Module 1: Introduction to ASP.NET

Practical No.1

Aim: - Create a simple web page to display the working of asp.net framework

Source Code: -

First.aspx:

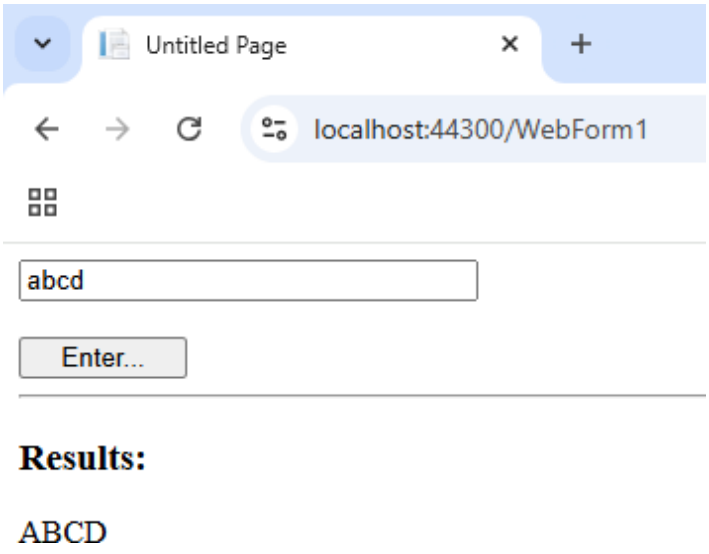
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication16.WebForm1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transition
al.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Untitled Page</title></head>
<body>
<form id="form1" runat="server">
<div>
<asp:TextBox ID="TextBox1" runat="server"
style="width:224px"></asp:TextBox>
<br /><br />
<asp:Button ID="Button1" runat="server" Text="Enter..."
style="width:85px" onclick="Button1_Click" />
<hr />
<h3> Results: </h3>
<span runat="server" id="changed_text" />
</div></form></body>  </html>
```

First.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication16
{

    public partial class WebForm1 :
System.Web.UI.Page
    {

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

protected void Button1_Click(object sender, EventArgs e)    {
string buf = TextBox1.Text;
changed_text.InnerHtml = buf.ToUpper();
}    }    }
```

Design:

Output:



Practical No.2

Aim: - Create a simple web page to show ASP.NET page Life cycle.

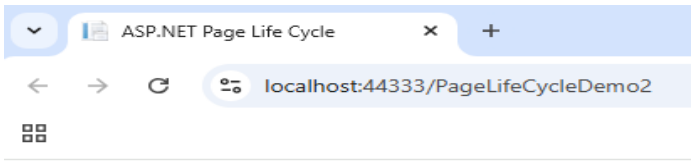
Source Code: - PageLifeCycleDemo2.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="PageLifeCycleDemo2.aspx.cs"
Inherits="PageLifeCycleDemo2.PageLifeCycleDemo2 " %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>ASP.NET Page Life Cycle</title></head>
<body>
<form id="form1" runat="server">
 <div><h1>ASP.NET Page Life Cycle</h1>
<asp:Label ID="lblMessage" runat="server"
Text="Lifecycle events will be logged here."></asp:Label>
</div></form></body></html>
```

PageLifeCycleDemo2.aspx.cs:

```
using System;
using System.Web.UI; namespace
PageLifeCycleDemo2
{

public partial class PageLifeCycleDemo2 : Page   {
protected void Page_PreInit(object sender, EventArgs
e)         {              Log("Page_PreInit");              }
protected void Page_Init(object sender, EventArgs e)
{             Log("Page_Init");}
protected void Page_InitComplete(object sender,
EventArgs e)            {
                          Log("Page_InitComplete");
}

protected override void OnPreLoad(EventArgs e)

{ base.OnPreLoad(e);
Log("OnPreLoad");
}

protected void Page_Load(object sender, EventArgs e)
{

Log("Page_Load");
}

protected override void OnLoadComplete(EventArgs e)
{

base.OnLoadComplete(e);
Log("OnLoadComplete");
}

protected override void OnPreRender(EventArgs e)
{

base.OnPreRender(e);
Log("OnPreRender");
}

protected override void
OnPreRenderComplete(EventArgs e)
{

base.OnPreRenderComplete(e);  Log("OnPreRenderComplete");
}

protected void Page_Unload(object sender, EventArgs e)
{

// Note: Response.Write or controls updates don't work in
Page_Unload.
// For demonstration, logging to Response is skipped here.
}

private void Log(string stage)
{

lblMessage.Text += $"{stage}<br />";
}     }     }
```

Output:-



ASP.NET Page Life Cycle

Lifecycle events will be logged here.Page_PreInit
Page_Init
Page_InitComplete
OnPreLoad
Page_Load
OnLoadComplete
OnPreRender
OnPreRenderComplete

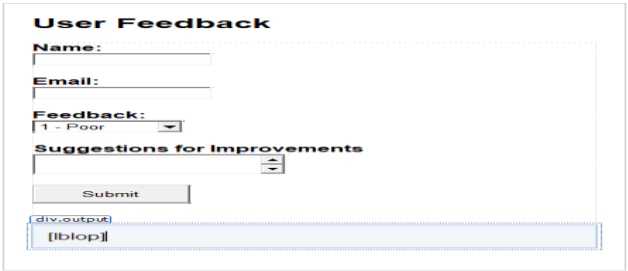
Practical No.3

Aim: - Design a web form to accept user input (e.g., name, email, and feedback) and display it after submission.

Source Code: - Feedbackform.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Feedbackform.aspx.cs"
Inherits="FeedbackForm.Feedback" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>User Feedback Form</title>
<style>
body { font-family: Arial, sans-serif; margin: 40px; }
.container { width: 400px; margin: auto; padding: 20px;
border: 1px solid #ccc; border-radius: 10px; } label { font-
weight: bold; }
.output { margin-top: 20px; padding: 10px; border: 1px solid
#ddd; background-color: #f9f9f9; }
</style></head>
<body>
<div class="container">
<h2>User Feedback</h2>
<form id="form1" runat="server">
<label for="txtName">Name:</label><br />
<asp:TextBox ID="txtname"
runat="server"></asp:TextBox><br /><br />
<label for="txtEmail">Email:</label><br />
<asp:TextBox ID="txtemail"
runat="server"></asp:TextBox><br /><br />
<label for="txtFeedback">Feedback:</label><br />
<asp:DropDownList ID="ddlRating" runat="server">
<asp:ListItem Text="1 - Poor" Value="1">
</asp:ListItem>
<asp:ListItem Text="2 - Fair" Value="2">
</asp:ListItem>
<asp:ListItem Text="3 - Good" Value="3">
</asp:ListItem>
<asp:ListItem Text="4 - Very Good" Value="4">
</asp:ListItem>
<asp:ListItem Text="5 - Excellent" Value="5">
</asp:ListItem>
</asp:DropDownList>
<br /> <br />
<asp:Label ID="Label1" runat="server"
Text="Suggestions for Improvements"
Font-Bold="True"></asp:Label><br />
<asp:TextBox ID="txtfeedback" runat="server"
TextMode="MultiLine"></asp:TextBox><br /><br />
<asp:Button ID="btnsubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" Height="32px" Width="112px"
/><br /><br />
<div class="output">
<asp:Label ID="lblop" runat="server"
Visible="False"></asp:Label>
</div></form></div></body></html>
```

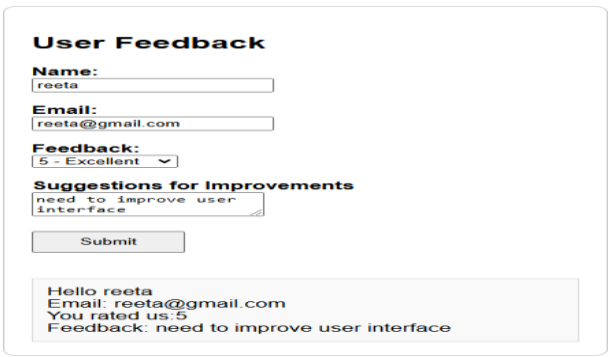
Design:



Feedbackform.aspx.cs:

```
using System;
using System.Collections.Generic; using System.Linq;
using System.Web; using System.Web.UI;
using System.Web.UI.WebControls; namespace FeedbackForm
{ public partial class Feedbackform : System.Web.UI.Page{ protected void
Page_Load(object sender, EventArgs e){
}
protected void btnSubmit_Click(object sender, EventArgs e){
string name = txtname.Text.Trim(); string email =
txtemail.Text.Trim();
string feedback = txtfeedback.Text.Trim(); lblop.Text = "Hello " +
txtname.Text + "<br/>" +
"Email: " + txtemail.Text + "<br/>" + "You rated us:"
+ddlRating.SelectedValue +"<br/>" +
"Feedback: " + txtfeedback.Text; lblop.Visible = true;
}}}
```

Output :



Practical No.4

Aim: - Implement a Cross Page Postback feature to pass data between two web pages.

Source Code: - WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication14.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"><title></title></head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Enter Your Name"></asp:Label>
   
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<br /><br /><br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Cross Page Post back" />
       
<br /><br /></div></form></body></html>
```

Design:

WebForm1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication1

{

    public partial class WebForm1 : System.Web.UI.Page
    {

    protected void Page_Load(object sender, EventArgs e){
                }
    protected void Button1_Click(object sender, EventArgs e)
    {

    Response.Redirect("WebForm2.aspx?name="+ TextBox1.Text  );
    }                       }
```

WebForm2.aspx:

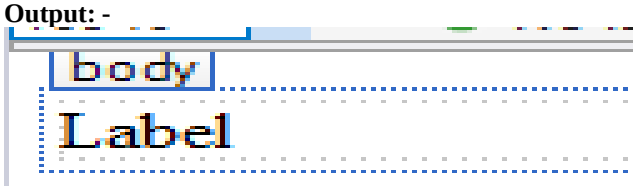
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="WebApplication14.WebForm2" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"><title></title></head>
<body>
<form id="form1" runat="server"><div>
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div></form></body></html>
```

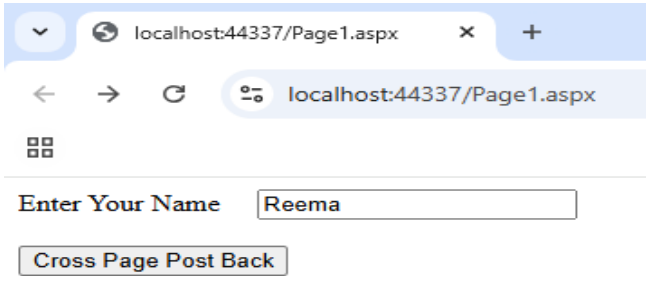
Design:

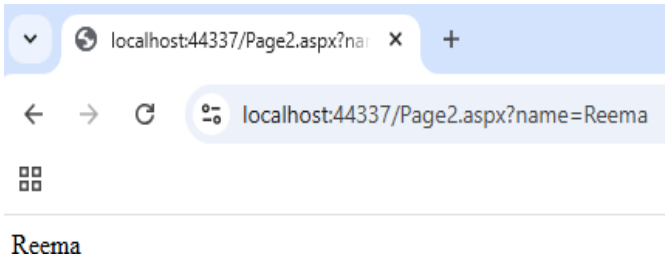
WebForm2.aspx.cs:

```
using System;
using System.Collections.Generic; using System.Linq;
using System.Web; using System.Web.UI;
using System.Web.UI.WebControls; namespace CrossPagePosting
{public partial class Page2 : System.Web.UI.Page
{

protected void Page_Load(object sender, EventArgs e)
{

Label1.Text = Request.QueryString["name"];
}    }    }
```

Output: -







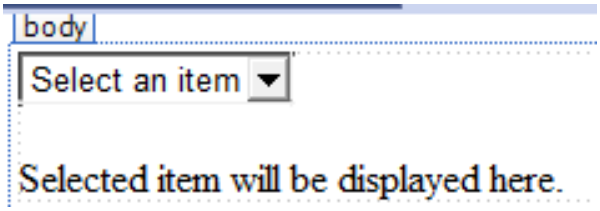
Practical No.5

Aim: Develop a web application demonstrating AutoPostback using a dropdown list to select and display items.

Source Code: - AutoPostBack.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="AutoPostBack.aspx.cs"
Inherits="AutoPostBackDemo.AutoPostBack" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"><title></title></head>
<body>
<form id="form1" runat="server">
<div>
<asp:DropDownList ID="ddlItems" runat="server"
AutoPostBack="True"
OnSelectedIndexChanged="ddlItems_SelectedIndexCh
anged">
<asp:ListItem Text="Select an item" Value="" />
<asp:ListItem Text="Apple" Value="Apple" />
<asp:ListItem Text="Banana" Value="Banana" />
<asp:ListItem Text="Cherry" Value="Cherry" />
<asp:ListItem Text="Date" Value="Date" />
</asp:DropDownList>
<br /><br />
<asp:Label ID="lblSelectedItem" runat="server"
Text="Selected item will be displayed here." />
</div>
</form>
</body>
</html>
```

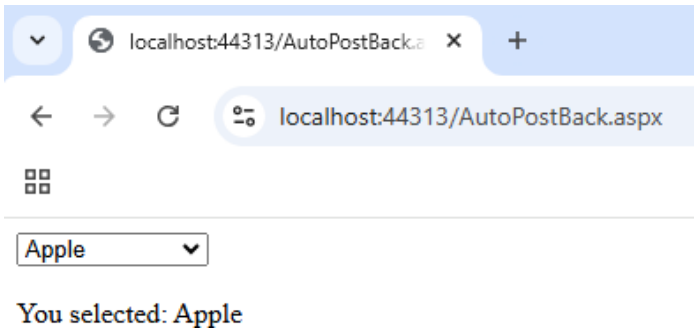
Design:



AutoPostBack.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace AutoPostBackDemo
{

public partial class AutoPostBack :
System.Web.UI.Page
{

protected void Page_Load(object sender, EventArgs e){
            }
protected void ddlItems_SelectedIndexChanged(object sender,
EventArgs e)
{

lblSelectedItem.Text = "You selected: " +
ddlItems.SelectedItem.Text;
}   }   }
```

Output: -



Practical No.6

Aim: - Create a simple calculator using server-side controls and operators in ASP.NET.

```
CodeBehind="Calculator.aspx.cs"
Inherits="Calculator.Calculator" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Simple Calculator</title>
<style>
        .calculator-container {
            width: 300px;
            margin: 0 auto;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 10px;
        }

        .calculator-container input { width:
            100%;
            padding:  10px;
            margin:  5px  0;
            font-size: 16px;
        }

        .calculator-container button {
            width: 100%;
            padding: 10px;
            font-size: 18px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 5px;
        }

        .calculator-container .operator-btn {
            background-color: #f44336;
        }

    </style>
</head>
<body>
<form id="form1" runat="server">
<div class="calculator-container">
<h2>Calculator</h2>
<!-- Input Fields for Numbers -->
```

Source Code: -

Calculator.aspx:

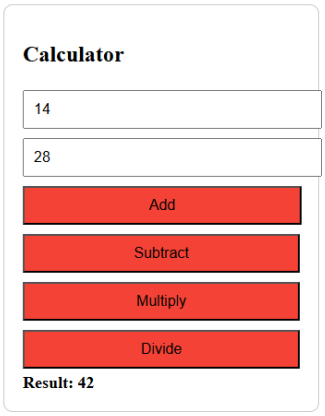
```
<%@ Page Language="C#" AutoEventWireup="true"
    <asp:TextBox ID="txtNumber1" runat="server"
    placeholder="Enter number 1"
    CssClass="calculator-input" />
    <asp:TextBox ID="txtNumber2" runat="server"
    placeholder="Enter number 2"
    CssClass="calculator-input" />

    <!-- Buttons for each operation -->
    <asp:Button ID="btnAdd" runat="server" Text="Add"
    OnClick="btnAdd_Click" CssClass="operator-btn"
    Width="302px" />
    <asp:Button ID="btnSubtract" runat="server"
    Text="Subtract"  OnClick="btnSubtract_Click"
    CssClass="operator-btn" />
    <asp:Button ID="btnMultiply" runat="server"
    Text="Multiply" OnClick="btnMultiply_Click"
    CssClass="operator-btn" />
    <asp:Button ID="btnDivide" runat="server"
    Text="Divide"  OnClick="btnDivide_Click"
    CssClass="operator-btn" />

    <!-- Output -->
    <asp:Label ID="lblResult" runat="server" Text="Result:
    " Font-Bold="True" Font-Size="Large"
    />

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

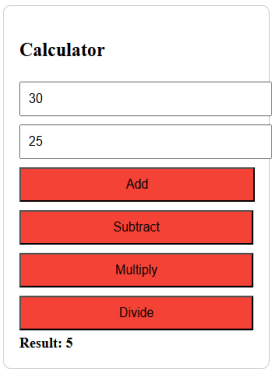
Calculator.aspx.cs:

```
using System;
using
System.Collections.Generic;
using System.Linq;
using System.Web;
using
System.Web.UI;
using System.Web.UI.WebControls;
namespace Calculator
{

public partial class Calculator : System.Web.UI.Page
{

protected void Page_Load(object sender, EventArgs
e){        }
// Event handler for the Add button
protected void btnAdd_Click(object sender, EventArgs e)
{

PerformCalculation("+");
```

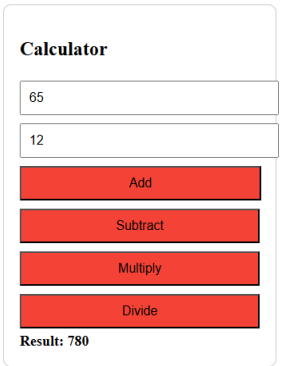
```
 }

// Event handler for the Subtract button protected
void btnSubtract_Click(object sender, EventArgs e)
{

PerformCalculation("-");
}


// Event handler for the Multiply button protected
void btnMultiply_Click(object sender, EventArgs e)
{

PerformCalculation("*");
}


// Event handler for the Divide button protected
void btnDivide_Click(object sender, EventArgs
e)
{

PerformCalculation("/");
}

private void PerformCalculation(string operatorSelected)
{

try
{

// Get the input values from the TextBoxes
double number1 =
Convert.ToDouble(txtNumber1.Text); double
number2 =
Convert.ToDouble(txtNumber2.Text); double
result = 0;

// Perform calculation based on the selected operator
switch (operatorSelected)
{

case "+":
result = number1 + number2;
break;
case "-":
result = number1 - number2;
break;
case "*":
result = number1 * number2;
break;
case "/":
if (number2 != 0)
{
```

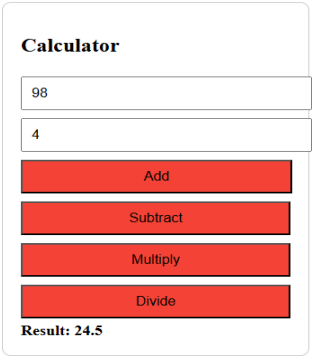
```
default
  lblResult.Text = "Please select a valid operator.";
  return;
  }

  // Display the result
  lblResult.Text = "Result: " + result.ToString();
  }

  catch (Exception ex)
  {

  // Handle any error (e.g., invalid input) and display a message
  lblResult.Text = "Error: " + ex.Message;
  }       }     }       }
```





Output:-





Practical No.7

Aim: - Design a Web Application for an Organization with Registration forms and advanced controls

user-form.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="user-form.aspx.cs"
Inherits="UserRegistration1.user_form" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
.auto-style1 {
     width:
     100%;
      }

.auto-style2 {
     width:
     278px;
       }

.auto-style3 {
      width:
      278px;
      height:
      23px;
        }

.auto-style4 {
      height:
      23px;
        }

</style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table class="auto-style1">
    <tr>   <td>
<asp:Label ID="Label1" runat="server" Text="User
Name"></asp:Label>
    </td>  <td>
    <asp:TextBox ID="username" runat="server"
required="true"></asp:TextBox></td>
    </tr>   <tr>  <td>
    <asp:Label ID="Label6" runat="server" Text="Email
ID"></asp:Label>
    </td>
    <td> <asp:TextBox ID="EmailID" runat="server"
TextMode="Email"></asp:TextBox></td>
    </tr> <tr>   <td>
    <asp:Label ID="Label2" runat="server"
Text="Password"></asp:Label></td>
    <td>
    <asp:TextBox ID="TextBox2" runat="server"
TextMode="Password"></asp:TextBox></td>
    </tr>  <tr>  <td>
    <asp:Label ID="Label3" runat="server" Text="Confirm
Password"></asp:Label></td> <td>
    <asp:TextBox ID="TextBox3" runat="server"
TextMode="Password"></asp:TextBox></td>
    </tr>  <tr> <td>
    <asp:Label ID="Label4" runat="server"
Text="Gender"></asp:Label></td>
    <td>
    <asp:RadioButton ID="RadioButton1" runat="server"
GroupName="gender" Text="Male" />
    <asp:RadioButton ID="RadioButton2" runat="server"
GroupName="gender" Text="Female"
/></td>  </tr>  <tr>   <td>
    <asp:Label ID="Label5" runat="server" Text="Select
Course"></asp:Label>s</td>
    <td>
    <asp:CheckBox ID="CheckBox1" runat="server"
Text="J2SEE" />
    <asp:CheckBox ID="CheckBox2" runat="server"
Text="J2EE" />
    <asp:CheckBox     ID="CheckBox3"     runat="server"
Text="Spring Framework" />
    </td> </tr>   <tr>  <td>  </td>   <td>   <br />
    <asp:Button     ID="Button1"     runat="server"
Text="Register"     CssClass="btn     btn-primary"
OnClick="Button1_Click"/>
    </td>  </tr>  </table>
    <asp:Label ID="message" runat="server"
Font-Size="Medium"  ForeColor="Red"></asp:Label>
        </div> </form>
```

```
<table class="auto-style1">
    <tr>
    <td class="auto-style2"><asp:Label
ID="ShowUserNameLabel" runat="server"
></asp:Label></td>
    <td>
    <asp:Label ID="ShowUserName" runat="server"
></asp:Label></td>   </tr>
    <tr>

<td class="auto-style2"><asp:Label
ID="ShowEmailIDLabel" runat="server"
></asp:Label></td>
    <td>
<asp:Label ID="ShowEmail" runat="server" >
</asp:Label></td>
    </tr>   <tr>
    <td class="auto-style3"><asp:Label
ID="ShowGenderLabel" runat="server" >
</asp:Label></td>
    <td class="auto-style4">
    <asp:Label ID="ShowGender" runat="server" >
</asp:Label></td>  </tr>
    <tr>
    <td class="auto-style2"><asp:Label
ID="ShowCourseLabel" runat="server" >
</asp:Label></td>
    <td>
    <asp:Label ID="ShowCourses" runat="server" >
</asp:Label></td>   </tr>
    </table>  </body></html>
```

User Name	
Email ID	
Password	
Confirm Password	
Gender	○ Male ○ Female
Select Courses	☐ J2SEE ☐ J2EE ☐ Spring Framework
	Register

[message]

[ShowUserNameLabel]	[ShowUserName]
[ShowEmailIDLabel]	[ShowEmail]
[ShowGenderLabel]	[ShowGender]
[ShowCourseLabel]	[ShowCourses]

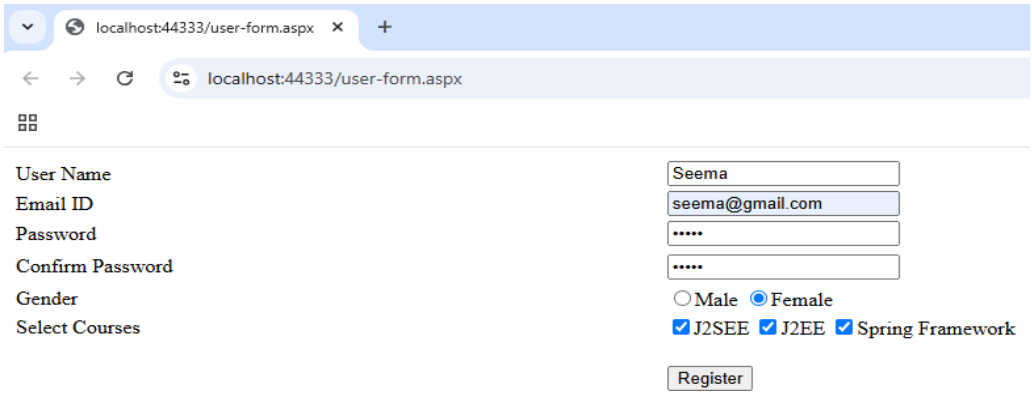
Source Code:

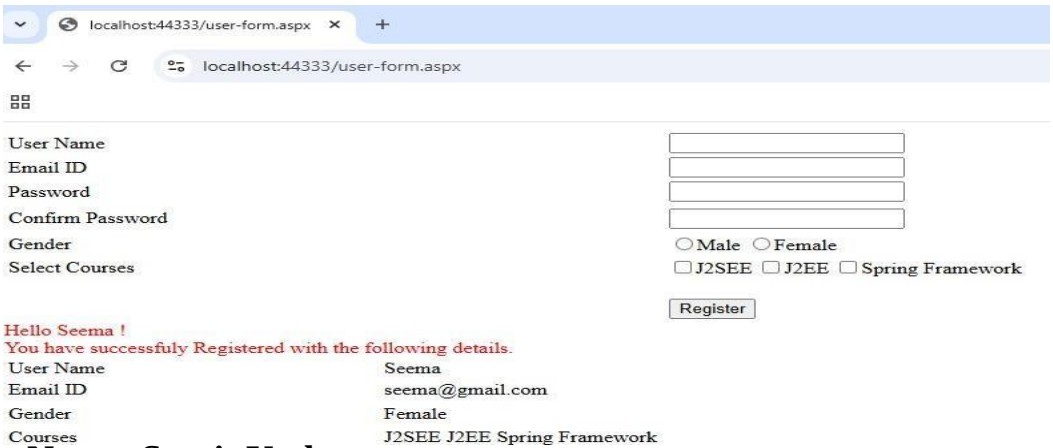
user-form.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace UserRegistration1
{

    public partial class user_form : System.Web.UI.Page
    {

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



    protected void Button1_Click(object sender, EventArgs e)
        {

        message.Text = "Hello " + username.Text + " ! ";
            message.Text = message.Text + " <br/> You
have successfuly Registered with the following details.";
            ShowUserName.Text = username.Text; ShowEmail.Text
            = EmailID.Text;
            if (RadioButton1.Checked)
            {

                ShowGender.Text = RadioButton1.Text;
            }

            else ShowGender.Text = RadioButton2.Text; var
            courses = "";
            if (CheckBox1.Checked)
            {

                    courses = CheckBox1.Text + " ";
              }

                if (CheckBox2.Checked)
                {

                  courses += CheckBox2.Text + " ";
              }

                if (CheckBox3.Checked)
                {

                 courses += CheckBox3.Text;
              }

            ShowCourses.Text = courses;
            ShowUserNameLabel.Text = "User Name";
            ShowEmailIDLabel.Text = "Email ID";
```

```
ShowGenderLabel.Text
= "Gender"; ShowCourseLabel.Text
= "Courses"; username.Text = "";
            EmailID.Text = "";
            RadioButton
            1.Checked = false;
            RadioButton
            2.Checked = false;
            CheckBox1. Checked =
            false; CheckBox2.
            Checked = false;
            CheckBox3. Checked =
            false;
        } }}
```

Output:





Module 2: Creating a User Interface (Controls and Master Page)

Practical No.8

Aim: - Design a registration form with validation controls for mandatory fields, email format, and passwords.

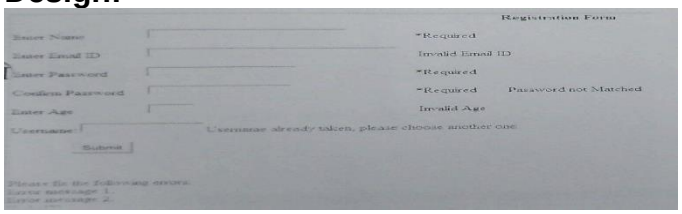
Source Code: -

WebForm1.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true" CodeBehind
"WebForm1.aspx.cs"
Inherits "ValidationControl Demo.
WebForm1" Unobtrusive
ValidationMode="None" %>
<!DOCTYPE html>
tml
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Labell" runat="server"
Font-Bold="True" Font-Kalic "False"
Text="Registration Form"></asp:Label>
<br />
<br/>
<asp:Label ID="Label2" runat="server"
Font-Bold "False" Text="Enter Name
"></asp:Label>

 &n
bsp;    &n
bsp;
<asp:TextBox ID="TextBox1"
runat="server"
Width="194px"></asp:TextBox>
<asp:RequiredField Validator
ID="RequiredFieldValidator1"
runat="server"
Control To Validate "TextBox1"
ErrorMessage-"*Required"></asp:Required
Field Vali
dator>
<br />
<br>
<asp:Label ID="Label3" runat="server"
Text="Enter Email ID"></asp:Label>
<asp:TextBox ID="TextBox2"
runat="server"
Width="245px"></asp:TextBox>
<asp:RegularExpression Validator
ID="RegularExpression Validator1"
runat="server"
ControlToValidate="TextBox2"
ErrorMessage "Invalid Email ID"
ValidationExpression
\w+([-]\w+)@\w+([-.]\w+

*\\w+([-.]\w+)"></asp:RegularExpressionVa
lidator
<br> <br/>
<asp:Label ID="Label4" runat="server"
Text "Enter Password"></asp:Label>

 & bsp;
<asp:TextBox ID="TextBox3"
runat="server"
Width="143px"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator2"
runat="server"
ControlToValidate "TextBox3"
ErrorMessage
Required"></asp:RequiredFieldVali
dator
<br/>
<br/>
<asp:Label ID="abel5" runat="server"
Text="Confirm Password"></asp:Label>

<asp:TextBox ID="TextBox4"
runat="server"
Width="138px"></asp:TextBox>
<asp:RequiredFieldValidator
ID="RequiredFieldValidator3"
runat="server"
ControlToValidate="TextBox4"
ErrorMessage"
Required"></asp:RequiredFieldVali dator>
```

```
<asp:Compare Validator
ID="Compare Validator1" runat="server"
ControlToCompare "TextBox3"
ControlToValidate="TextBox4"
ErrorMessage="Password not
Matched"></asp:Compare Validator>
<br />
<br/>
<asp:Label ID="Label6" runat="server"
Text "Enter Age"></asp:Label>
<asp:TextBox ID "TextBox5"
runat="server"
Width="41px"></asp:TextBox>
<asp:Range Validator ID="Range
Validator1" runat="server"
ControlToValidate="TextBox5"
ErrorMessage "Invalid Age" Maximum
Value="80"
Minimum Value="10"></asp:Range
Validator> <br>
<br />
<!--Custom validation for unique username
or email - ->
<label
for="txtUsername">Username:</label>
<asp:TextBox ID="txtUsername"

runat="server"
runat="server"></asp:TextBox>
<asp:CustomValidator ID="cvUsername"
Control To Validate="txtUsername"
OnServer Validate="ValidateUsername"
ErrorMessage="Username already taken.
piease choose another one"
ForeColor="Red"
Display "Dynamic" />
<br /><br />
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Submit"
/> <br/> <br/> <br />
<!-- Validation Summary ->>
<asp: ValidationSummary ID="vsSummary"
runat="server"
Header Text "Please fix the following
errors
:" ForeColor="Red" DisplayMode="List" />
runat="server"></asp:Label>
</div>
<asp:Label ID="Label7"
</form>
</body>
</html>
```

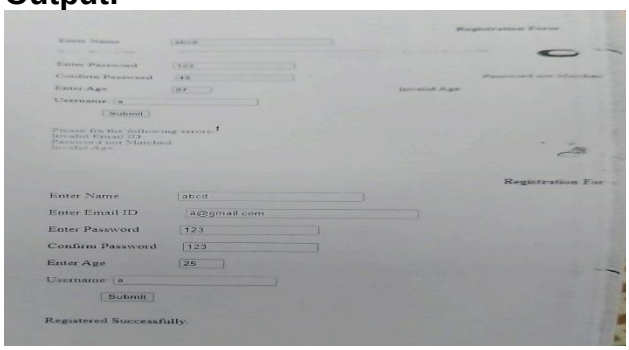
Design:



WebForm1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web; using System.Web.UI;
using System.Web.UI.WebControls
namespace ValidationControlDemo
public partial class WebForm
System.Web.ULPage
protected void Page Load(object sender,
EventArgs e) {
protected void Button1_Click(object
sender, EventArgs e)
1
Label 7. Text "Registered Successfully.";
protected void ValidateUsername(object
sender, Server ValidateEventArgs e)
// For demonstration, we simulate a
username check

// In a real-world scenario, this should check
the database for an existing username.
if (e.Value = "takenUsername")
{
e.IsValid false; // Custom validation failed
sername already taken)
}else {
// Custom validation succeeded
e.IsValid = true;
}
```

Output:



Practical No.9

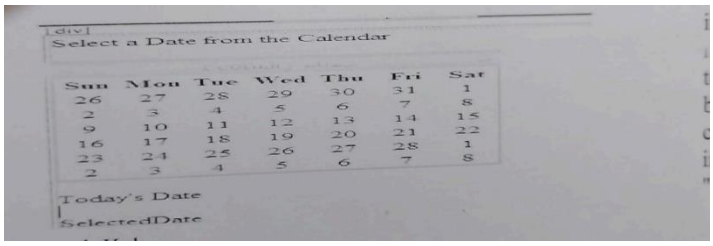
Aim: Create a web page with an interactive calendar control to select a date and display it in a label. Display Current Month's Calendar. Show Some image in the cell along with a message.

Source Code: -

Calendar.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true" CodeBehind
"Calendar.aspx.cs"
Inherits "CalendarControl Demo.Calendar
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"><title></title>
</head>
body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Labell" runat="server"
Text="Select a Date from the
Calendar"></asp:Label>
<br/>
<br>
<asp:Calendar ID="Calendar1"
runat="server" OnSelectionChanged
"Calendar1_SelectionChanged"
OnDayRender-"Calendar!
DayRender"></asp:Calendars
<br/>

<asp:Label ID="Label2" runat="server"
Text="Today's Date"></asp:Label>
<br>
<br/>
<asp:Label ID="Label3" runat="server"
Text="SelectedDate"></asp:Label>
</div>
</form>
</body>
</html>
```

Design:

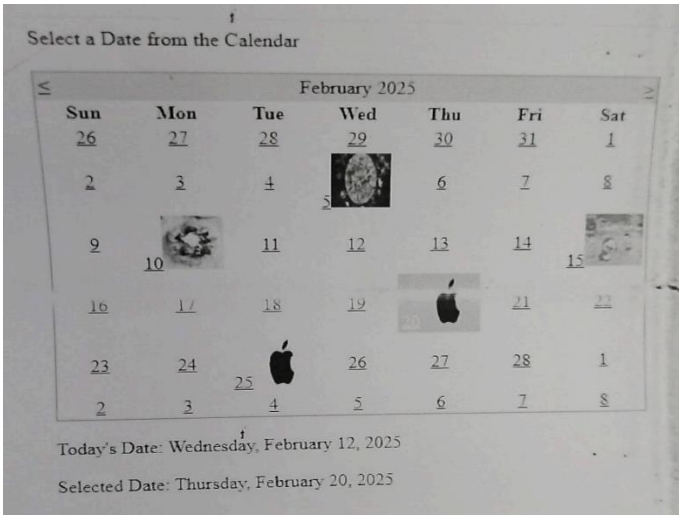


Calendar.aspx.cs:

```
using System,
using System.Collections.Generic;
using System.Linq:
using System. Web:
using System.Web.UI;
using System.Web.UI.WebControls
namespace CalendarControlDemo

public partial class Calendar:
System.Web.UI.Page
protected void Page Load(object sender,
EventArgs
c) if (!IsPostBack)
{
Calendarl Selected Date Date Time Today.
Label2.Text "Today's Date:"+
Date Time. Today. ToString("D");
{
{ protected void Calendar!
SelectionChanged(object
sender, EventArgs e)
Label2. Text "Today's Date:"+
Date Time. Today. ToString("D"); Inhall
Tata
Calendar {. SelectedDate. ToString("D");
{
protected void Calendar{_DayRender(object
sender,
DayRenderEventArgs e)
{ if (le.Day IsOther Month)
int day e.Day. Date. Day:
string imagePath":
string tooltip Text;
// Update the image paths and tooltip text as
needed
switch (day)

case 5:
imagePath/Images/download.jpg"; Use
tooltip Text "Holiday":
break;
case 10:
```

```
imagePath="-/Images/Mexiçan-Pizza-Thum
bnail-11.jpg"
tooltipText = "Meeting Day!"; break;
case 15:
imagePath =
"~/Images/anime-doraemon-wallpaper-previ
ew.jpg"; tooltipText = "Birthday";
break;
case 20:
imagePath = "~/Images/Apple-Logo.png";
tooltipText = "Event Day!";
break:
case 25:
imagePath = "~/Images/Apple-Logo.png",
tooltipText = "Fun day";
Orcan,
}
if (!string.IsNullOrEmpty(imagePath))
Add the image and tooltip text to the day
cell
e.Cell.ToolTip = tooltipText;
Image image = new Image

{
ImageUrl = imagePath,
Alternate Text = tooltipText,
Width = 50,
Height = 50
};
e.Cell.Controls.Add(image);
}} }}}
```

Output:



Practical No.10

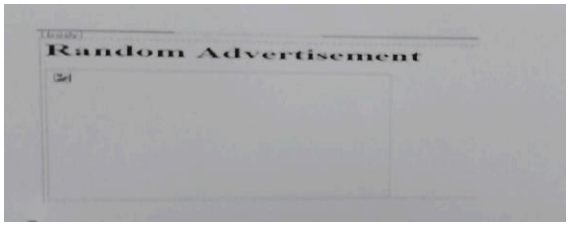
Aim: - Design an ASP.NET Application to Display Random Advertisements using ADRotator Control

Source Code: -

ADRotator.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind "ADRotator.aspx.cs"
Inherits="ADRotatorDemo.ADRotator" %>
<!DOCTYPE html>
<html lang="en">
<head runat="server">
<meta charset="utf-8" />
<title>Random Advertisement
Display</title>
</head>
<body>
Gorm id="form1" runat="server">
<div><h1>Random Advertisement</h1>
<!-- ADRotator Control to Display Ads -->
asp: ADRotator ID="AdRotator1"
runat="server"
AdvertisementFile="~/XMLFilel.xml"
Width="300" Height="250" />
</div></form></body></html>
```

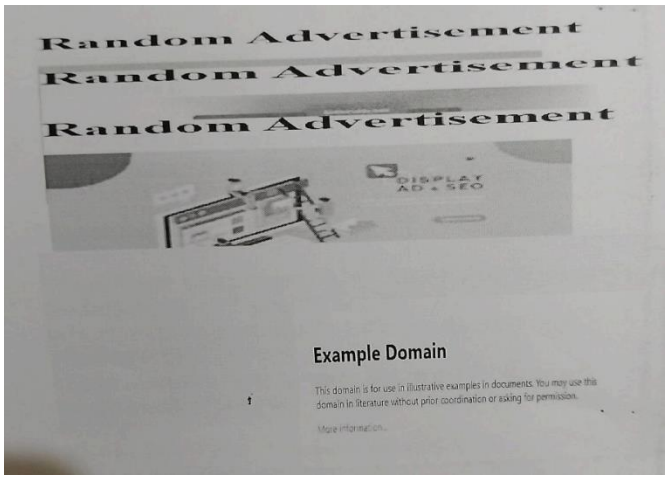
Design:



XMLFile1.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<Advertisements>
<Ad>
<NavigateUrl>http://example.com/ad1</Na
vigateUrl>
<ImageUrl>~/Images/ad1.jpg</ImageUrl>
<Alternate Text>Ad 1</AlternateText>
</Ad>
<Ad>
<ImageUrl>~/Images/ad2.jpg</ImageUrl>
<NavigateUrl>http://example.com/ad2</Na
vigateUrl>
<Alternate Text>Ad 2</AlternateText>
<Ad>
<ImageUrl>~/Images/ad3.jpg</ImageUrl>
<NavigateUrl>http://example.com/ad3</Na
vigateUrl>
<Alternate Text>Ad 3</Alternate Text>
</Ad>
</Advertisements>
```

Output:



When click on an ad image will display the particular website page which you given in url.

Practical No.11

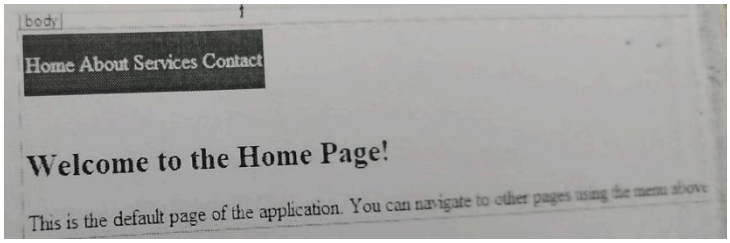
Aim: - Implement a navigation menu to link multiple pages of your web application.

Source Code: -

Default.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true" CodeBehind
"Default.aspx.cs"
Inherits "Navigation MenuDemo.Default"
%>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Navigation Menu Examples/title>
<style>
menu
Background-color: #333;
overflow: hidden;
.menu a
color: white;
padding: 14px 20px:
text-decoration: none;
float: left;
}
menu a:hover
{
background-color: #111;
</style></head>
body
form id="form1" runat="server">
<div><!- Menu Control -->
<asp:Menu ID="Menul" runat="server"
CssClass="menu"
Orientation="Horizontal">
<Items>
<asp:Menultem Text="Home"
NavigateUrl="~/Home.aspx" />

<asp:Menultem Text="About"
NavigateUrl="~/About.aspx" />
<asp:Menultem Text="Services"
NavigateUrl="~/Services.aspx" />
<asp:Menultem Text="Contact"
NavigateUrl="/Contact.aspx" />
</Items>
</asp:Menu>
<div style="padding-top: 20px;">
<h2>Welcome to the Home Page!<12>
<p>This is the default page of the
application. You can navigate to other pages
using the menu above.</p>
</div></div></form></body></html>
```

Design:

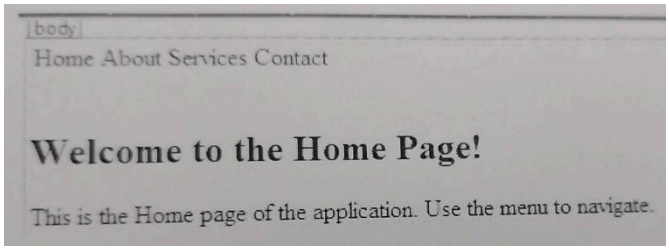


Home.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind "Home.aspx.cs"
Inherits-"Navigation MenuDemo.Home"
%>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head
runat="server"><title>Home</title></head>
<body>
<form id="form1" runat="server">
<div><!-- Menu Control -->
<asp:Menu ID="Menul" runat="server"
CssClass="menu"
Orientation="Horizontal">
<Items>
<asp:Menultem Text="Home"
NavigateUrl/Home.aspx" />
<asp:Menultem Text="About"
NavigateUrl="/About.aspx">
<asp:Menultem Text="Services"
NavigateUrl="/Services.aspx" />
asp:Menultem Text="Contact"
NavigateUrl="/Contact.aspx" />
```

```
<Items>
</asp:Menu>
<div style="padding-top: 20px;">
        <h2>Welcome to the Home Page!</h2>
        <p>This is the Home page of the
        application. Use the
        menu to navigate.</p>
        </div>div></form> </body></html>
```

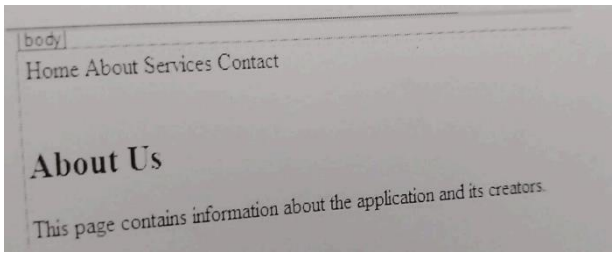
Design:



About.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind "About.aspx.cs"
Inherits "Navigation MenuDemo. About"
%>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head
runat="server"><title>About</title></head>
<body>
rm id="form1" runat="server">
<div><!- Menu Control -->
<asp:Menu ID="Menul" runat="server"
CssClass="menu" Orientationy
"Horizontal">
<Items>
/> NavigateUrl="~/About.aspx"
<asp:Menultem Text "Home"
NavigateUrl"~/Home.aspx" />
<asp:Menultem Text About"
<asp:Menultem Text="Services"
NavigateUrl="-/Services.aspx" />
<asp:Menultem Text="Contact"
NavigateUrl="~/Contact.aspx" />
</Items>
</asp:Menu>
<div style="padding-top: 20px;">
<h2>About Us</h2>
>This page contains information about the
application and its creators.</p>
</div></div></form></body></html>
```

Design:



Services.aspx:

```
%@ Page Language="C#"
AutoEventWireuptrue
Code Behind "Services.aspx.co
Inherits "Navigation MenuDemo Services">
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Services</title>
</head
<body>

<form id="form1" runat="server">
<div><!- Menu Control->
<asp:Menu ID "Menul" runat="server"
CssClass"menu" Orientation "Horizontal">
<Items
<asp:Menultem Text="Home"
NavigateUrl/Home.aspx" />
<asp:Menultem Text"About"
NavigateUrl"~/About.aspx">
<asp:Menultem Text="Services"
NavigateUrl="/Services.aspx" />
<asp:Menultem Text="Contact"
NavigateUrl="/Contact.aspx">
</items>
</asp:Menu>
<div style="padding-top: 20px;">
<h2>Our Services</h2>
<p>This page outlines the services provided
by our
application.</p>
</div></div>form></body></html>
```

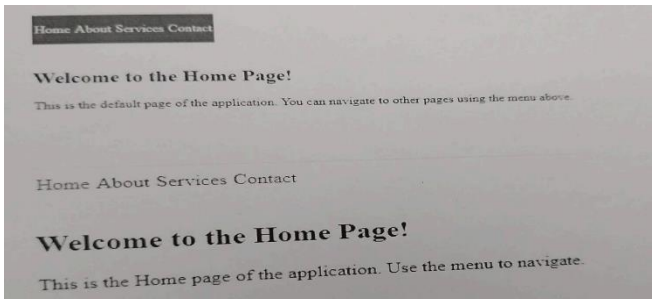
Design:

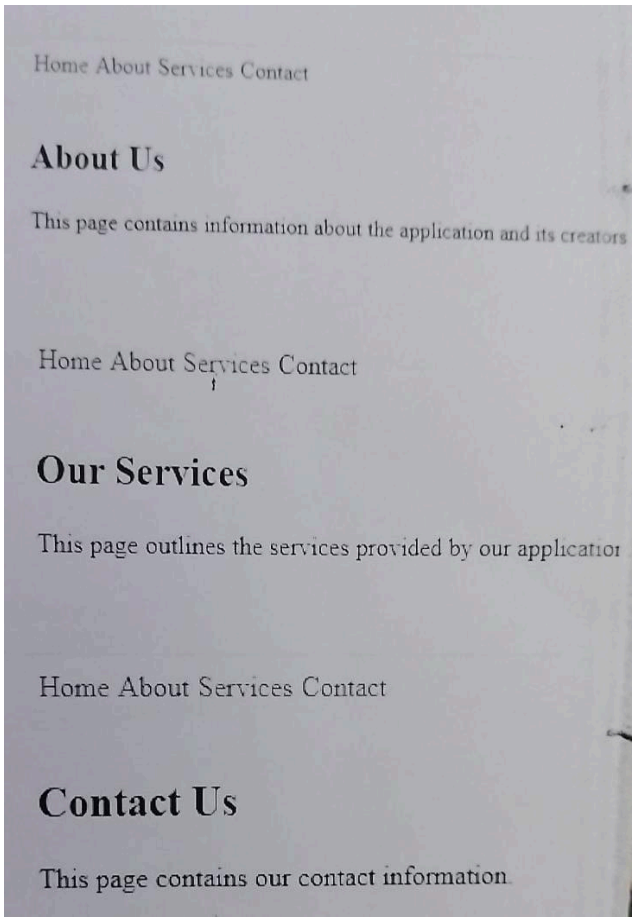
Contact.aspx:

```
<%@ Page Language="C#"
AutoEventWireup "tr
CodeBehind "Contact.aspx.cs"
Inherits "Navigation MenuDemo Contact
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Contact</title>
</head>
<body>
<form id="form1" runat="server">
<div><!-- Menu Control -->
<asp:Menu ID="Menu1" runat="server"
CssClass="menu"
Orientation="Horizontal">
<Items>
<asp:Menultem Text="Home"
NavigateUrl="~/Home.aspx" />
<asp:Menultem Text="About"
NavigateUrl="~/About.aspx" />
<asp:Menultem Text="Services"

NavigateUrl="~/Services.aspx" />
<asp:Menultem Text="Contact"
NavigateUrl="~/Contact.aspx" />
</Items>
</asp:Menu>
div style="padding-top: 20px;">
<h2>Contact Us</h2>
<p>This page contains our contact
information.</p>
</div></div></form></body></html>
```

Design:

Output:





Practical No.12

Aim:- Develop a file upload application that saves the uploaded file and displays its name and size.

Source Code: -

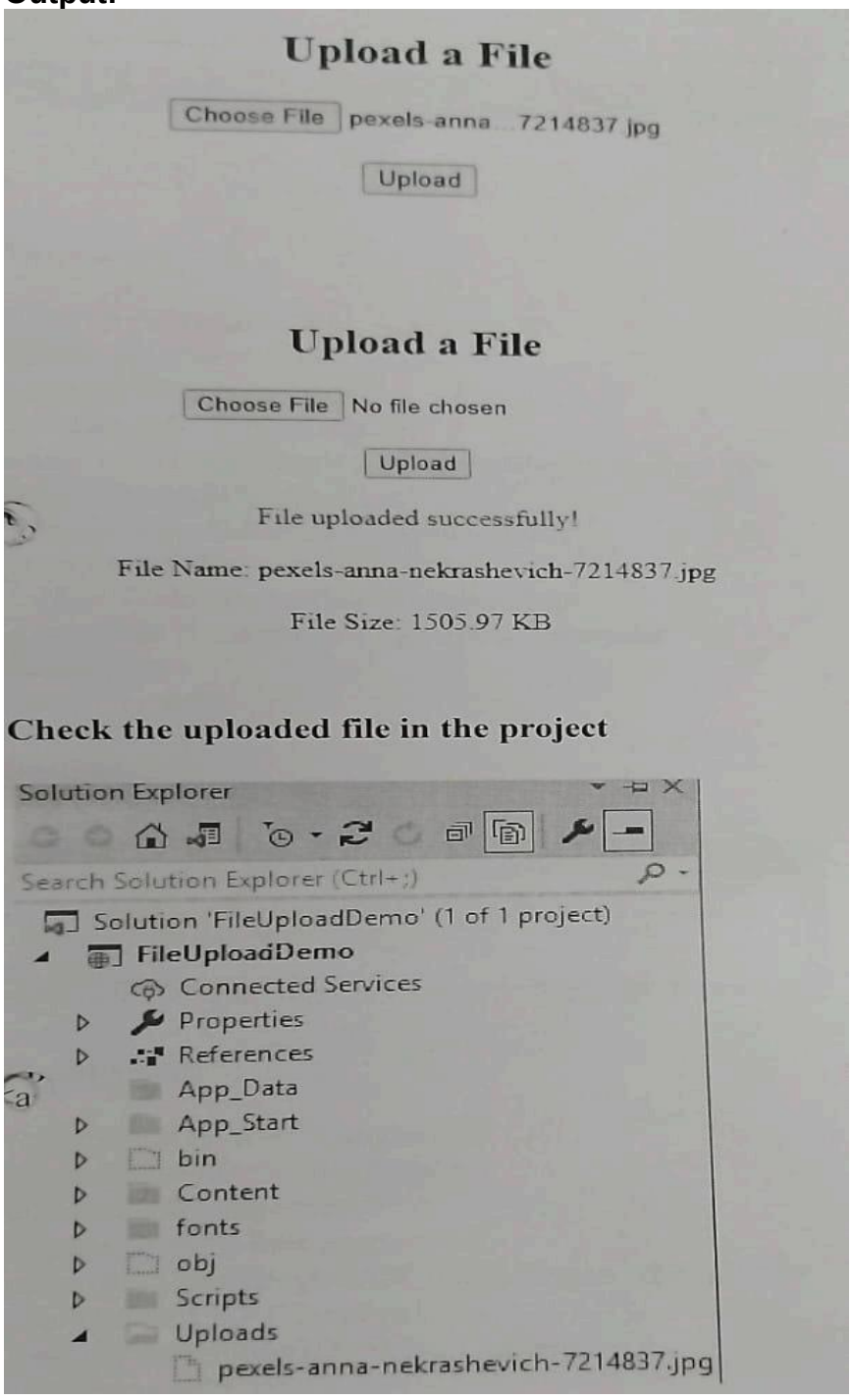
FileUpload.aspx:

```
<%@ Page Language "C#"
AutoEventWireup="true" CodeBehind
"FileUpload.aspx.cs"
Inherits "FileUploadDemo.FileUpload" %>
<!DOCTYPE html>
<html lang="en">
<head runat="server">
<title>File Upload</title>
</head>
<body>
form id="form1" runat="server">
<div style="text-align:center;">
<h2>Upload a File</h2>
<asp:FileUpload ID="FileUpload1"
runat="server" />
<br /><br />
<asp:Button ID="btnUpload" runat="server"
Text="Upload" OnClick="btnUpload_Click"
/>
<br /><br />
<asp:Label ID="lblMessage" runat="server"
ForeColor="Green"></asp:Label>
<br/><br/>
<asp:Label ID="IblFileName"
runat="server"></asp:Label>
<br /><br />
<asp:Label ID="IbIFileSize"
runat="server"></asp:Label>
</div></form></body></html>
```

FileUpload.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.IO;
public partial class FileUpload:
{
protected void Page_Load(object sender,
EventArgs {}
protected void btnUpload_Click(object
sender,
EventArgs e)
    if (FileUpload1.HasFile)
    (try
    // Define upload path
string uploadFolder
Server.MapPath("~/Uploads/");
// Ensure the upload directory exists
if (!Directory Exists(uploadFolder))
{ Directory.CreateDirectory(uploadFolder);
// Get file details
string fileName=
Path.GetFileName(FileUpload1.FileName):
string filePath Path. Combine(uploadFolder,
fileName);
double fileSizeKB
FileUpload1.FileContent.Length
/1024.0;
// Save the file to the server
FileUpload1.SaveAs(filePath);
// Display uploaded file details
IblMessage.Text="File uploaded
successfully!"; IblFileName.Text "File
Name:" + fileName;
IblFileSize. Text "File Size:" +
fileSizeKB.ToString("F2")+" KB":
} catch (Exception ex)
lblMessage.ForeColor = System.Drawing.
Color.Red; IblMessage. Text "Error: " +
ex.Message;
}}
else
{
IblMessage ForeColor
System.Drawing.Color.Red; IblMessage.
Text "Please select a file to upload.";
} }}}
```

Output:


Upload a File
Choose File
pexels-anna...7214837.jpg
Upload
Upload a File
Choose File
No file chosen
Upload
File uploaded successfully!
File Name: pexels-anna-nekrashevich-7214837.jpg
File Size: 1505.97 KB
Check the uploaded file in the project
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'FileUploadDemo' (1 of 1 project)
FileUploadDemo
Connected Services
Properties
References
App_Data
App_Start
bin
Content
fonts
obj
Scripts
Uploads
pexels-anna-nekrashevich-7214837.jpg

Practical No.13

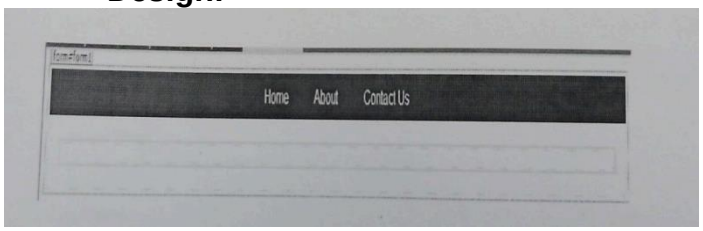
Aim: Create a Master Page for a website and add different content pages for Home, About, and Contact Us.

Source Code: -

Site.Master:

```
<%@ Master Language="C#"
AutoEventWireup="true
CodeBehind "Site.master.cs"
Inherits="MasterPagewithContentPages.Site
Master" %>
<!DOCTYPE html>
<html lang="en">
<head runat="server">
<title>My Website</title>
<style>
body { font-family: Arial, sans-serif; }
navbar { background-color: #333; padding:
10px;
text-align: center; } /
navbar a { color: white; padding: 14px
20px;
text-decoration: none; }
navbar a:hover{ background-color:
#575757; }
.content { padding: 20px; }
.footer { background-color: #333; color:
white;
text-align: center; padding: 10px; position:
fixed;
bottom: 0; width: 100%; }
</style></head>
<body>
<form id="form1" runat="server">
<div class="navbar">
<a href="Home.aspx">Home</a>
<a href="About.aspx">About</a>
<a href="Contact.aspx">Contact Us</a>
</div>
<div class="content">
<asp:ContentPlaceHolder
ID="MainContent"
runat="server"></asp:ContentPlaceHolder>
</div>
<div class="footer">copy; 2025 My
Website. All rights reserved.
</div></form></body></html>
```

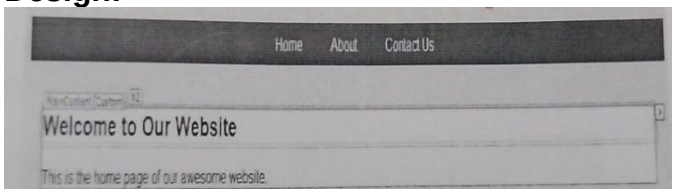
Design:



Home.aspx:

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="Home.aspx.cs"
Inherits="MasterPagewithContentPages.Ho
me" %>
<asp:Content ID="Content1"
ontentPlaceHolderID="MainContent"
runat="server">
<h2>Welcome to Our Website</h2>
<p>This is the home page of our awesome
website.</p>
</asp:Content>
```

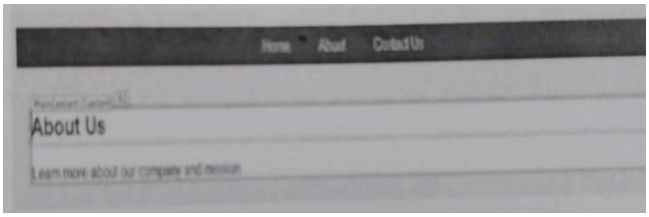
Design:



About.aspx:

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind "About.aspx.cs"
Inherits="MasterPagewithContentPages.Abo
ut1" %>
<asp:Content ID="Content1"
ContentPlaceHolderID="MainContent"
runat="server">
<h2>About Us</h2>
<p>Learn more about our company and
mission.</p>
</asp:Content>
```

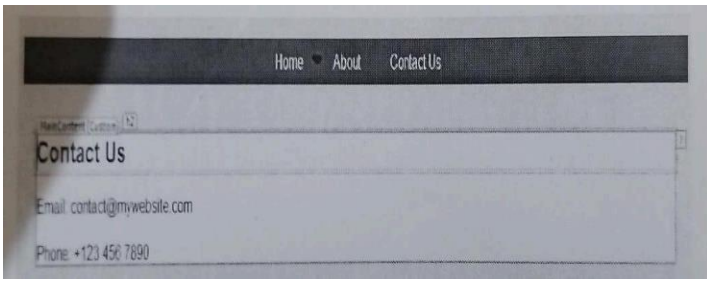
Design:



Contact.aspx:

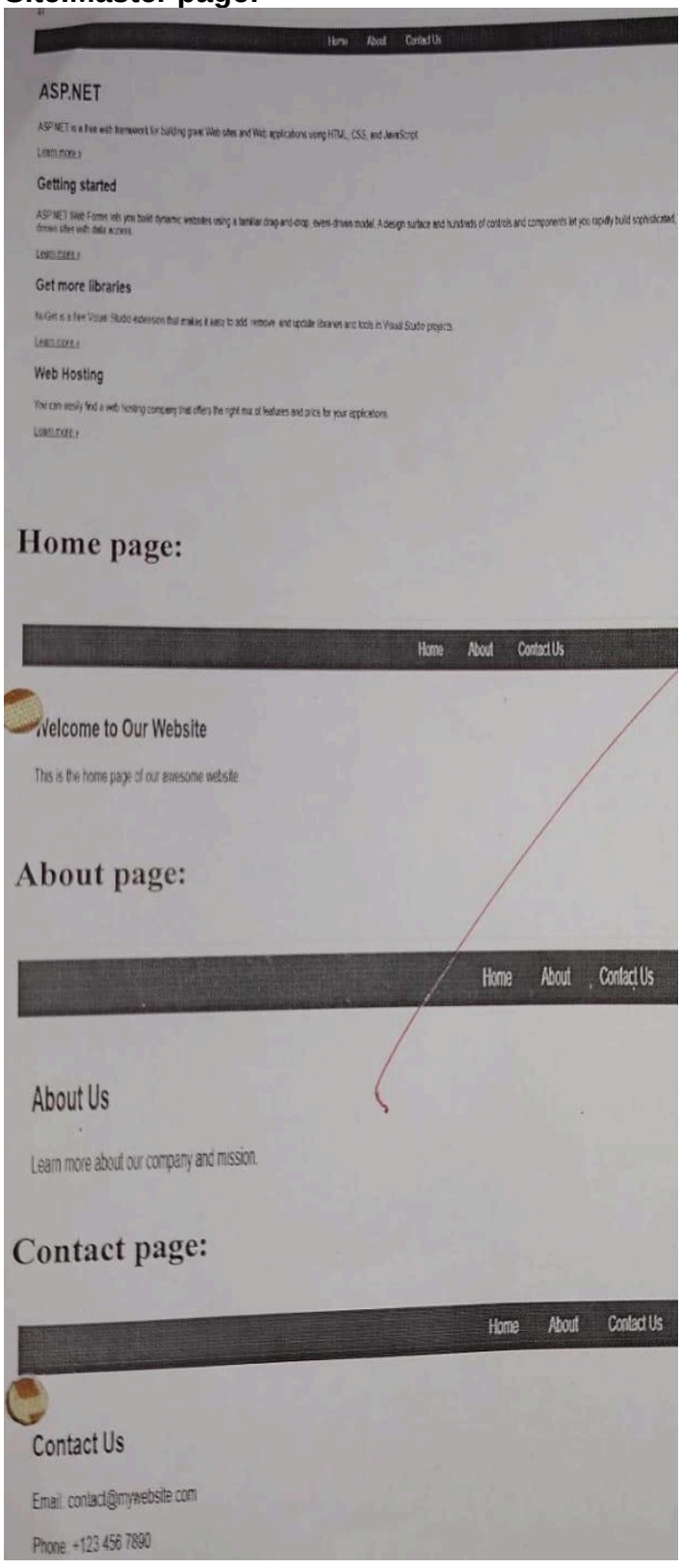
```
<%@ Page Title="Contact" Language="C#"
MasterPageFile="~/Site.Master"
AutoEventWireup="true"
CodeBehind="Contact.aspx.cs"
Inherits="MasterPagewithContentPages.Con
tact" %>
<asp:Content ID="BodyContent"
ContentPlaceHolderID="MainContent"
runat="server">
<h2>Contact Us</h2>
<p>Email: contact@mywebsite.com</p>
<p>Phone: +123 456 7890</p>
</asp:Content>
```

Design:



Output:

Site.Master page:



Name: Sumit Kadam **Roll no : 16**

Module 3

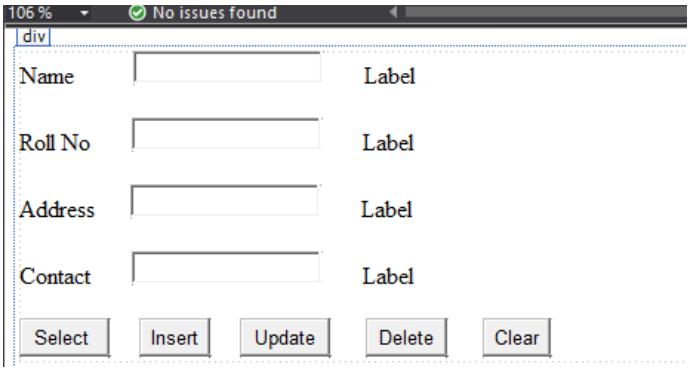
Practical: 14

Aim : Develop a Web Application to demonstrate a connection oriented architecture . Create StudentInfo (RollNo,Name,Address,Phone no)table and add,delete,update and select records.

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="StudentInfo.aspx.cs"
Inherits="StudentInfo_connected_architect
ure.StudentInfo" %>
<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server"
Text="Name"></asp:Label>
       
;  
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
      
<asp:Label ID="Labelname" runat="server"
Text="Label"></asp:Label>
<br />
<br />
<asp:Label ID="Label3" runat="server"
Text="Roll No"></asp:Label>
      
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
      
<asp:Label ID="Labelrollno"
runat="server" Text="Label"></asp:Label>
<br />
<br />
<asp:Label ID="Label5" runat="server"
Text="Address"></asp:Label>
     
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
      
<asp:Label ID="Labeladd" runat="server"
Text="Label"></asp:Label>
<br />
<br />
<asp:Label ID="Label7" runat="server"
Text="Contact"></asp:Label>
      
<asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
      
<asp:Label ID="Labelcont" runat="server"
Text="Label"></asp:Label>
<br />
<br />
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Select " />
    
<asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="Insert" />
    
<asp:Button ID="Button3" runat="server"
Text="Update" OnClick="Button3_Click1" />
     
<asp:Button ID="Button4" runat="server"
Text="Delete" OnClick="Button4_Click1" />
     
<asp:Button ID="Button5" runat="server"
Text="Clear" OnClick="Button5_Click1" />
   
</div>
</form>
</body>
</html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

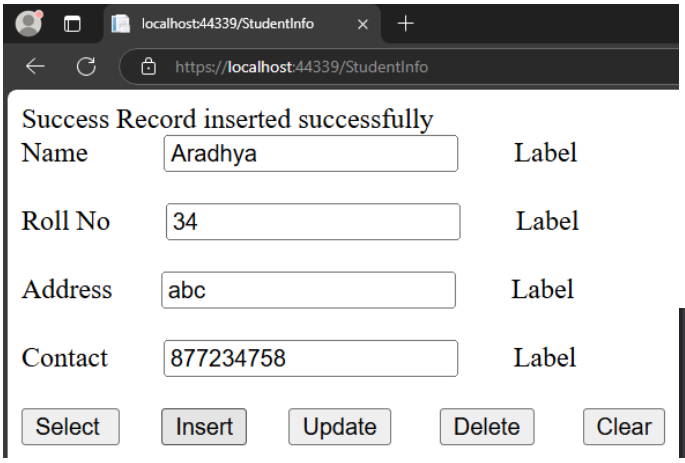
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

namespace
StudentInfo_connected_architecture
{
public partial class StudentInfo :
System.Web.UI.Page
{
SqlConnection con = new SqlConnection();
SqlDataReader dr;
SqlCommand cmd;
protected void Page_Load(object sender,
EventArgs e)
{
con.ConnectionString = "Data Source =
(localdb)\\MSSQLLocalDB; Initial Catalog
=StudentData; Integrated Security = True;
Connect Timeout = 30; Encrypt = False;
TrustServerCertificate = False;
ApplicationIntent = ReadWrite;
MultiSubnetFailover = False";
con.Open();
Response.Write("Success");

}

protected void Button2_Click(object
sender, EventArgs e)
{
string nm, ad, phn;
int roll;
nm = TextBox1.Text;
roll = Convert.ToInt16(TextBox2.Text);
ad = TextBox3.Text;
phn = TextBox4.Text;
string qry = "insert into student(sname,
sroll, sadd, scont) values ('" + nm +
"'," + roll + ",'" + ad + "','" + phn +
"')";
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("\n Record inserted
successfully");

}

protected void Button1_Click(object
sender, EventArgs e)
{
int roll =
Convert.ToInt16(TextBox2.Text);
cmd = new SqlCommand("select * from
student where sroll='" + roll + "'",
con);
dr = cmd.ExecuteReader();
if (dr.Read())
{
Labelname.Text = "Name: " +
dr[0].ToString();
Labelrollno.Text = "Roll No: " +
dr[1].ToString();
Labeladd.Text = "Address: " +
dr[2].ToString();
Labelcont.Text = "Contact: " +
dr[3].ToString();
}
dr.Close();
Response.Write("\n Record selected.");
}


protected void Button3_Click1(object
sender, EventArgs e)
{
string nm, ad, phn;
int roll;
```

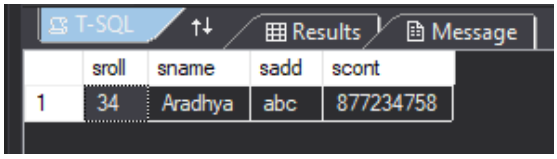
```
nm = TextBox1.Text;
roll = Convert.ToInt16(TextBox2.Text);
ad = TextBox3.Text;
phn = TextBox4.Text;
string qry = "update student set scont='"
+ phn + "' where sroll=" + roll;
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("\n Record updated
successfully");
}

protected void Button4_Click1(object
sender, EventArgs e)
{
int roll;
roll = Convert.ToInt16(TextBox2.Text);
string qry = "delete from student where
sroll=" + roll;
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("\n Record deleted
successfully");
}

protected void Button5_Click1(object
sender, EventArgs e)
{
TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";
}
}
}
```

Sql Queries:

```
Create database StudentData;
use StudentData;
Create Table student
(
sroll varchar(200),
sname varchar(200),
sadd varchar(200),
scont varchar(200),
);
select * from student;
```

Output: Insert :





Select :

Success Record selected.

Name: Aradhya — Name: 34

Roll No: 34 — Roll No: Aradhya

Address: abc — Address: abc

Contact: 877234758 — Contact: 877234758

Select | Insert | Update | Delete | Clear

Update :

Success Record updated successfully

Name: Aradhya — Name: 34

Roll No: 34 — Roll No: Aradhya

Address: abc — Address: abc

Contact: 6666 — Contact: 877234758

Select | Insert | Update | Delete | Clear

Success Record selected.

Name: Aradhya — Name: 34

Roll No: 34 — Roll No: Aradhya

Address: abc — Address: abc

Contact: — Contact: 6666

Select | Insert | Update | Delete | Clear

Delete :

Success Record deleted successfully

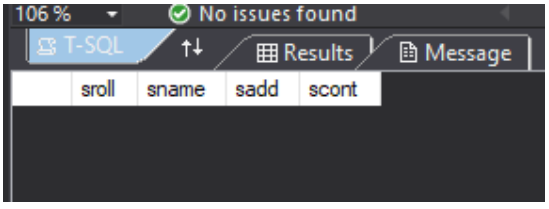
Name: Aradhya — Name: 34

Roll No: 34 — Roll No: Aradhya

Address: abc — Address: abc

Contact: — Contact: 6666

Select | Insert | Update | Delete | Clear



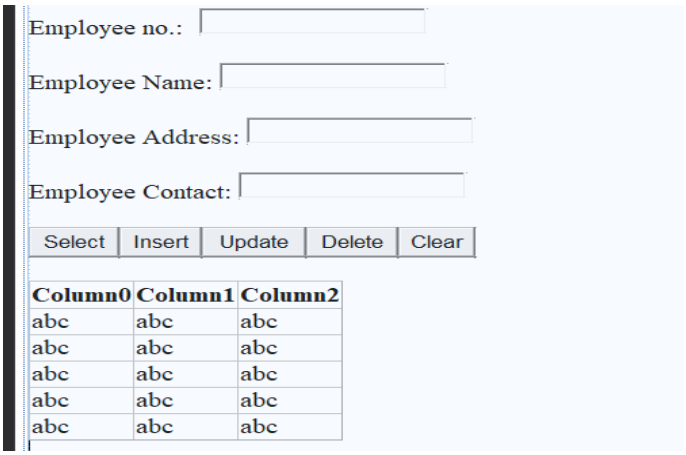
Practical: 15

Aim : Develop a Web Application to demonstrate a disconnected architecture. Create Employee (empno, empname, empaddress, empphoneno) table and add, delete, update and select records.

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Prac15_disconnected.WebForm1"
%>

<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<br />
<br />
<asp:Label ID="Label1" runat="server"
Text="Employee no."></asp:Label> :
<asp:TextBox ID="TextBox1" runat="server"
style="margin-left: 7px"></asp:TextBox>
<br />
<br />
<asp:Label ID="Label3" runat="server"
Text="Employee Name"></asp:Label> :
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Label ID="Label5" runat="server"
Text="Employee Address"></asp:Label> :
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Label ID="Label7" runat="server"
Text="Employee Contact"></asp:Label> :
<asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Select" />
<asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="Insert" />
<asp:Button ID="Button3" runat="server"
Text="Update" OnClick="Button3_Click1" />
<asp:Button ID="Button4" runat="server"
Text="Delete" OnClick="Button4_Click1" />
<asp:Button ID="Button5" runat="server"
Text="Clear" OnClick="Button5_Click1" />
<br />
<br />
<asp:GridView ID="GridView1"
runat="server" AutoGenerateColumns="True"
></asp:GridView>
<br />
<br />
</div>
</form>
</body>
</html>
```

Design: Aspx file :



Name: Sumit Kadam **Roll no : 16**

Code behind file : .aspx.cs :

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

namespace Prac15_disconnected
{
public partial class WebForm1 :
System.Web.UI.Page
{
SqlConnection sqlcon = new
SqlConnection();
SqlCommand sqlcmd = new SqlCommand();
void Connection()
{
sqlcon.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=EmployeeDB;Integrated
Security=True";
sqlcmd.Connection = sqlcon;
sqlcon.Open();
}
void populate()
{
SqlDataAdapter sda = new
SqlDataAdapter("Select * from emptable",
sqlcon);
sqlcon.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=EmployeeDB;Integrated
Security=True";
sqlcon.Open();
DataSet ds = new DataSet();
sda.Fill(ds);
GridView1.DataSource = ds;
GridView1.DataBind();
sqlcon.Close();
}
protected void Page_Load(object sender,
EventArgs e)
{
if (!IsPostBack)
{
populate();
}
}


protected void Button1_Click(object
sender, EventArgs e)
{
Connection();
string Select_Query = "SELECT * FROM
emptable WHERE empno='" +
TextBox1.Text.Trim() + "'";
SqlDataAdapter sda = new
SqlDataAdapter(Select_Query, sqlcon);
DataSet ds = new DataSet();
sda.Fill(ds);
GridView1.DataSource = ds;
GridView1.DataBind();
sqlcon.Close();
}



protected void Button2_Click(object
sender, EventArgs e)
{
Connection();
string Insert_Query = "INSERT INTO
emptable(empno, empname, empaddress,
empcontactno) VALUES('" +
TextBox1.Text.Trim() + "', '" +
TextBox2.Text.Trim() + "', '" +
TextBox3.Text.Trim() + "', '" +
TextBox4.Text.Trim() + "')";
sqlcmd.CommandText = Insert_Query;
sqlcmd.ExecuteNonQuery();
Response.Write("Record Inserted
Successfully");
sqlcon.Close();
populate();
}

protected void Button3_Click1(object
sender, EventArgs e)
{
Connection();
string Update_Query = "UPDATE emptable
SET empname='" + TextBox2.Text.Trim() +
"', empaddress='" + TextBox3.Text.Trim()
+ "', empcontactno='" +
TextBox4.Text.Trim() + "' WHERE empno='"
+ TextBox1.Text.Trim() + "'";
sqlcmd.CommandText = Update_Query;
sqlcmd.ExecuteNonQuery();
Response.Write("Record Updated
Successfully");
sqlcon.Close();
populate();
}

protected void Button4_Click1(object
sender, EventArgs e)
{
Connection();
string Delete_Query = "DELETE FROM
emptable WHERE empno='" +
TextBox1.Text.Trim() + "'";
```

```
sqlcmd.CommandText = Delete_Query;
sqlcmd.ExecuteNonQuery();
Response.Write("Record Deleted
Successfully");
sqlcon.Close();
populate();
}

protected void Button5_Click1(object
sender, EventArgs e)
{
TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";
}

}
}
```

Sql Queries:

```
CREATE DATABASE EmployeeDB;
USE EmployeeDB;

CREATE TABLE emptable (
[empno]     INT           NOT NULL,
[empname]       VARCHAR (50) NULL,
[empaddress]    VARCHAR (50) NULL,
[empcontactno] VARCHAR (50) NULL
);
select * from emptable;
```

Output:

Insert :

Record Inserted Successfully

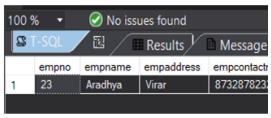
Employee no. : 23

Employee Name : Aradhya

Employee Address : Virar

Employee Contact : 8732878232

Select Insert Update Delete Clear



Update :

Record Updated Successfully

Employee no. : 23

Employee Name : Aradhya .P

Employee Address : Virar W

Employee Contact : 9077745221

Select Insert Update Delete Clear

empno	empname	empaddress	empcontactno
23	Aradhya .P	Virar W	9077745221

Select :

Employee no. : 23

Employee Name :

Employee Address :

Employee Contact :

Select Insert Update Delete Clear

empno	empname	empaddress	empcontactno
23	Aradhya	Virar	8732878232

Delete :

Record Deleted Successfully

Employee no. : 23

Employee Name :

Employee Address :

Employee Contact :

Select Insert Update Delete Clear

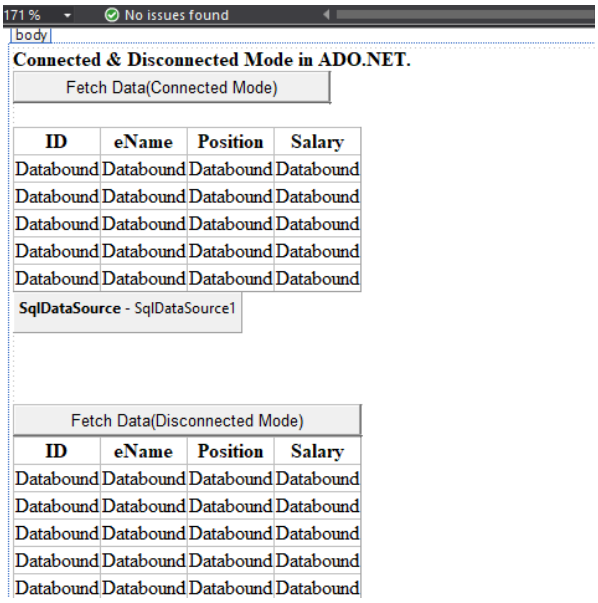
Practical: 16

Aim : Create a web application to connect to a database using ADO.NET and display data in a GridView.(connected and disconnected).

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="DataGridView.aspx.cs"
Inherits="Prac16.DataGridView" %>

<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<strong>Connected &amp;amp; Disconnected Mode
in ADO.NET.</strong><br />
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Fetch
Data(Connected Mode)" />
<br />
<br />
<asp:GridView ID="GridView1"
runat="server"
AutoGenerateColumns="False"
DataKeyNames="ID" >
<Columns>
<asp:BoundField DataField="ID"
HeaderText="ID" InsertVisible="False"
ReadOnly="True" SortExpression="ID" />
<asp:BoundField DataField="eName"
HeaderText="eName" SortExpression="eName"
/>
<asp:BoundField DataField="Position"
HeaderText="Position"
SortExpression="Position" />
<asp:BoundField DataField="Salary"
HeaderText="Salary"
SortExpression="Salary" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:EmployeeDBConnectionStr
ing %>" SelectCommand="SELECT * FROM
[Employees3]"></asp:SqlDataSource>
<br />
<br />
<br />
<asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="Fetch
Data(Disconnected Mode)" />
<br />
<asp:GridView ID="GridView2"
runat="server"
AutoGenerateColumns="False"
DataKeyNames="ID" >
<Columns>
<asp:BoundField DataField="ID"
HeaderText="ID" InsertVisible="False"
ReadOnly="True" SortExpression="ID" />
<asp:BoundField DataField="eName"
HeaderText="eName" SortExpression="eName"
/>
<asp:BoundField DataField="Position"
HeaderText="Position"
SortExpression="Position" />
<asp:BoundField DataField="Salary"
HeaderText="Salary"
SortExpression="Salary" />
</Columns>
</asp:GridView>
<br />
<br />
</div>
</form>
</body>
</html>
```

Design: Aspx file



Code behind file : .aspx.cs :

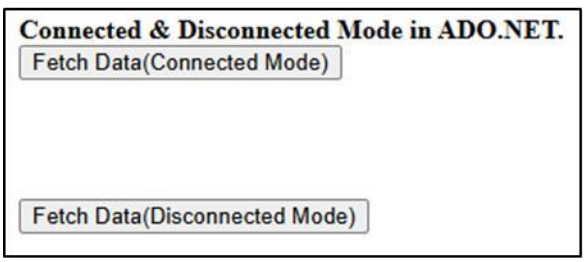
```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Web.UI;

namespace Prac16
{
public partial class DataGridView :
System.Web.UI.Page
{
private string connectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=EmployeeDB;Integrated
Security=True";

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

protected void Button1_Click(object
sender, EventArgs e)
{
using (SqlConnection conn = new
SqlConnection(connectionString))
{
string query = "SELECT * FROM
Employees2";
SqlCommand cmd = new SqlCommand(query,
conn);

conn.Open();
using (SqlDataReader reader =
cmd.ExecuteReader())
{
GridView1.DataSource = reader;
GridView1.DataBind();
}
}
}

protected void Button2_Click(object
sender, EventArgs e)
{
using (SqlConnection conn = new
SqlConnection(connectionString))
{
string query = "SELECT * FROM
Employees2";
SqlDataAdapter adapter = new
SqlDataAdapter(query, conn);
DataSet ds = new DataSet();

adapter.Fill(ds);

GridView2.DataSource = ds.Tables[0];
GridView2.DataBind();
}}}}
```

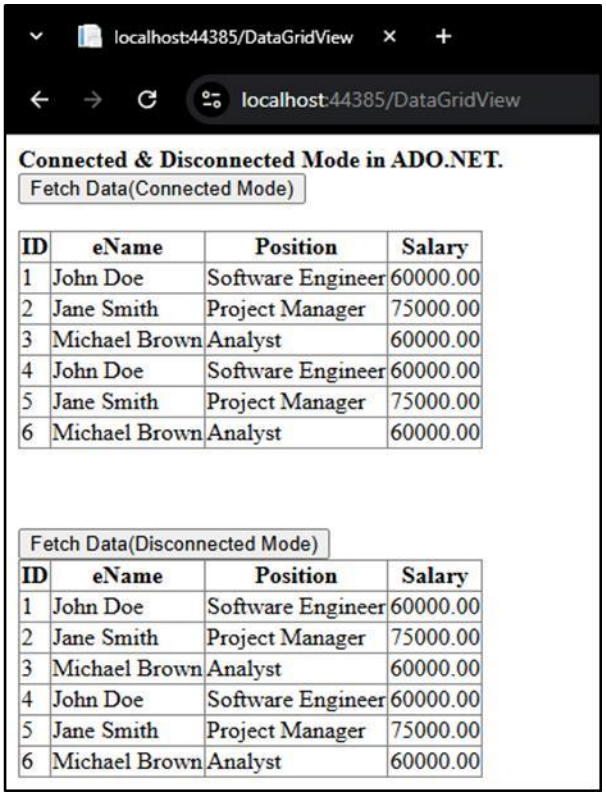
Sql Queries:

```
CREATE DATABASE EmployeeDB;
USE EmployeeDB;

Create Table Employees3
(
ID INT PRIMARY KEY IDENTITY,
```

```
eName NVARCHAR(100),
Position NVARCHAR(100),
Salary DECIMAL(10,2)
);

INSERT INTO Employees3 (eName, Position, Salary)
VALUES
('John Doe', 'Software Engineer', 60000.00),
('Jane Smith', 'Project Manager', 75000.00),
('Michael Brown', 'Analyst', 60000.00);


select * from Employees3;
```

Output:





Practical: 17

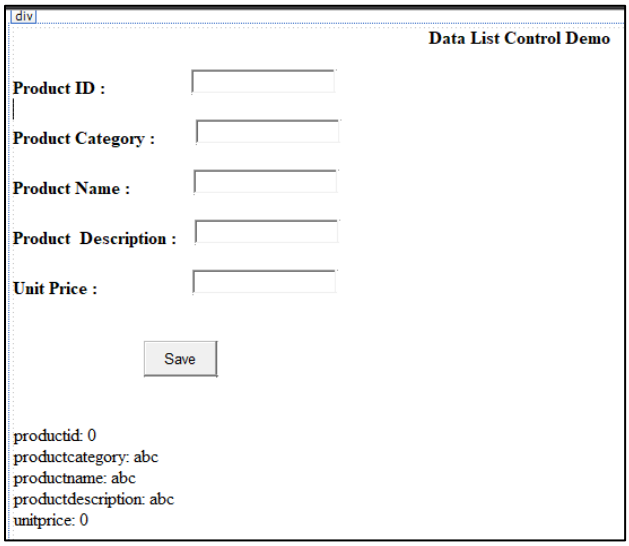
Aim : Design a Web Application to demonstrate Data Bound Control – DataList.

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="DataList.aspx.cs"
Inherits="Prac17.DataList" %>

<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
       
;      &nbs
p;      &nb
sp;      &n
bsp;      &
nbsp;      
       
;      &nbs
p;      &nb
sp;      &n
bsp;      &
nbsp;      
       
; <strong>   Data List Control
Demo<br />
<br />
Product ID
:      &nbs
p;      &nb
sp;     </strong>
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
<strong>Product Category
:      &nbs
p;  </strong>
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
<br />
<strong>Product Name
:      &nbs
p;      
</strong>
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
<strong>Product  Description
:    </strong>
<asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
<br />
<br />
<strong>Unit Price
:      &nbs
p;      &nb
sp;      
</strong>
<asp:TextBox ID="TextBox5"
runat="server"></asp:TextBox>
<br />
<br />
<br                                    />
       
;      &nbs
p;      &nb
sp;      &n
bsp;
<asp:Button ID="Button1" runat="server"
Height="32px" OnClick="Button1_Click"
Text="Save" Width="65px" />
<br />
<br />
<asp:DataList ID="DataList1"
runat="server"
DataSourceID="SqlDataSource1">
<ItemTemplate>
productid:
<asp:Label ID="productidLabel"
runat="server" Text='<%#
Eval("productid") %>' />
<br />
productcategory:
<asp:Label ID="productcategoryLabel"
runat="server" Text='<%#
Eval("productcategory") %>' />
<br />
productname:
<asp:Label ID="productnameLabel"
runat="server" Text='<%#
Eval("productname") %>' />
<br />
productdescription:
```

```
<asp:Label ID="productdescriptionLabel"
runat="server" Text='<%#
Eval("productdescription") %>' />
<br />
unitprice:
<asp:Label ID="unitpriceLabel"
runat="server" Text='<%#
Eval("unitprice") %>' />
<br />
<br />
</ItemTemplate>
</asp:DataList>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:ProductmasterConnection
String %>" SelectCommand="SELECT * FROM
[Productmaster]"></asp:SqlDataSource>
<br />
</div>
</form>
</body>
</html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

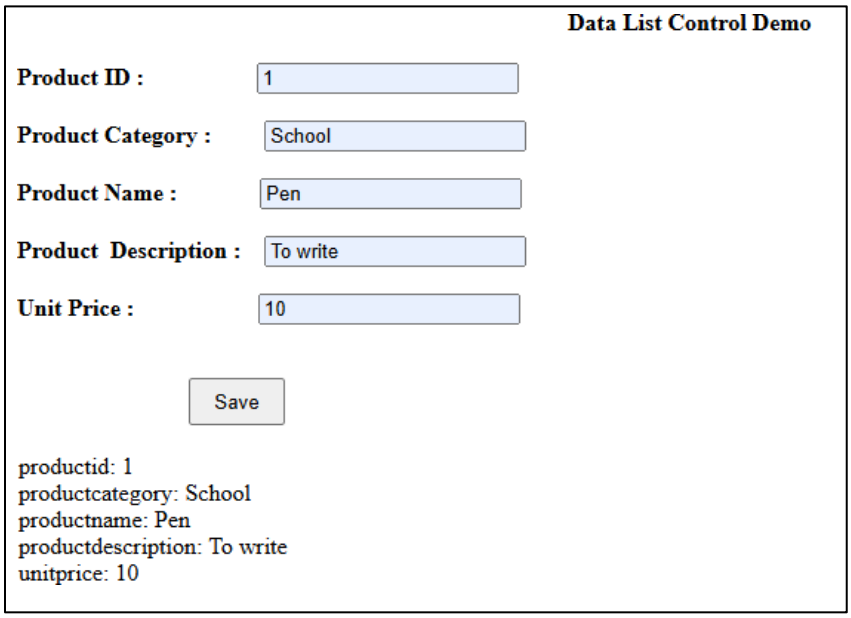
```
using System;
using System.Data.SqlClient;
using System.Web.UI;

namespace Prac17
{public partial class DataList :
System.Web.UI.Page
{
SqlConnection con = new SqlConnection();
SqlCommand cmd;

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

protected void Button1_Click(object
sender, EventArgs e)
{
int pid, unitp;
try
{
pid = Convert.ToInt32(TextBox1.Text);
string pcat = TextBox2.Text;
string pnm = TextBox3.Text;
string pdis = TextBox4.Text;
unitp = Convert.ToInt32(TextBox5.Text);

con.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=Productmaster;Integrated
Security=True";
con.Open();

string qry = "INSERT INTO Productmaster
(productid, productcategory, productname,
productdescription, unitprice) VALUES
(@pid, @pcat, @pnm, @pdis, @unitp)";
cmd = new SqlCommand(qry, con);
cmd.Parameters.AddWithValue("@pid", pid);
cmd.Parameters.AddWithValue("@pcat",
pcat);
```

```
cmd.Parameters.AddWithValue("@pnm", pnm);
cmd.Parameters.AddWithValue("@pdis",
pdis);
cmd.Parameters.AddWithValue("@unitp",
unitp);
cmd.ExecuteNonQuery();

con.Close();
DataList1.DataBind();
}
catch (Exception ex)
{
Response.Write("Error: " + ex.Message);
if(con.State ==
System.Data.ConnectionState.Open)
{
con.Close();
}
}
}}
```

Sql Queries:

```
CREATE DATABASE Productmaster;
USE Productmaster;
Create Table Productmaster
(
productid int,
productcategory VARCHAR(10),
productname VARCHAR(10),
productdescription VARCHAR(10),
unitprice int
);
select * from Productmaster;
```

Output:



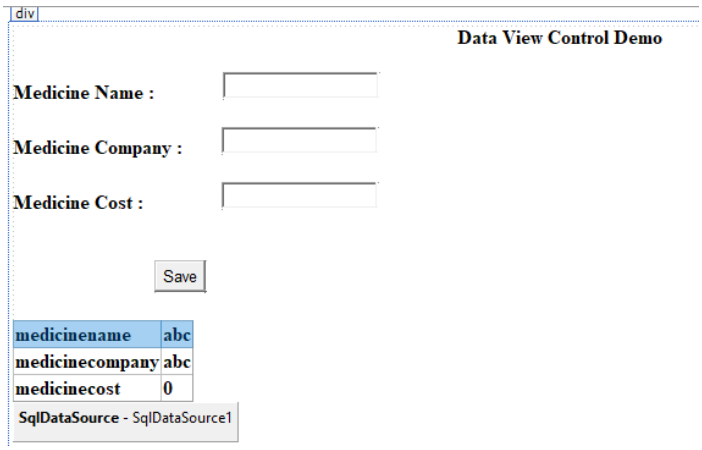
Practical: 18

Aim : Design a Web Application to demonstrate Data Bound Control-DetailsView

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Prac18.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<strong>    Data View
Control Demo<br />
<br />
Medicine Name :  
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
Medicine Company
:      &nbs
p;
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
<br />
Medicine Cost :  
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
<br />
<asp:Button ID="Button1" runat="server"
Text="Save" OnClick="Button1_Click" />
<br />
<br />
<asp:DetailsView ID="DetailsView1"
runat="server"
DataSourceID="SqlDataSource1"
Height="50px" Width="125px"
AutoGenerateRows="False">
<Fields>
<asp:BoundField DataField="medicinename"
HeaderText="medicinename"
SortExpression="medicinename" />
<asp:BoundField
DataField="medicinecompany"
HeaderText="medicinecompany"
SortExpression="medicinecompany" />
<asp:BoundField DataField="medicinecost"
HeaderText="medicinecost"
SortExpression="medicinecost" />
</Fields>
</asp:DetailsView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:medConnectionString %>"
SelectCommand="SELECT * FROM
[medicine]"></asp:SqlDataSource>
<br />
<br />
</strong>
</div></form></body>
</html>
```

Design: Aspx file :



Name: Sumit Kadam **Roll no : 16**

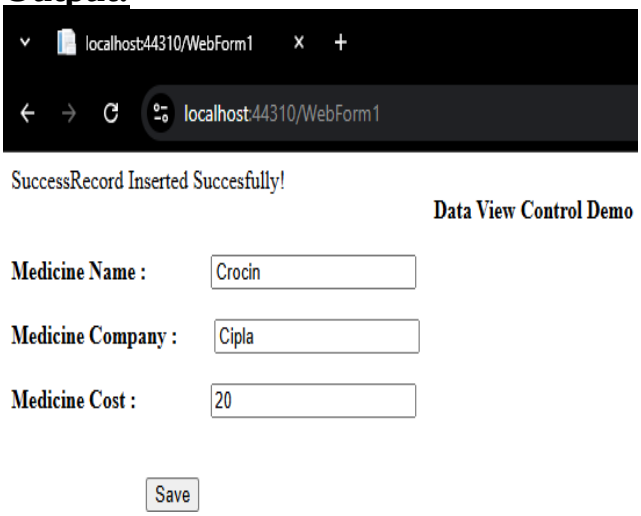
Code behind file : .aspx.cs :

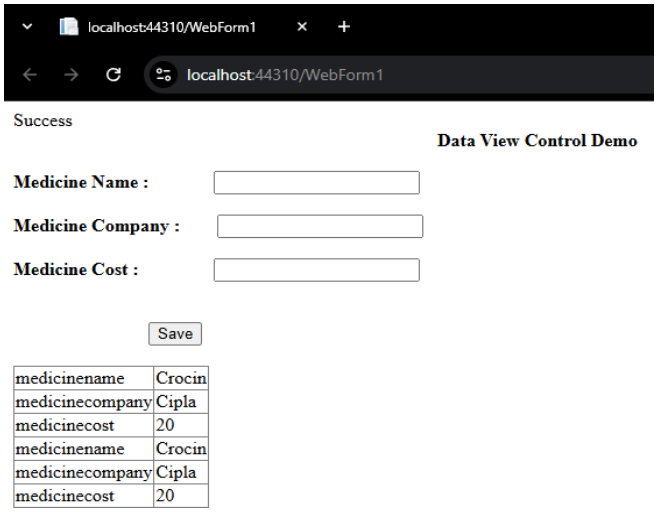
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Prac18
{
public partial class WebForm1 :
System.Web.UI.Page
{
SqlConnection con = new SqlConnection();
SqlCommand cmd;
protected void Page_Load(object sender,
EventArgs e)
{
con.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=med;Integrated Security=True";
con.Open();
Response.Write("Success");
}
protected void Button1_Click(object
sender, EventArgs e)
{
string mnm, mcomp;
int mcost;
mnm = TextBox1.Text;
mcomp = TextBox2.Text;
mcost = Convert.ToInt32(TextBox3.Text);

string qry = "insert into
medicine(medicinename,medicinecompany,med
icinecost)values('" + mnm + "','" + mcomp
+ "'," + mcost + ")";
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("Record Inserted
Succesfully!");

}
}
}
```

Sql Queries:

```
create table medicine2(
medicinename varchar(100),
medicinecompany varchar(100),
medicinecost int
);
select * from medicine2;
```

Output:





Name: Sumit Kadam **Roll no : 16**

Practical: 19

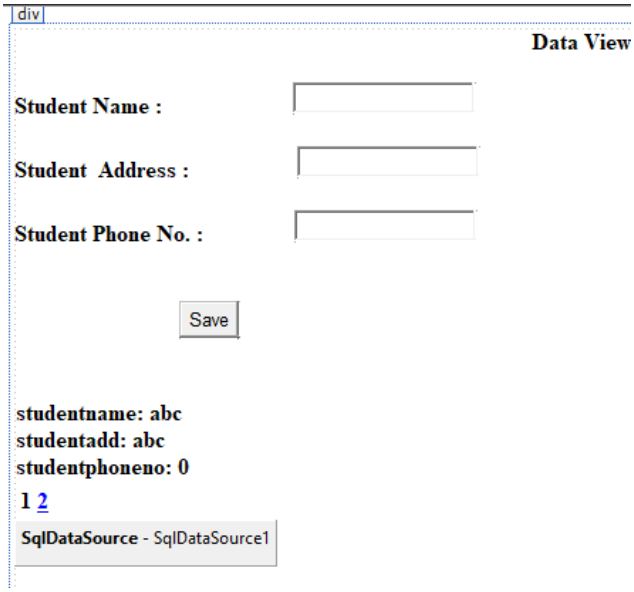
Aim : Design a Web Application to demonstrate Data Bound Control- FormView

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="prac19.WebForm1" %>

<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
       
;      &nbs
p;      &nb
sp;      &n
bsp;      &
nbsp;      
       
;      &nbs
p;      &nb
sp;      &n
bsp;      &
nbsp;      
    
<strong>    Data View
Control Demo<br />
<br />
Student Name
:      &nbs
p;      &nb
sp;     
  
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
Student  Address
:      &nbs
p;      &nb
sp;    
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
<br />
Student Phone No.
:      &nbs
p;      &nb
sp;  <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
<br                                    />
       
;      &nbs
p;      &nb
sp;      &n
bsp;
<asp:Button ID="Button1" runat="server"
Text="Save" OnClick="Button1_Click" />
<br />
<br />
<br />
<asp:FormView ID="FormView1"
runat="server" AllowPaging="True"
DataSourceID="SqlDataSource1">
<EditItemTemplate>
studentname:
<asp:TextBox ID="studentnameTextBox"
runat="server" Text='<%#
Bind("studentname") %>' />
<br />
studentadd:
<asp:TextBox ID="studentaddTextBox"
runat="server" Text='<%#
Bind("studentadd") %>' />
<br />
studentphoneno:
<asp:TextBox ID="studentphonenoTextBox"
runat="server" Text='<%#
Bind("studentphoneno") %>' />
<br />
<asp:LinkButton ID="UpdateButton"
runat="server" CausesValidation="True"
CommandName="Update" Text="Update" />
 <asp:LinkButton
ID="UpdateCancelButton" runat="server"
CausesValidation="False"
CommandName="Cancel" Text="Cancel" />
</EditItemTemplate>
<InsertItemTemplate>
studentname:
<asp:TextBox ID="studentnameTextBox"
runat="server" Text='<%#
Bind("studentname") %>' />
<br />
studentadd:
```

```
<asp:TextBox ID="studentaddTextBox"
runat="server" Text='<%#
Bind("studentadd") %>' />
<br />
studentphoneno:
<asp:TextBox ID="studentphonenoTextBox"
runat="server" Text='<%#
Bind("studentphoneno") %>' />
<br />
<asp:LinkButton ID="InsertButton"
runat="server" CausesValidation="True"
CommandName="Insert" Text="Insert" />
 <asp:LinkButton
ID="InsertCancelButton" runat="server"
CausesValidation="False"
CommandName="Cancel" Text="Cancel" />
</InsertItemTemplate>
<ItemTemplate>
studentname:
<asp:Label ID="studentnameLabel"
runat="server" Text='<%#
Bind("studentname") %>' />
<br />
studentadd:
<asp:Label ID="studentaddLabel"
runat="server" Text='<%#
Bind("studentadd") %>' />
<br />
studentphoneno:
<asp:Label ID="studentphonenoLabel"
runat="server" Text='<%#
Bind("studentphoneno") %>' />
<br />

</ItemTemplate>
</asp:FormView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:studentdb_1ConnectionSt
ring2 %>" SelectCommand="SELECT * FROM
[student2]"></asp:SqlDataSource>
<br />
<br />
</strong>
</div>
</form>
</body>
</html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

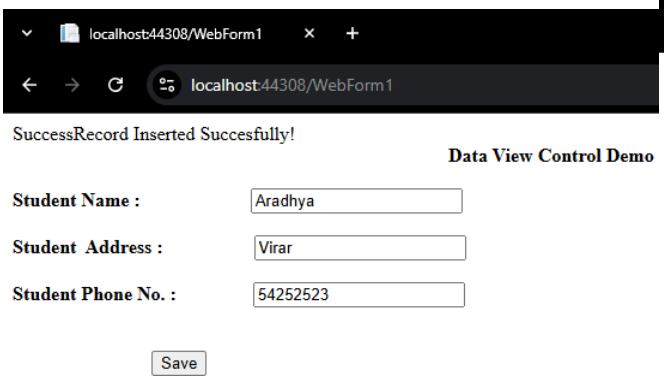
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace prac19
{
public partial class WebForm1 :
System.Web.UI.Page
{
SqlConnection con = new SqlConnection();
SqlCommand cmd;
```

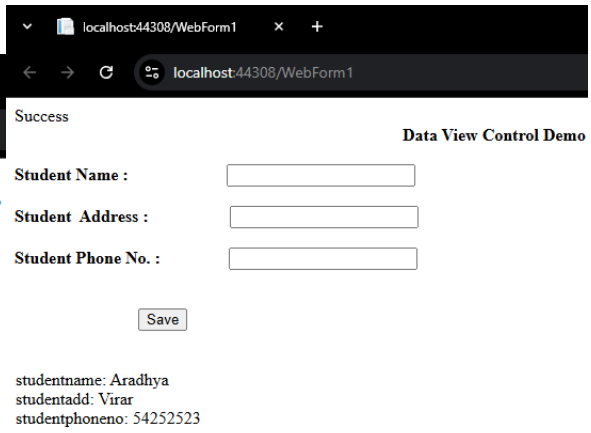
```
protected void Page_Load(object sender,
EventArgs e)
{
con.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=studentdb_1;Integrated
Security=True";
con.Open();
Response.Write("Success");
}


protected void Button1_Click(object
sender, EventArgs e)
{
string mnm, mcomp;
int mcost;
mnm = TextBox1.Text;
mcomp = TextBox2.Text;
mcost = Convert.ToInt32(TextBox3.Text);

string qry = "insert into
student2(studentname,studentadd,studentph
oneno )values('" + mnm + "','" + mcomp +
"'," + mcost + ")";
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("Record Inserted
Succesfully!");

}
}
}
```

Sql Queries:

```
create database studentdb_1;
use studentdb_1;
create table student2(
studentname varchar(100),
studentadd varchar(100),
studentphoneno int
);
select * from student2;
```

Output:





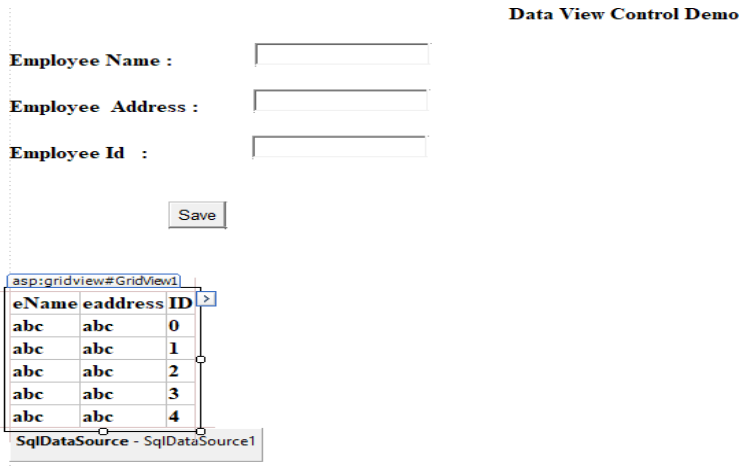
Practical: 20

Aim : Design a Web Application to demonstrate Data Bound Control- GridView

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="prac20.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
       
; 
<strong>    Data View
Control Demo<br />
<br />
Employee Name
:      &nbs
p;      &nb
sp;
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
Employee  Address
:      &nbs
p;  
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
Employee Id  
:   
     
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<asp:Button ID="Button1" runat="server"
Text="Save" OnClick="Button1_Click" />
<br />
<asp:GridView ID="GridView1"
runat="server"
AutoGenerateColumns="False"
DataSourceID="SqlDataSource1">
<Columns>
<asp:BoundField DataField="eName"
HeaderText="eName" SortExpression="eName"
/>
<asp:BoundField DataField="eaddress"
HeaderText="eaddress"
SortExpression="eaddress" />
<asp:BoundField DataField="ID"
HeaderText="ID" SortExpression="ID" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:empployeeConnectionStri
ng %>" SelectCommand="SELECT * FROM
[employee_0]"></asp:SqlDataSource>
<br />
</strong></div></form></body>
</html>
```

Design: Aspx file :



Name: Sumit Kadam **Roll no : 16**

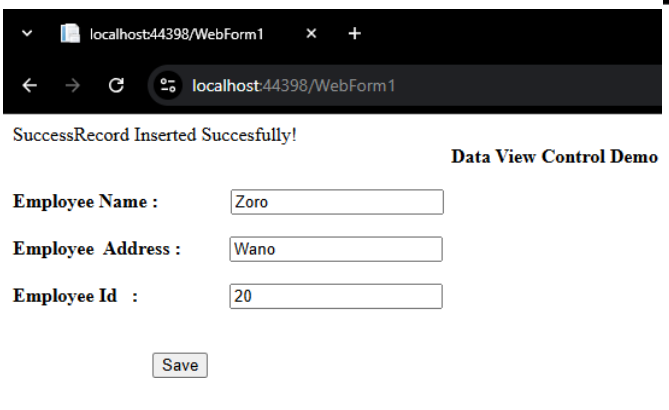
Code behind file : .aspx.cs :

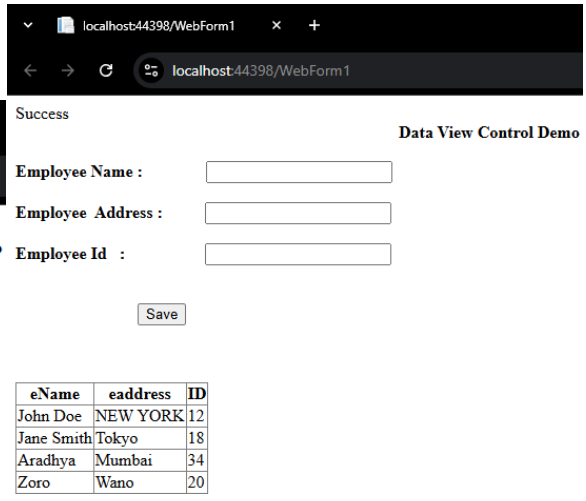
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace prac20
{public partial class WebForm1 :
System.Web.UI.Page
{
SqlConnection con = new SqlConnection();
SqlCommand cmd;
protected void Page_Load(object sender,
EventArgs e)
{
con.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=empployee;Integrated
Security=True";
con.Open();
Response.Write("Success");}

protected void Button1_Click(object
sender, EventArgs e)
{
string ename, eadd;
int e_id;
ename = TextBox1.Text;
eadd = TextBox2.Text;
e_id = Convert.ToInt32(TextBox3.Text);

string qry = "insert into
employee_0(eName,eaddress,ID )values('" +
ename + "','" + eadd + "'," + e_id + ")";
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("Record Inserted
Succesfully!");
}}}
```

Sql Queries:

```
create database empdb_a;use empdb_a;

create table employee(
emp_name varchar(100),
emp_add varchar(100),
emp_id int
);
select * from employee;
```

Output:





Practical: 21

Aim : Design a Web Application to demonstrate Data Bound Control- ListView

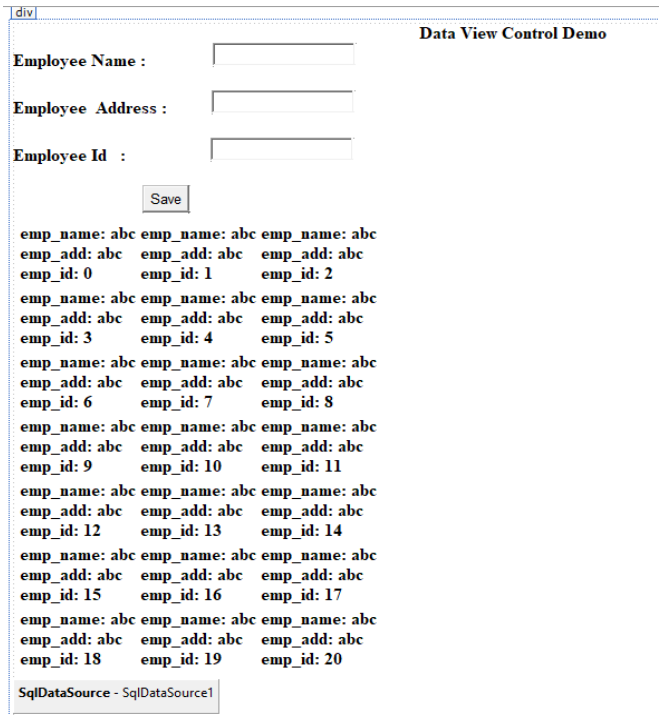
Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="pract21.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
       &
nbsp;      &n
bsp;      &nb
sp;      &nbs
p;       
;  <strong>   
Data View Control Demo<br />
Employee Name
:       
       
<asp:TextBox ID="t1"
runat="server"></asp:TextBox>
<br />
<br />
Employee  Address
:       
 
<asp:TextBox ID="t2"
runat="server"></asp:TextBox>
<br />
<br />
Employee Id  
:       
     
      <asp:TextBox
ID="t3" runat="server"></asp:TextBox>
<br />
       &
nbsp;   
<asp:Button ID="Button1" runat="server"
Text="Save" OnClick="Button1_Click" />
<br />
<asp:ListView ID="ListView1" runat="server"
DataSourceID="SqlDataSource1"
GroupItemCount="3">
<AlternatingItemTemplate>
<td runat="server" style="">emp_name:
<asp:Label ID="emp_nameLabel"
runat="server" Text='<%# Eval("emp_name")
%>' />
<br />
emp_add:
<asp:Label ID="emp_addLabel" runat="server"
Text='<%# Eval("emp_add") %>' />
<br />
emp_id:
<asp:Label ID="emp_idLabel" runat="server"
Text='<%# Eval("emp_id") %>' />
<br />
</td>
</AlternatingItemTemplate>
<EditItemTemplate>
<td runat="server" style="">emp_name:
<asp:TextBox ID="emp_nameTextBox"
runat="server" Text='<%# Bind("emp_name")
%>' /><br />
emp_add:
<asp:TextBox ID="emp_addTextBox"
runat="server" Text='<%# Bind("emp_add")
%>' /><br />
emp_id:
<asp:TextBox ID="emp_idTextBox"
runat="server" Text='<%# Bind("emp_id") %>'
/><br />
<asp:Button ID="UpdateButton"
runat="server" CommandName="Update"
Text="Update" /><br />
<asp:Button ID="CancelButton"
runat="server" CommandName="Cancel"
Text="Cancel" /><br />
</td>
</EditItemTemplate>
<EmptyDataTemplate>
<table runat="server" style="">
<tr>
<td>No data was returned.</td>
</tr>
</table>
</EmptyDataTemplate>
<EmptyItemTemplate>
<td runat="server" />
</EmptyItemTemplate>
<GroupTemplate>
<tr id="itemPlaceholderContainer"
runat="server">
<td id="itemPlaceholder"
runat="server"></td>
</tr>
</GroupTemplate>
<InsertItemTemplate>
<td runat="server" style="">emp_name:
<asp:TextBox ID="emp_nameTextBox"
runat="server" Text='<%# Bind("emp_name")
%>' /><br />
emp_add:
```

Name: Sumit Kadam **Roll no : 16**

```
<asp:TextBox ID="emp_addTextBox"
runat="server" Text='<%# Bind("emp_add")
%>' /><br />
emp_id:
<asp:TextBox ID="emp_idTextBox"
runat="server" Text='<%# Bind("emp_id") %>'
/><br />
<asp:Button ID="InsertButton"
runat="server" CommandName="Insert"
Text="Insert" /><br />
<asp:Button ID="CancelButton"
runat="server" CommandName="Cancel"
Text="Clear" />
<br /></td>
</InsertItemTemplate>
<ItemTemplate>
<td runat="server" style="">emp_name:
<asp:Label ID="emp_nameLabel"
runat="server" Text='<%# Eval("emp_name")
%>' /><br />
emp_add:
<asp:Label ID="emp_addLabel" runat="server"
Text='<%# Eval("emp_add") %>' /><br />
emp_id:
<asp:Label ID="emp_idLabel" runat="server"
Text='<%# Eval("emp_id") %>' /><br />
</td>
</ItemTemplate><LayoutTemplate>
<table runat="server">
<tr runat="server">
<td runat="server">
<table id="groupPlaceholderContainer"
runat="server" border="0" style="">
<tr id="groupPlaceholder" runat="server">
</tr>
</table>
</td></tr>
<tr runat="server">
<td runat="server" style=""></td>
</tr>
</table>
</LayoutTemplate>
<SelectedItemTemplate>
<td runat="server" style="">emp_name:
<asp:Label ID="emp_nameLabel"
runat="server" Text='<%# Eval("emp_name")
%>' />
<br />
emp_add:
<asp:Label ID="emp_addLabel" runat="server"
Text='<%# Eval("emp_add") %>' />
<br />
emp_id:
<asp:Label ID="emp_idLabel" runat="server"
Text='<%# Eval("emp_id") %>' />
<br />
</td>
</SelectedItemTemplate>
</asp:ListView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:empdb_bConnectionString3
%>" SelectCommand="SELECT * FROM
[employee]"></asp:SqlDataSource>
<br />
</strong>
</div></form>
</body></html>
```

Design: Aspx file :



Name: Sumit Kadam **Roll no : 16**

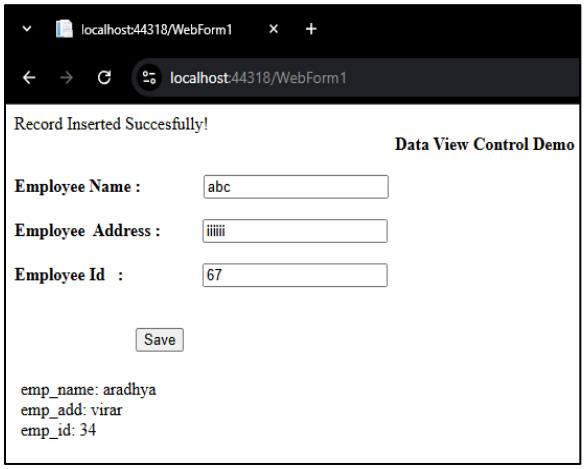
Code behind file : .aspx.cs :

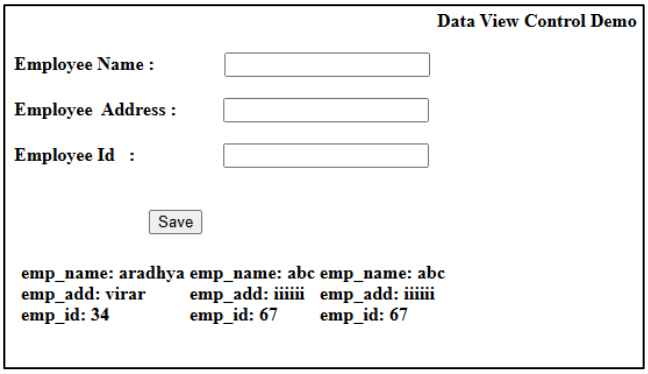
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace pract21
{public partial class WebForm1 :
System.Web.UI.Page
{SqlConnection con = new SqlConnection();
SqlCommand cmd;
protected void Page_Load(object sender,
EventArgs e)
{
con.ConnectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=empdb_b;Integrated
Security=True";
con.Open();}
protected void Button1_Click(object
sender, EventArgs e)
{string ename, eadd;
int e_id;
ename = t1.Text;
eadd = t2.Text;
e_id = Convert.ToInt32(t3.Text);
string qry = "insert into
employee(emp_name,emp_add,emp_id
)values('" + ename + "','" + eadd + "',"
+ e_id + ")";
cmd = new SqlCommand(qry, con);
cmd.ExecuteNonQuery();
Response.Write("Record Inserted
Succesfully!");
}}
}
```

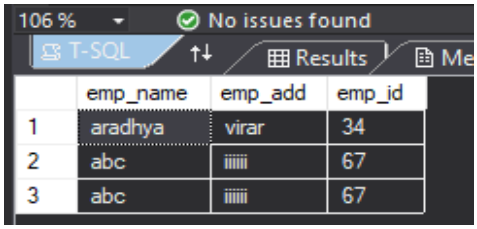
Sql Queries:

```
create database empdb_b;
use empdb_b;create table employee(emp_name varchar(100),emp_add varchar(100),emp_id int);
select * from employee;
```

Output:







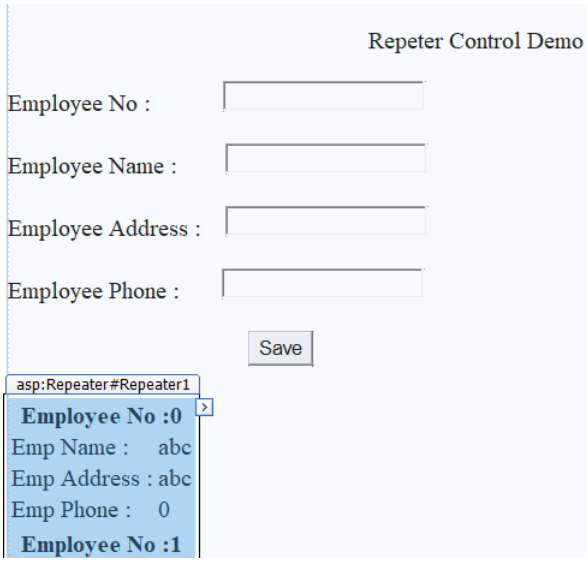
Practical: 22

Aim : Design a Web Application to demonstrate Data Bound Control- Repeater

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="repeater.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"&gt;
<head runat="server">
    <title></title></head><body>
    <form id="form1" runat="server">
        <div><br />
       
;
<asp:Label ID="Label5" runat="server"
Text="Repeter Control Demo"></asp:Label>
<br /><br />
<asp:Label ID="Label1" runat="server"
Text="Employee No :"></asp:Label>
       
;    
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox><br /><br />
<asp:Label ID="Label2" runat="server"
Text="Employee Name :"></asp:Label>
       
;
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox><br />
<asp:Label ID="Label3" runat="server"
Text="Employee Address :"></asp:Label>
               
            <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox><br />
            <asp:Label ID="Label4"
runat="server" Text="Employee Phone
:"></asp:Label>
      
            <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
            <br /><br />
       
;      &nbs
p;  
            <asp:Button ID="Button1"
runat="server" OnClick="Button1_Click"
Text="Save" /><br /><br />
<asp:Repeater ID="Repeater1"
runat="server"
DataSourceID="SqlDataSource1">
                <ItemTemplate>
                    <div class
="Repeater1">
                        <table>
                            <tr><th
colspan ="2">Employee No
:<%#Eval("empno") %></th></tr>
                            <tr><td> Emp
Name :</td><td> <%#Eval("empname")
%></td></tr>
                            <tr><td> Emp
Address :</td><td> <%#Eval("empadd")
%></td></tr>
                            <tr><td> Emp
Phone :</td><td> <%#Eval("empphon")
%></td></tr>
                        </table>
                    </div>
                </ItemTemplate>
            </asp:Repeater>
            <asp:SqlDataSource
ID="SqlDataSource1" runat="server"
ConnectionString="<%$
ConnectionStrings:empdbConnectionString
%>" SelectCommand="SELECT * FROM
[employee]"></asp:SqlDataSource>
        </div>
    </form></body></html>
```

Design: Aspx file :



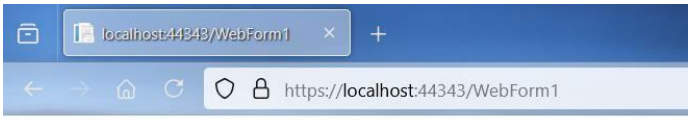
Code behind file : .aspx.cs :

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

namespace repeater
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        SqlConnection con = new SqlConnection();
        SqlCommand cmd;

        protected void Page_Load(object sender, EventArgs e)
        {
            con.ConnectionString = "Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=empdb;Integrated Security=True";
            con.Open();
}

        protected void Button1_Click(object sender, EventArgs e)
        {
            string empno, empname, empadd, empphon;
            empno = TextBox1.Text;
            empname = TextBox2.Text;
            empadd = TextBox3.Text;
            empphon = TextBox4.Text;
            string qry = "insert into employee(empno,empname,empadd,empphon) values ('" + empno + "', '" + empname + "', '" + empadd + "', '" + empphon + "')";
            cmd = new SqlCommand(qry, con);
            cmd.ExecuteNonQuery();
            Response.Write("Record Inserted Successfully");
        }
    }
}
```

Sql Queries:

```
create database empdb;
use empdb;
create table employee(
empno int,
empname varchar(100),
empadd varchar(100),
empphon int);

select * from employee;
```

Output:



Record Inserted Successfully

Repeter Control Demo

Employee No : 1

Employee Name : Aradhya

Employee Address : Virar

Employee Phone : 899999991

Save

Repeter Control Demo

Employee No :

Employee Name :

Employee Address :

Employee Phone :

Save

Employee No :1

Emp Name : Aradhya

Emp Address : Virar

Emp Phone : 899999991

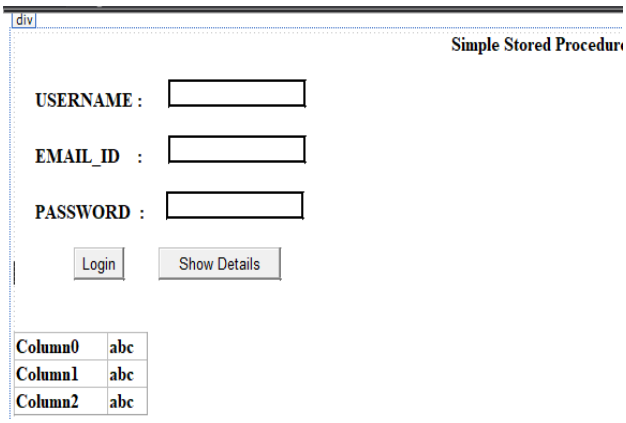
Practical: 23

Aim : Develop a Stored Procedure to fetch the employee details them using a Details View Control

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="SimpleStoredProcedure.aspx.cs
"
Inherits="Pract23.SimpleStoredProcedure"
%>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head><body>
<form id="form1" runat="server">
<div>
     
<strong>Simple Stored Procedure Demo<br
/><br />
     USERNAME
:     
<asp:TextBox ID="TextBox1" runat="server"
BorderStyle="Inset"></asp:TextBox>
<br />
    
EMAIL_ID   
:     
<asp:TextBox ID="TextBox2" runat="server"
BorderStyle="Inset"></asp:TextBox>
<br />
     PASSWORD 
:    
<asp:TextBox ID="TextBox3" runat="server"
BorderStyle="Inset"></asp:TextBox>
<br />
<br />
       
;      
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Login" />
       
;
<asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="Show
Details" /><br />
<asp:DetailsView ID="DetailsView1"
runat="server" Height="50px"
Width="125px">
</asp:DetailsView>
<br /><br />
</strong></div>
</form></body></html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

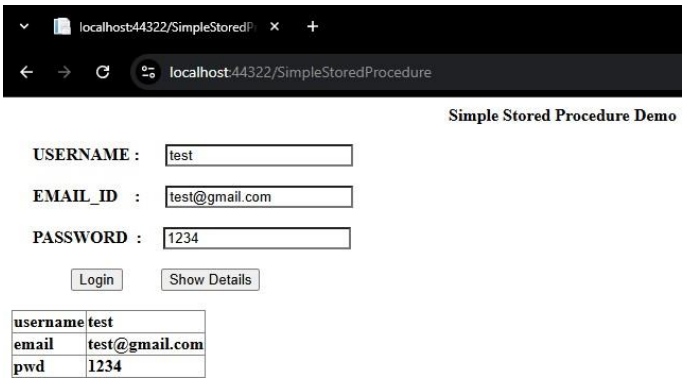
```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Pract23
```

```
{public partial class
SimpleStoredProcedure :
System.Web.UI.Page
{protected void Page_Load(object sender,
EventArgs e)
{}
protected void Button1_Click(object
sender, EventArgs e)
{string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=UserRegistration;Integrated
Security=True";
SqlConnection cn = new
SqlConnection(str);
SqlCommand cmd = new
SqlCommand("storlogin", cn);
cmd.CommandType =
CommandType.StoredProcedure;
cn.Open();
SqlParameter p1 = new
SqlParameter("username", TextBox1.Text);
SqlParameter p2 = new
SqlParameter("email", TextBox2.Text);
SqlParameter p3 = new SqlParameter("pwd",
TextBox3.Text);
cmd.Parameters.Add(p1);
cmd.Parameters.Add(p2);
cmd.Parameters.Add(p3);
cmd.ExecuteReader();
Response.Write(" LOGIN
Successfully.....");
cn.Close();
}
protected void Button2_Click(object
sender, EventArgs e)
{

string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=UserRegistration;Integrated
Security=True";
SqlConnection cn = new
SqlConnection(str);
SqlDataAdapter sde = new
SqlDataAdapter("Select * from
registration", cn);
DataSet ds = new DataSet();
sde.Fill(ds);
DetailsView1.DataSource = ds;
DetailsView1.DataBind();}}}
```

Sql Queries:

```
create database UserRegistration;

use UserRegistration;

CREATE TABLE registration(
username varchar(100) NULL,
email varchar(100) NULL,
pwd varchar(100) NULL);

select * from registration;
delete from registration;

Go

create procedure storlogin(
@username varchar(40),
@email varchar(50),
@pwd varchar(20))
As
BEGIN
insert into registration
values(@username,@email,@pwd)
END
```

Output:



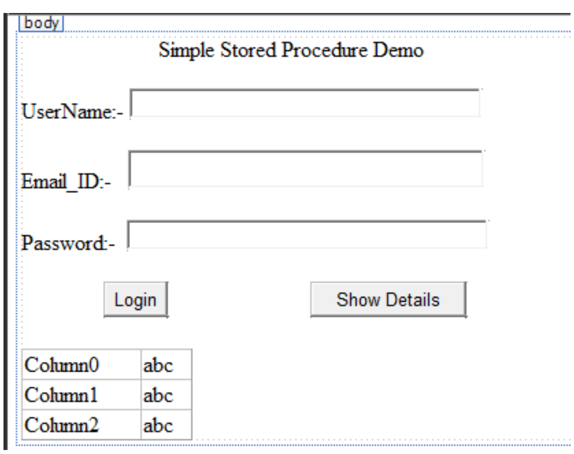
Practical: 24

Aim : Design a Web Application to demonstrate the working of parameterized stored procedure

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="StoredProcedure.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head><body>
<form id="form1" runat="server">
<div>
                         Simple Stored
Procedure Demo<br />
<br />
UserName:-
<asp:TextBox ID="TextBox1" runat="server"
Width="249px"></asp:TextBox>
<br /><br />
Email_ID:-  
<asp:TextBox ID="TextBox2" runat="server"
Height="22px"
Width="252px"></asp:TextBox>
<br /><br />
Password:- 
<asp:TextBox ID="TextBox3" runat="server"
Width="256px"></asp:TextBox>
<br /><br />
              
<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Login" />
                         
<asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="Show Details"
/>
<br />
<asp:DetailsView ID="DetailsView1"
runat="server" Height="50px" Width="125px">
</asp:DetailsView>
</div>
</form>
</body>
</html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ParametrizdStoreProcedure
{
public partial class WebForm1 :
System.Web.UI.Page
{
protected void Page_Load(object sender,
EventArgs e){}
```

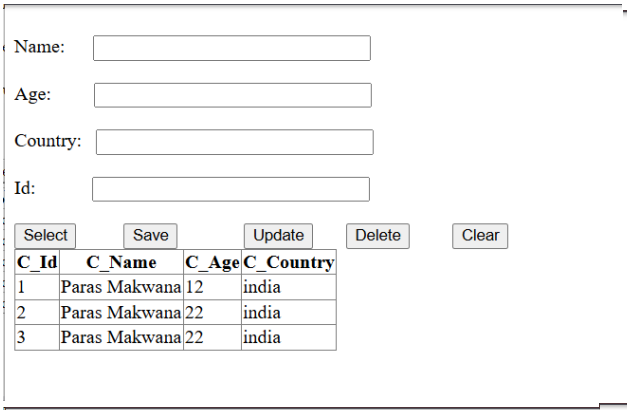
```
protected void Button1_Click(object sender,
EventArgs e)
{string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=CustomerDB;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertifica
te=False;ApplicationIntent=ReadWrite;MultiSu
bnetFailover=False";
using (SqlConnection cn = new
SqlConnection(str))
{
using (SqlCommand cmd = new
SqlCommand("ParameterizedStoredProcedure",
cn))
{
cmd.CommandType =
CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Action",
"Select");
cmd.Parameters.AddWithValue("@Name",
string.IsNullOrWhiteSpace(TextBox1.Text) ?
(object)DBNull.Value : TextBox1.Text);
cmd.Parameters.AddWithValue("@Age",
string.IsNullOrWhiteSpace(TextBox2.Text) ?
(object)DBNull.Value :
Convert.ToInt32(TextBox2.Text));
cmd.Parameters.AddWithValue("@Country",
string.IsNullOrWhiteSpace(TextBox3.Text) ?
(object)DBNull.Value : TextBox3.Text);
SqlDataAdapter da = new
SqlDataAdapter(cmd);
DataTable dt = new DataTable();
da.Fill(dt);

if (dt.Rows.Count > 0)
{GridView1.DataSource = dt;  // Bind data to
GridView
GridView1.DataBind();       // Refresh GridView
}
else
{Response.Write("No records found.");
GridView1.DataSource = null;
GridView1.DataBind();
}}}}

protected void Button2_Click(object sender,
EventArgs e)
{string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=CustomerDB;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertifica
te=False;ApplicationIntent=ReadWrite;MultiSu
bnetFailover=False";
SqlConnection cn = new SqlConnection(str);
SqlCommand cmd = new
SqlCommand("ParameterizedStoredProcedure",
cn);
cmd.CommandType =
CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Action",
"Insert");
cmd.Parameters.AddWithValue("@Name",
TextBox1.Text);
cmd.Parameters.AddWithValue("@Age",
TextBox2.Text);
cmd.Parameters.AddWithValue("@Country",
TextBox3.Text);
cn.Open();
cmd.ExecuteNonQuery();
Response.Write("Record Inserted
Successfully. ... ");
GridView1.DataBind();
cn.Close();}

protected void Button3_Click(object sender,
EventArgs e)
{string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=CustomerDB;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertifica
te=False;ApplicationIntent=ReadWrite;MultiSu
bnetFailover=False";
SqlConnection cn = new SqlConnection(str);
SqlCommand cmd = new
SqlCommand("ParameterizedStoredProcedure",
cn);
cmd.CommandType =
CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Action",
"Update");
cmd.Parameters.AddWithValue("@Name",
TextBox1.Text);
cmd.Parameters.AddWithValue("@Age",
TextBox2.Text);
cmd.Parameters.AddWithValue("@Country",
TextBox3.Text);
cmd.Parameters.AddWithValue("@Id",
TextBox4.Text);
cn.Open();
```

```
cmd.ExecuteNonQuery();
Response.Write("Record Updated sucessfully
.....");
GridView1.DataBind();
cn.Close();}

protected void Button4_Click(object sender,
EventArgs e)
{string str = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=CustomerDB;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertifica
te=False;ApplicationIntent=ReadWrite;MultiSu
bnetFailover=False";
SqlConnection cn = new SqlConnection(str);
SqlCommand cmd = new
SqlCommand("ParameterizedStoredProcedure",
cn);
cmd.CommandType =
CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Action",
"Delete");
cmd.Parameters.AddWithValue("@Id",
TextBox4.Text);
cn.Open();
cmd.ExecuteNonQuery();
Response.Write("Record Deleted
Successfully. ... ");
GridView1.DataBind();
cn.Close();}

protected void Button5_Click(object sender,
EventArgs e)
{TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";}
}
}
```

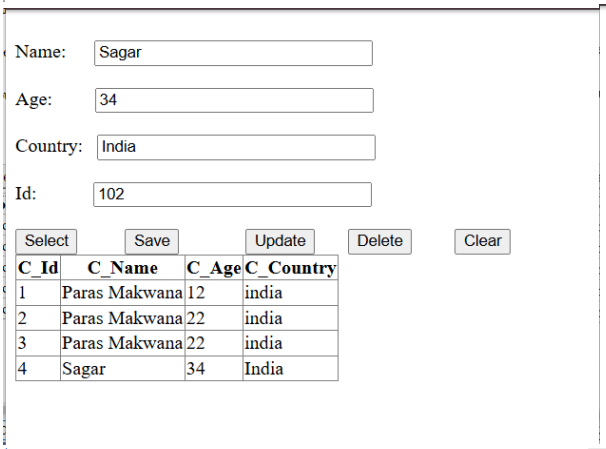
Sql Queries:

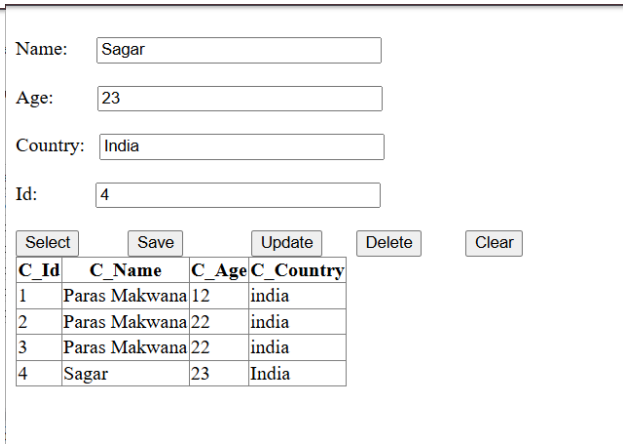
```
create database UserRegistration;
use UserRegistration;

create table registration
(
username varchar(100)NULL,
email varchar(100)NULL,
pwd varchar(100)NULL,
);

select * from registration;
delete  from registration;
Go

create procedure storlogin
(
@username varchar(40),
@email varchar(50),
@pwd varchar(20)
)
AS
BEGIN
insert into registration values (@username,@email,@pwd)
END
```

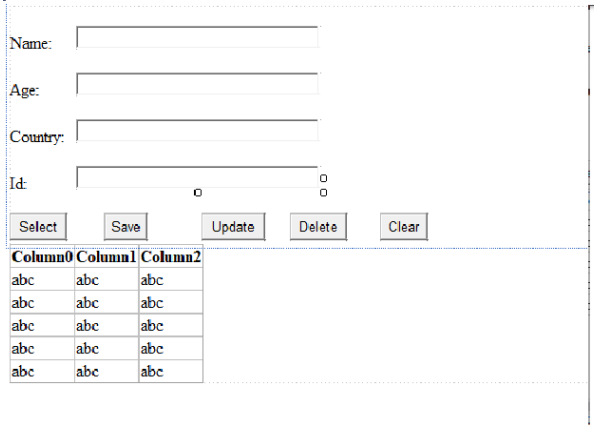
Output:

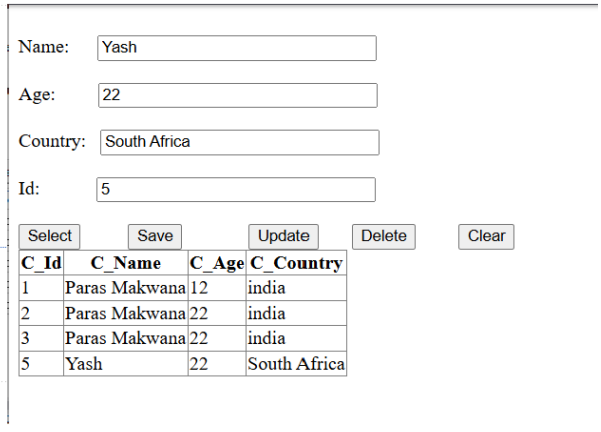

Name:
Age:
Country:
Id:
Select
Save
Update
Delete
Clear
C_Id C_Name C_Age C_Country
1 Paras Makwana 12 india
2 Paras Makwana 22 india
3 Paras Makwana 22 india


Record Inserted Successfully.....
Name: Yash
Age: 22
Country: South Africa
Id: 5
Select
Save
Update
Delete
Clear


Name: Sagar
Age: 34
Country: India
Id: 102
Select
Save
Update
Delete
Clear
C_Id C_Name C_Age C_Country
1 Paras Makwana 12 india
2 Paras Makwana 22 india
3 Paras Makwana 22 india
4 Sagar 34 India


Name: Sagar
Age: 23
Country: India
Id: 4
Select
Save
Update
Delete
Clear
C_Id C_Name C_Age C_Country
1 Paras Makwana 12 india
2 Paras Makwana 22 india
3 Paras Makwana 22 india
4 Sagar 23 India


Name:
Age:
Country:
Id:
Select
Save
Update
Delete
Clear
Column0 Column1 Column2
abc abc abc
abc abc abc
abc abc abc
abc abc abc
abc abc abc


Name: Yash
Age: 22
Country: South Africa
Id: 5
Select
Save
Update
Delete
Clear
C_Id C_Name C_Age C_Country
1 Paras Makwana 12 india
2 Paras Makwana 22 india
3 Paras Makwana 22 india
5 Yash 22 South Africa

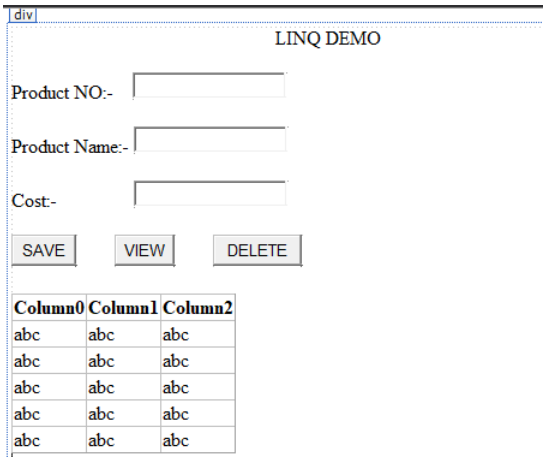
Practical: 25

Aim : Design a Web Application to display the use of LINQ.

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="prac25.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head><body>
<form id="form1" runat="server">
<div>
       
;      &nbs
p;      
LINQ DEMO<br /><br />
Product NO:-   
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br /><br />
Product Name:- <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
Cost:-
       
;      &nbs
p;  <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1" runat="server"
Text="SAVE" OnClick="Button1_Click" />

       
;
<asp:Button ID="Button2" runat="server"
Text="VIEW" OnClick="Button2_Click" />

       
;
<asp:Button ID="Button3" runat="server"
Text="DELETE" OnClick="Button3_Click" />
<br /><asp:GridView ID="GridView1"
runat="server">
</asp:GridView><br />
</div></form></body>
</html>
```

Design: Aspx file :



Code behind file : .aspx.cs :

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

namespace prac25
{public partial class WebForm1 :
System.Web.UI.Page
{protected void Page_Load(object sender,
EventArgs e)
{}
```

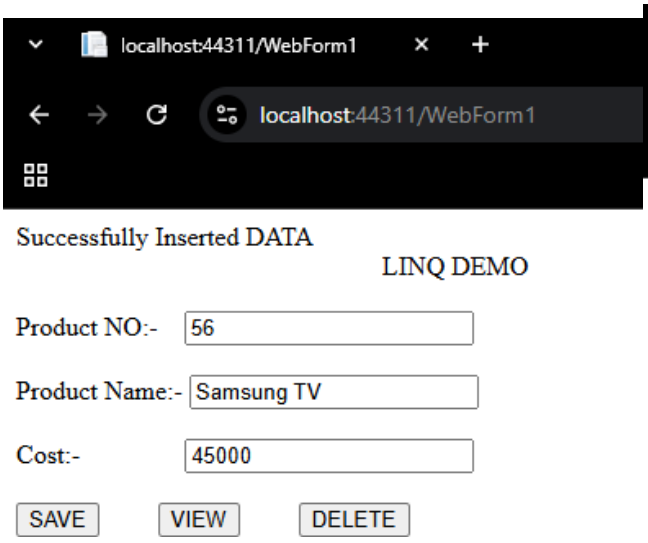
Name: Sumit Kadam **Roll no : 16**

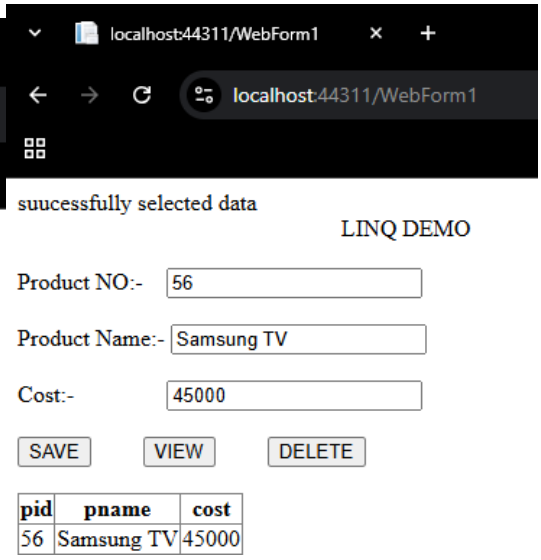
```
protected void Button1_Click(object
sender, EventArgs e)
{var d = new DataClasses1DataContext();
var c = new product1();
c.pid = Convert.ToInt32(TextBox1.Text);
c.pname = TextBox2.Text;
c.cost = TextBox3.Text;
d.product1s.InsertOnSubmit(c);
d.SubmitChanges();
Response.Write("Successfully Inserted
DATA");}
protected void Button2_Click(object
sender, EventArgs e)
{var d = new DataClasses1DataContext();
var a = from products in d.product1s
select products;
GridView1.DataSource = a;
GridView1.DataBind();
Response.Write("suucessfully selected
data");}
protected void Button3_Click(object
sender, EventArgs e)
{var d = new DataClasses1DataContext();
var c = new product1();
var deleteproduct = from products in
d.product1s
where products.pid ==
Convert.ToInt32(TextBox1.Text)
select products;
foreach (var detail in deleteproduct)
{
d.product1s.DeleteOnSubmit(detail);
}
d.SubmitChanges();
Response.Write("successfully deleted ");}
}}
```

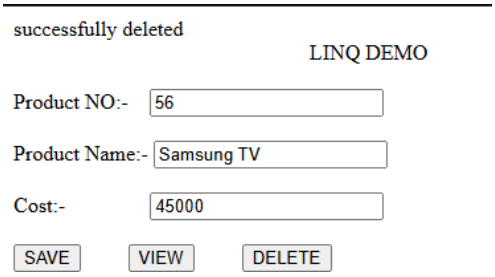
Sql Queries:

```
create database ProductDB;use ProductDB;
create table product1(pid int NOT NULL PRIMARY KEY,pname varchar(200),cost varchar(100));
select * from product1;
```

Output:







Name: Sumit Kadam **Roll no : 16**

Practical: 26

Aim : Design a Web Application to demonstrate LINQ with object data set.

Class File

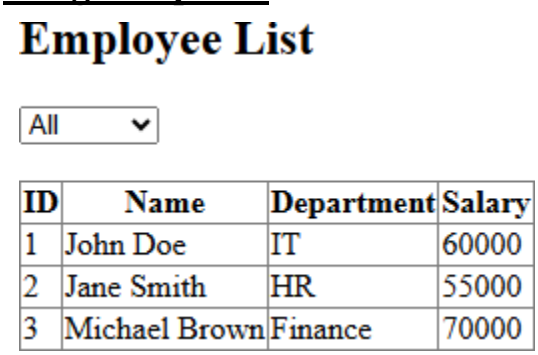
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace LinQ.App_Code
{
    public class Employee
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Department { get; set; }
        public int Salary { get; set; }

        internal void InsertEmployee(Employee newEmployee)
        {
            throw new NotImplementedException();
        }
    }
}
```

Design Code:

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>LINQ with Object Data Source</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Employee List</h2>

        <!-- DropDownList for department filter -->
        <asp:DropDownList ID="ddlDepartment" runat="server" AutoPostBack="true" OnSelectedIndexChanged="ddlDepartment_SelectedIndexChanged">
            <asp:ListItem Text="All" Value="" />
            <asp:ListItem Text="IT" Value="IT" />
            <asp:ListItem Text="HR" Value="HR" />
            <asp:ListItem Text="Finance" Value="Finance" />
        </asp:DropDownList>

        <!-- ObjectDataSource for fetching data -->
        <br />
        <asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
            TypeName="EmployeeData"
            SelectMethod="GetEmployees"
            DeleteMethod="DeleteEmployee"
            InsertMethod="InsertEmployee"
            UpdateMethod="UpdateEmployee">
            <SelectParameters>
                <asp:ControlParameter Name="department" ControlID="ddlDepartment" PropertyName="SelectedValue" Type="String" />
            </SelectParameters>
        </asp:ObjectDataSource>

        <br />
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
            BorderWidth="1" AllowPaging="true" AllowSorting ="true" DataKeyNames="Id">
            <Columns>
                <asp:BoundField DataField="Id" HeaderText="ID" ReadOnly="true" />
                <asp:BoundField DataField="Name" HeaderText="Name" />
                <asp:BoundField DataField="Department" HeaderText="Department" />
                <asp:BoundField DataField="Salary" HeaderText="Salary" />
            </Columns>
        </asp:GridView>
```

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

Design: Aspx file



ID	Name	Department	Salary
1	John Doe	IT	60000
2	Jane Smith	HR	55000
3	Michael Brown	Finance	70000

Code behind file : .aspx.cs :

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

namespace LINQwithObjecDataSource
{
    public partial class
LINQwithObjectDataSource :
System.Web.UI.Page
    {
        public static List<Employee>
employees = new List<Employee>
        {
            new Employee { Id = 1, Name =
"John Doe", Department ="IT", Salary =
60000 },
            new Employee { Id = 2, Name =
"Jane Smith", Department ="HR", Salary =
55000 },
            new Employee { Id = 3, Name =
"Michael Brown", Department ="Finance",
Salary = 70000 },
        };

        //Method to fetch employees (with
optional department filter)
        public List<Employee>
GetEmployees(string department)
        {
            return
string.IsNullOrEmpty(department)
            ? employees
            : employees.Where(emp =>
emp.Department == department).ToList();
        }

        //Insert a new employee
        public void
InsertEmployee(Employee employee)
        {
            employee.Id = employees.Max(e
=> e.Id) + 1; // auto increament id
            employees.Add(employee);
        }

        //update an existing employee
        public void
UpdateEmployee(Employee employee)
        {
            var existingEmployee =
employees.FirstOrDefault(emp => emp.Id ==
employee.Id);
            if(existingEmployee != null)
            {
                existingEmployee.Name =
employee.Name;

existingEmployee.Department =
employee.Department;
                existingEmployee.Salary =
employee.Salary;
            }
        }
```

```
        //Delete an employee by ID
        public void DeleteEmployee(int
id)
        {
            var employee =
employees.FirstOrDefault(emp => emp.Id ==
id);
            if(employee != null)
            {

employees.Remove(employee);
            }
        }
        protected void Page_Load(object
sender, EventArgs e)
        {
            if(!IsPostBack)
            {

ddlDepartment_SelectedIndexChanged(null,
null);
                BindData();
            }
        }
        private void BindData(string
department = "")
        {
            var filteredData =
string.IsNullOrEmpty(department)
                ? employees
                : employees.Where(emp =>
emp.Department == department).ToList();
            GridView1.DataSource =
filteredData;
            GridView1.DataBind();
        }

        protected void
ddlDepartment_SelectedIndexChanged(object
sender, EventArgs e)
        {
            //GridView1.DataBind();
            string selectedDepartment =
ddlDepartment.SelectedValue;
            BindData(selectedDepartment);
        }


    }
}
```

Output:

Employee List

All

ID	Name	Department	Salary
1	John Doe	IT	60000
2	Jane Smith	HR	55000
3	Michael Brown	Finance	70000

Employee List

HR

ID	Name	Department	Salary
2	Jane Smith	HR	55000

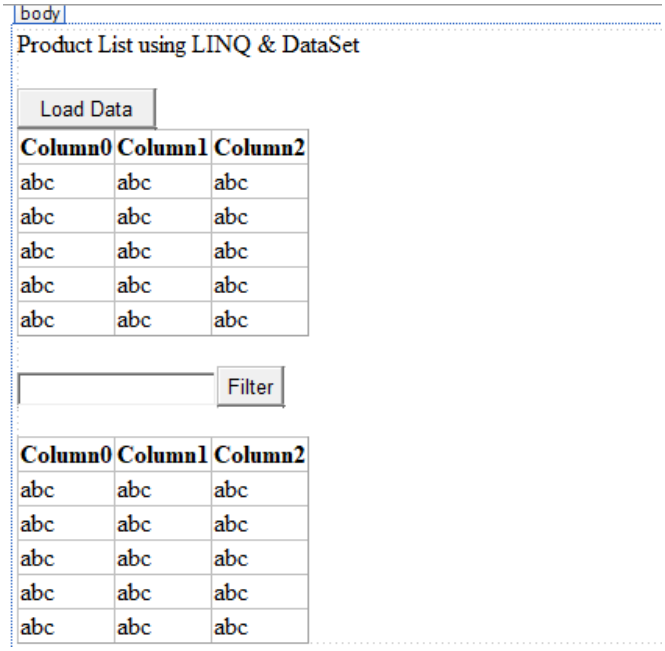
Practical: 27

Aim : Design a Web Application to demonstrate LINQ with data set.

Design Code:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="prac27.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head><body>
<form id="form1" runat="server">
<div>
Product List using LINQ &amp; DataSet<br
/><br />
<asp:Button ID="Button1" runat="server"
Text="Load Data" OnClick="Button1_Click"
/><br />
<asp:GridView ID="GridView1"
runat="server">
</asp:GridView>
<br />
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<asp:Button ID="Button2" runat="server"
Text="Filter" OnClick="Button2_Click" />
<br />
<asp:GridView ID="GridView2"
runat="server">
</asp:GridView>
</div></form>
</body>
</html>
```

Design: Aspx file :

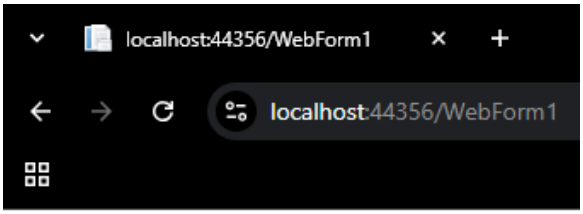


Code behind file : .aspx.cs :

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

namespace prac27
{public partial class WebForm1 :
System.Web.UI.Page
{
static DataSet ds = new DataSet();
protected void Page_Load(object sender,
EventArgs e){}

private void LoadDataSet()
{
DataTable dt = new DataTable("Products");

dt.Columns.Add("ProductID", typeof(int));
dt.Columns.Add("ProductName",
typeof(string));
dt.Columns.Add("Price", typeof(double));
dt.Columns.Add("Category",
typeof(string));

// Adding sample data
dt.Rows.Add(1, "Laptop", 900,
"Electronics");
dt.Rows.Add(2, "Smartphone", 700,
"Electronics");
dt.Rows.Add(3, "Table", 120,
"Furniture");
dt.Rows.Add(4, "Shirt", 25, "Clothing");
dt.Rows.Add(5, "Book", 15, "Stationery");
```

```
// Add DataTable to DataSet
ds.Tables.Add(dt);
}
protected void Button1_Click(object
sender, EventArgs e)
{
if (ds.Tables.Count == 0) // Load data
only once
{
LoadDataSet();
}
// Bind data to GridView
GridView1.DataSource =
ds.Tables["Products"];
GridView1.DataBind();
}

protected void Button2_Click(object
sender, EventArgs e)
{
if (ds.Tables.Count == 0)
{
Response.Write("Load data first!");
return;
}
string filterCategory =
TextBox1.Text.Trim();

// Use LINQ to filter DataTable
var filteredProducts = from product in
ds.Tables["Products"].AsEnumerable()
where product.Field<string>("Category")
!= null &&
product.Field<string>("Category")
.ToLower().Contains(filterCategory.ToLowe
r())
select product;

// Convert the filtered results back to a
DataTable
if (filteredProducts.Any())
{
GridView2.DataSource =
filteredProducts.CopyToDataTable();
}
else
{
GridView2.DataSource = null; // If no
results, clear GridView
Response.Write("No records found.");
}
GridView2.DataBind();
}}
}
```

Output:



Product List using LINQ & DataSet

Load Data

ProductID	ProductName	Price	Category
1	Laptop	900	Electronics
2	Smartphone	700	Electronics
3	Table	120	Furniture
4	Shirt	25	Clothing
5	Book	15	Stationery

Electronics Filter

ProductID	ProductName	Price	Category
1	Laptop	900	Electronics
2	Smartphone	700	Electronics

Practical No:28

Aim : Design a Web Page to perform CRUD Operations using SQLDATASource

Design Code:

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

<!DOCTYPE html>
<html>
<head>
    <title>CRUD Operations with
SqlDataSource</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Manage Employees</h2>
        <!-- SqlDataSource for CRUD
Operations -->
        <asp:SqlDataSource
            ID="SqlDataSource1"
            runat="server"
            ConnectionString="<%$
ConnectionStrings:EmployeeDBConnectionStr
ing %>"
            SelectCommand="SELECT * FROM
Employees"
            InsertCommand="INSERT INTO
Employees (Name, Position, Salary) VALUES
(@Name, @Position, @Salary)"
            UpdateCommand="UPDATE
Employees SET Name=@Name,
Position=@Position, Salary=@Salary WHERE
Id=@Id"
            DeleteCommand="DELETE FROM
Employees WHERE Id=@Id">
            <InsertParameters>
                <asp:Parameter
Name="Name" Type="String" />
                <asp:Parameter
Name="Position" Type="String" />
                <asp:Parameter
Name="Salary" Type="Decimal" />
            </InsertParameters>
            <UpdateParameters>
                <asp:Parameter
Name="Name" Type="String" />
                <asp:Parameter
Name="Position" Type="String" />
                <asp:Parameter
Name="Salary" Type="Decimal" />
                <asp:Parameter Name="Id"
Type="Int32" />
            </UpdateParameters>
            <DeleteParameters>
                <asp:Parameter Name="Id"
Type="Int32" />
            </DeleteParameters>
        </asp:SqlDataSource>
        <!-- GridView for CRUD -->
        <asp:GridView
            ID="GridView1"
            runat="server"
            AutoGenerateColumns="False"
            DataKeyNames="Id"
            DataSourceID="SqlDataSource1"
            AllowPaging="True"
            AllowSorting="True"
            AutoGenerateEditButton="True"
AutoGenerateDeleteButton="True">
            <Columns>
                <asp:BoundField
DataField="Id" HeaderText="Id"
ReadOnly="True" SortExpression="Id"
InsertVisible="False" />
                <asp:BoundField
DataField="Name" HeaderText="Name"
SortExpression="Name" />
                <asp:BoundField
DataField="Position"
HeaderText="Position"
SortExpression="Position" />
                <asp:BoundField
DataField="Salary" HeaderText="Salary"
SortExpression="Salary" />
            </Columns>
        </asp:GridView>

        <h3>Add New Employee</h3>
        <table>
            <tr>
                <td>Name:</td>
                <td><asp:TextBox
ID="txtName"
runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td>Position:</td>
                <td><asp:TextBox
ID="txtPosition"
runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td>Salary:</td>
```

```
                <td><asp:TextBox
ID="txtSalary"
runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td colspan="2">
                    <asp:Button
ID="btnInsert" runat="server" Text="Add
Employee" OnClick="btnInsert_Click" />
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

Design: Aspx file :

Manage Employees

	Id	Name	Position	Salary
Edit Delete	1	John Doe	Software Engineer	60000.00
Edit Delete	2	Jane Smith	Project Manager	75000.00
Edit Delete	3	Michael Brown	Analyst	50000.00
Edit Delete	4	John Doe	Software Engineer	60000.00
Edit Delete	5	Jane Smith	Project Manager	75000.00
Edit Delete	6	Michael Brown	Analyst	50000.00

Add New Employee

Name:
Position:
Salary:
Add Employee

Code behind file : .aspx.cs :

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

namespace CRUDoperationusingSQLDataSource
{
    public partial class WebForm1 :
System.Web.UI.Page
    {
        protected void Page_Load(object
sender, EventArgs e)
        {
        }
        protected void
btnInsert_Click(object sender, EventArgs
e)
        {
SqlDataSource1.InsertParameters["Name"].D
efaultValue = txtName.Text;

SqlDataSource1.InsertParameters["Position
"].DefaultValue = txtPosition.Text;
SqlDataSource1.InsertParameters["Salary"]
.DefaultValue = txtSalary.Text;
            SqlDataSource1.Insert();
            // Clear input fields
            txtName.Text = "";
            txtPosition.Text = "";
            txtSalary.Text = "";
            // Refresh GridView
            GridView1.DataBind();
        }
    }
```

Output:

Manage Employees

	Id	Name	Position	Salary
Edit Delete	1	John Doe	Software Engineer	60000.00
Edit Delete	2	Jane Smith	Project Manager	75000.00
Edit Delete	3	Michael Brown	Analyst	50000.00
Edit Delete	4	John Doe	Software Engineer	60000.00
Edit Delete	5	Jane Smith	Project Manager	75000.00
Edit Delete	6	Michael Brown	Analyst	50000.00
Edit Delete	7	Ayush Tambe	Project Manager	75000.00

Add New Employee

Name: Ayush Tambe
Position: Project Manager
Salary: 75000
Add Employee

Practical 29

Aim: Create a Web application demonstrating client-side state management using all the techniques.

Webform1.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="ClientsideSessionManagement.a
spx.cs"
Inherits="ClientsideSessionManagement.Cli
entsideSessionManagement" %>

<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="height: 733px; width:
965px">
                <asp:Label ID="Label1"
runat="server" Font-Bold="True" Font-
Size="Larger" Text="Hidden Form Field
"></asp:Label>
                <br />
                <asp:HiddenField
ID="HiddenField1" runat="server"
Value="10000" />
                <br />
       <asp:Button ID="Button2"
runat="server" OnClick="Button2_Click"
Text="View Hidden Form Field" />

      
                <asp:Label ID="Label2"
runat="server" Text="Label"></asp:Label>
                <br />
                <br />
                <asp:Label ID="Label3"
runat="server" Font-Bold="True" Font-
Size="Larger"
Text="ViewState"></asp:Label>
                <br />
                <br />
                <asp:Button ID="Button3"
runat="server" OnClick="Button3_Click"
Text="Create ViewState" />
                  
                <asp:TextBox
ID="TextBox1"
runat="server"></asp:TextBox>
                <br />
                <br />
                <asp:Button ID="Button4"
runat="server" OnClick="Button4_Click"
Text="Get View State" />
                <br />
                <br />
                <asp:Label ID="Label4"
runat="server" Text="Label"></asp:Label>
                <br />
                <br />
                <asp:Label ID="Label5"
runat="server" Font-Bold="True" Font-
Size="Larger" Text="Query
String"></asp:Label>
                <br />
                <br />
                <asp:TextBox
ID="TextBox2"
runat="server"></asp:TextBox>

     
                <asp:Button ID="Button5"
runat="server" OnClick="Button5_Click"
Text="Send" Width="61px" />
                <br />
                <br />
                <h2>Reading/Writing
Cookies</h2><br/>
    Cookie Name  <asp:TextBox
ID="NameField"
runat="server"></asp:TextBox><br/><br/>
                <br/>
    <asp:Button ID="Button1"
runat="server" Text="Write Cookie"
        onclick="Button1_Click"
/>   
                <asp:Button ID="Button6"
runat="server" OnClick="Button6_Click"
Text="Read Cookies" />
                <br/><br/>
                <br/>
                <asp:Label ID="Label6"
runat="server" Text="Label"></asp:Label>
                <br/>
        </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs:

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

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

namespace ClientsideSessionManagement
{
    public partial class
ClientsideSessionManagement :
System.Web.UI.Page
    {
        protected void Page_Load(object
sender, EventArgs e)
        {
            //lnkRead.NavigateUrl =
"WebForm2.aspx?cookie = " +
NameField.Text.ToString();
        }

        protected void
Button2_Click(object sender, EventArgs e)
        {
            Label2.Text =
HiddenField1.Value;
        }

        protected void
Button1_Click(object sender, EventArgs e)
        {

Response.Cookies["name"].Value =
NameField.Text;

Response.Cookies["name"].Expires =
DateTime.Now.AddMinutes(1);
            Label6.Text = "Cookie
Created";
            NameField.Text = "";
        }

        protected void
Button3_Click(object sender, EventArgs e)
        {
            ViewState["userid"] =
TextBox1.Text;
            Response.Write("ViewState
Created Successfully");
        }

        protected void
Button4_Click(object sender, EventArgs e)
        {
            if (ViewState["userid"] !=
null)
            {
                Label4.Text =
ViewState["userid"].ToString();
            }
        }

        protected void
Button5_Click(object sender, EventArgs e)
        {

Response.Redirect("GetQueryStringShow.asp
x?data=" + TextBox2.Text);
        }

        protected void
Button6_Click(object sender, EventArgs e)
        {
            if (Request.Cookies["name"]
== null)
            {
                NameField.Text = "No
cookie found";
            }
            else
            {
                NameField.Text =
Request.Cookies["name"].Value;
            }
        }
    }
}
```

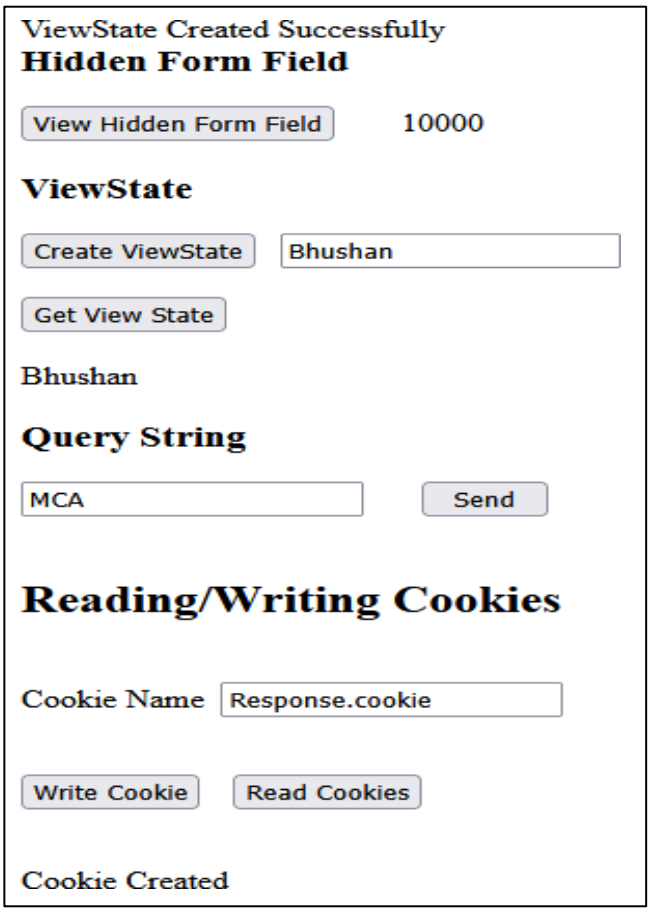
WebForm2.aspx

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="ClientsideSessionManagement.Web
Form2" %>

<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
   
            <asp:Button ID="Button1"
runat="server" OnClick="Button1_Click"
Text="Get Query String" />
            <br />
        </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 ClientsideSessionManagement
{
    public partial class WebForm2 :
System.Web.UI.Page
    {
        protected void Page_Load(object
sender, EventArgs e)
        {

        }

        protected void
Button1_Click(object sender, EventArgs e)
        {

if(Request.QueryString["data"]!=null)
            {
                string data =
Request.QueryString["data"].ToString();
                TextBox1.Text = data;
            }
        }
    }
}
```

Design:



Practical 30

Aim: Create a Web application demonstrating server-side state management using all the techniques.

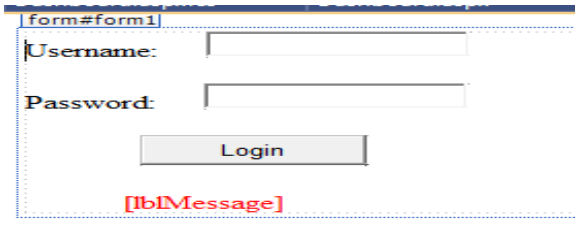
Login.aspx:

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="Login.aspx.cs"
Inherits="ServersideSessionManagement.Log
in" %>
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
</head>
<body>
<form id="form1" runat="server" >
<asp:Label ID="lblUsername"
runat="server"
Text="Username:"></asp:Label>

<asp:TextBox ID="txtUsername"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Label ID="lblPassword"
runat="server"
Text="Password:"></asp:Label>

<asp:TextBox ID="txtPassword"
runat="server"
TextMode="Password"></asp:TextBox>
<br />
<br />



        
;       &nbs
p;
<asp:Button ID="btnLogin"
runat="server" Text="Login"
OnClick="btnLogin_Click" style="margin
left: 0px" Width="110px" />
 <br />
<br />

        
;       &nbs
p;
<asp:Label ID="lblMessage"
runat="server"
ForeColor="Red"></asp:Label>
</form>
</body>
</html>
```

Login.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ServersideSessionManagement
{
public partial class Login :
System.Web.UI.Page
{
protected void Page_Load(object
sender, EventArgs e)
{
}
protected void
btnLogin_Click(object sender, EventArgs
e)
{
string username =
txtUsername.Text;
string password =
txtPassword.Text;
// Hardcoded authentication
(Replace with database check)
if (username == "admin" &&
password == "password")
{
Session["User"] =
username;
Response.Redirect("Dashboard.aspx");
}
else
{
lblMessage.Text =
"Invalid username or password.";
}
}
}
}
```

Design:



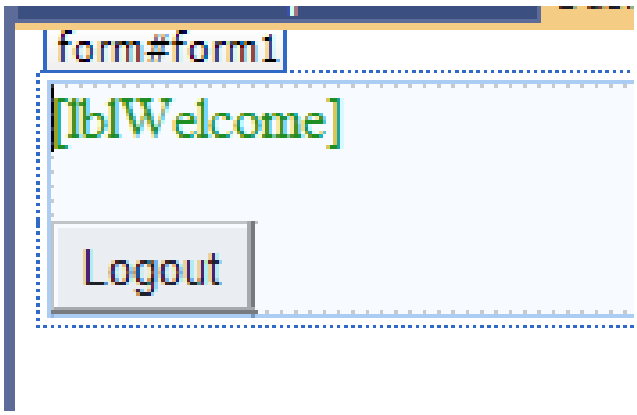
Dashboard.aspx

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="Dashboard.aspx.cs"
Inherits="ServersideSessionManagement.Das
hboard" %>
<!DOCTYPE html>
<html>
<head>
<title>Dashboard</title>
</head>
<body>
<form id="form1" runat="server">
<asp:Label ID="lblWelcome"
runat="server"
ForeColor="Green"></asp:Label>
<br />
<br />
<asp:Button ID="btnLogout"
runat="server" Text="Logout"
OnClick="btnLogout_Click" />
</form>
</body>
</html>
```

Dashboard.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ServersideSessionManagement
{
public partial class Dashboard :
System.Web.UI.Page

{
        protected void Page_Load(object
sender, EventArgs e)
{
if (Session["User"] == null)
{
Response.Redirect("Login.aspx");
}
else
{
lblWelcome.Text =
"Welcome, " + Session["User"].ToString();
}
}
protected void
btnLogout_Click(object sender, EventArgs
e)
{
Session.Clear();
Response.Redirect("Login.aspx");
}
}
}
```

Design:



Practical 31

Design Asp.Net Web Application to demonstrate Ajax controls.

```
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <br />     <br />

            Enter email:-
       
;    
            <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
            <br />   <br />

            Enter phone no:-
     
            <asp:TextBox ID="TextBox3"
runat="server"> </asp:TextBox>
            <br />  <br />

                 
;      &nbs
p;      &nb
sp; 
            <asp:Button ID="Button1"
runat="server" onclick="Button1_Click"
Text="SUBMIT" />
            <br />  <br />

            <asp:ScriptManager
ID="ScriptManager1"
runat="server"></asp:ScriptManager>

            <asp:UpdatePanel
ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Timer
ID="Timer1" runat="server"
Interval="1000" OnTick="Timer1_Tick">
                    </asp:Timer>
                    <br />

                    <asp:Panel ID="Panel1"
runat="server" BackColor="#3399FF"
style="margin-left: 40px" Width="168px"
Height="62px" BorderStyle="Double">
                       &nb
sp;      &n
bsp;      
                    <br                    />
                       &nb
sp;      &n
bsp;     

                    <asp:Label
ID="Label1" runat="server"
Text="Label"></asp:Label>
    </asp:Panel>
    </ContentTemplate>
    </asp:UpdatePanel>

        </div>        <br />
            &n
        bsp;&nb
sp;  
        ; 
        ; Name
        :
        <asp:Label ID="Label2"
runat="server"></asp:Label>
        <br />
        <br />
     &nbs
p; 
;   Email ID :
        <asp:Label ID="Label3"
runat="server"></asp:Label>
        <br />
        <br />
     &nbs
p; 
;   Phone : 
        <asp:Label ID="Label4"
runat="server"></asp:Label>
        </form>
    </body>
    </html>
```

Ajaxcontrols.a

spx using

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

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

        }

        protected void
Button1_Click(object sender,
EventArgs e)
```

Name: Sumit Kadam **Roll no : 16**

```
{
    Label2.Text         =
    TextBox1.Text;
    Label3.Text         =
    TextBox2.Text;
    Label4.Text         =
    TextBox3.Text;
}
```

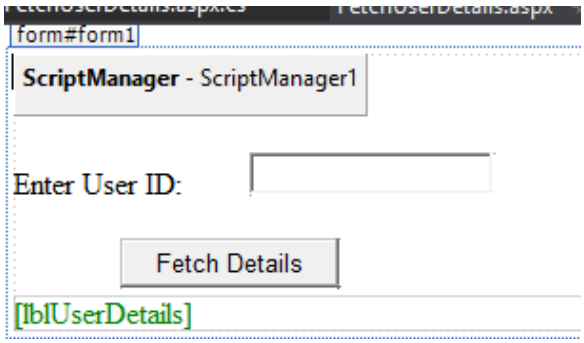
```
        protected void
    Timer1_Tick(object sender, EventArgs e)
        {
Label1.Text =
DateTime.Now.ToLongTimeString();
}
}
}
```

Output :

Practical 32

Implement an AJAX-enabled web form to fetch user details without reloading the page.

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="FetchUserDetails.aspx.c
s"
Inherits="AJAXEnabledApplication.Fe
tchUse rDetails" %>

<!DOCTYPE html>
<html>
<head>
<title>AJAX Fetch User
Details</title></head><body><fo
rm id="form1" runat="server">
<asp:ScriptManager
ID="ScriptManager1"
runat="server"></asp:ScriptManager>

<br />

<asp:Label ID="lblUserId"
runat="server" Text="Enter User
ID:"></asp:Label>

       
; 
<asp:TextBox ID="txtUserId"
runat="server"></asp:TextBox>

<br />
<br />

;
<asp:Button ID="btnFetch"
runat="server" Text="Fetch Details"
OnClick="btnFetch_Click" />
<br />
<asp:UpdatePanel
ID="UpdatePanel1" runat="server">
<ContentTemplate>
<asp:Label
ID="lblUserDetails" runat="server"
ForeColor="Green"></asp:Label>
</ContentTemplate>
</asp:UpdatePanel>
</form>
</body>
</html>
```

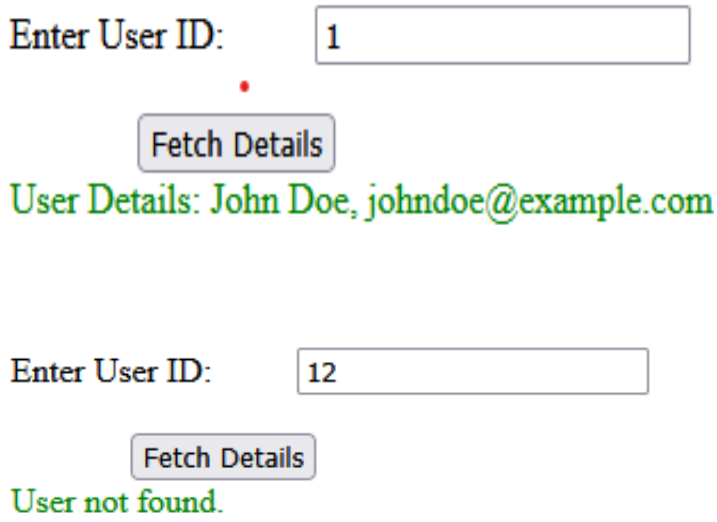



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace AJAXEnabledApplication
{
public partial class FetchUserDetails
: System.Web.UI.Page
{
private Dictionary<int, string>
users = new Dictionary<int, string>
{
{1, "John Doe,
johndoe@example.com"},
{2, "Jane Smith,
janesmith@example.com"},
{3, "Mike Johnson,
mikejohnson@example.com"}
};
protected void Page_Load(object
sender, EventArgs e)
{
}
protected void
btnFetch_Click(object sender, EventArgs
e)
{
if
(int.TryParse(txtUserId.Text, out int
userId) && users.ContainsKey(userId))
{
lblUserDetails.Text =
$"User Details: {users[userId]}";
}
else
{
lblUserDetails.Text =
"User not found.";
        }
    }
```

Name: Sumit Kadam **Roll no : 16**

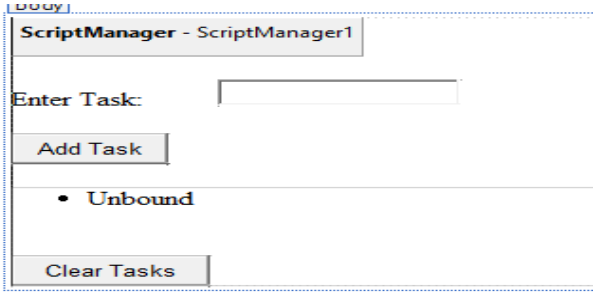
```
}
}
```

Output:


Enter User ID:
1
Fetch Details
User Details: John Doe, johndoe@example.com
Enter User ID:
12
Fetch Details
User not found.

Practical 33

Develop a simple to-do list application using AJAX controls and update panels.



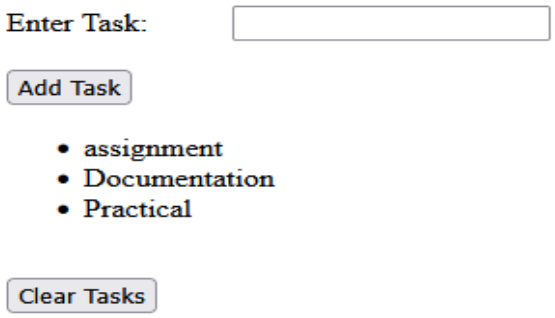

```
<%
@ Page Language="C#"
 AutoEventWireup="true" CodeBehind="To
 DoList.aspx.cs"
 Inherits="SimpleTo_DoListusingAJAXControl
 s.To_DoList" %>
    <!DOCTYPE html>
    <html>
    <head>
        <title>AJAX To-Do List</title>
    </head>
    <body>
        <form id="form1" runat="server">
            <asp:ScriptManager
    ID="ScriptManager1"
    runat="server"></asp:ScriptManager>

            <br />

            <asp:Label ID="lblTask"
    runat="server" Text="Enter
    Task:"></asp:Label>
                 &nb
    sp;   
            <asp:TextBox ID="txtTask"
    runat="server"></asp:TextBox>
             
            <br />
            <br />
            <asp:Button ID="btnAdd"
    runat="server" Text="Add Task"
    OnClick="btnAdd_Click" />

            <br />

            <asp:UpdatePanel
    ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:BulletedList
    ID="lstTasks"
    runat="server"></asp:BulletedList>
                <br />
                <asp:Button ID="btnClear"
    runat="server" Text="Clear Tasks"
    OnClick="btnClear_Click" />
                </ContentTemplate>
            </asp:UpdatePanel>
        </form>
    </body>
    </html>
```

To-DoList.aspx.cs

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

namespace
SimpleTo_DoListusingAJAXControls
{
public partial class To_DoList :
System.Web.UI.Page
{
protected void Page_Load(object
sender, EventArgs e)
{
if (!IsPostBack)
{
Session["Tasks"] = new
List<string>();
}
}
protected void
btnAdd_Click(object sender, EventArgs e)
{
List<string> tasks =
(List<string>)Session["Tasks"];
if
(!string.IsNullOrWhiteSpace(txtTask.Text)
)
{
```

```
        tasks.Add(txtTask.Text);
        Session["Tasks"] = tasks;
        txtTask.Text = "";
        }
        LoadTasks();
        }
        protected void
        btnClear_Click(object sender, EventArgs
        e)
        {
        Session["Tasks"] = new
        List<string>();
        LoadTasks();
        }
        private void LoadTasks()
        {
        lstTasks.Items.Clear();
        List<string> tasks =
        (List<string>)Session["Tasks"];
        foreach (string task in
        tasks)
        {
lstTasks.Items.Add(task);
                }
            }
        }
    }
}
```

Output:



Practical 34

Aim: Demonstrte the use of Global.asax file for handling application-level event

Global.asax:

```
<%@ Application
Codebehind="Global.asax.cs"
Inherits="UseofGlobalasaxFileDemo.Global"
Language="C#" %>
<script runat="server">
void Application_Start(object sender,
EventArgs e)
{
// Runs when the application
starts
Application["TotalVisitors"] = 0;
Application["ActiveUsers"] = 0;
}
void Session_Start(object sender,
EventArgs e)
{
// Runs when a new session starts
if (Application["TotalVisitors"]
== null)


{
Application["TotalVisitors"]
= 0;
}
Application["TotalVisitors"] =
(int)Application["TotalVisitors"] + 1;
if (Application["ActiveUsers"] ==
null)
{
Application["ActiveUsers"] =
0;
}
Application["ActiveUsers"] =
(int)Application["ActiveUsers"] + 1;
}
void Application_Error(object sender,
EventArgs e)
{
// Handles global errors and logs
them
Exception ex =
Server.GetLastError();

System.IO.File.AppendAllText(Server.MapPa
th("~/Errors.log"), ex.ToString());



}
void Session_End(object sender,
EventArgs e)
{
// Runs when a session ends
Application["ActiveUsers"] =
(int)Application["ActiveUsers"] - 1;
}
void Application_End(object sender,
EventArgs e)
{
// Runs when the application
shuts down
}
</script>
```

Globalasax.aspx:

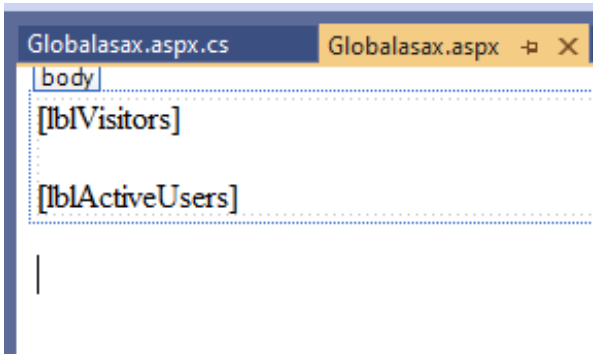
```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="Globalasax.aspx.cs"
Inherits="UseofGlobalasaxFileDemo.Globala
sax" %>
<!DOCTYPE html>
<html>
<head>
<title>Global.asax Example</title>
</head>
<body>
<form id="form1" runat="server">
<asp:Label ID="lblVisitors"
runat="server"></asp:Label>
<br />
<br />
<asp:Label ID="lblActiveUsers"
runat="server"></asp:Label>
</form>
</body>
</html>
```

Globalasax.aspx.cs:

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


namespace UseofGlobalasaxFileDemo
{
public partial class Globalasax :
System.Web.UI.Page
{
protected void Page_Load(object
sender, EventArgs e)
{
lblVisitors.Text = "Total
Visitors: " +
Application["TotalVisitors"].ToString();
lblActiveUsers.Text = "Active
```

```
Users: " +
Application["ActiveUsers"].ToString();
}
}
}
```

Design:



Output:

Total Visitors: 3

Active Users: 3

Module 5: Web Services and WCF

Practical No. 35

Aim: Create a XML-based web service to fetch product details and consume it in a web application.

Code:

WebService1.asmx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
namespace ASMX
{
/// <summary>
/// Summary description for WebService1
/// </summary>
[WebService(Namespace = "ASMX")]
[WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)]
[System.ComponentModel.ToolboxItem(false)]
// To allow this Web Service to be called from
script, using ASP.NET AJAX, uncomment the
following line.
// [System.Web.Script.Services.ScriptService]
public class WebService1 :
System.Web.Services.WebService
{
[WebMethod]
public List<Product> GetAllProducts()
{
return new List<Product>
{
new Product { Id = 1, Name = "Laptop",
Price = 1000 },
new Product { Id = 2, Name = "Phone",
Price = 500 },
new Product { Id = 3, Name = "Tablet",
Price = 300 }
}; } } }
public class Product
{
public int Id { get; set; }
public string Name { get; set; }
public decimal Price { get; set; }
}
```

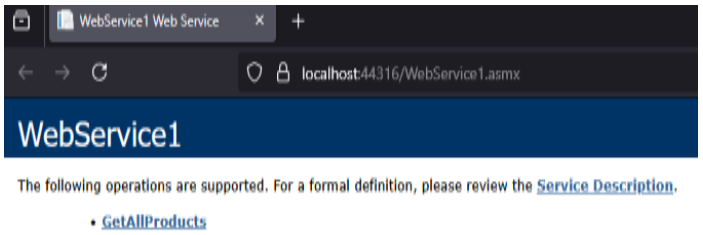
XMLbasedWebService.aspx.cs

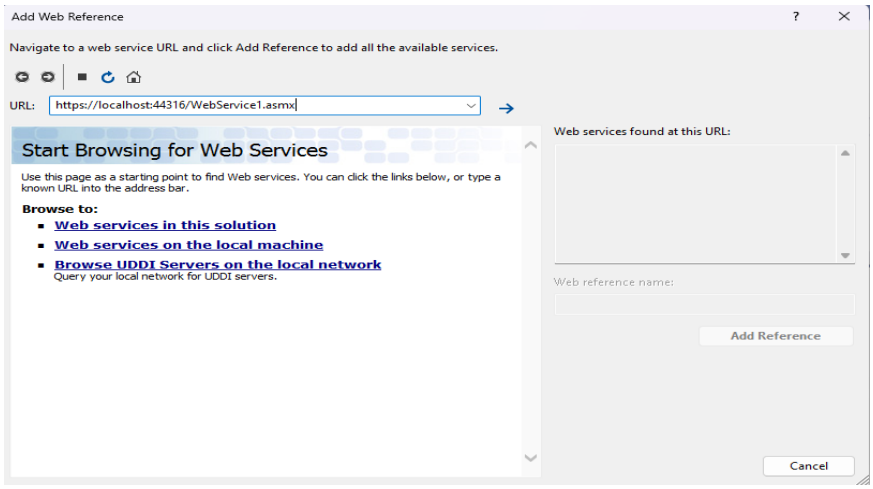
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ASMX
{
public partial class WebForm1 :
System.Web.UI.Page
{
protected void Page_Load(object sender,
EventArgs e)
{
if(!IsPostBack)
{
WebService1 service = new
WebService1();
var products = service.GetAllProducts();
GridView1.DataSource = products;
GridView1.DataBind();
} } }
```

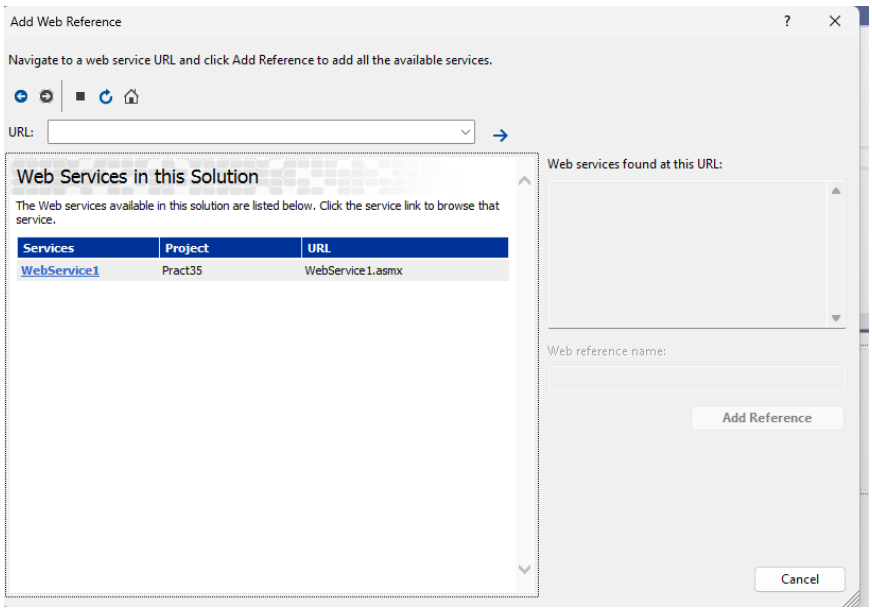
XMLbasedWebService.aspx

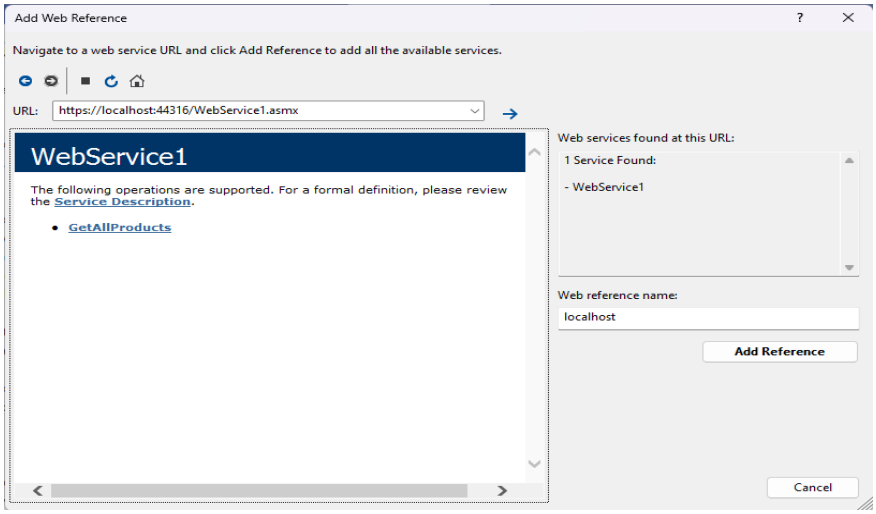
```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="XMLbasedWebService.aspx.cs"
Inherits="ASMX.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body style="height: 277px">
<form id="form1" runat="server">
<div>
<asp:GridView ID="GridView1"
runat="server" AutoGenerateColumns="True">
</asp:GridView>
</div>
</form>
</body>
</html>
```

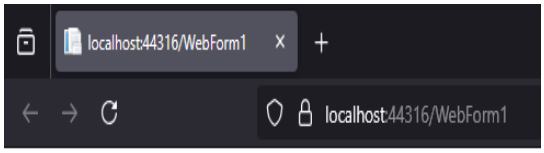
Output:











Id	Name	Price
1	Laptop	1000
2	Phone	500
3	Tablet	300

Practical No. 36

Aim: Create an xml based web service to create a calculator and consume it in a website

Code:

CalculatorService.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
namespace XMLbasedWebService
{
/// <summary>
/// Summary description for CalculatorService
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)]
[System.ComponentModel.ToolboxItem(false)]
// To allow this Web Service to be called from
script, using ASP.NET AJAX, uncomment the
following line.
// [System.Web.Script.Services.ScriptService]
public class CalculatorService :
System.Web.Services.WebService
{
[WebMethod]
public int Add(int a, int b) { return a + b; }
[WebMethod]
public int Subtract(int a, int b)
{ return a - b; }
[WebMethod]
public int Multiply(int a, int b)
{ return a * b; }
[WebMethod]
public double Divide(int a, int b)
{
if (b == 0) throw new
DivideByZeroException("Division by zero.");
return (double)a / b;
}
}
}
```

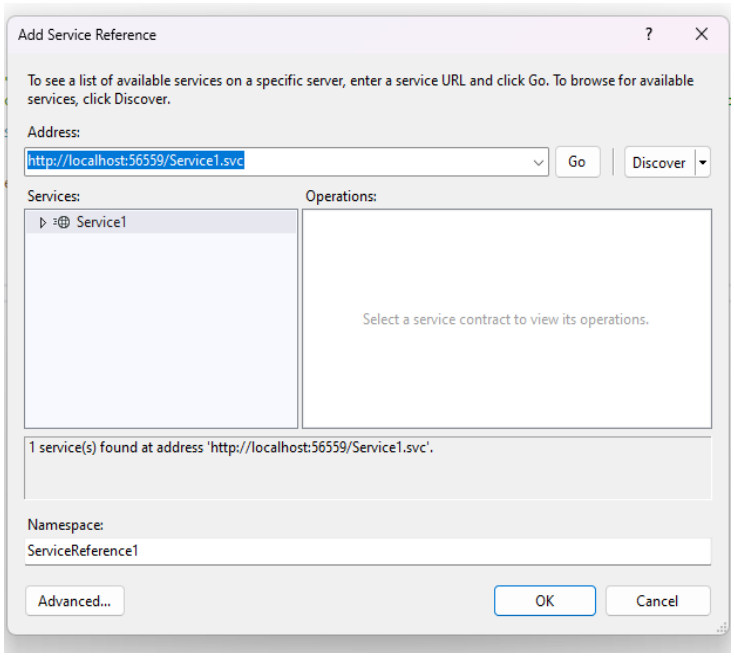
XMLbasedWebServicewithCalculator.aspx

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="XMLbasedWebServicewithCalculator.
aspx.cs"



Inherits="XMLbasedWebService.XMLbasedWebSer
vicewithCalculator" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
</head>
<body>
<form id="form1" runat="server">
<div>
Enter No:-<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
Enter 2nd No:-<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />
<br />
    &
nbsp;
<asp:DropDownList ID="DropDownList1"
runat="server" >
<asp:ListItem>Add</asp:ListItem>
<asp:ListItem>Sub</asp:ListItem>
<asp:ListItem>Multi</asp:ListItem>
<asp:ListItem>Divide</asp:ListItem>
</asp:DropDownList>
<br />
<br />
    &
nbsp;
<asp:Button ID="Button1" runat="server"
Text="Calculate" OnClick="Button1_Click" />
<br />
<br />

<asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

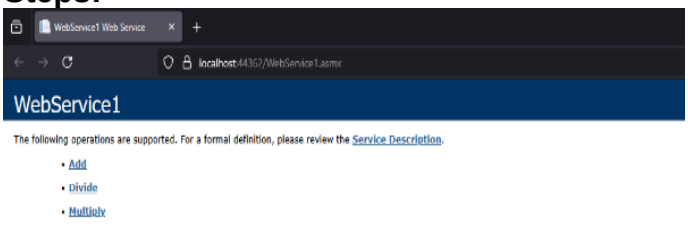
XMLbasedWebServicewithCalculator.aspx.cs

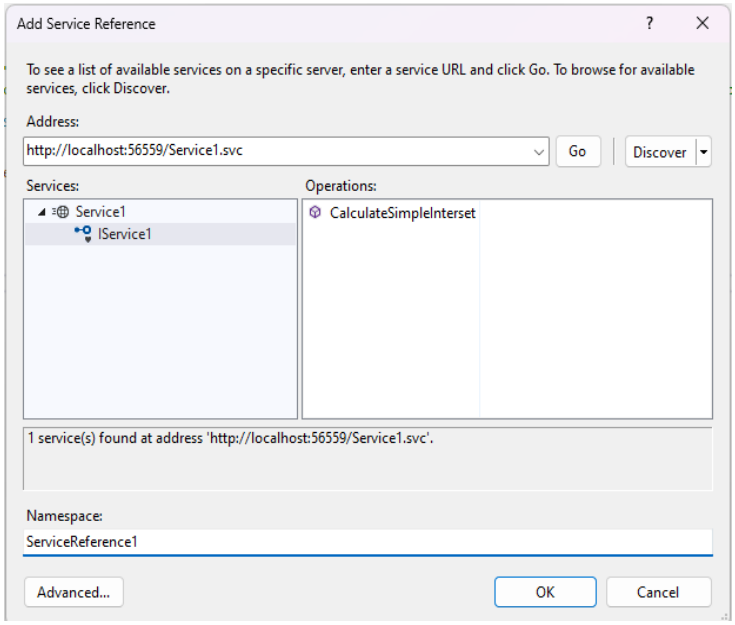
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace XMLbasedWebService
{

public partial class
XMLbasedWebServicewithCalculator :
System.Web.UI.Page
{
protected void Page_Load(object sender,
EventArgs e)
{}
protected void Button1_Click(object sender,
EventArgs e)
{
int a = int.Parse(TextBox1.Text);
int b = int.Parse(TextBox2.Text);
var service = new
```

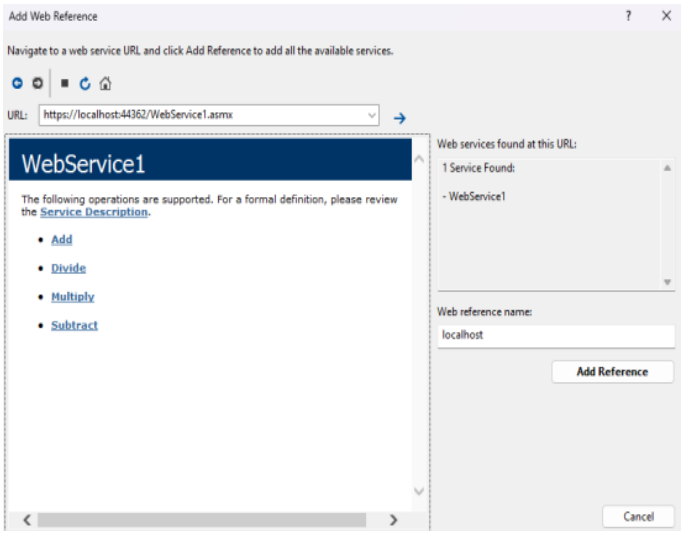
```
XMLbasedWebService.CalculatorService();
string operation =
DropDownList1.SelectedValue;
double result = 0;
switch (operation)
{
case "Add":
result = service.Add(a, b);
break;
case "Subtract":
result = service.Subtract(a, b);
break;
case "Multiply":
result = service.Multiply(a, b);
break;
case "Divide":
result = service.Divide(a, b);
break;
}
Label1.Text = "Result: " + result.ToString();
}
}
}
```



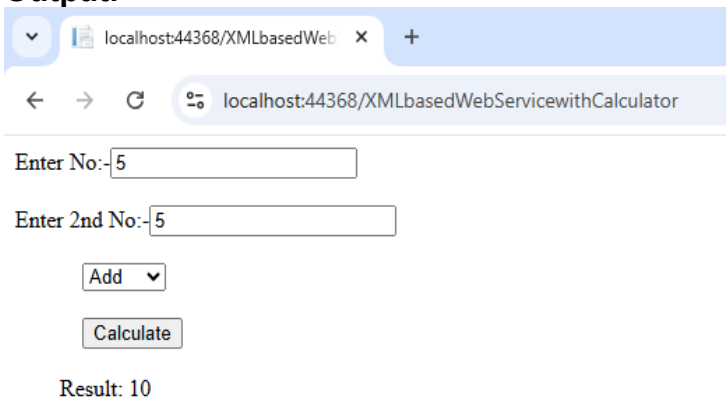
Steps:







Output:



Name: Sumit Kadam **Roll no : 16**

Practical No. 37

Aim : Develop a database-drives WCF service to fetch and update user details.

Database query:

```
create database userdb21;
use userdb21;
create table user2(
id INT PRIMARY KEY IDENTITY,
name NVARCHAR(100),
Email NVARCHAR(100),
Age INT);
insert into user2(NAme,Email,Age) VALUES
('harsh','harsh@example.com',20);
```

Code:

UserDTO.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace WcfService2.DTOs
{
public class UserDTO
{
public int id { get; set; }
public string name { get; set; }
public string Email { get; set; }
public int Age { get; set; }
}  }
```

IService1.cs

```
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using WcfService2.DTOs;
namespace WcfService2
{
[ServiceContract]
public interface IService1
{

[OperationContract]
UserDTO GetUserById(int id);
[OperationContract]
bool UpdateUser(UserDTO user);
}  }
```

Model1.cs

```
namespace WcfService2
{
using System;
using System.Collections.Generic;
public partial class User2
{
public int Id { get; set; }
public string Name { get; set; }
public string Email { get; set; }
public Nullable<int> Age { get; set; }
}  }
```

Service1.svc

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using WcfService2.DTOs;
namespace WcfService2
{
// NOTE: You can use the "Rename" command on the
"Refactor" menu to change the class name "Service1" in
code, svc and config file together.
// NOTE: In order to launch WCF Test Client for
testing this service, please select Service1.svc or
Service1.svc.cs at the Solution Explorer and start
debugging.
public class Service1 : IService1
{
userdb21Entities db = new userdb21Entities();
public UserDTO GetUserById(int id)
{
var user = db.user2.Find(id);
if (user == null) return null;
return new UserDTO
{
id = user.id,
name = user.name,
Email = user.Email,
Age = (int)user.Age
};  }


public bool UpdateUser(UserDTO user)
{
var existing = db.user2.Find(user.id);
if (existing == null) return false;
existing.name = user.name;
existing.Email = user.Email;
existing.Age = user.Age;
db.SaveChanges();
return true;
}
}
}
```

user2.cs

```
namespace WcfService2
{
using System;
```

```
using System.Collections.Generic;
public partial class user2
{
public int id { get; set; }
public string name { get; set; }
public string Email { get; set; }
public Nullable<int> Age { get; set; }
}
}
```

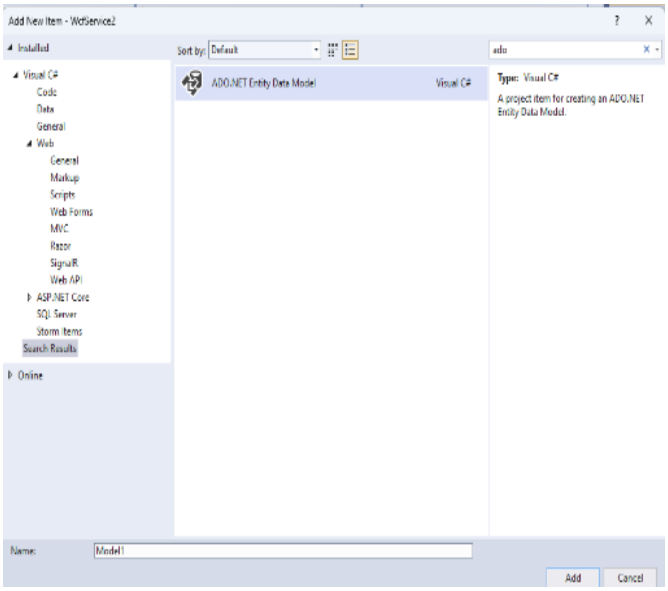
DatabseDrivenWCFService.aspx

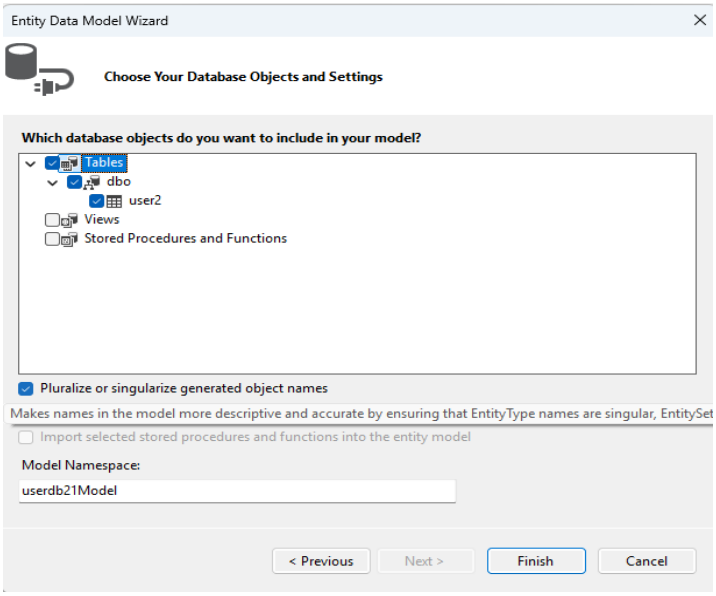
```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeBehind="DatabseDrivenWCFService.aspx.
cs" Inherits="WcfService2.WebForm1" %>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
Enter User ID:
<asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
<br />
<br />
Name:

;
<asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
<br />

<br />
Email:
;      &nbs
p;
<asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
<br />
<br />
Age:      &
nbsp;

<asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1"
runat="server" Text="Update" />
<br />
<br />
<asp:Label ID="Label1"
runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

DatabseDrivenWCFService.aspx.cs

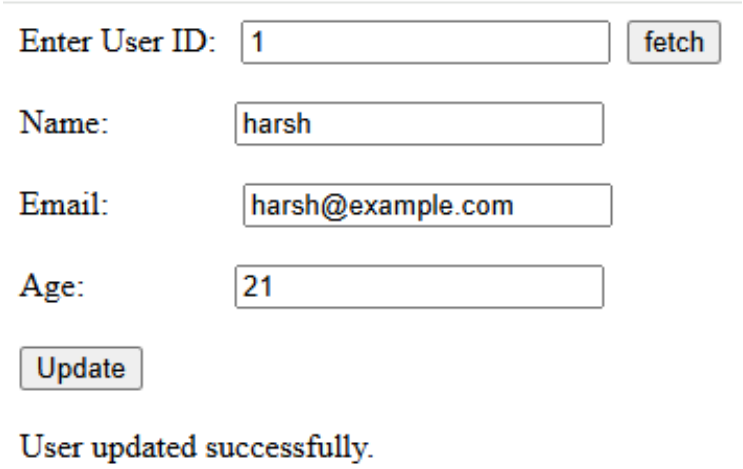
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WcfService2.DTOs;
namespace WcfService2
{
public partial class WebForm1 : System.Web.UI.Page
{
Service1 userService = new Service1();
protected void Page_Load(object sender,
EventArgs e)
{
}
protected void Button2_Click(object sender,
EventArgs e)
{
int id = int.Parse(TextBox1.Text);
var user = userService.GetUserById(id);
if (user != null)
{
TextBox2.Text = user.name;
TextBox3.Text = user.Email;
TextBox4.Text = user.Age.ToString();
Label1.Text = "User found.";
}
else
{

Label1.Text = "User not found.";
}
}
protected void Button1_Click(object sender,
EventArgs e)
{
var user = new UserDTO
{
id = int.Parse(TextBox1.Text),
name = TextBox2.Text,
Email = TextBox3.Text,
Age = int.Parse(TextBox4.Text)
};
bool success = userService.UpdateUser(user);
Label1.Text = success ? "User updated
successfully." : "Update failed.";
} }}
```

Steps:





Output:



Name: Sumit Kadam **Roll no : 16**

Practical No. 38

Aim: Create and consume the WCF service to calculate simple interest

Code:

Service1.svc

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
namespace WcfService1
{
// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.
// NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc or Service1.svc.cs at the Solution Explorer and start debugging.
public class Service1 : IService1
{
public double CalculateSimpleInterest(double principal, double rate, double time)
{
return (principal * rate * time) / 100;
} } }
```

Iservice1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
namespace WcfService1
{
// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.
[ServiceContract]
public interface IService1
{
[OperationContract]
double
CalculateSimpleInterest(double principal, double rate, double time);
} }
```

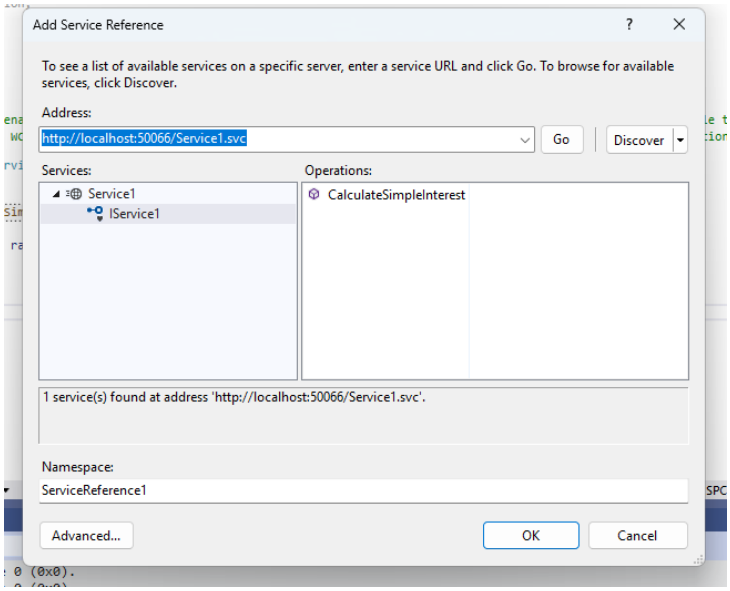
SimpleInterest.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="SimpleInterest.aspx.cs" Inherits="WcfService1.SimpleInterest" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
Enter Principal:                   
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<br />
<br />
Enter Rate of Interest:       
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<br />
<br />
Enter Year:                     
<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
<br />
<br />
<br />
                     
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Calculate Interest" />
<br />
<br />
<br />
                     
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

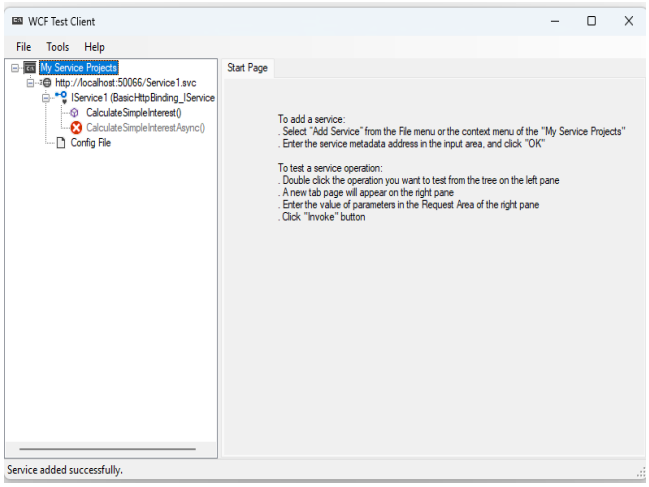
SimpleInterest.aspx.cs

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

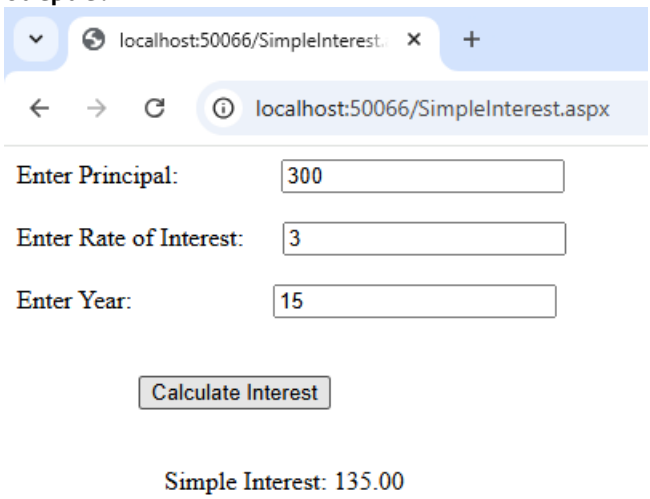
```
using System.Web.UI.WebControls;
namespace WcfService1
{
public partial class SimpleInterest :
System.Web.UI.Page
{
CalculateSimpleInterest.Service1Client
obj = new
CalculateSimpleInterest.Service1Client();
protected void Page_Load(object
sender, EventArgs e)
{
}
protected void
Button1_Click(object sender, EventArgs e)
{
double principal =
Convert.ToDouble(TextBox1.Text);
double rate =
Convert.ToDouble(TextBox2.Text);
double time =
Convert.ToDouble(TextBox3.Text);
double interest=(principal *
rate * time)/ 100;
Label1.Text = "Simple
Interest: " + interest.ToString("F2");
}  }
}
```



Step:



Output:

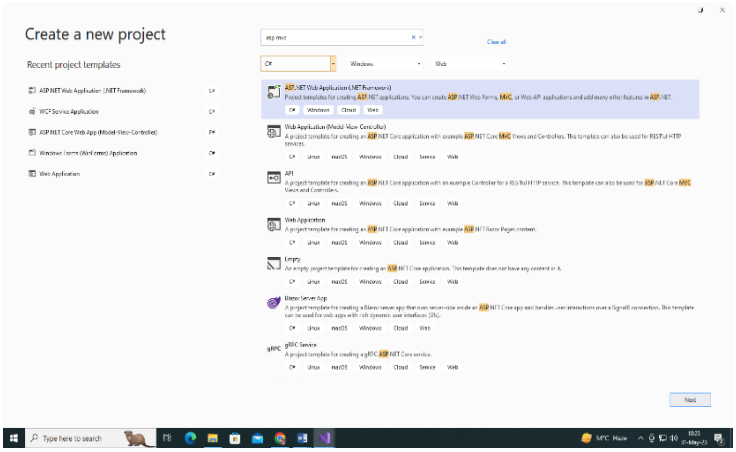


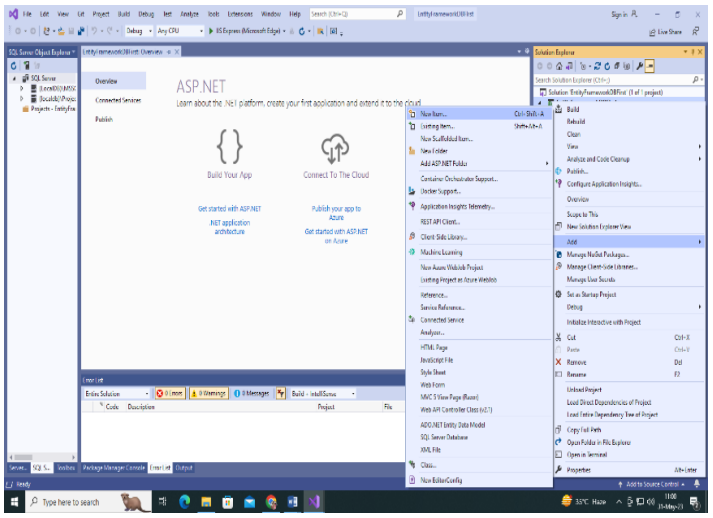
Module 6: ASP.NET Core MVC Framework

Practical No: 39

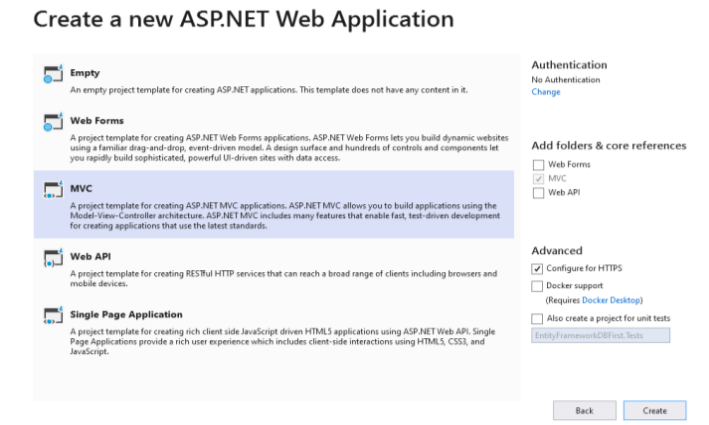
Aim:- Build websites to demonstrate the working of entity framework

1. Create a new project

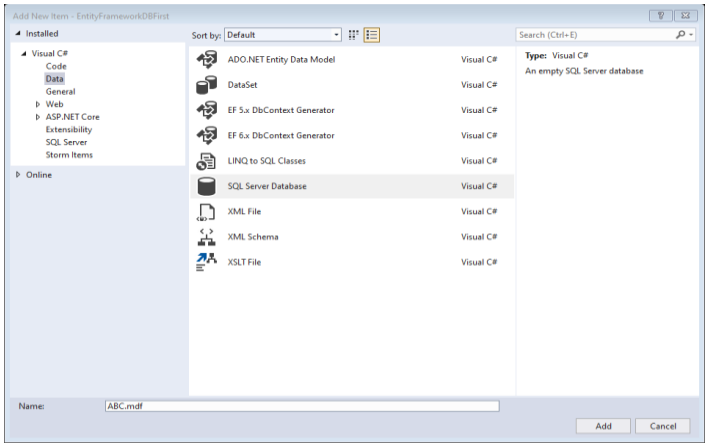


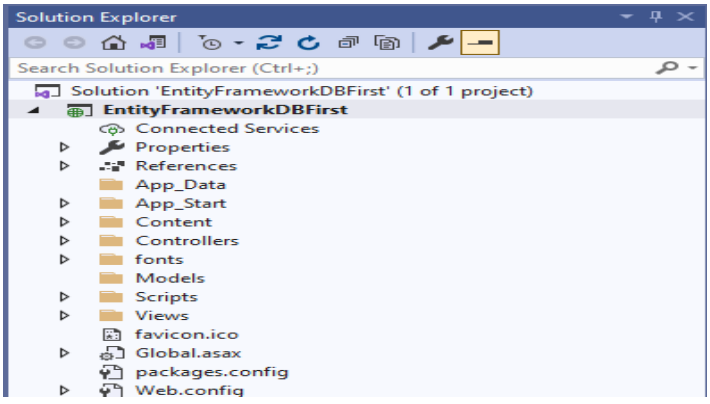


2. Select MVC and click on Create

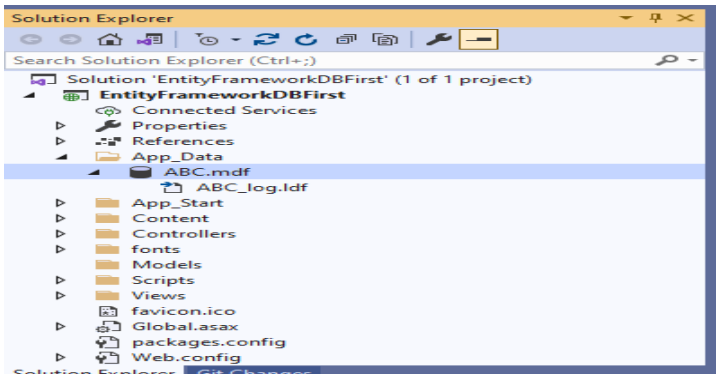


Select SQL Server Database





Give the name ABC.mdf
Click on add



Right Click on project->add->new Item ->

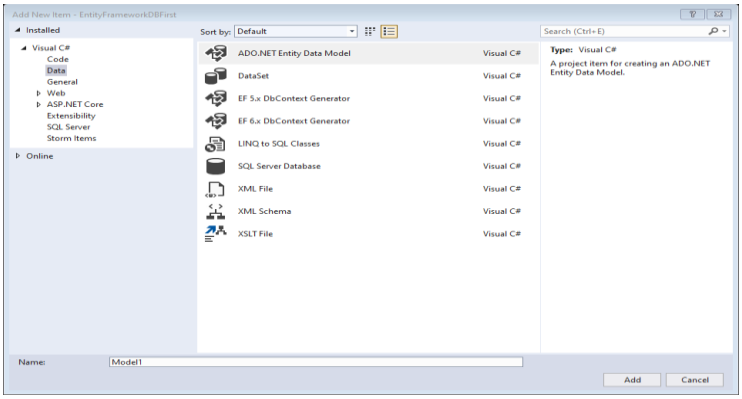
Double click on ABC.mdf

Name: Sumit Kadam **Roll no : 16**

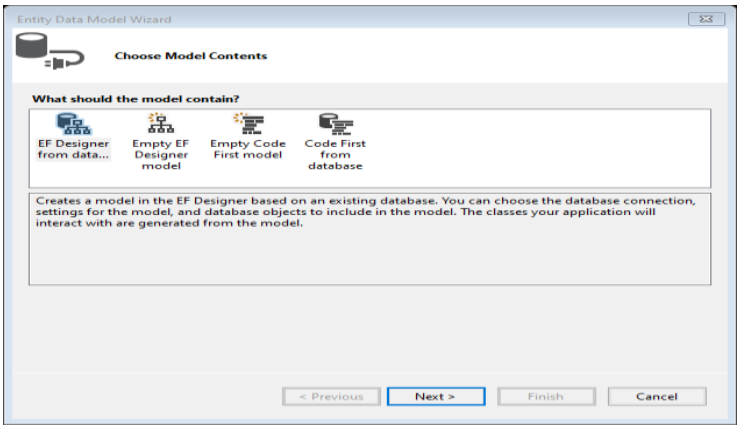
In server explorer Right click on ABC.mdf->new Query->Write Query to create table (table should contain Primary key compulsory)

```
CREATE TABLE [dbo].[product] (
[pid] INT NOT NULL,
[pname] VARCHAR (100) NULL,
[pcost] VARCHAR (100) NULL,
PRIMARY KEY CLUSTERED ([pid] ASC)
);
```

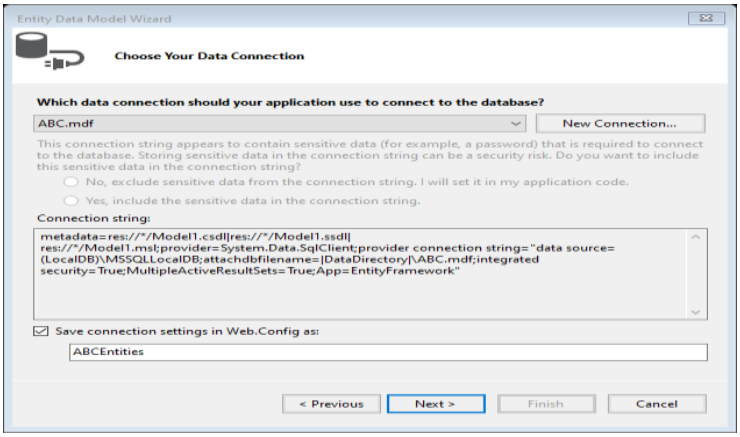
Right click on project -> add -> new item



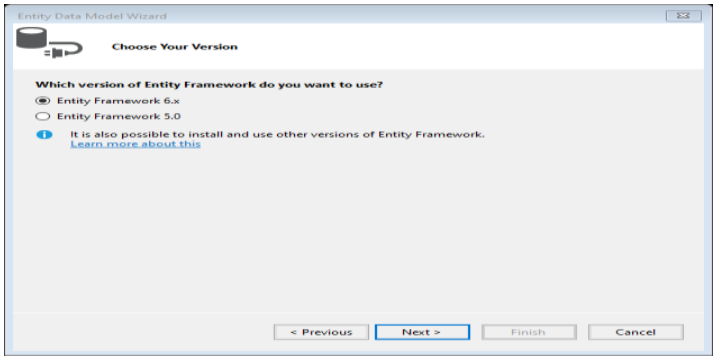
Click on add



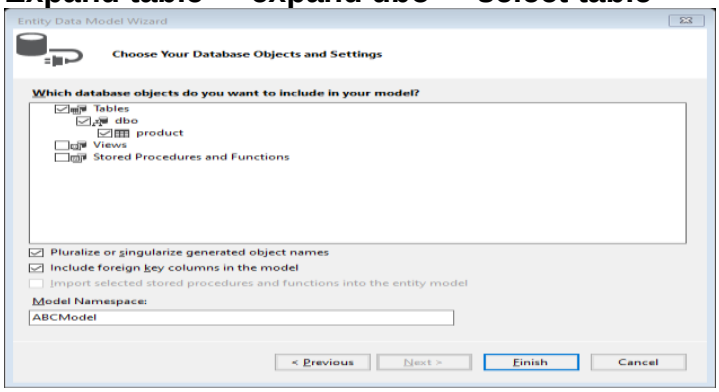
Click on next



Click on next

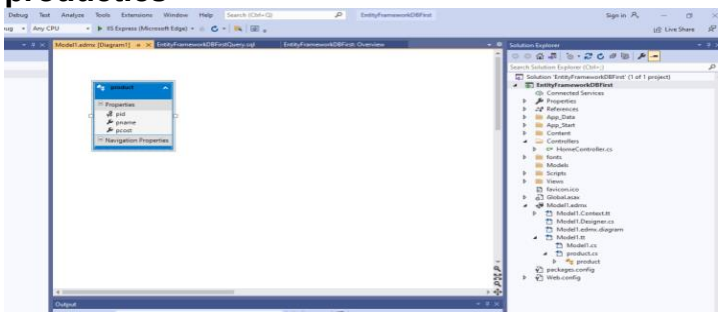


Click on next
Expand table -> expand dbo -> select table

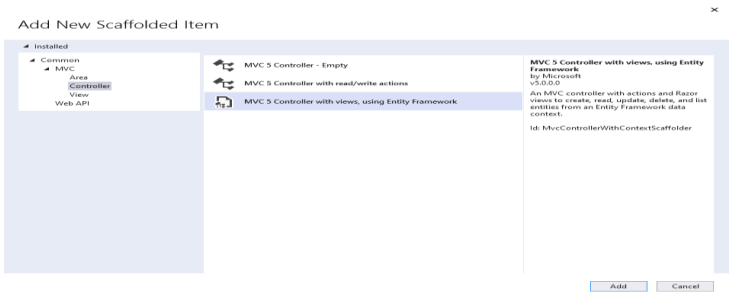


Click on Finish

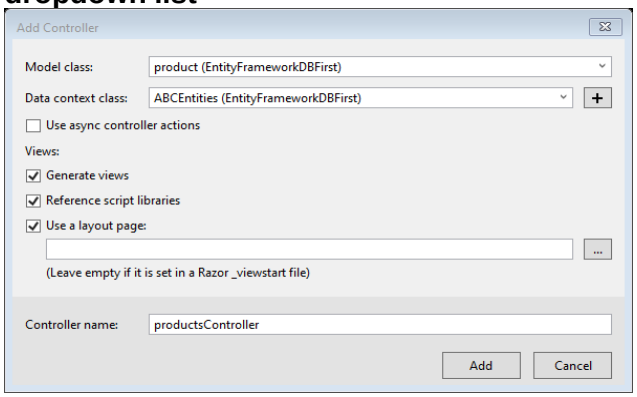
It will auto generate code – Model1.edmx – Expand
Model1.edmx – Expand Model1.tt -> it will contain product.cs



Right click on Controller->add->Controller
Select MVC5 Controller with views, using Entity Framework

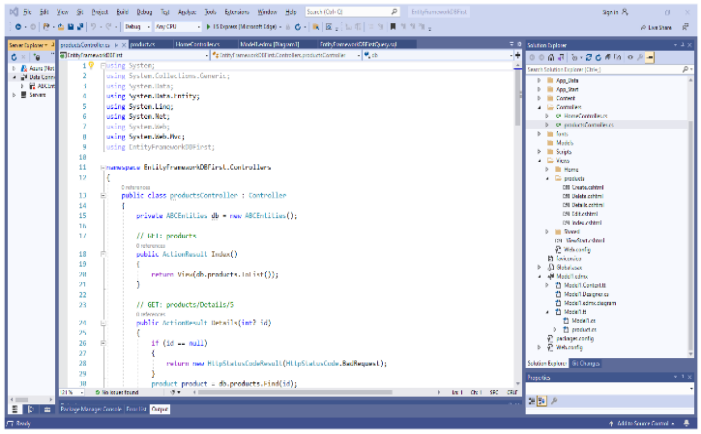


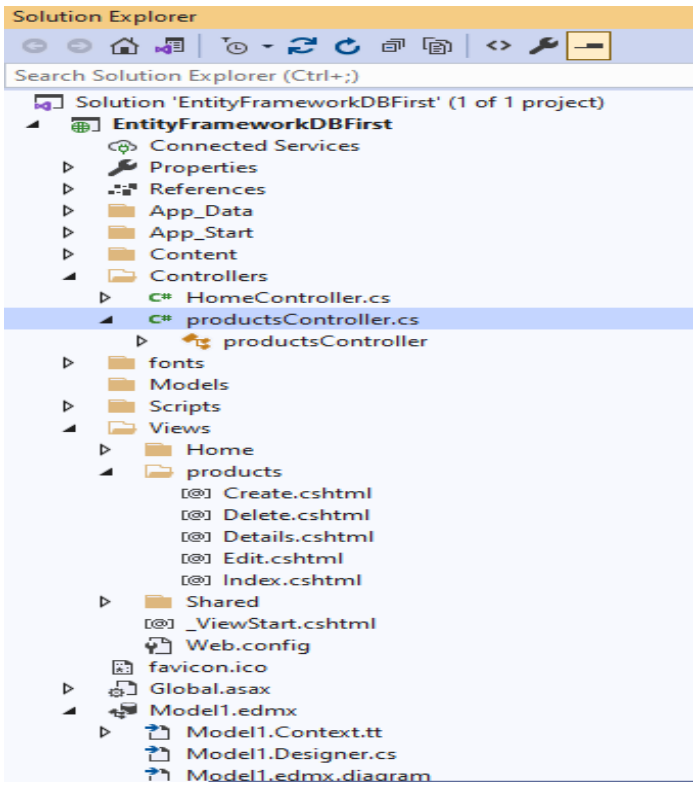
Click on add select the displayed items from dropdown list



Click on add (If any error re build project then add controller)

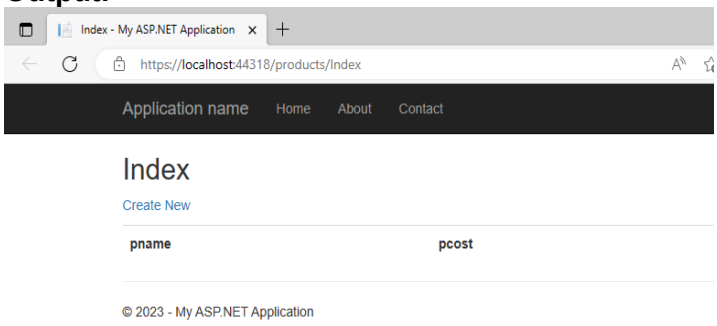
It will auto generate code for productsController.cs and also generate views for product.



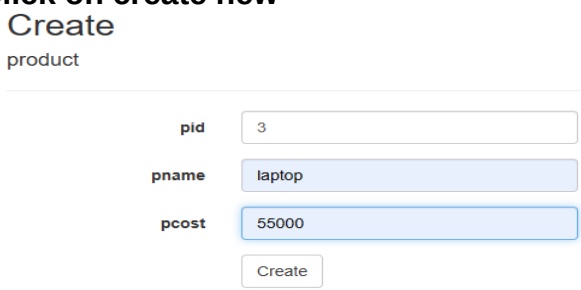


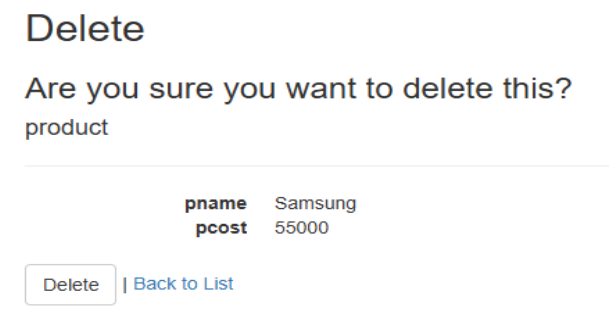
Expand View Products -> Index.cshtml-> Run project

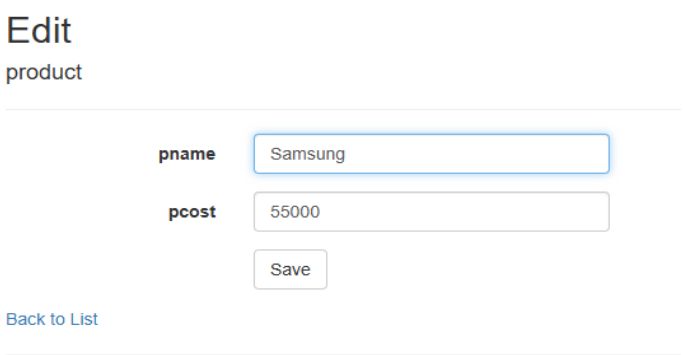
Output:



Click on create new







Practical No: 40

Aim: Create an simple ASP.NET Core MVC application with models, views, and controllers to display book info.

Code:

In **model** make

Book.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace Prac40.Models
{
public class Book
{
public int Id { get; set; }
public string Title { get; set; }
public string Author { get; set; }
public string Genre { get; set; }
public int Year { get; set; }
}
}
```

In **Contollers** folder create file with name

BooksController.cs

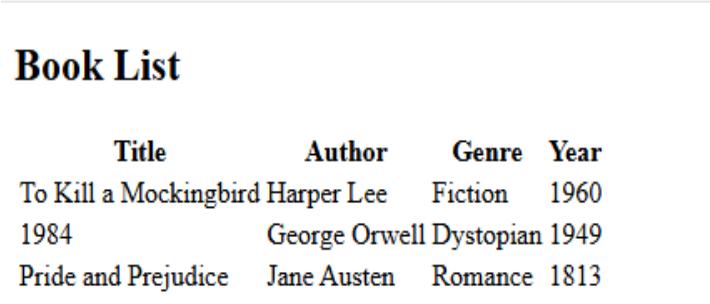
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Prac40.Models;
namespace Prac40.Controllers
{
public class BooksController : Controller
{
public ActionResult Index()
{
var books = new List<Book>
{
new Book { Id= 1,Title= "To Kill a Mockingbird" ,
Author=
"Harper Lee", Genre= "Fiction", Year= 1960 },
new Book { Id= 2, Title= "1984", Author= "George
Orwell"
, Genre= "Dystopian", Year= 1949 },
new Book { Id= 3, Title= "Pride and Prejudice" , Author=
"Jane Austen", Genre= "Romance", Year= 1813 }
};
return View(books);
}
}
}
```

In **View/Books** create

index.cshtml

```
@{
Layout = null;
}
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title></title>
</head>
<body>
<div>
@model List<Prac40.Models.Book>
<h2> Book List</h2>
<table class="table table-bordered">
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Genre</th>
<th>Year</th>
</tr>
</thead>
<tbody>
@foreach (var book in Model)
{
<tr>
<td>@book.Title</td>
<td>@book.Author</td>
<td>@book.Genre</td>
<td>@book.Year</td>
</tr>
}
</tbody>
</table>
</div>
</body>
</html>
```

Output:



Book List

Title	Author	Genre	Year
To Kill a Mockingbird	Harper Lee	Fiction	1960
1984	George Orwell	Dystopian	1949
Pride and Prejudice	Jane Austen	Romance	1813

Practical No: 41

Aim: Design a form with data annotations for validation and display appropriate error messages.

Code:

In **models** create file
UserFormModel.cs

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

namespace Prac41.Models
{
public class UserFormModel

{
[Required(ErrorMessage = "Name is required.")]
[StringLength(100, ErrorMessage = "Name can't be longer
than 100 characters.")]
public string Name { get; set; }

[Required(ErrorMessage = "Email is required.")]
[EmailAddress(ErrorMessage = "Enter a valid email.")]
public string Email { get; set; }

[Required(ErrorMessage = "Age is required.")]
[Range(18, 100, ErrorMessage = "Age must be between 18
and
100.      ")]
101.
public int Age { get; set; }
}
}
```

In **Contollers folder** create file with name
UserController.cs

```
using Prac41.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace Prac41.Controllers
{
public class UserController : Controller
{
public ActionResult Create()
{
return View();
}
[HttpPost]
public ActionResult Create(UserFormModel model)
{
if (ModelState.IsValid)
{
return RedirectToAction("Success");
}
return View(model);
}
public ActionResult Success()
{
return View();
}
}
}
```

In **View/User** create
Create.cshtml

```
@{
Layout = null;
}
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title></title>
</head>
<body>
<div>
@model Prac41.Models.UserFormModel
<h2>User Form</h2>
@using (Html.BeginForm())
{
@Html.AntiForgeryToken()
<div>
@Html.LabelFor(m => m.Name)
@Html.TextBoxFor(m => m.Name)
@Html.ValidationMessageFor(m => m.Name, "", new {
@class = "text-danger" })
</div>
<div>
@Html.LabelFor(m => m.Email)
@Html.TextBoxFor(m => m.Email)
@Html.ValidationMessageFor(m => m.Email, "", new {
@class = "text-danger" })
</div>
<div>
@Html.LabelFor(m => m.Age)
@Html.TextBoxFor(m => m.Age)
@Html.ValidationMessageFor(m => m.Age, "", new {
@class = "text-danger" })
</div>
<button type="submit">Submit</button>
}
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}

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

Success.cshtml

```
@{
Layout = null;
}
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title></title>
</head>
<body>
<div>
<h2>Form submitted successfully!</h2>
</div>
</body>
</html>
```

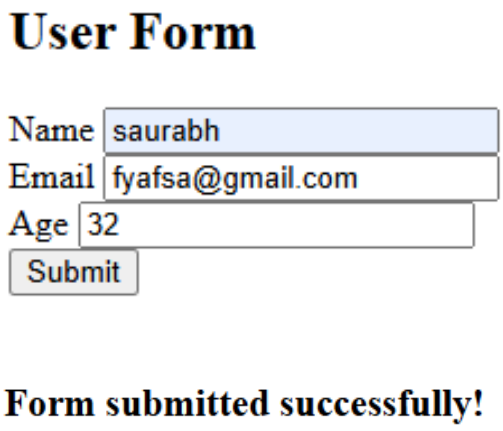
In App_Start folder open BundleConfig.cs

```
bundles.Add(new
ScriptBundle("~/bundles/jqueryval").Include(
"~/Scripts/jquery.validate*"));
```

In **View/Web.config** (under <configuration>):

```
<appSettings>
<add key="ClientValidationEnabled" value="true" />
<add key="UnobtrusiveJavaScriptEnabled" value="true" />
</appSettings>
```

Output:



Form submitted successfully!

Practical No: 42

Aim: Implement CRUD operations using Entity Framework Code-First approach for emp data.

Code:

Install Entity Framework (if not already installed):

Open Package manager console and type

Install-Package EntityFramework

In model add file

Employee.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;
namespace EmpCRUDApp.Models
{
public class Employee
{
[Key]
public int EmpId { get; set; }
[Required]
[StringLength(100)]
public string Name { get; set; }
[Required]
public string Department { get; set; }
[Required]
[Range(10000, 100000)]
public decimal Salary { get; set; }
}
}
```

EmpDbContext.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;
namespace EmpCRUDApp.Models
{
public class EmpDbContext : DbContext
{
public DbSet<Employee> Employees { get; set; }
public EmpDbContext() : base("DefaultConnection") { }
}
}
```

Open **Web.config** in Views folder

Add under

<configuration><connectionStrings>:

CopyEdit

```
<connectionStrings>
<add name="DefaultConnection"
connectionString="Data
Source=(localdb)\MSSQLLocalDB;Initial
Catalog=EmpDb;Integrated Security=True;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

Open Package Manager Console

Run:

Enable-Migrations

Add-Migration InitialCreate

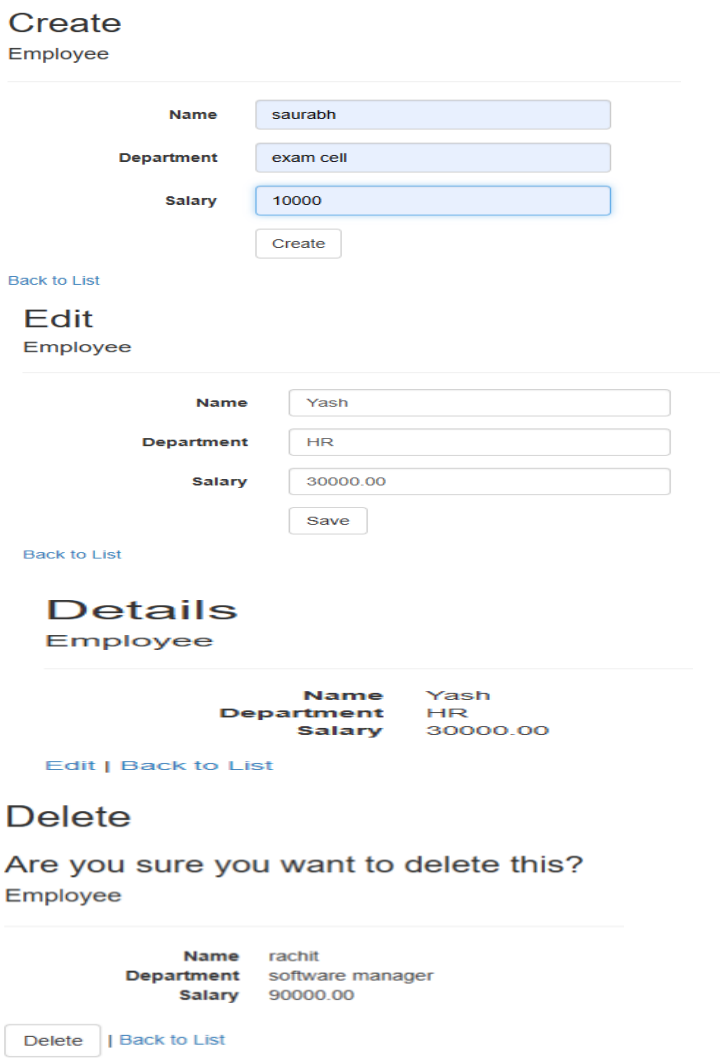
Update-Database

In **Controller folder** make file

1. Right-click the **Controllers** folder → **Add > Controller**
2. Choose **MVC 5 Controller with views, using Entity Framework**
3. Select:
   - Model class: Employee
   - Data context class: EmpDbContext
4. Click **Add**

Output:

https://localhost:44333/Employees



Practical No: 43

Aim: Design a web page to Create partial views

Code:

In the **Models** folder:
Employee.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;
namespace PartialViewDemo.Models
{
public class Employee
{
[Key]
public int EmpId { get; set; }
[Required]
public string Name { get; set; }
public string Department { get; set; }
public decimal Salary { get; set; }
}
}
```

In the **Controllers** folder:
EmployeeController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using PartialViewDemo.Models;
namespace PartialViewDemo.Controllers
{
public class EmployeeController : Controller
{
public ActionResult Create()
{
return View();
}
[HttpPost]
public ActionResult Create(Employee emp)
{
if (ModelState.IsValid)
{
// Save logic here (to DB or memory)
ViewBag.Message = "Employee created!";
ModelState.Clear();
}
return View();
}
}
}
```

In View/_EmployeeForm.cshtml

```
@model PartialViewDemo.Models.Employee
<div>
<div class="form-group">
@Html.LabelFor(m => m.Name)
@Html.TextBoxFor(m => m.Name, new { @class = "form
control" })
</div>
<div class="form-group">
@Html.LabelFor(m => m.Department)
@Html.TextBoxFor(m => m.Department, new { @class =
"form-control" })
</div>
<div class="form-group">
@Html.LabelFor(m => m.Salary)
@Html.TextBoxFor(m => m.Salary, new { @class = "form
control" })
</div>
</div>
```

In View/Create.cshtml

```
@model PartialViewDemo.Models.Employee
<h2>Create Employee</h2>
@using (Html.BeginForm())
{
@Html.AntiForgeryToken()
@Html.Partial("_EmployeeForm", Model)
<button type="submit" class="btn btn-primary">Save</button>
}
@if (ViewBag.Message != null)
{
<div class="alert alert-success">@ViewBag.Message</div>
}
```

Output: