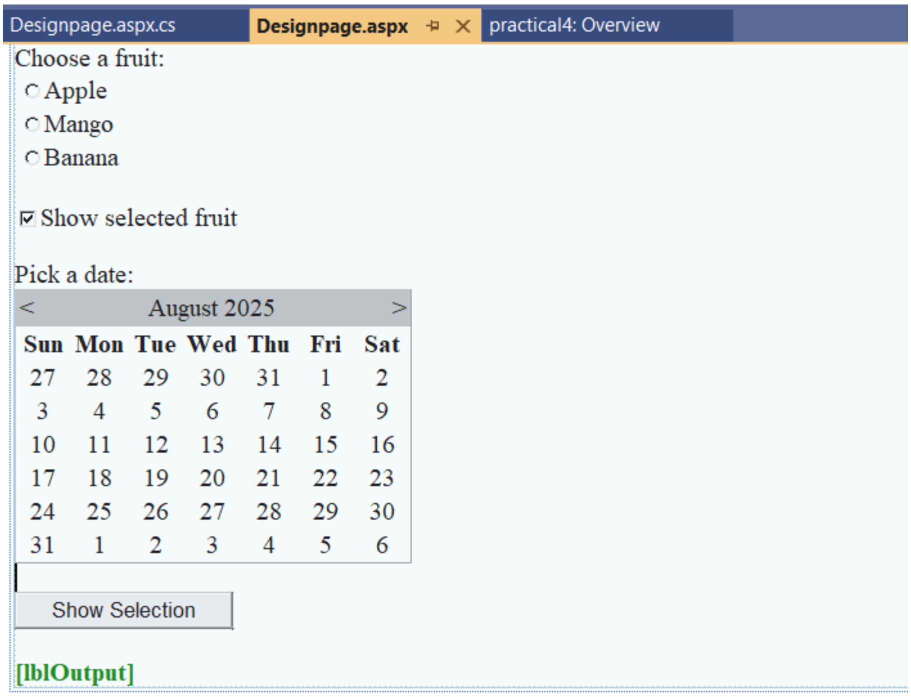
Create a simple web page with various server controls to demonstrate setting and use of their properties. (Example: AutoPostBack)

design



Choose a fruit:

○ Apple
○ Mango
○ Banana

☑ Show selected fruit

Pick a date:

<			August 2025			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

[lblOutput]

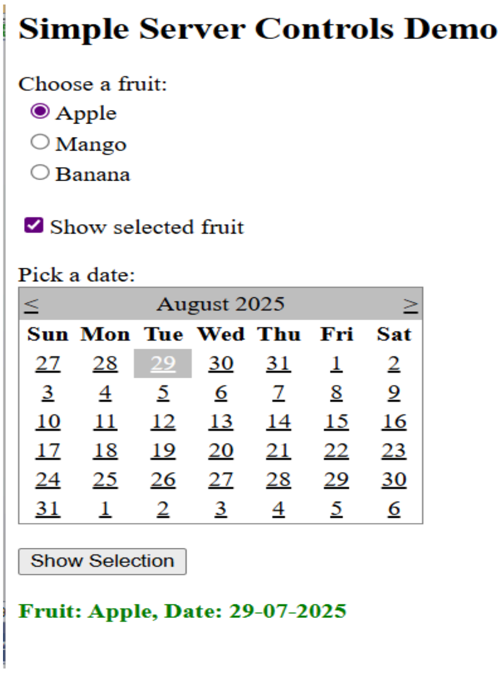
aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Designpage.aspx.cs"
Inherits="practical4.designpage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Simple Server Controls Demo</h2>

    <!-- RadioButtonList -->
    <asp:Label ID="Label1" runat="server" Text="Choose a fruit:"></asp:Label><br />
    <asp:RadioButtonList ID="rblFruits" runat="server">
      <asp:ListItem>Apple</asp:ListItem>
      <asp:ListItem>Mango</asp:ListItem>
      <asp:ListItem>Banana</asp:ListItem>
    </asp:RadioButtonList>
    <br />

    <!-- CheckBox -->
    <asp:CheckBox ID="chkShow" runat="server" Text="Show selected fruit" Checked="true"
/>
    <br /><br />

    <!-- Calendar -->
    <asp:Label ID="Label2" runat="server" Text="Pick a date:"></asp:Label><br />
    <asp:Calendar ID="calDate" runat="server"></asp:Calendar>
    <br />

    <!-- Button -->
    <asp:Button ID="btnShow" runat="server" Text="Show Selection"
OnClick="btnShow_Click" />
    <br /><br />

    <!-- Output Label -->
    <asp:Label ID="lblOutput" runat="server" ForeColor="Green"
Font-Bold="true"></asp:Label>

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

aspx.cs

```
using System;


namespace practical4
{
  public partial class designpage : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnShow_Click(object sender, EventArgs e)
    {
      if (chkShow.Checked)
```

```
            {
                string fruit = rblFruits.SelectedValue;
                string date = calDate.SelectedDate == DateTime.MinValue
                            ? "No date selected"
                            : calDate.SelectedDate.ToShortDateString();

                lblOutput.Text = $"Fruit: {fruit}, Date: {date}";
            }
            else
            {
                lblOutput.Text = "Display is turned off.";
            }
        }
    }
}
```



Simple Server Controls Demo

Choose a fruit:

● Apple
○ Mango
○ Banana

☑ Show selected fruit

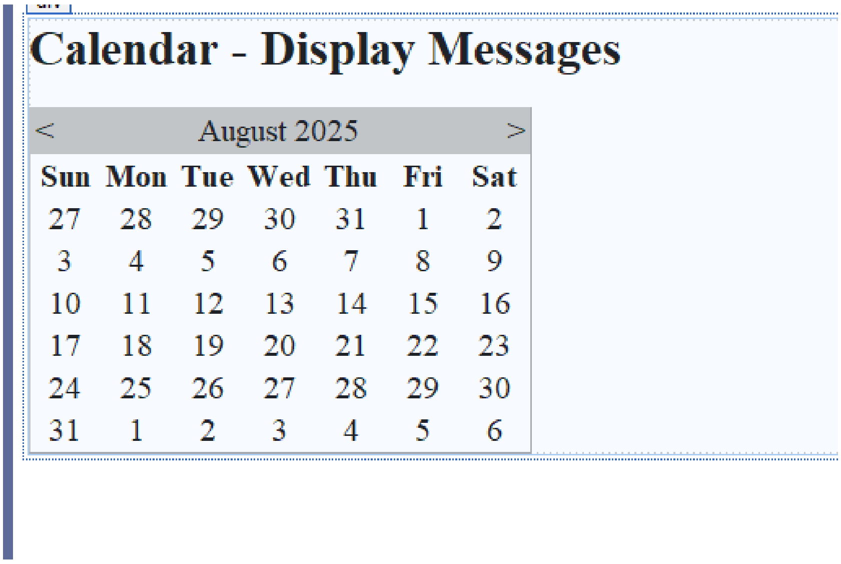
Pick a date:

< August 2025 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Fruit: Apple, Date: 29-07-2025

b. Demonstrate the use of Calendar control to perform following operations.
i. Display messages in a calendar control



Calendar - Display Messages

<			August 2025			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

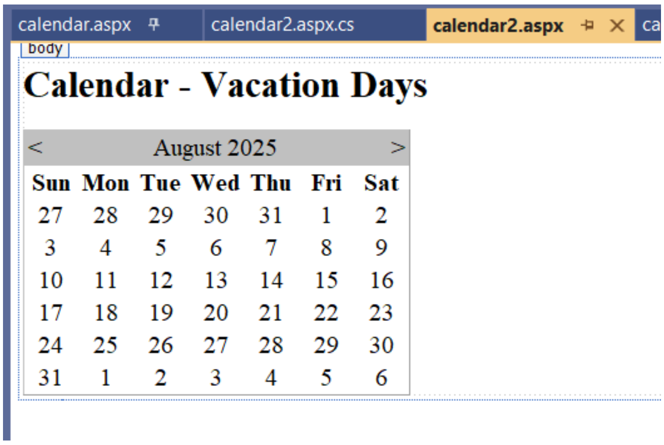
Aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="calendar.aspx.cs"
Inherits="practical4.calendar" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Calendar - Display Messages</h2>
            <asp:Calendar ID="Calendar1" runat="server"
OnDayRender="Calendar1_DayRender"></asp:Calendar>
        </div>
    </form>
</body>
</html>
```

aspx.cs

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

namespace practical4
{
    public partial class calendar : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
        {
            if (e.Day.Date == DateTime.Today)
            {
                e.Cell.Controls.Add(new Literal { Text = "<br/><span style='color:red;'>Today</span>"
});
            }

            if (e.Day.Date == new DateTime(DateTime.Today.Year, DateTime.Today.Month, 15))
            {
                e.Cell.Controls.Add(new Literal { Text = "<br/><span
style='color:green;'>Meeting</span>" });
            }
        }
    }
}
```

ii. Display vacation in a calendar control



Calendar - Vacation Days

August 2025

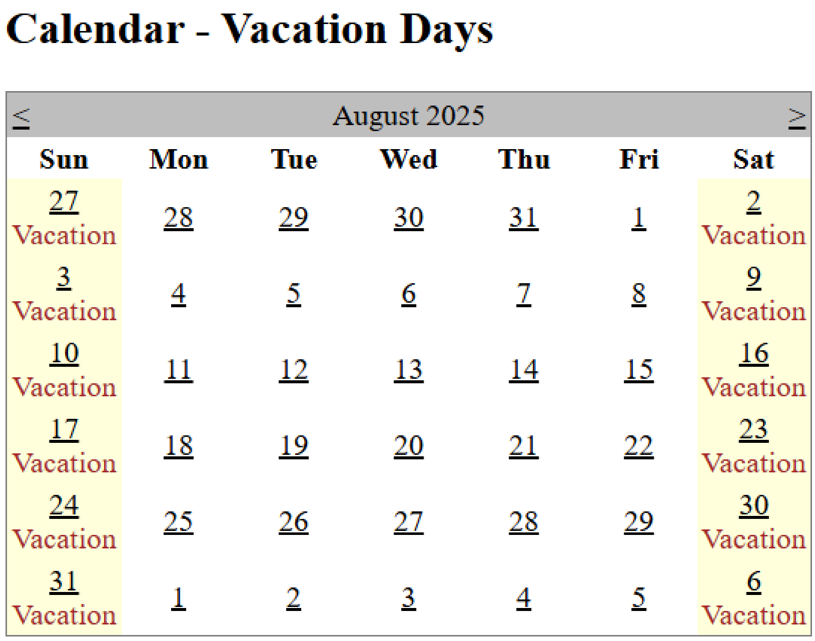
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="calendar2.aspx.cs"
Inherits="practical4.calendar2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Calendar - Vacation Days</h2>
            <asp:Calendar ID="Calendar1" runat="server"
OnDayRender="Calendar1_DayRender"></asp:Calendar>
        </div>
    </form>
</body>
</html>
```

aspx.cs

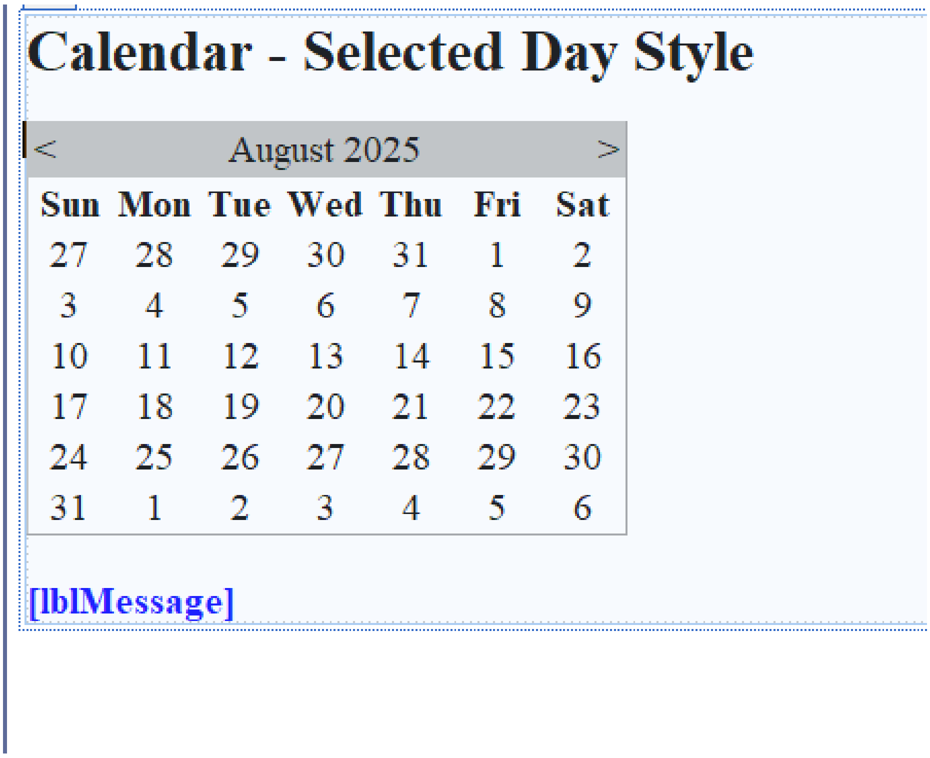
```
using System;

using System.Web.UI.WebControls;

namespace practical4
{
    public partial class calendar2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
        {
            if (e.Day.IsWeekend)
            {
                e.Cell.BackColor = System.Drawing.Color.LightYellow;
                e.Cell.Controls.Add(new Literal { Text = "<br/><span
style='color:brown;'>Vacation</span>" });
            }
        }
    }
}
```



Calendar - Vacation Days

<			August 2025			>
Sun	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**
27 Vacation	28	29	30	31	1	2 Vacation
3 Vacation	4	5	6	7	8	9 Vacation
10 Vacation	11	12	13	14	15	16 Vacation
17 Vacation	18	19	20	21	22	23 Vacation
24 Vacation	25	26	27	28	29	30 Vacation
31 Vacation	1	2	3	4	5	6 Vacation

iii. Selected day in a calendar control using style



Calendar - Selected Day Style

<			August 2025			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

[lblMessage]

Aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="calendar3.aspx.cs"
Inherits="practical4.calendar3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Calendar - Selected Day Style</h2>
            <asp:Calendar ID="Calendar1" runat="server"
            OnSelectionChanged="Calendar1_SelectionChanged">
        </asp:Calendar>
        <br />
        <asp:Label ID="lblMessage" runat="server" ForeColor="Blue"
Font-Bold="true"></asp:Label>
```

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

aspx.cs

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

namespace practical4
{
    public partial class calendar3 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Calendar1_SelectionChanged(object sender, EventArgs e)
        {
            lblMessage.Text = "You selected: " + Calendar1.SelectedDate.ToShortDateString();

            Calendar1.SelectedDayStyle.BackColor = System.Drawing.Color.LightBlue;
            Calendar1.SelectedDayStyle.ForeColor = System.Drawing.Color.DarkBlue;
            Calendar1.SelectedDayStyle.Font.Bold = true;
        }
    }
}
```

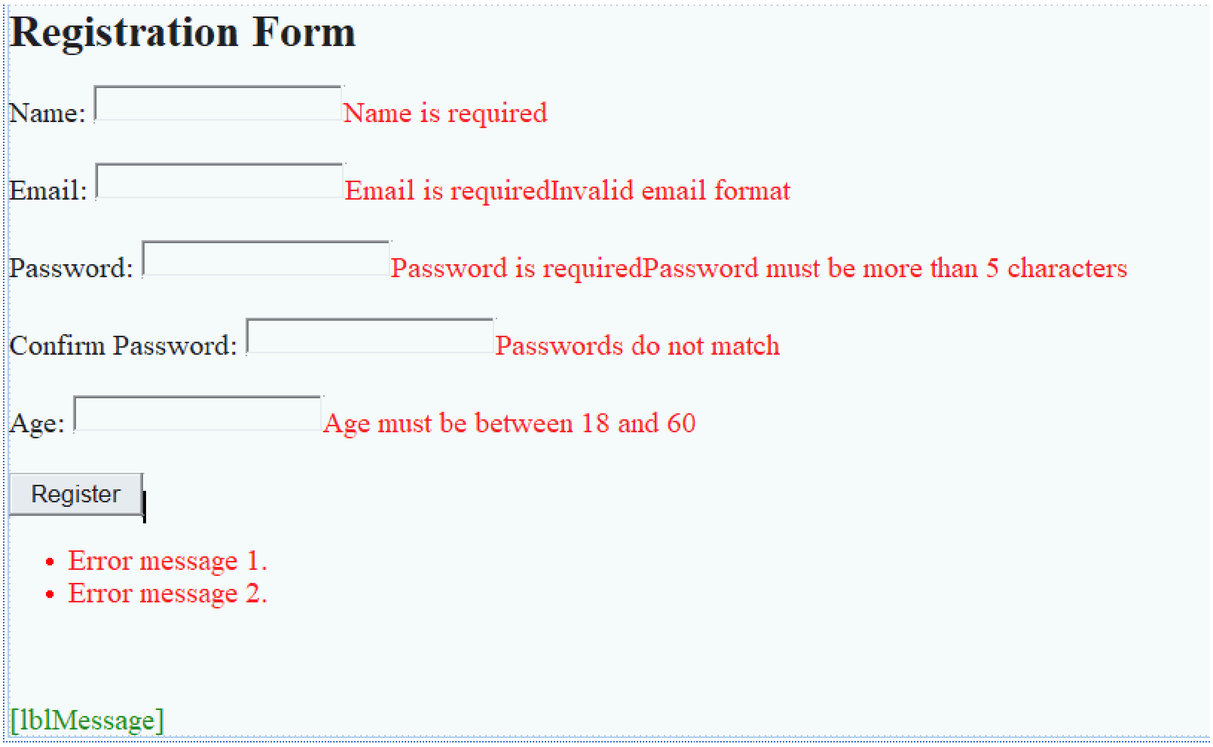
Calendar - Selected Day Style

<			August 2025			>
Sun	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	**21**	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

You selected: 21-08-2025

Practical 5

Working with Form Controls
a. Create a Registration form to demonstrate use of various Validation controls.



Registration Form

Name: Name is required

Email: Email is requiredInvalid email format

Password: Password is requiredPassword must be more than 5 characters

Confirm Password: Passwords do not match

Age: Age must be between 18 and 60

- Error message 1.
- Error message 2.

[lblMessage]

Aspx

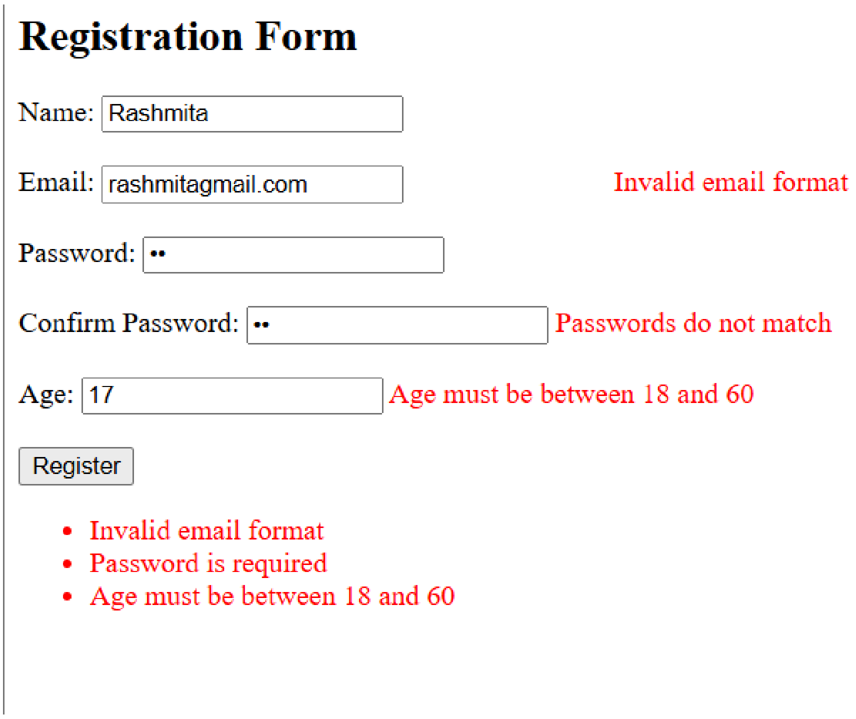
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="firstone.aspx.cs"
Inherits="Practical5.firstone" %>

<!DOCTYPE html>

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

        <h2>Registration Form</h2>

        <!-- Name -->
        Name:
        <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="rfvName" runat="server"
            ControlToValidate="txtName"
```

```
      ErrorMessage="Name is required"
      ForeColor="Red" />
    <br /><br />

    <!-- Email -->
    Email:
    <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="rfvEmail" runat="server"
      ControlToValidate="txtEmail"
      ErrorMessage="Email is required"
      ForeColor="Red" />
    <asp:RegularExpressionValidator ID="revEmail" runat="server"
      ControlToValidate="txtEmail"
      ErrorMessage="Invalid email format"
      ForeColor="Red"
      ValidationExpression="\w+@\w+\.\w+" />
    <br /><br />

    <!-- Password -->
    Password:
    <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
    <asp:RequiredFieldValidator ID="rfvPassword" runat="server"
      ControlToValidate="txtPassword"
      ErrorMessage="Password is required"
      ForeColor="Red" />
    <asp:CustomValidator ID="cvPasswordLength" runat="server"
      ControlToValidate="txtPassword"
      ErrorMessage="Password must be more than 5 characters"
      ForeColor="Red"
      OnServerValidate="cvPasswordLength_ServerValidate" />
    <br /><br />

    <!-- Confirm Password -->
    Confirm Password:
    <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
    <asp:CompareValidator ID="cvPasswords" runat="server"
      ControlToValidate="txtConfirmPassword"
      ControlToCompare="txtPassword"
      ErrorMessage="Passwords do not match"
      ForeColor="Red" />
    <br /><br />

    <!-- Age -->
```

```
        Age:
        <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
        <asp:RangeValidator ID="rvAge" runat="server"
            ControlToValidate="txtAge"
            MinimumValue="18"
            MaximumValue="60"
            Type="Integer"
            ErrorMessage="Age must be between 18 and 60"
            ForeColor="Red" />
        <br /><br />

        <!-- Submit Button -->
        <asp:Button ID="btnSubmit" runat="server"
            Text="Register"
            OnClick="btnSubmit_Click" />
        <asp:ValidationSummary ID="vsSummary" runat="server"
            ForeColor="Red" />
        <br /><br />

        <!-- Message Label -->
        <asp:Label ID="lblMessage" runat="server" ForeColor="Green"></asp:Label>

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

aspx.cs

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

namespace Practical5
{
    public partial class firstone : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
        }
        protected void cvPasswordLength_ServerValidate(object source,
System.Web.UI.WebControls.ServerValidateEventArgs args)
        {
            // Validate if password length > 5
            args.IsValid = args.Value.Length > 5;
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                lblMessage.Text = "Registration successful!";
            }
        }
    }
}
```



Registration Form

Name: Rashmita

Email: rashmitagmail.com Invalid email format

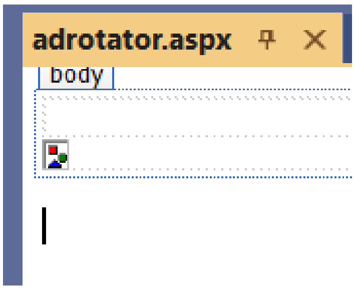
Password: ••

Confirm Password: •• Passwords do not match

Age: 17 Age must be between 18 and 60

- Invalid email format
- Password is required
- Age must be between 18 and 60

b. Create Web Form to demonstrate use of Ad rotator Control



Aspx

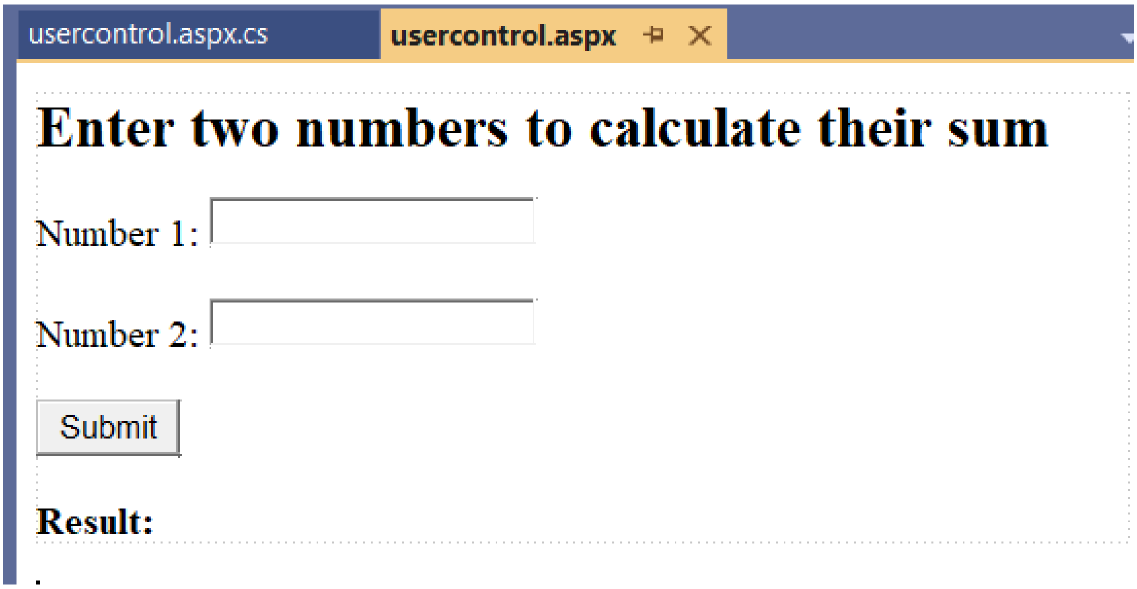
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="adrotator.aspx.cs"
Inherits="richcontrols.adrotator" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
        <asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/XMLFile1.xml"
OnAdCreated="AdRotator1_AdCreated1" />
    </form>
</body>
</html>
```

Xmlfile

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
            <Ad>
                        <ImageUrl>/images/p1.jpg</ImageUrl>
                        <NavigateUrl>https://www.google.com</NavigateUrl>
                        <AlternateText>The photo is about colors</AlternateText>
            </Ad>
            <Ad>
                        <ImageUrl>/images/p2.jpg</ImageUrl>
```

```
			<NavigateUrl>https://www.google.com</NavigateUrl>
			<AlternateText>The photo is about colors</AlternateText>
		</Ad>
		<Ad>
			<ImageUrl>/images/p3.jpg</ImageUrl>
			<NavigateUrl>https://www.google.com</NavigateUrl>
			<AlternateText>The photo is about colors</AlternateText>
		</Ad>
</Advertisements>
```

c. Create Web Form to demonstrate use of User Controls.



Enter two numbers to calculate their sum

Number 1:

Number 2:

Result:

.

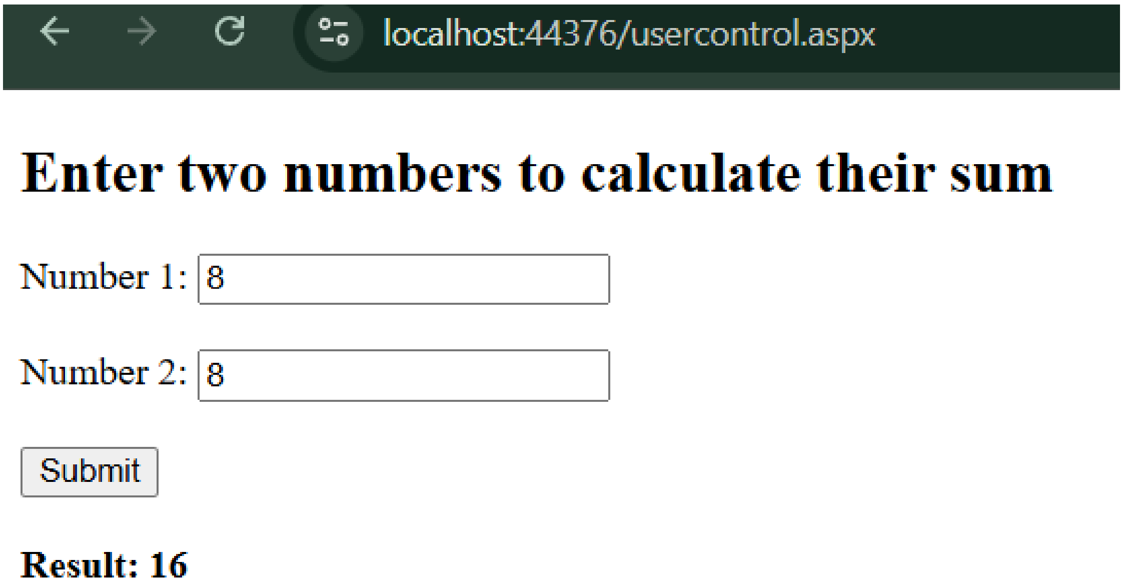
Aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="usercontrol.aspx.cs"
Inherits="Practical5.usercontrol" %>

<!DOCTYPE html>

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

```
    <form id="form1" runat="server">
        <div>
            <h2>Enter two numbers to calculate their sum</h2>


            <label for="Label1">Number 1:</label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />


            <label for="Label2">Number 2:</label>
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br /><br />


            <asp:Button ID="Button1" runat="server" Text="Submit" OnClick="Button1_Click" /><br
/><br />


            <asp:Label ID="LabelResult" runat="server" Text="Result: "
Font-Bold="true"></asp:Label>
        </div>
    </form>

</body>
</html>
```

aspx.cs

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

namespace Practical5
{
    public partial class usercontrol : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            int number1 = Convert.ToInt32(TextBox1.Text);
            int number2 = Convert.ToInt32(TextBox2.Text);


            int sum = number1 + number2;


            LabelResult.Text = "Result: " + sum.ToString();

        }
    }
}
```



Enter two numbers to calculate their sum

Number 1: 8

Number 2: 8

Result: 16

Practical 6
Working with Navigation, Beautification and Master page.
a. Create Web Form to demonstrate use of Website Navigation controls and Site Map.

CODE:

Web.Sitemap

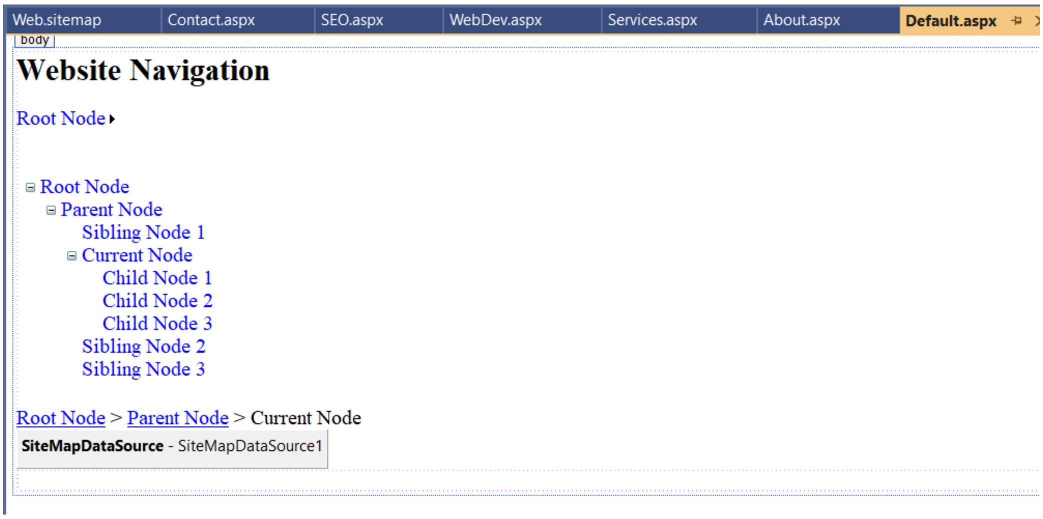
```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
        <siteMapNode title="Home" url="~/Default.aspx">
                <siteMapNode title="About" url="~/About.aspx" />
                <siteMapNode title="Services" url="~/Services.aspx">
                        <siteMapNode title="Web Development" url="~/WebDev.aspx" />
                        <siteMapNode title="SEO" url="~/SEO.aspx" />
                </siteMapNode>
                <siteMapNode title="Contact" url="~/Contact.aspx" />
        </siteMapNode>
</siteMap>
```

Aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="PRACTICAL6.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Website Navigation</h2>

      <asp:Menu ID="Menu1" runat="server"
DataSourceID="SiteMapDataSource1"></asp:Menu>
      <br />
      <asp:TreeView ID="TreeView1" runat="server"
DataSourceID="SiteMapDataSource1"></asp:TreeView>
      <br />
      <asp:SiteMapPath ID="SiteMapPath1" runat="server"></asp:SiteMapPath>

      <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
      <div>
      </div>
```

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

OUTPUT:



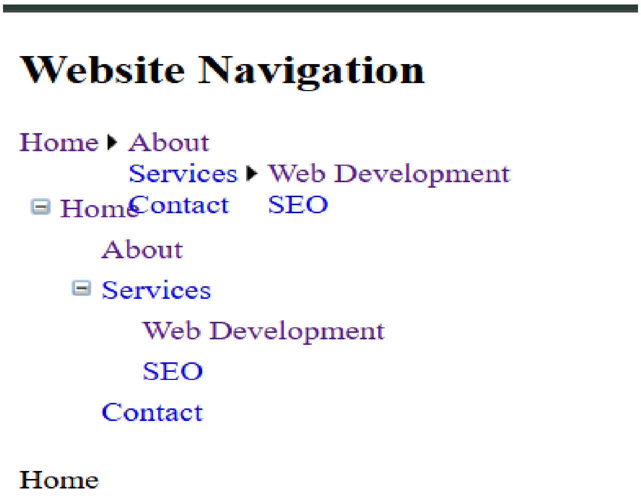
Website Navigation

Root Node ▸

⊟ Root Node
  ⊟ Parent Node
      Sibling Node 1
    ⊟ Current Node
        Child Node 1
        Child Node 2
        Child Node 3
      Sibling Node 2
      Sibling Node 3

Root Node > Parent Node > Current Node

SiteMapDataSource - SiteMapDataSource1



Website Navigation

Home ▸ About
Services ▸ Web Development
Contact SEO

⊟ Home
    About
  ⊟ Services
      Web Development
      SEO
    Contact

Home

b. Create a web application to demonstrate the use of Master Page with applying Styles and Themes for page beautification.

Code:

Site.Master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="PRACTICAL6.Site" %>

<!DOCTYPE html>
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Master Page Example</title>
   <link href="App_Themes/Default/Style.css" rel="stylesheet" />
</head>
<body>
   <form id="form1" runat="server">
      <div style="background-color:lightblue; padding:10px;">
         <h2>My Website Header</h2>
      </div>

      <asp:ContentPlaceHolder ID="MainContent" runat="server"></asp:ContentPlaceHolder>

      <div style="background-color:lightgray; padding:10px;">
         Footer © 2025
      </div>
   </form>
</body>
</html>
```

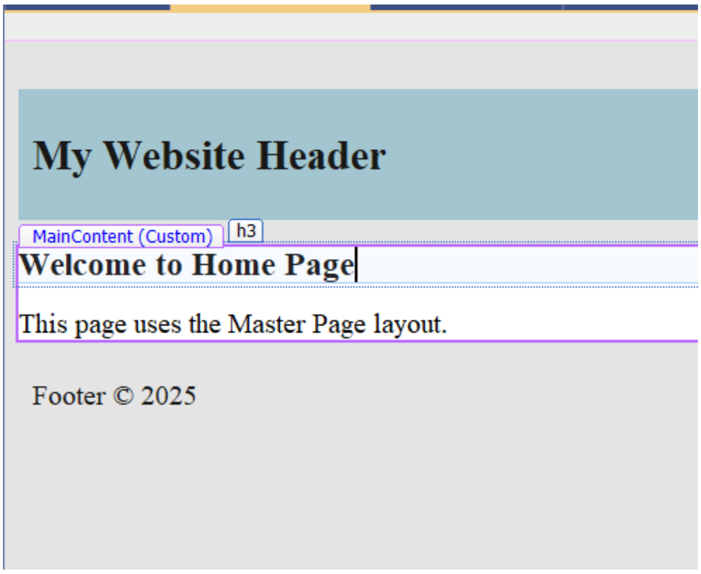
Home.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="HOME.aspx.cs" Inherits="PRACTICAL6.HOME" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
   <h3>Welcome to Home Page</h3>
   <p>This page uses the Master Page layout.</p>
</asp:Content>
```

Style.css

```
body {
   font-family: Arial;
   background-color: #f9f9f9;
}

h3 {
   color: darkblue;
}
```

Output:

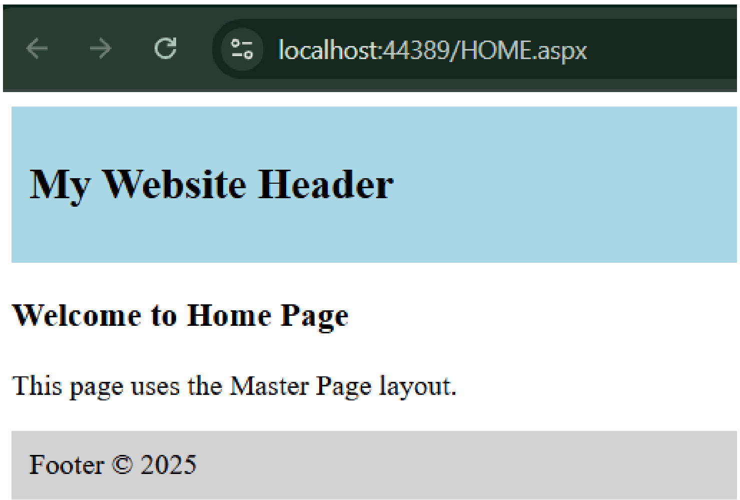


My Website Header

Welcome to Home Page

This page uses the Master Page layout.

Footer © 2025



My Website Header

Welcome to Home Page

This page uses the Master Page layout.

Footer © 2025

c. Create a web application to demonstrate various states of ASP.NET Pages.

CODE:
ASPX

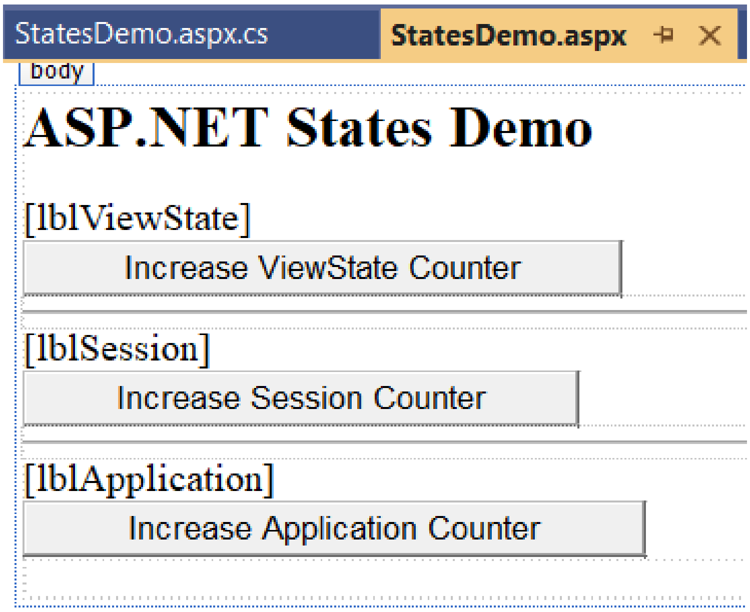
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StatesDemo.aspx.cs"
Inherits="PRACTICAL6.StatesDemo" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>ASP.NET States Demo</h2>

        <asp:Label ID="lblViewState" runat="server"></asp:Label><br />
        <asp:Button ID="btnViewState" runat="server" Text="Increase ViewState Counter"
OnClick="btnViewState_Click" />
        <hr />

        <asp:Label ID="lblSession" runat="server"></asp:Label><br />
        <asp:Button ID="btnSession" runat="server" Text="Increase Session Counter"
OnClick="btnSession_Click" />
        <hr />

        <asp:Label ID="lblApplication" runat="server"></asp:Label><br />
        <asp:Button ID="btnApplication" runat="server" Text="Increase Application Counter"
OnClick="btnApplication_Click" />
        <div>
        </div>
    </form>
</body>
</html>
```

ASPX.CS

```
using System;

namespace PRACTICAL6
{
    public partial class StatesDemo : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (ViewState["Counter"] == null)
                ViewState["Counter"] = 0;

            if (Session["Counter"] == null)
                Session["Counter"] = 0;

            if (Application["Counter"] == null)
                Application["Counter"] = 0;
            lblViewState.Text = "ViewState Counter: " + ViewState["Counter"].ToString();
            lblSession.Text = "Session Counter: " + Session["Counter"].ToString();
            lblApplication.Text = "Application Counter: " + Application["Counter"].ToString();
        }

        protected void btnViewState_Click(object sender, EventArgs e)
        {
            ViewState["Counter"] = (int)ViewState["Counter"] + 1;
        }

        protected void btnSession_Click(object sender, EventArgs e)
        {
            Session["Counter"] = (int)Session["Counter"] + 1;
        }

        protected void btnApplication_Click(object sender, EventArgs e)
        {
            Application["Counter"] = (int)Application["Counter"] + 1;
        }
    }
}
```

OUTPUT:



ASP.NET States Demo

[lblViewState]

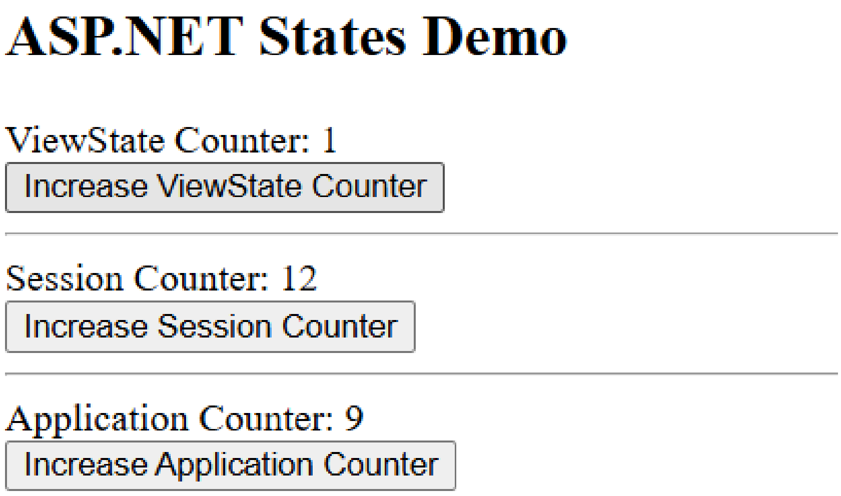
Increase ViewState Counter

[lblSession]

Increase Session Counter

[lblApplication]

Increase Application Counter



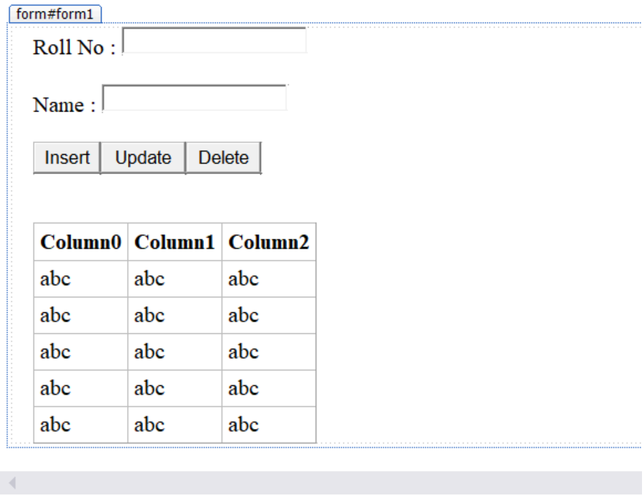
ASP.NET States Demo

ViewState Counter: 1

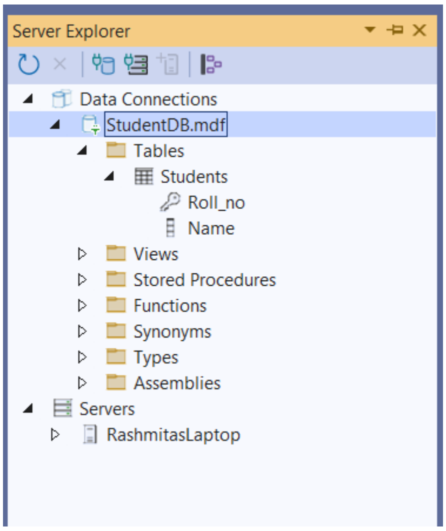
Session Counter: 12

Application Counter: 9

Working with Database (Connected Data Access)
Write a web application to perform CRUD operation.



Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc



Aspx

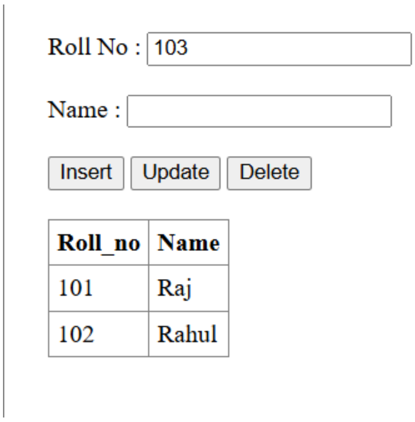
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="practical7_o_.WebForm1" %>

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

```
        <div style="margin:20px;">

            <!-- Roll No -->
            <asp:Label ID="Label1" runat="server" Text="Roll No : "></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br /><br />

            <!-- Name -->
            <asp:Label ID="Label2" runat="server" Text="Name : "></asp:Label>
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br /><br />

            <!-- CRUD Buttons -->
            <asp:Button ID="Button1" runat="server" Text="Insert" OnClick="Button1_Click" />
            <asp:Button ID="Button2" runat="server" Text="Update" OnClick="Button2_Click" />
            <asp:Button ID="Button3" runat="server" Text="Delete" OnClick="Button3_Click" />
            <br /><br />

            <!-- GridView -->
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="True"
                          BorderStyle="Solid" BorderWidth="1px" CellPadding="5" />
        </div>
    </form>
</body>
</html>
```

aspx.cs

```
using System;
using System.Data.SqlClient;
using System.Web.UI;

namespace practical7_o_
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        string conString = @"Data 
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Rashmita\OneDrive\Document
s\StudentDB.mdf;Integrated Security=True";
```

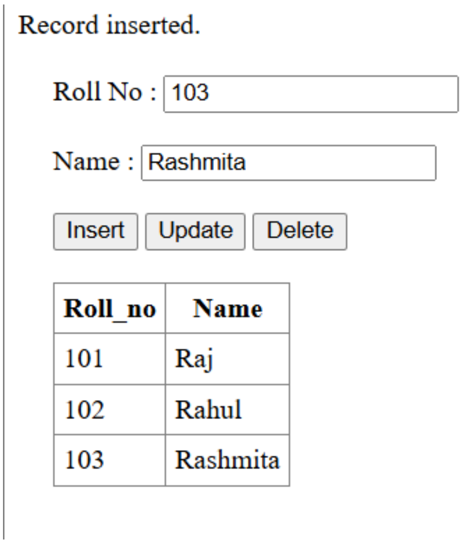
```
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGrid();
            }
        }

        void BindGrid()
        {
            using (SqlConnection con = new SqlConnection(conString))
            {
                SqlCommand cmd = new SqlCommand("SELECT * FROM Students", con);
                con.Open();
                SqlDataReader reader = cmd.ExecuteReader();
                GridView1.DataSource = reader;
                GridView1.DataBind();
            }
        }

        // INSERT
        protected void Button1_Click(object sender, EventArgs e)
        {
            using (SqlConnection con = new SqlConnection(conString))
            {
                SqlCommand cmd = new SqlCommand("INSERT INTO Students VALUES(@Roll,
@Name)", con);
                cmd.Parameters.AddWithValue("@Roll", TextBox1.Text);
                cmd.Parameters.AddWithValue("@Name", TextBox2.Text);
                con.Open();
                int c = cmd.ExecuteNonQuery();

                Response.Write(c > 0 ? "Record inserted." : "Error inserting.");
            }
            BindGrid();
        }

        // UPDATE
        protected void Button2_Click(object sender, EventArgs e)
        {
            using (SqlConnection con = new SqlConnection(conString))
            {
                SqlCommand cmd = new SqlCommand("UPDATE Students SET Name=@Name
WHERE Roll_no=@Roll", con);
                cmd.Parameters.AddWithValue("@Roll", TextBox1.Text);
                cmd.Parameters.AddWithValue("@Name", TextBox2.Text);

                con.Open();
                int c = cmd.ExecuteNonQuery();

                Response.Write(c > 0 ? "Record updated." : "Error updating.");
            }
            BindGrid();
        }

        // DELETE
        protected void Button3_Click(object sender, EventArgs e)
        {
            using (SqlConnection con = new SqlConnection(conString))
            {
                SqlCommand cmd = new SqlCommand("DELETE FROM Students WHERE
Roll_no=@Roll", con);
                cmd.Parameters.AddWithValue("@Roll", TextBox1.Text);

                con.Open();
                int c = cmd.ExecuteNonQuery();

                Response.Write(c > 0 ? "Record deleted." : "Error deleting.");
            }
            BindGrid();
        }
    }
}
```



Roll No : 103

Name :

Roll_no	Name
101	Raj
102	Rahul

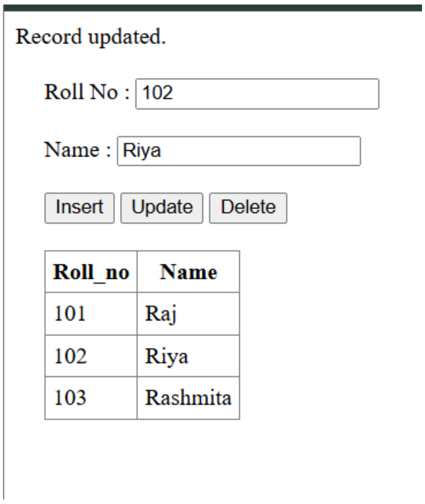


Record inserted.

Roll No : 103

Name : Rashmita

Roll_no	Name
101	Raj
102	Rahul
103	Rashmita

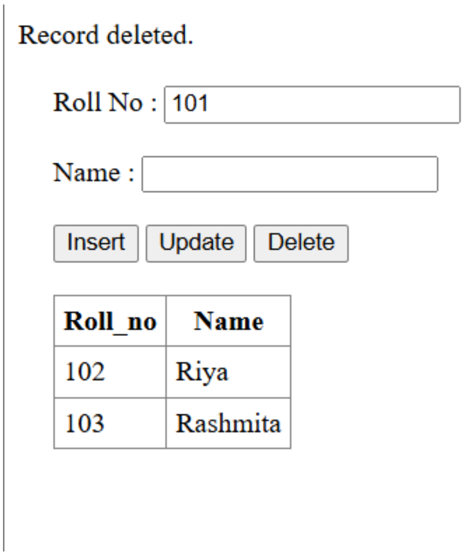


Record updated.

Roll No : 102

Name : Riya

Roll_no	Name
101	Raj
102	Riya
103	Rashmita

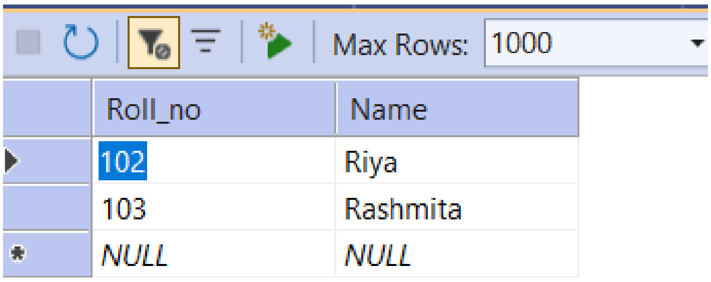


Record deleted.

Roll No : 101

Name :

Roll_no	Name
102	Riya
103	Rashmita



Max Rows: 1000

Roll_no	Name
102	Riya
103	Rashmita
NULL	*NULL*

Create a web application to display Using Disconnected Data Access and

Data Binding using Grid View.

Aspx

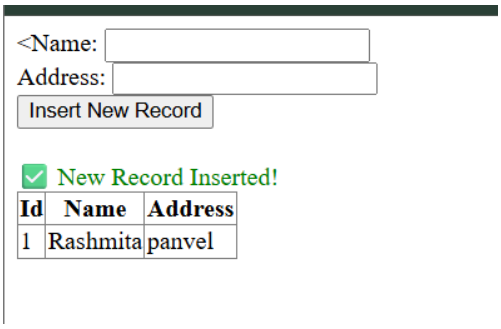
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication4.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Disconnected Data Access Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <<asp:Label Text="Name: " runat="server" />
        <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
        <br />

        <asp:Label Text="Address: " runat="server" />
        <asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>
        <br />

        <asp:Button ID="Button1" runat="server" Text="Insert New Record"
OnClick="Button1_Click1" />
        <br /><br />

        <asp:Label ID="lblMessage" runat="server" ForeColor="Green"></asp:Label>
        <br />

        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="True" />
        </div>
    </form>
</body>
</html>
```

aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace WebApplication4
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    string connectionStr = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Rashmita\OneDrive\Document
s\StudentDB.mdf;Integrated Security = True;";

    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        BindGrid();
      }
    }

    private void BindGrid()
    {
      using (SqlConnection con = new SqlConnection(connectionStr))
      {
        string query = "SELECT * FROM Details";
        SqlDataAdapter da = new SqlDataAdapter(query, con);
        DataSet ds = new DataSet();
        da.Fill(ds, "Details");

        GridView1.DataSource = ds;
        GridView1.DataBind();
      }
    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
      using (SqlConnection conn = new SqlConnection(connectionStr))
      {
        SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Details", conn);
        SqlCommandBuilder builder = new SqlCommandBuilder(da);

        DataSet ds = new DataSet();
        da.Fill(ds, "Details");

        DataRow newRow = ds.Tables["Details"].NewRow();
        newRow["Name"] = txtName.Text.Trim();
        newRow["Address"] = txtAddress.Text.Trim();

        ds.Tables["Details"].Rows.Add(newRow);
```

```
                // Save changes back to DB
                da.Update(ds, "Details");

                lblMessage.Text = "✅ New Record Inserted!";
                BindGrid();

                // Clear input fields
                txtName.Text = "";
                txtAddress.Text = "";
            }
        }
    }
}
```

output

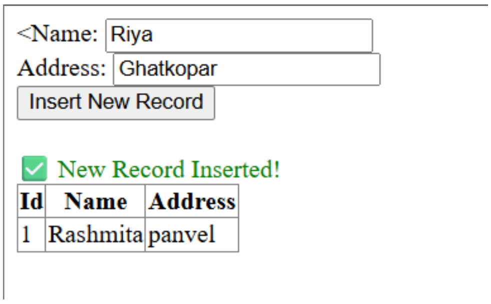


<Name:

Address:

✅ New Record Inserted!

Id	Name	Address
1	Rashmita	panvel

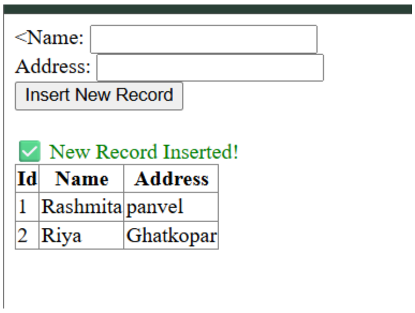


<Name: Riya

Address: Ghatkopar

✅ New Record Inserted!

Id	Name	Address
1	Rashmita	panvel



<Name:

Address:

✅ New Record Inserted!

Id	Name	Address
1	Rashmita	panvel
2	Riya	Ghatkopar

Practical 9

Working with GridView control

a. Create a web application to demonstrate use of GridView button column and GridView events.
b. Create a web application to demonstrate GridView paging and Create your own table format using GridView.

Code :

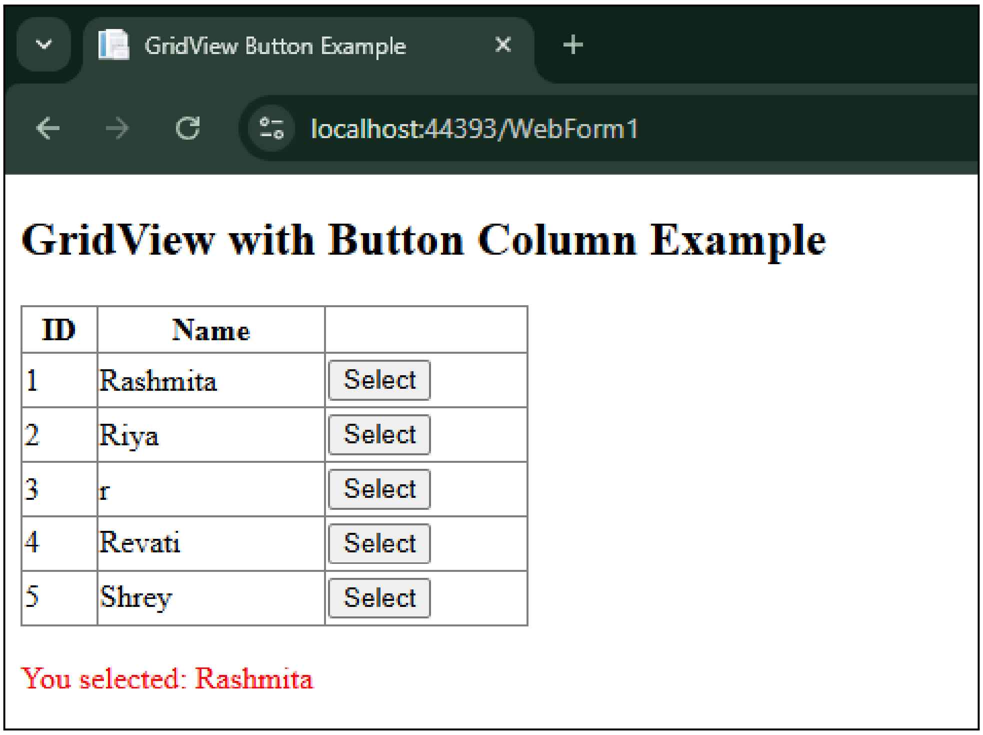
Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication3.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>GridView Button Example</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>GridView with Button Column Example</h2>
      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowCommand="GridView1_RowCommand" Height="165px" Width="261px">
         <Columns>
            <asp:BoundField DataField="ID" HeaderText="ID" />
            <asp:BoundField DataField="Name" HeaderText="Name" />
            <asp:ButtonField Text="Select" CommandName="SelectRow" ButtonType="Button" />
         </Columns>
      </asp:GridView>
      <br />
      <asp:Label ID="lblMessage" runat="server" ForeColor="Red" />
   </form>
</body>
</html>
```

webform.aspx.cs

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

namespace WebApplication3
{
    public partial class WebForm1 : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                var dt = new DataTable();
                dt.Columns.Add("ID");
                dt.Columns.Add("Name");
                dt.Rows.Add("1", "Rashmita");
                dt.Rows.Add("2", "Riya");
                dt.Rows.Add("3", "r");
                dt.Rows.Add("4", "Revati");
                dt.Rows.Add("5", "Shrey");

                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }

        protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
        {
            if (e.CommandName == "SelectRow")
            {
                // ButtonField sets CommandArgument to the row index automatically
                int rowIndex = Convert.ToInt32(e.CommandArgument);
                string selectedName = GridView1.Rows[rowIndex].Cells[1].Text;
                lblMessage.Text = "You selected: " + selectedName;
            }
        }
    }
}
```

Output :



GridView with Button Column Example

ID	Name	
1	Rashmita	Select
2	Riya	Select
3	r	Select
4	Revati	Select
5	Shrey	Select

You selected: Rashmita

b.

Code :

Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>GridView Paging Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>GridView Paging with Table Format</h2>
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
            Height="160px" Width="185px" AllowPaging="True" PageSize="5"
            OnPageIndexChanging="GridView1_PageIndexChanging">
            <Columns>
                <asp:BoundField DataField="ID" HeaderText="ID" />
                <asp:BoundField DataField="Name" HeaderText="Name" />
            </Columns>
        </asp:GridView>
    </form>
</body>
</html>
```

webform.aspx.cs

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

namespace WebApplication
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
```

```
            LoadGridData();
        }
    }

    private void LoadGridData()
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("ID");
        dt.Columns.Add("Name");

        dt.Rows.Add("1", "Rashmita");
        dt.Rows.Add("2", "Kavya");
        dt.Rows.Add("3", "Anjali");
        dt.Rows.Add("4", "TT");
        dt.Rows.Add("5", "JK");
        dt.Rows.Add("6", "RU");
        dt.Rows.Add("7", "SP");
        dt.Rows.Add("8", "Minal");
        dt.Rows.Add("9", "Shrey");

        GridView1.DataSource = dt;
        GridView1.DataBind();
    }

    protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        GridView1.PageIndex = e.NewPageIndex;   // Update the page index
        LoadGridData();                         // Reload data to reflect the new page
    }
  }
}
```