default1.aspx where the tilde character (~) is used to indicate root of the our Web site.
- We can change~/Products/new/Default1.aspx to ~/Products/Default1.aspx by making an entry in the web.config file of our web application.
- To do this, we require to include the <urlMapping> section inside of the <system.web> section. Here's an example to do the mapping that we want :

```
<system.web>
  <urlMappings enabled="true">
    <add url="~/Default.aspx" mappedUrl="~/Products/new/Default1.aspx"/>
  </urlMappings>
</system.web>
```

- Inside of the <urlMappings> element we can add many elements using <add> tag.
- Each of these <add> include a url attribute which is used to state the URL that the user will enter, and a mappedUrl attribute, which state the actual URL within the application to deliver to the entered URL.

2.6.2(b) URL Routing

Q. What is URL routing? (2 Marks)

- URL Routing is new feature in ASP.Net 4.0.
- ASP.NET routing permits us to use URL due to which we does not need to map to specific files in a Web site.
- So we can use URLs that describe action of user.
- The ASP.NET MVC framework and ASP.NET Dynamic Data use routing to provide features that are used only in MVC applications and in Dynamic Data applications.
- For example, a request for http://server/application/Products.aspx?id=4 maps to a file that is named Products.aspx that include code and markup for rendering a response to the browser.
- The Web page uses the query string which use id=4 to determine what type of data to show.
- In ASP.NET routing, we can store URL patterns that map to request-handler files.
- ASP.NET routing do not require adding names of those files in the URL.
- In addition, we can add placeholders in a URL pattern so that variable data can be passed to the request handler without using a query string.
- ASP.Net 4.0 URL Routing enables mapping between search engine's optimized URL to physical Web Form pages.

Syllabus Topic : SiteMapPath Control

2.6.3 SiteMapPath Control

Q. Explain siteMapPath control in detail. (4 Marks)

- To access webpage from another webpage SiteMapPath web control is used.
- This control is used mainly for navigation and shows the map of the site related to its web pages.
- This map contains the pages of the particular website and shows the name of those pages.
- We can click on that particular page in the Site Map to go to that page.
- SiteMapPath control is used to show links for connecting to URLs of other pages.
- The SiteMapPath control uses SiteMapProvider property which is useful for accessing data from databases and it stores the information in a data source.
- The SiteMapPath web control find out navigation data from a site map.
- This data contains information about the pages in our website like the URL, title, description, and place in the navigation hierarchy.
- The presentation of the SiteMapPath control is as follows :

Root Node->Child Node

Public Properties of SiteMapPath class

1. **ParentLevelsDisplayed** : Number of levels of parent nodes are shown by this property.
2. **RenderCurrentNodeAsLink** : This property is used to state whether or not the site navigation node that represents the current page is running as a hyperlink.
3. **PathSeperator** : This property states the string that shows the SiteMapPath nodes in the rendered navigation path.

Style properties of the SiteMapPath class

1. **CurrentNodeStyle** : This property is used to specify the style which is used to show the text for the current node.
2. **RootNodeStyle** : This property determine the style which is for the root node style text.
3. **NodeStyle** : This property determines the style which is used to display text for all nodes in the site navigation path.

- Now we can use the SiteMapPath control using following steps :

Step 1 : Open Microsoft Visual Studio 2010.

Step 2 : Select File->New->Web Site.

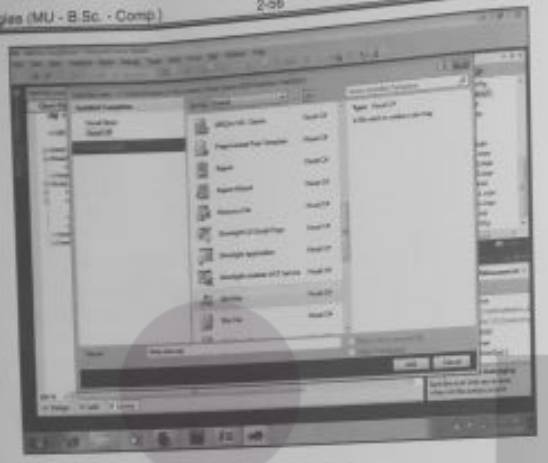
Step 3 : Select ASP.NET Web Site and name it.

Step 4 : Add two web forms to the application named MyWeb1.aspx and AddWeb2.aspx by performing the following steps.

Step 5 : In same way add the AddWeb2.aspx web form to the application. After that we have to add the Site Map file into the project. The Site Map file is the XML file and has the extension .sitemap. The steps are as follows.

Step 6 : Right-click the application in the Solution Explorer window and then click the Add New Item option from the context menu.

Step 7 : Select the Site Map Template from the Templates Pane. Note that, by default, the file has the name web.sitemap.

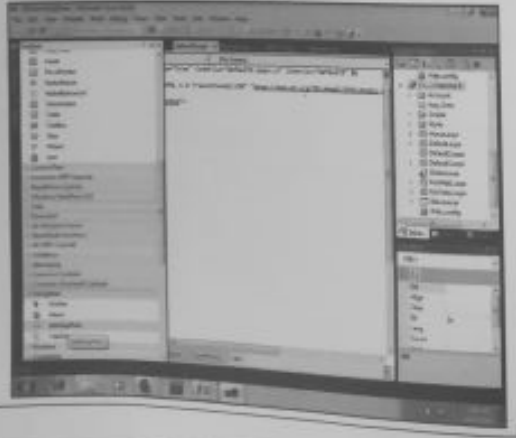


Step 8 : Now click the Add button to add the sitemap file to our application.

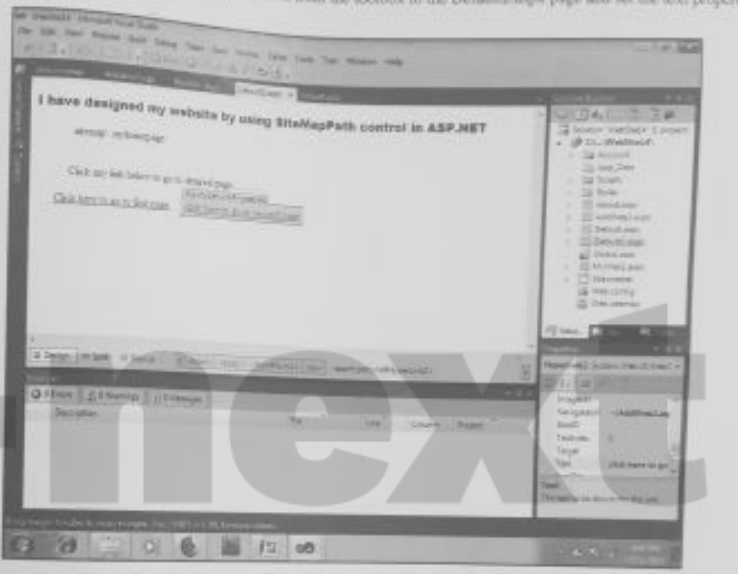
Step 9 : Now we can check that the sitemap file has been included in our project and we can see it in the Solution Explorer. And now we have to set the URL and title attributes in the sitemap file.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
    <siteMapNode url="Default2.aspx" title="mythomepage"  description="" >
      <siteMapNode url="Mywebl.aspx" title="myfirstpage"  description="" />
      <siteMapNode url="AddWeb2.aspx" title="mysecondpage"  description="" />
    </siteMapNode>
</siteMap>
```

Step 10 : Now Add one SiteMapPath control on the Default2.aspx page from the navigation tab of the toolbox.



Step 11 : Add three labels and two hyperlink controls from the toolbox to the Default2.aspx page and set the text property of the control.



Step 12 : The source code of the Default2.aspx page is as follows :

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<html>
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <asp:Label ID="Label1" runat="server"
    Text="I have designed my website by using
SiteMapPath control in ASP.NET"
    Font-Bold="True" Font-Names="Arial Black" Font-
Size="Large"></asp:Label>
  <br />
  <br />
          
    
  <asp:Label ID="Label2" runat="server"
Text="sitemap:"></asp:Label>
     
  <asp:SiteMapPath ID="SiteMapPath1"
runat="server">
  </asp:SiteMapPath>
  <br />
  <br />
  <br />

         
  
  <asp:Label ID="Label3" runat="server"
    Text="Click any link below  to go to desired
page.....">
  <br />
  <br />
```

```
             
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/MyWeb1.aspx">Click here to go to first
page< asp:HyperLink>
            
    <asp:HyperLink ID="HyperLink2" runat="server"
NavigateUrl="~/AddWeb2.aspx"><click here to go to
second page</asp:HyperLink>
    </div>
    </form>
</body>
</html>
```

Step 13 : Now in the design mode of the MyWeb1.aspx page, add the same control but add only one hyperlink control and set its NavigateUrl property to AddWeb2.aspx then click ok.

The source code of the MyWeb1.aspx page is as follows :

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="MyWeb1.aspx.cs" Inherits="MyWeb1" %>

<html>
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
           
        <asp:Label ID="Label1" runat="server" Font-
Bold="True" Font-Names="Arial Black"
            Font-Size="Large"
            Text="Thank you for clicking.This is My First
Webpage.....">
        </asp:Label>
        <br />
        <br />

         
         
     
                 
        <asp:Label ID="Label2" runat="server"
Text="SiteMap:">
        </asp:Label>
               
        <asp:SiteMapPath ID="SiteMapPath1"
runat="server">
        </asp:SiteMapPath>
        <br />
        <br />

         
         
    
        <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/AddWeb2.aspx">click here to go to
mywebpage2</asp:HyperLink>
    </div>
    </form>
</body>
</html>
```

Step 14 : Now in the design mode of the AddWeb2.aspx page, add the same controls as in myweb1.aspx and set the NavigateUrl property of the hyperlink as Default2.aspx then click ok.

The source code of AddWeb2.aspx is as follows :

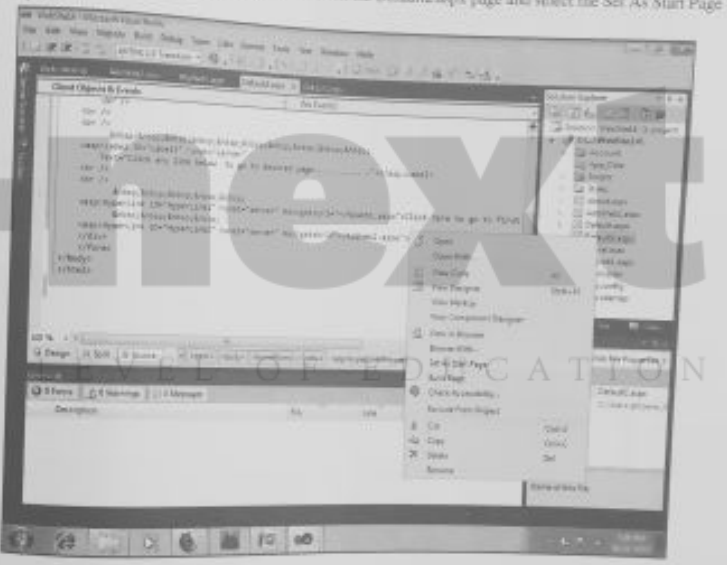
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="AddWeb2.aspx.cs" Inherits="AddWeb2" %>

<html>
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label ID="Label1" runat="server" Font-
Bold="True" Font-Names="Arial Black"
            Font-Size="Large"
            Text="Thanx for Clicking.This is my Second
webpage by using SiteMapPath control.....">
        </asp:Label>
    </div>

<p>        &
nbsp;     
         
        <asp:Label ID="Label2" runat="server"
Text="SiteMap:"></asp:Label>
            
        <asp:SiteMapPath ID="SiteMapPath1"
runat="server">
```



```
        </asp:SiteMapPath>
    </p>
    <p> </p>

<p>        &
nbsp;     
         
         
     
```

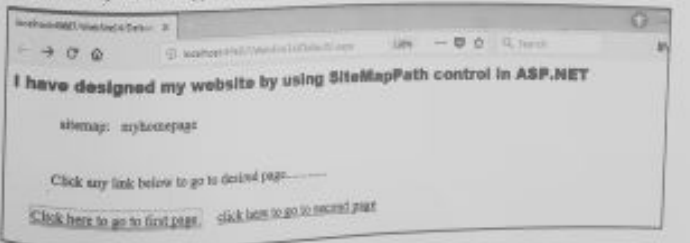
```
           
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Default2.aspx">Click here to go to Home
page</asp:HyperLink> 
    </p>
    </form>
</body>
</html>
```

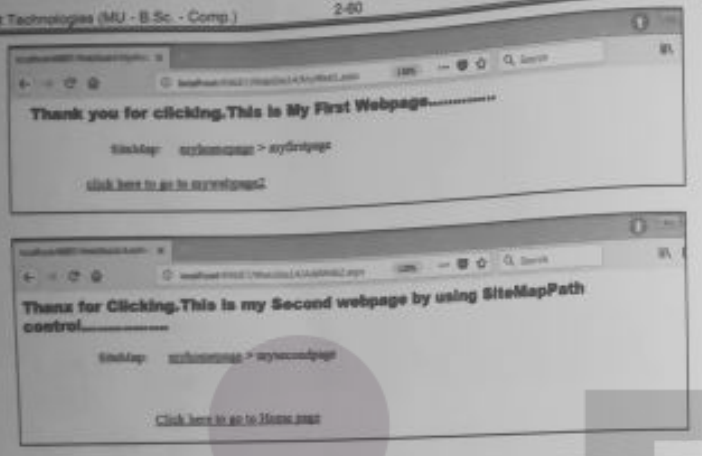
Step 15 : Now in the Solution Explorer window, right click on the Default2.aspx page and select the Set As Start Page option.



Step 16 : Press the F5 key to run the application.

Output





Syllabus Topic : TreeView Control

2.6.4 TreeView Control

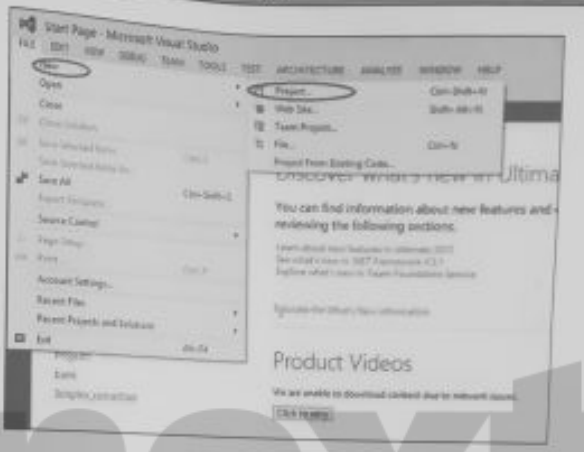
Q. How to create TreeView ? **(4 Marks)**

- The TreeView Web control is used to show the hierarchical data in a tree structure.
- A TreeView is a group of TreeNode objects.
- The contents of the TreeView control can be stated directly in the control :
  1. XML file
  2. Web.sitemap file
  3. Database table
- Following is Constructure of TreeViewControl :

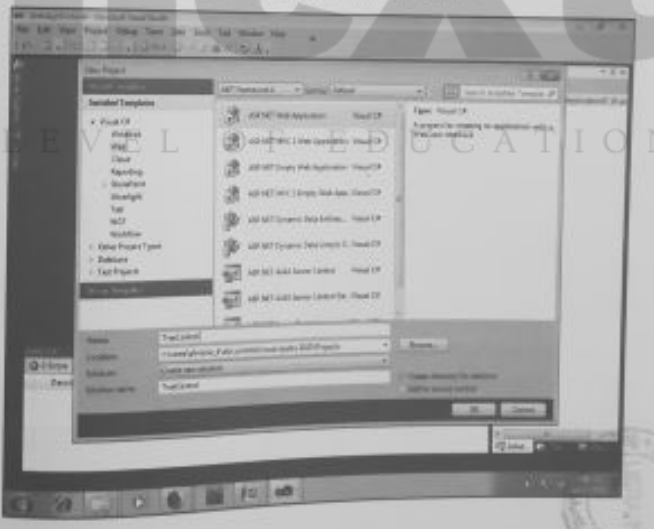
  Treeview() which is used to initialize new object of Treeview class.

☞ **Steps to create a TreeView Control in ASP.NET**

Step 1 : Open Microsoft Visual Studio then select "File" -> "New" -> "Project..."

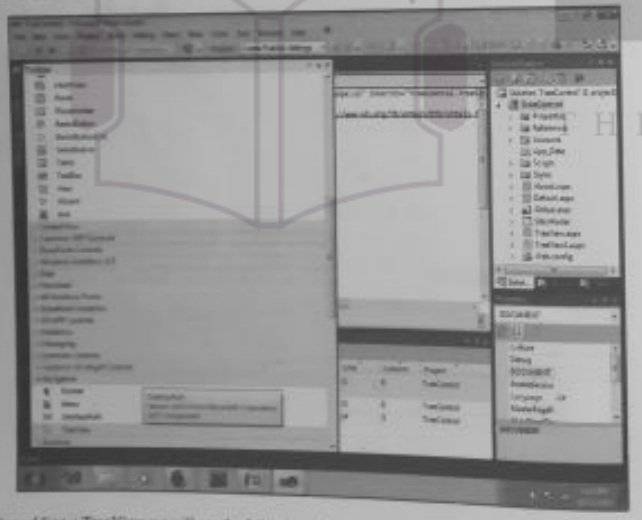


Step 2 : Add an ASP.NET Web Forms Application and give it the name TreeControl.



Step 3 : Open Solution Explorer then add a Webform and give it a name such as "TreeView1.aspx".

Step 4 : Now drag and drop a TreeViewControl from the ToolBox.



And after adding a TreeView we will see the following code :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TreeView1.aspx.cs"
Inherits="TreeControl.TreeView1" %>
```



```
<html>
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <d<asp:TreeView ID="TreeView1" runat="server"></asp:TreeView>
  js>
  </div>
  </form>
</body>
</html>
```

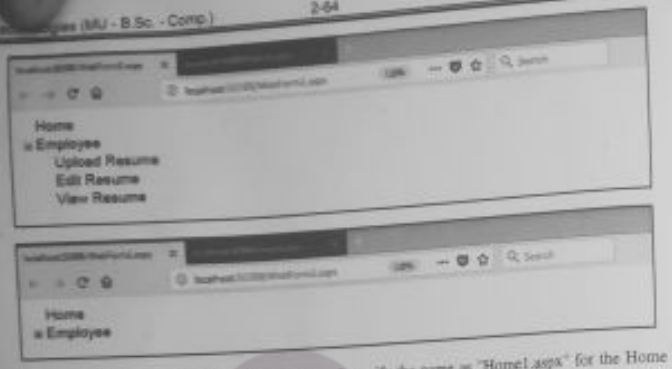
Create Tree view

Step 5 : Write the following code inside the <Nodes></Nodes> element and create Nodes of the TreeView in the "TreeView1.aspx" page as in the following.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="TreeControl.WebForm2" %>
<html>
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div style="font-family: Arial">
      <asp:TreeView runat="server" ID="TreeView1">
        <Nodes>
          <asp:TreeNode Text="Home" NavigateUrl="~/Home1.aspx" Target="_blank" />
          <asp:TreeNode Text="Employee" NavigateUrl="~/Emp.aspx" Target="_blank">
            <asp:TreeNode Text="Upload Resume" NavigateUrl="~/Upload_Resume1.aspx" Target="_blank" />
            <asp:TreeNode Text="Edit Resume" NavigateUrl="~/Edit_Resume1.aspx" Target="_blank" />
            <asp:TreeNode Text="View Resume" NavigateUrl="~/ViewResume1.aspx" Target="_blank" />
          </asp:TreeNode>

        </Nodes>
      </asp:TreeView>
    </div>
  </form>
</body>
</html>
```

Create tree node which content hyperlink to different pages like Home1.aspx etc.

Output



Now before going further we add one new webform and specify the name as "Home1.aspx" for the Home link of the TreeView.

Home1.aspx.cs

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Home1.aspx.cs" Inherits="TreeControl.Home1" %>

<html>
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
      <asp:Label ID="HomePage" runat="server" Text="Welcome To Home Page" BackColor="Yellow"
ForeColor="#000000"></asp:Label>
  </div>
  </form>
</body>
</html>
```

Output

Syllabus Topic : Menu Control

2.6.5 Menu Control

Q. Write note on menu control. (4 Marks)

- The Menu control permits the multiple display options adding a static display.
- It also provides dynamic display where part of the menu appear according to position of mouse on screen.
- This control also provides a combination of static and dynamic display modes.
- When we use menu control with root items, static mode is used.
- For menu control with child menu items, we can use dynamic mode.
- We can made setting of ASP.NET Menu control in the designer with static links to our pages or we can bind

menu control automatically to a hierarchical data source such as an XmlDataSource or a SiteMapDataSource control.

- Its properties like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. are applied using style properties of <table>, < tr>, <td> tags.

Following are some important properties that are very useful

Properties of Menu Control	
DataSourceID	This property used to state the data source to be used.
Text	This property used to state the text to display in the menu.
Tooltip	This property used to state the tooltip of the menu item when we fire mouse over.
Value	This property used to states the nondisplayed value (usually unique id to use in server side events)
NavigateUrl	This property is used to state the target location to shift the user when menu item is clicked. If not set we can use MenuItemClick event to decide what to do.
Target	If NavigationUrl property is set, it indicates where to open the target location (in new window or same window).
Selectable	If false, this item can't be selected.
Image Url	This property is used to state the image that appears next to the menu item.
Image Tool Tip	This property is used to state the tooltip text to display for image next to the item.
Pop Out Image Url	This property is used to state the image that is displayed right to the menu item when it has some subitems.
Target	NavigationUrl property is set to state target location to open the file
Static Menu Style	This property is used to Set the style of the parent box which contains all menu items .
Dynamic Menu Style	This property is used to Set the style of the parent box which contains dynamic menu items.
Static Menu Item Style	This property is used to Set the style of the every static menu items.
Dynamic Menu Item Style	This property is used to Set the style of the every dynamic menu item.
Static Selected Style	This property is used to the style of the selected static items.
Dynamic Selected Style	This property is used to set the style of the selected dynamic items.
Static Hover Style	This property is used to set the mouse overing style of the static items.
Dynamic Hover Style	This property is used to Set the mouse overing style of the dynamic items and subitems.
Target	NavigationUrl property is set to state target location to open the file

Program 2.6.1

Create a program using menu control to show different menus such as Home,About,Gallary etc.

Solution :

Program using menu control to show different menus such as Home,About,Gallary etc.

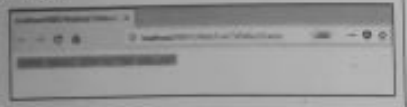
Default2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
```

Create menu control

```
<div>
<asp:Menu ID="Menu1" runat="server"
BackColor="#B5C7DE" DynamicHorizontalOffset="2"

Font-Names="Verdana" Font-Size="0.8em"

ForeColor="#284E98" Orientation="Horizontal"

StaticSubMenuIndent="10px">

<Items>
<asp:MenuItem Text="HOME" Value="home"
NavigateUrl="~/home.aspx">
</asp:MenuItem>
<asp:MenuItem Text="ABOUT" Value="about"
NavigateUrl="~/about1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="CONTACTNO" Value="contact"
NavigateUrl="~/contact1.aspx">
</asp:MenuItem>
<asp:MenuItem Text="GALLARY" Value="contact"
NavigateUrl="~/gallary.aspx">
</asp:MenuItem>
</Items>

</asp:Menu>

</div>
</form>
</body>
</html>
```

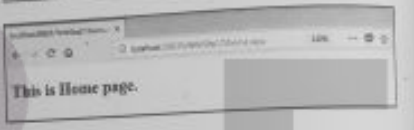
Output



Home.aspx

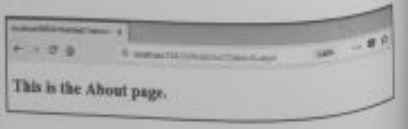
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="home.aspx.cs" Inherits="home" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <h2>This is Home page.</h2>
  </div>
  </form>
</body>
</html>
```



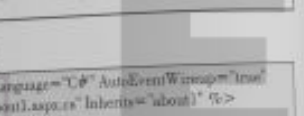
about1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="about1.aspx.cs" Inherits="about1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <h2>This is the About page.</h2>
  </div>
  </form>
</body>
</html>
```



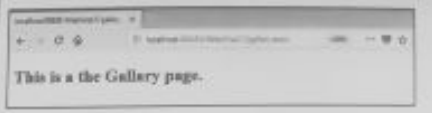
Contact1.aspx

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <h2>This is the contact page.</h2>
  </div>
  </form>
</body>
</html>
```



gallary.aspx

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <h2>This is a the  Gallary page.</h2>
  </div>
  </form>
</body>
</html>
```



Review Questions

Q. 1 Write short note on : List Controls. **(Refer Section 2.1.3)** **(4 Marks)**

Q. 2 State the page life cycle with diagram. **(Refer Section 2.1.6)** **(4 Marks)**

Q. 3 Write short note on view state. **(Refer Section 2.2.1)** **(4 Marks)**

Q. 4 Explain cookies in details. **(Refer Section 2.2.4)** **(4 Marks)**

Q. 5 Explain session in detail. **(Refer Section 2.2.5)** **(4 Marks)**

Q. 6 Explain the application state in detail ? **(Refer Section 2.2.7)** **(4 Marks)**

Q. 7 What is use of validation controls? List the different validation controls. **(Refer Section 2.3.1)** **(4 Marks)**

Q. 8 List all HTML5 validation controls and explain any 2 HTML validation controls in details. **(Refer Section 2.3.4)** **(4 Marks)**

Q. 9 List all rich controls and explain any one rich control in detail. **(Refer Section 2.4)** **(4 Marks)**

Q. 10 Write short note on Multiview control. **(Refer Section 2.4.3)** **(4 Marks)**

Q. 11 Explain themes and master pages in detail. **(Refer Section 2.5)** **(4 Marks)**

Q. 12 What is concept of master pages and content pages and how to connect them? **(Refer Section 2.5.4)** **(4 Marks)**

Q. 13 Explain siteMapPath control in detail. **(Refer Section 2.6.3)** **(4 Marks)**

Q. 14 Write note on menu control. **(Refer Section 2.6.5)** **(4 Marks)**

□□□