Advanced Web Technologies(AWT) Lab (MCAL25)

INDEX

Name of the faculty: Ganesh Bhagwat

7	Design a webpage to demonstrate the working of a simple stored procedure.		CO2	
8	Design a webpage to demonstrate the working of parameterized stored procedure.		CO2	
9	Design a webpage to display the use of LINQ.		CO2	
10	Build websites to demonstrate the working of entity frameworks in dot net.		CO3	
11	Design Web Applications using Client Side Session Management		C03	
12	Design Web Applications using Server Side Session Management Techniques		CO3	
13	Build a web page using AJAX Controls.		CO3	
14	Build a web application to create and use web service in ASP.net		CO3	

15	Build a web application to create and WCF service in ASP.net		CO3	
16	Design web application using MVC framework		CO4	

PRACTICAL 1: Design a Web Application for an Organization with Registration forms and advanced controls.

CODE:

RegistrationForm.aspx:

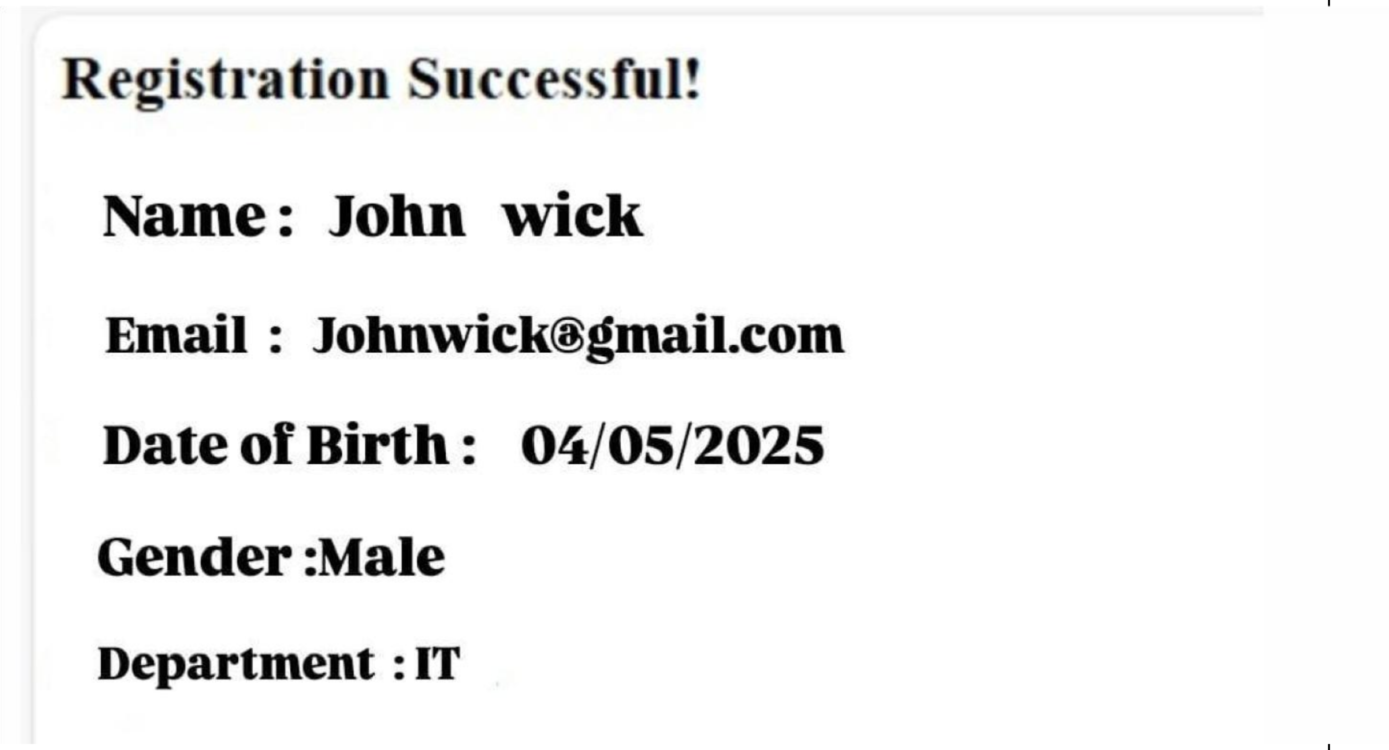
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="Practical1.RegistrationForm" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Registration Form</title>
  <link rel="stylesheet" type="text/css" href="style1.css" />
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Registration Form</h2>
      <table>
        <tr>
          <td><label for="txtFirstName">First Name:</label></td>
          <td><asp:TextBox ID="txtFirstName" runat="server" CssClass="form-
control"></asp:TextBox></td>
        </tr>
        <tr>
          <td><label for="txtLastName">Last Name:</label></td>
          <td><asp:TextBox ID="txtLastName" runat="server" CssClass="form-
control"></asp:TextBox></td>
        </tr>
        <tr>
          <td><label for="txtEmail">Email:</label></td>
          <td><asp:TextBox ID="txtEmail" runat="server" CssClass="form-
control"></asp:TextBox></td>
        </tr>
        <tr>
          <td><label for="txtDOB">Date of Birth:</label></td>
          <td><asp:TextBox ID="txtDOB" runat="server" TextMode="Date" CssClass="form-
control"></asp:TextBox></td>
        </tr>
        <tr>
          <td><label for="ddlGender">Gender:</label></td>
          <td>
            <asp:DropDownList ID="ddlGender" runat="server" CssClass="form-control">
              <asp:ListItem Text="Select Gender" Value=""></asp:ListItem>
               <asp:ListItem Text="Male" Value="Male"></asp:ListItem>
              <asp:ListItem Text="Female" Value="Female"></asp:ListItem>
              <asp:ListItem Text="Other" Value="Other"></asp:ListItem>
```

```
                    </asp:DropDownList>
                </td>
            </tr>
            <tr>
                <td><label>Department:</label></td>
                <td>
                    <asp:RadioButtonList ID="rblDepartment" runat="server">
                        <asp:ListItem Text="HR" Value="HR"></asp:ListItem>
                       <asp:ListItem Text="IT" Value="IT"></asp:ListItem>
                        <asp:ListItem        Text="Finance"         Value="Finance"></asp:ListItem>
                    </asp:RadioButtonList>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:CheckBox ID="chkTerms" runat="server" Text="I accept the terms and
conditions!" />
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <asp:Button ID="btnSubmit" runat="server" Text="Register"
OnClick="btnSubmit_Click" />
                </td>
            </tr>
        </table>
      </div>
   </form>
</body>
</html>
```

RegistrationForm.aspx.cs:

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

namespace Practical1
{
        public partial class RegistrationForm : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {

                }
        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            if (chkTerms.Checked)
            {
                string              firstName              =
                txtFirstName.Text; string lastName =
                txtLastName.Text;   string   email   =
                txtEmail.Text;      string      dob      =
                txtDOB.Text;
                string  gender  =  ddlGender.SelectedValue;  string
                department = rblDepartment.SelectedValue;

                // Display confirmation message
                Response.Write($"<h3>Registration Successful!</h3>");
                Response.Write($"<p>Name: {firstName} {lastName}</p>");
                Response.Write($"<p>Email: {email}</p>");
                Response.Write($"<p>Date of Birth: {dob}</p>");
                Response.Write($"<p>Gender: {gender}</p>");
                Response.Write($"<p>Department: {department}</p>");
            }
            else
            {
                Response.Write("<h3 style='color:red'>Please accept the terms and conditions.</h3>");
            }

        }
    }
}
```

OUTPUT:



Registration Successful!

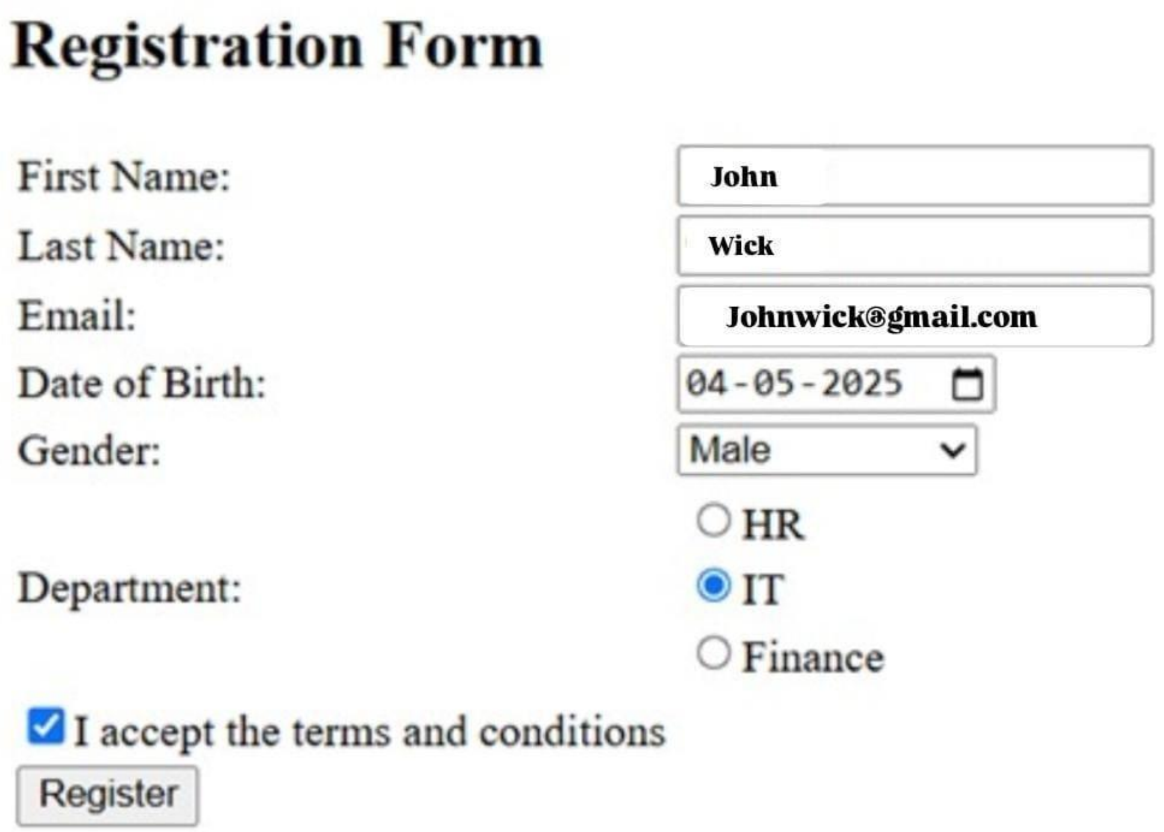
Name : John wick

Email : Johnwick@gmail.com

Date of Birth : 04/05/2025

Gender :Male

Department : IT



Registration Form

First Name:	John
Last Name:	Wick
Email:	Johnwick@gmail.com
Date of Birth:	04-05-2025
Gender:	Male
Department:	○ HR ● IT ○ Finance

☑ I accept the terms and conditions

PRACTICAL 2: Create a website using the master page concept.

CODE:

MasterPage.master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="MasterPage.master.cs"
Inherits="Practical2.MasterPage" %>

<!DOCTYPE html>
<html>
<head>
   <title>Master</title>
   <link rel="stylesheet" href="styles.css" />
</head>
<body>
   <div class="header">
      <h1>Welcome to My Website!</h1>
      <nav>
         <a href="Home.aspx">Home</a>
         <a href="About.aspx">About</a>
         <a href="Contact.aspx">Contact</a>
      </nav>
   </div>

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

   <div class="footer">
      <p>&copy; 2025 My Website. All Rights Reserved.</p> </div>
</body>
</html>
```

Home.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
AutoEventWireup="true" CodeBehind="Home.aspx.cs" Inherits="Practical2.Home" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
   <h2>Welcome to the Home Page</h2>
   <p>This is the main content of the Home page.</p> </asp:Content>
```

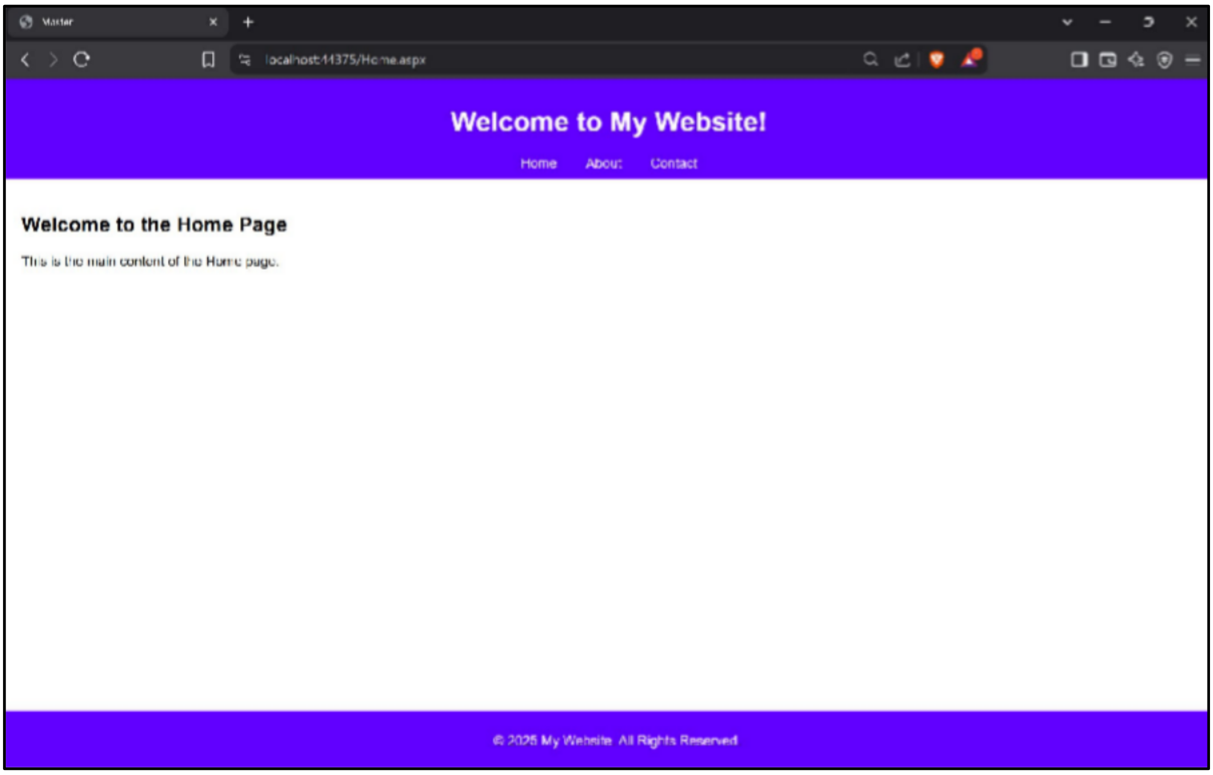
About.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
AutoEventWireup="true" CodeBehind="About.aspx.cs" Inherits="Practical2.About" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
   <h2>About Us</h2>
   <p>Learn more about our organization on this page.</p>
</asp:Content>
```

Contact.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
AutoEventWireup="true" CodeBehind="Contact.aspx.cs" Inherits="Practical2.Contact" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
   <h2>Contact Us</h2>
   <p>Feel free to reach out to us via this page.</p>
</asp:Content>
```

OUTPUT:

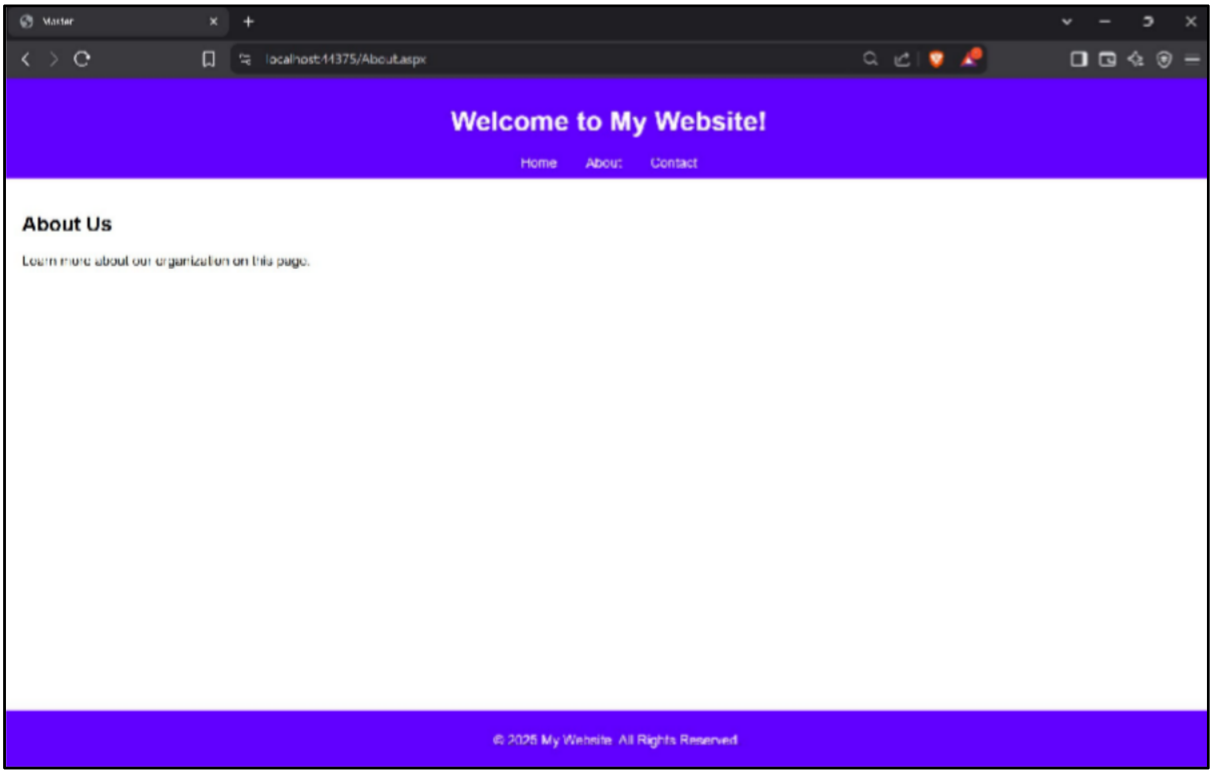


Welcome to My Website!

Welcome to the Home Page

This is the main content of the Home page.

© 2025 My Website. All Rights Reserved.



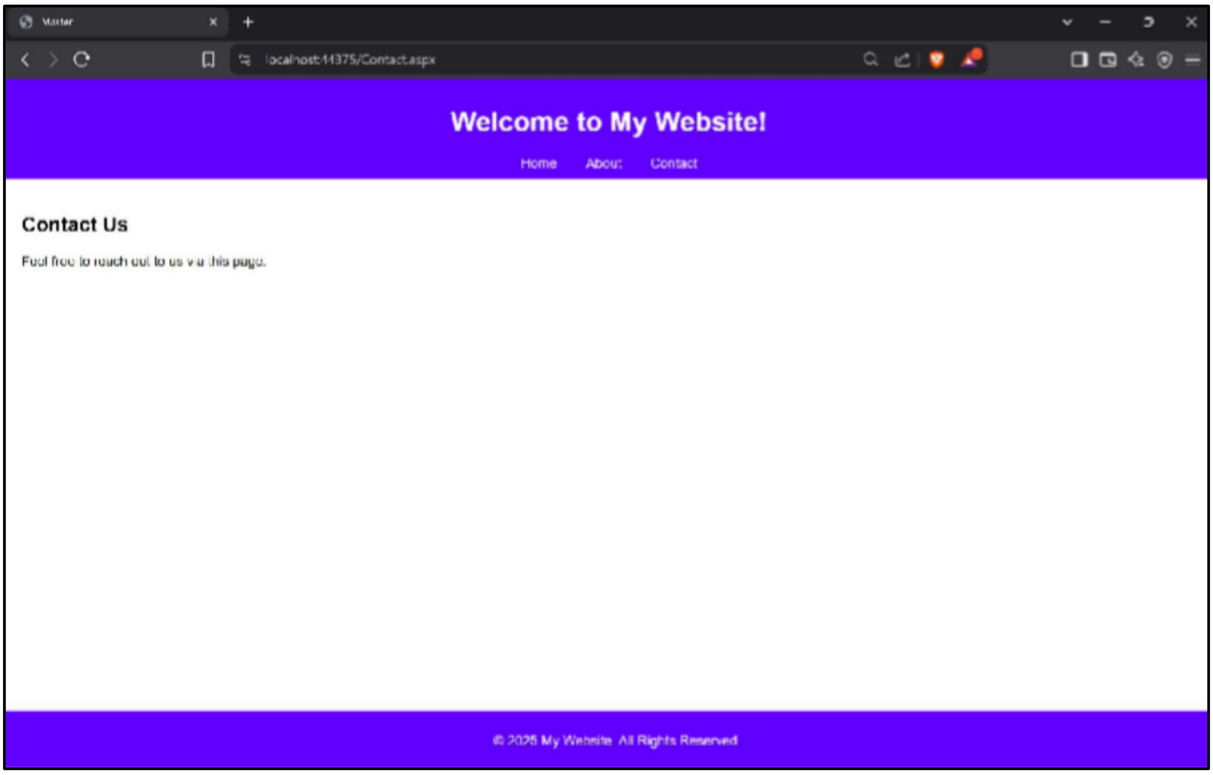
Welcome to My Website!

Home About Contact

About Us

Learn more about our organization on this page.

© 2025 My Website. All Rights Reserved



Welcome to My Website!

Home About Contact

Contact Us

Feel free to reach out to us via this page.

© 2025 My Website. All Rights Reserved

PRACTICAL 3: Design a Web Application for an Organization with Registration forms and advanced controls.

1. AdRotator Control in ASP.NET

CODE:

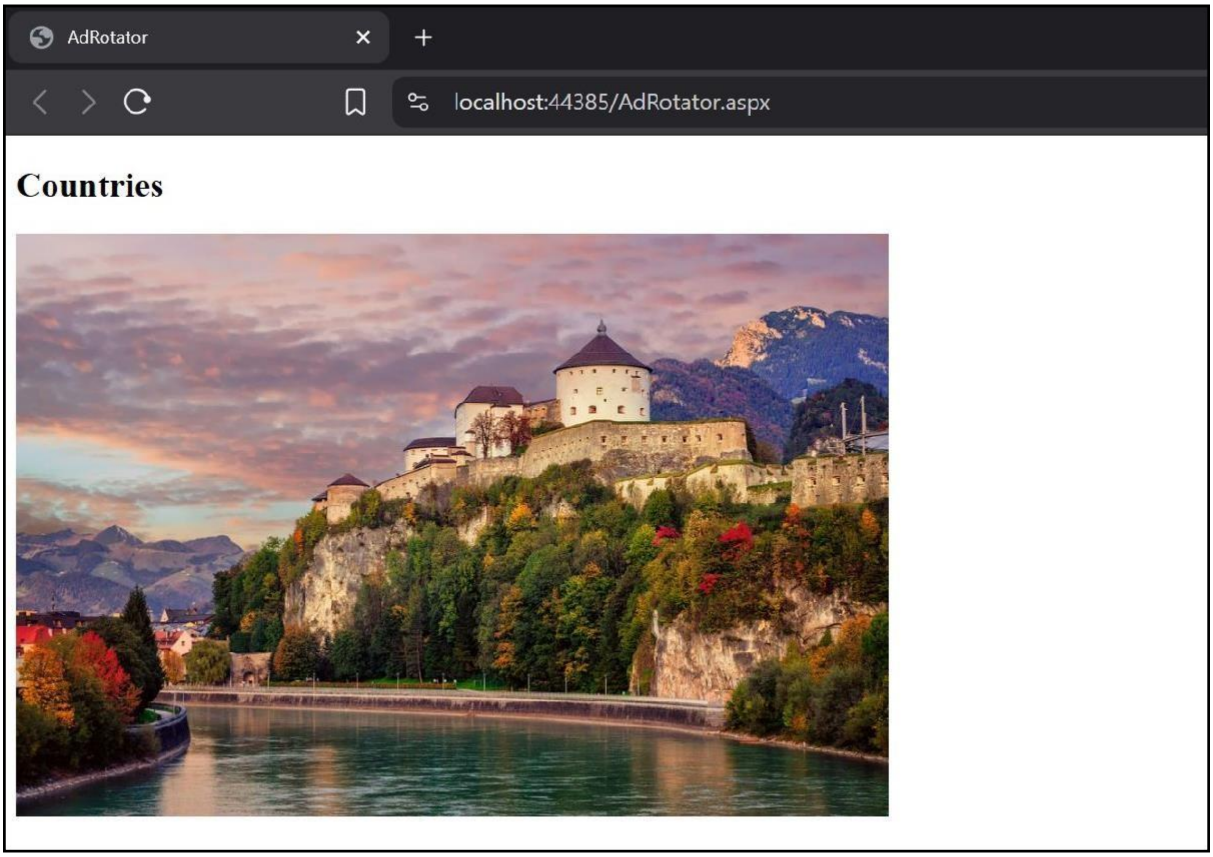
AdRotator.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="AdRotator.aspx.cs"
Inherits="Practical3.AdRotator" %>

<!DOCTYPE html>
<html>
<head>
  <title>AdRotator</title>
</head>
<body>
  <h2>Countries</h2>
  <asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/Ads.xml" Width="600px"
  Height="400px"/>
</body>
</html>
```

Ads.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
        <Ad>
                <ImageUrl>Images/ad1.jpg</ImageUrl>
                <NavigateUrl>https://www.example2.com</NavigateUrl>
                <AlternateText>Austria</AlternateText>
                <Impressions>50</Impressions>
        </Ad>
        <Ad>
                <ImageUrl>Images/ad2.jpg</ImageUrl>
                <NavigateUrl>https://www.example1.com</NavigateUrl>
                <AlternateText>Argentina</AlternateText>
                <Impressions>30</Impressions>
        </Ad>
</Advertisements>
```

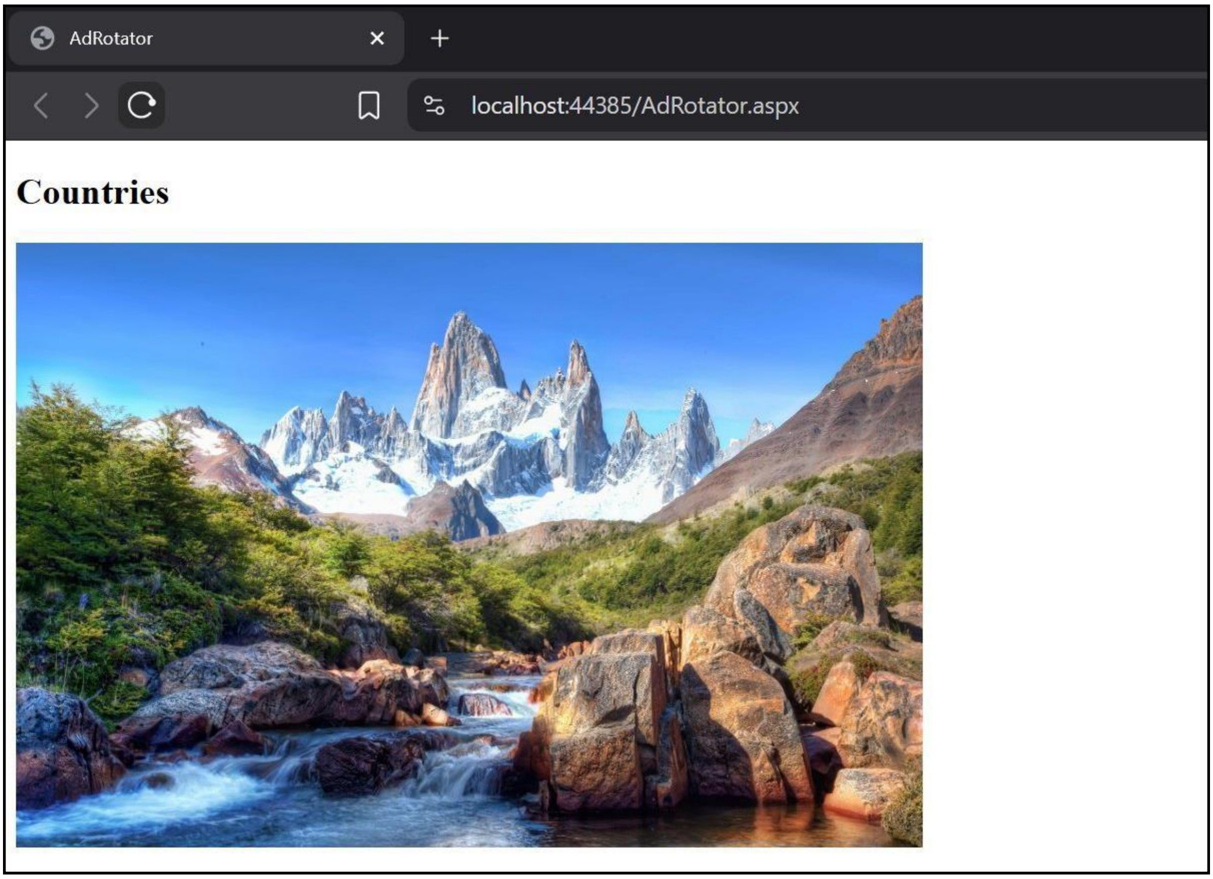
OUTPUT:



AdRotator

localhost:44385/AdRotator.aspx

Countries



AdRotator

localhost:44385/AdRotator.aspx

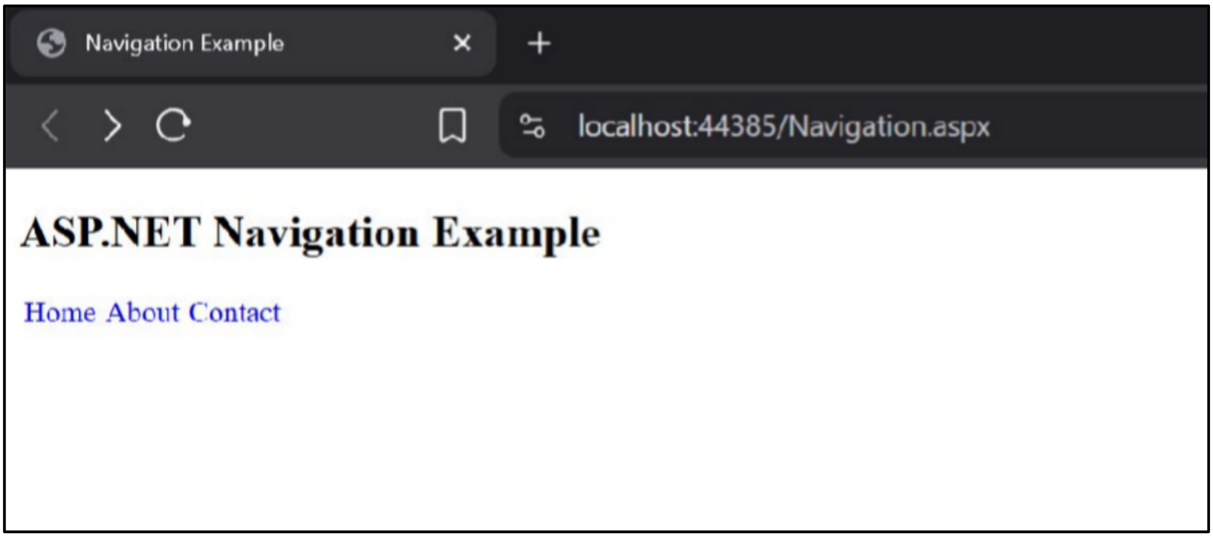
Countries

2. Navigation Control (Menu Navigation)

CODE:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Navigation.aspx.cs"
Inherits="Practical3.Navigation" %>
<!DOCTYPE html>
<html>
<head runat="server">
<title>Navigation Example</title>
</head>
<body>
    <form id="form1" runat="server">
    <h2>ASP.NET Navigation Example</h2>
    <asp:Menu ID="Menu1" runat="server" Orientation="Horizontal">
        <Items>
            <asp:MenuItem Text="Home" NavigateUrl="Home.aspx"/>
            <asp:MenuItem Text="About" NavigateUrl="About.aspx"/>
            <asp:MenuItem Text="Contact" NavigateUrl="Contact.aspx"/>
        </Items>
    </asp:Menu>
    </form>
</body>
</html>
```

OUTPUT:



ASP.NET Navigation Example

Home About Contact

3. File Upload in ASP.NET

CODE:

FileUpload.aspx:

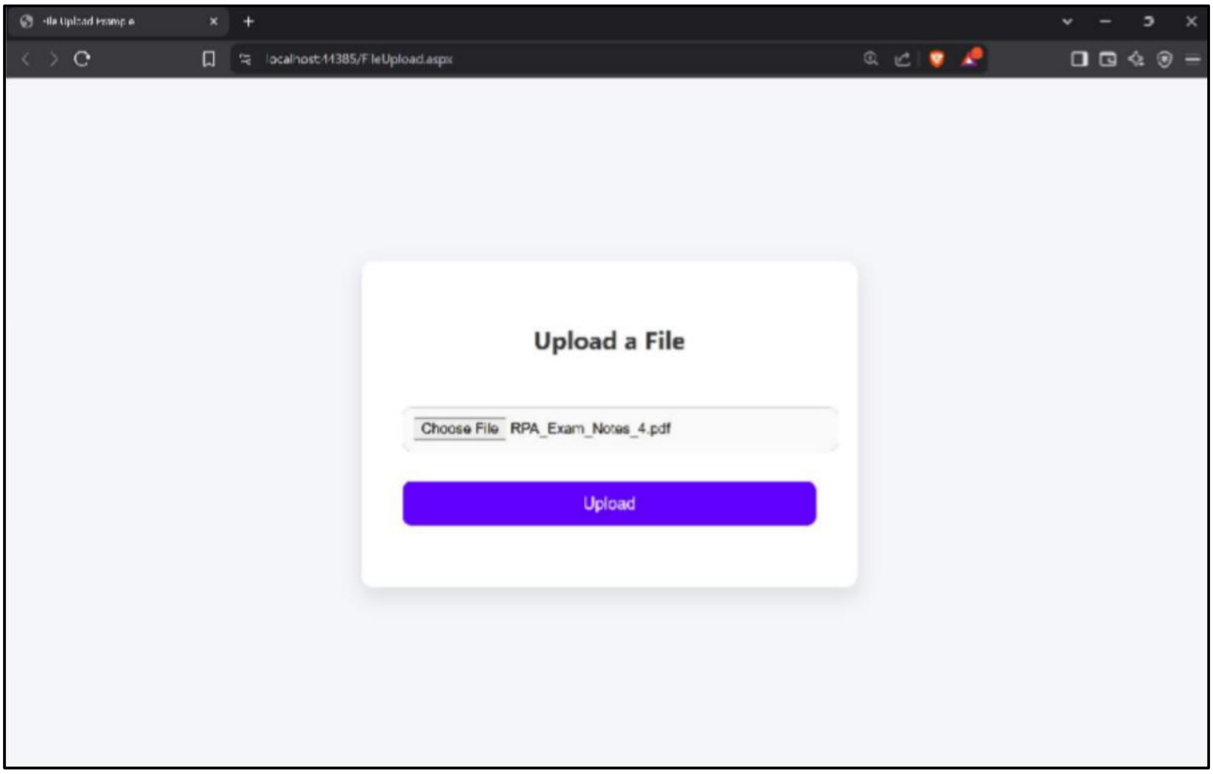
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="FileUpload.aspx.cs"
Inherits="Practical3.FileUpload" %>

<!DOCTYPE html>
<html>
<head runat="server">
  <title>File Upload Example</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <form id="form1" runat="server">
  <h2>Upload a File</h2>
  <asp:FileUpload ID="FileUploadControl" runat="server" />
  <asp:Button  ID="UploadButton"  runat="server"  Text="Upload"  OnClick="UploadButton_Click"  />
  <br />
  <asp:Label     ID="StatusLabel"     runat="server"     Text=""></asp:Label>
    </form>
</body>
</html>
```

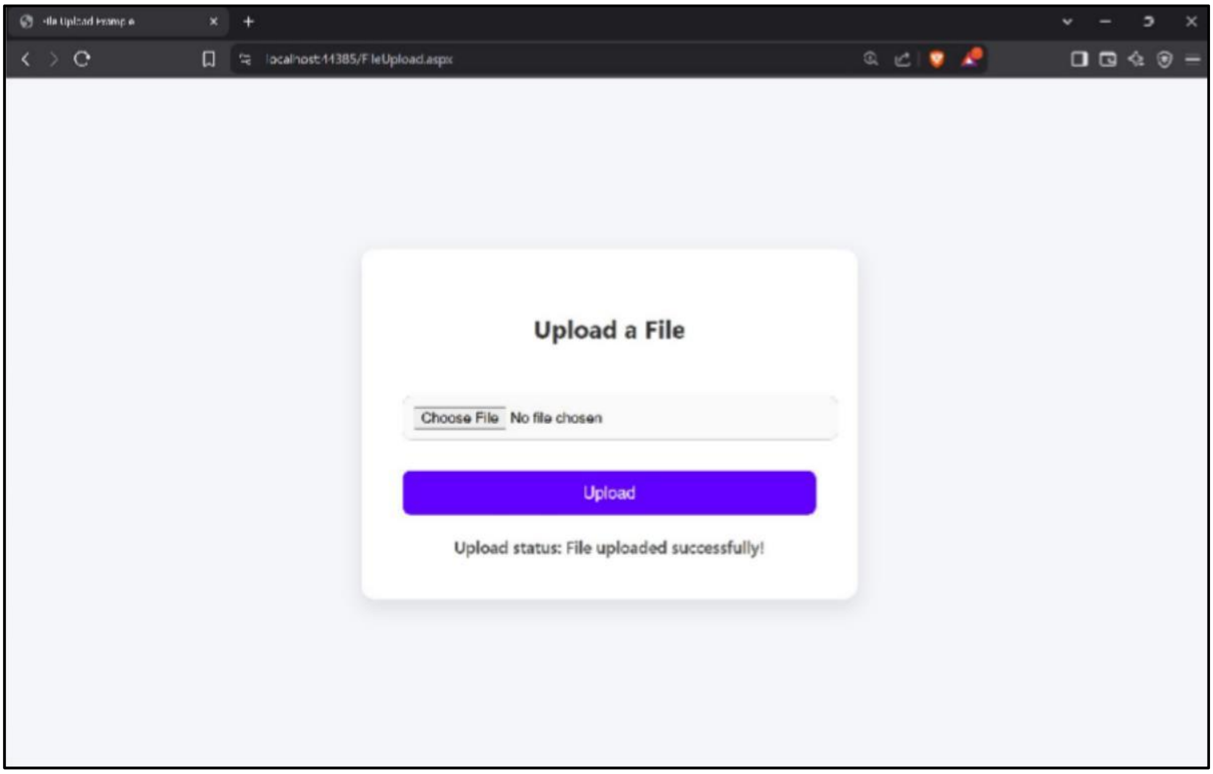
FileUpload.aspx.cs:

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

namespace Practical3
{
        public partial class FileUpload : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {

                }
    protected void UploadButton_Click(object sender, EventArgs e)
    {
      if (FileUploadControl.HasFile)
      {
        try
        {
```

```
                    string filename = Path.GetFileName(FileUploadControl.FileName);
                    FileUploadControl.SaveAs(Server.MapPath("~/Uploads/") + filename);
                    StatusLabel.Text = "Upload status: File uploaded successfully!";
                }
                catch (Exception ex)
                {
                    StatusLabel.Text = "Upload status: Error - " + ex.Message;
                }
            }
            else
            {
                StatusLabel.Text = "Upload status: No file selected.";
            }
        }
    }
}
```

OUTPUT:





Upload a File

Upload status: File uploaded successfully!

PRACTICAL 4: Webpage Demonstrating Connection-Oriented Architecture (ASP.NET Web Forms with SQL Server Database)

CODE:

Page.aspx:

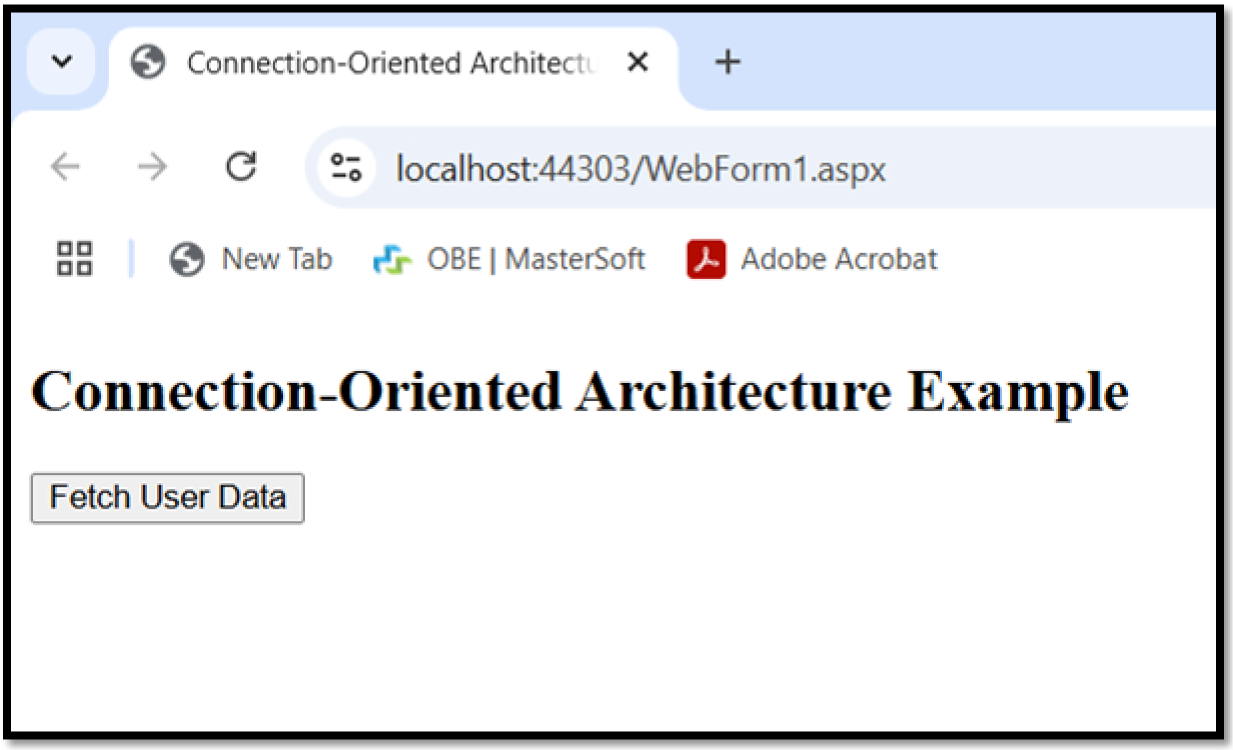
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical4.Page" %>

<!DOCTYPE html>
<html>
<head runat="server">
  <title>DB</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
   <form id="form1" runat="server">
     <h2>Database Example</h2>

     <asp:Button ID="FetchDataButton" runat="server" Text="Fetch User Data"
OnClick="FetchDataButton_Click" />
     <br /><br />

     <asp:GridView ID="UsersGridView" runat="server" AutoGenerateColumns="true"
BorderWidth="1" />

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

Page.aspx.cs:

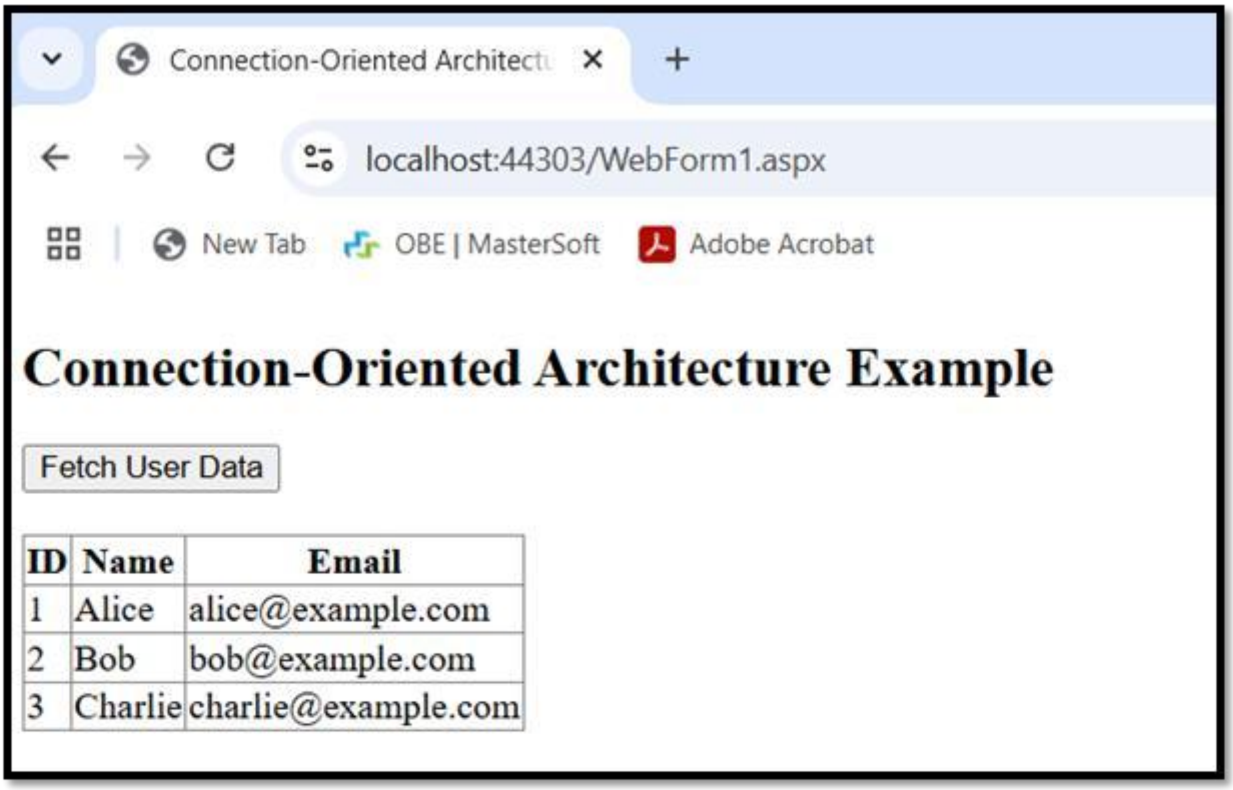
```
using System;
using
System.Collections.Generic;
using System.Data.SqlClient;
using System.Data; using
System.Linq; using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical4
{
        public partial class Page : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {

                }
    protected void FetchDataButton_Click(object sender, EventArgs e)
```

```
        {
            string connectionString = "Data Source=KAPIL;Initial Catalog=dbprac;Integrated
Security=True";

            // Create a connection object
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                try
                {
                    conn.Open(); // Open the connection

                    // SQL query to fetch data
                    string query = "SELECT * FROM Users";
                    SqlDataAdapter da = new SqlDataAdapter(query, conn);
                    DataTable dt = new DataTable();
                    da.Fill(dt);

                    // Bind data to GridView
                    UsersGridView.DataSource = dt;
                    UsersGridView.DataBind();
                }
                catch (Exception ex)
                {
                    Response.Write("<script>alert('Error: " + ex.Message + "');</script>");
                }
            }

        }
    }
}
```

OUTPUT:



Connection-Oriented Architecture Example



Connection-Oriented Architecture Example

ID	Name	Email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

PRACTICAL 5: Webpage Demonstrating Disconnected Architecture (ASP.NET Web Forms with SQL Server Database)

CODE:

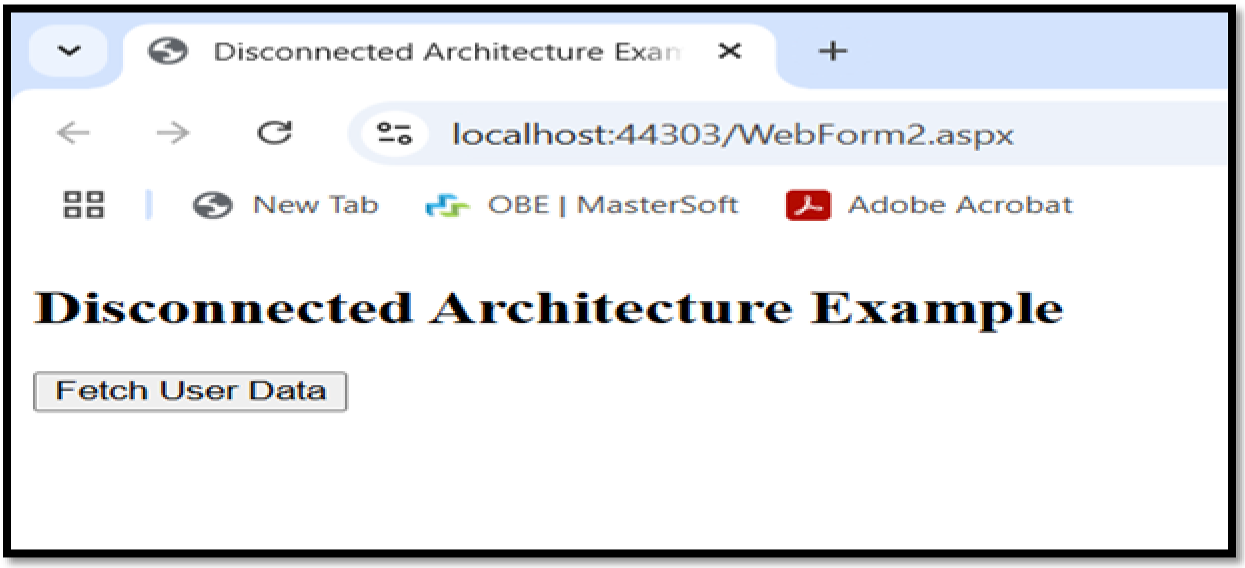
Page.aspx:

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

<!DOCTYPE html>
<html>
<head runat="server">
  <title>Disconnected Architecture Example</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <form id="form1" runat="server">
    <h2>Disconnected Architecture Example</h2>

    <asp:Button ID="FetchDataButton" runat="server" Text="Fetch User Data"
OnClick="FetchDataButton_Click" />
    <br /><br />

    <asp:GridView ID="UsersGridView" runat="server" AutoGenerateColumns="true"
BorderWidth="1" />
  </form>
</body>
</html>
```

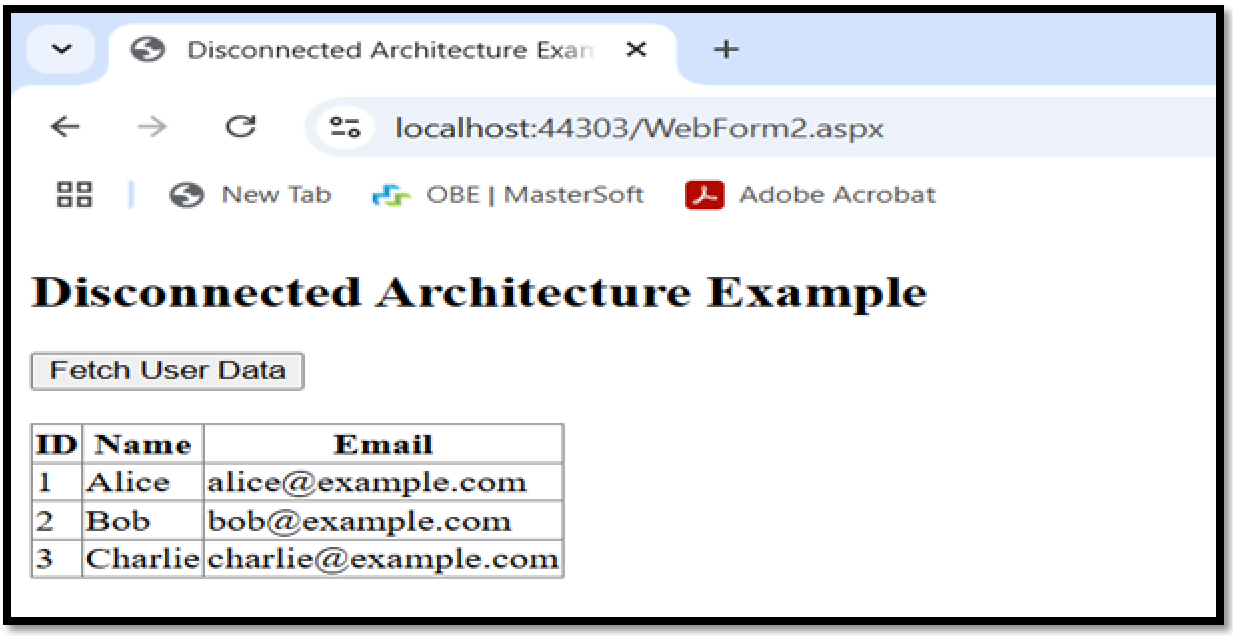
Page.aspx.cs:

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

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

                }
    protected void FetchDataButton_Click(object sender, EventArgs e)
    {
      string connectionString = "Data Source=KAPIL;Initial Catalog=dbprac;Integrated
Security=True";

      // Create objects for disconnected architecture
      SqlDataAdapter da;
      DataSet ds = new DataSet();

      try
      {
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
          // SQL query to fetch data string query
          = "SELECT * FROM Users"; da = new
          SqlDataAdapter(query, conn);

          // Fill dataset with data from the database
          da.Fill(ds, "Users");
        } // Connection is closed after this block

        // Bind data to GridView (data remains in memory)
        UsersGridView.DataSource = ds.Tables["Users"];
        UsersGridView.DataBind();
      }
      catch (Exception ex)
      {
        Response.Write("<script>alert('Error: " + ex.Message + "');</script>");
      }
    }
  }
}
```

OUTPUT:



Disconnected Architecture Example



Disconnected Architecture Example

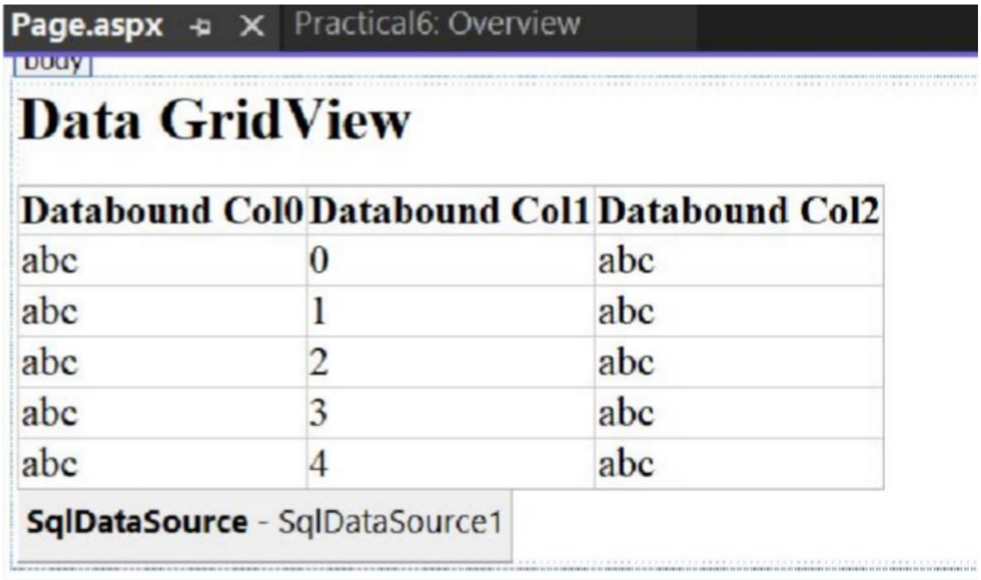
ID	Name	Email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

PRACTICAL 6: Create a webpage that demonstrates the use of data bound controls of ASP.NET.

CODE:

Page.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical6.Page" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Data GridView</h2>
         <p>
            <asp:GridView     ID="GridView1"     runat="server"     DataSourceID="SqlDataSource1">
            </asp:GridView>
            <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:dbpracConnectionString %>"
ProviderName="<%$      ConnectionStrings:dbpracConnectionString.ProviderName
%>" SelectCommand="SELECT * FROM [Users]"></asp:SqlDataSource> </p>
       </div>
   </form>
</body>
</html>
```

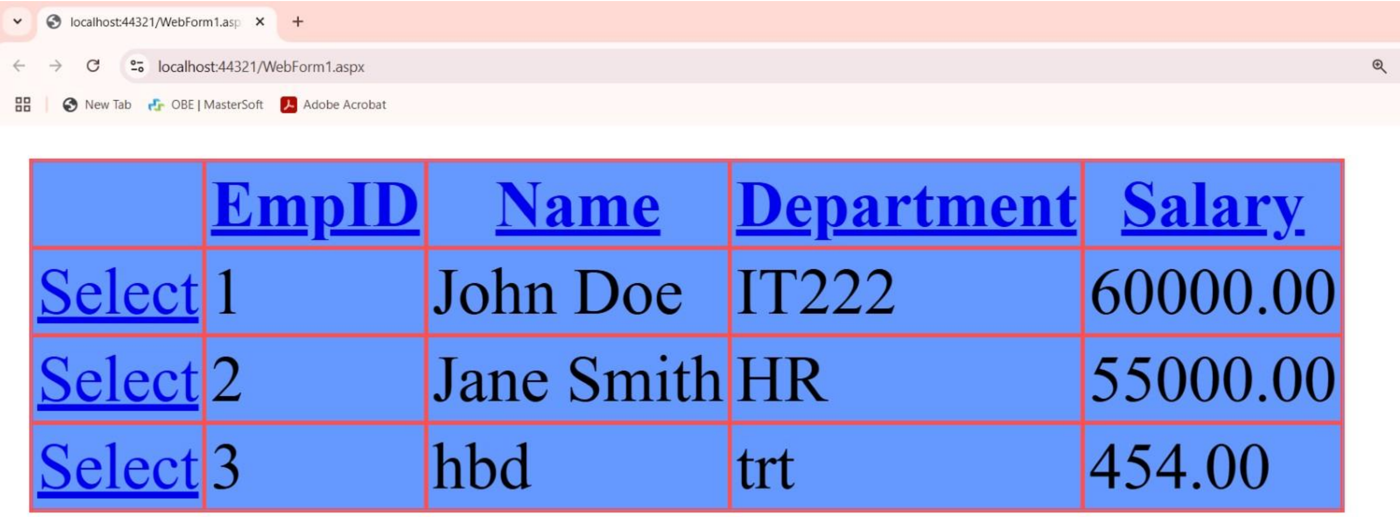


Data GridView

Databound Col0	Databound Col1	Databound Col2
abc	0	abc
abc	1	abc
abc	2	abc
abc	3	abc
abc	4	abc

SqlDataSource - SqlDataSource1

OUTPUT:



	EmpID	Name	Department	Salary
Select	1	John Doe	IT222	60000.00
Select	2	Jane Smith	HR	55000.00
Select	3	hbd	trt	454.00

PRACTICAL 7: Design a webpage to demonstrate the working of a simple stored procedure.

CODE:

Procedure Script:

```
CREATE PROCEDURE GetUsers
AS
BEGIN
   SELECT Id, Name, Email FROM Users END
```

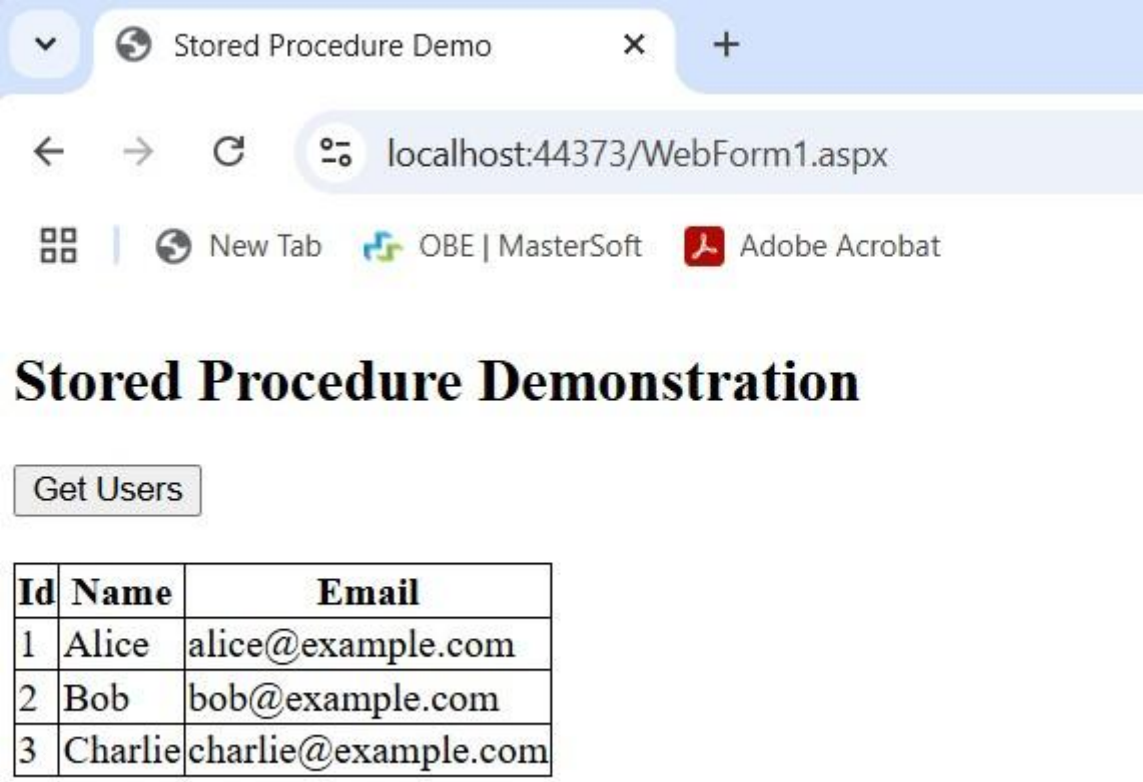
Page.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical7.Page" %>

<!DOCTYPE html>
<html>
<head>
   <title>Stored Procedure Demo</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Stored Procedure Demonstration</h2>

      <asp:Button ID="btnGetUsers" runat="server" Text="Get Users" OnClick="btnGetUsers_Click" />
      <br /><br />

      <asp:GridView ID="GridViewUsers" runat="server" AutoGenerateColumns="true"
BorderColor="Black" BorderWidth="1px" />
   </form>
</body>
</html>
```

Page.aspx.cs:

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

namespace Practical7
{
        public partial class Page : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {
                    }
        protected void btnGetUsers_Click(object sender, EventArgs e)
        {
          string connStr = "Data Source=KAPIL;Initial Catalog=dbprac;Integrated Security=True";

            using (SqlConnection conn = new SqlConnection(connStr))
            {
                using (SqlCommand cmd = new SqlCommand("GetUsers", conn))
                {
                    cmd.CommandType          =          CommandType.StoredProcedure;
                    conn.Open();

                    SqlDataAdapter da = new SqlDataAdapter(cmd);
                    DataTable dt = new DataTable();
                    da.Fill(dt);

                    GridViewUsers.DataSource = dt;
                    GridViewUsers.DataBind();

                }
            }
        }
    }
}
```

OUTPUT:



Stored Procedure Demonstration

Id	Name	Email
1	Alice	alice@example.com
2	Bob	bob@example.com
3	Charlie	charlie@example.com

PRACTICAL 8: Design a webpage to demonstrate the working of parameterized stored procedure.

CODE:

Stored Prodecure:

```
CREATE PROCEDURE GetEmployeesByDepartment
   @DepartmentName VARCHAR(100)
AS
BEGIN
   SELECT * FROM Employees WHERE Department = @DepartmentName; END;
```

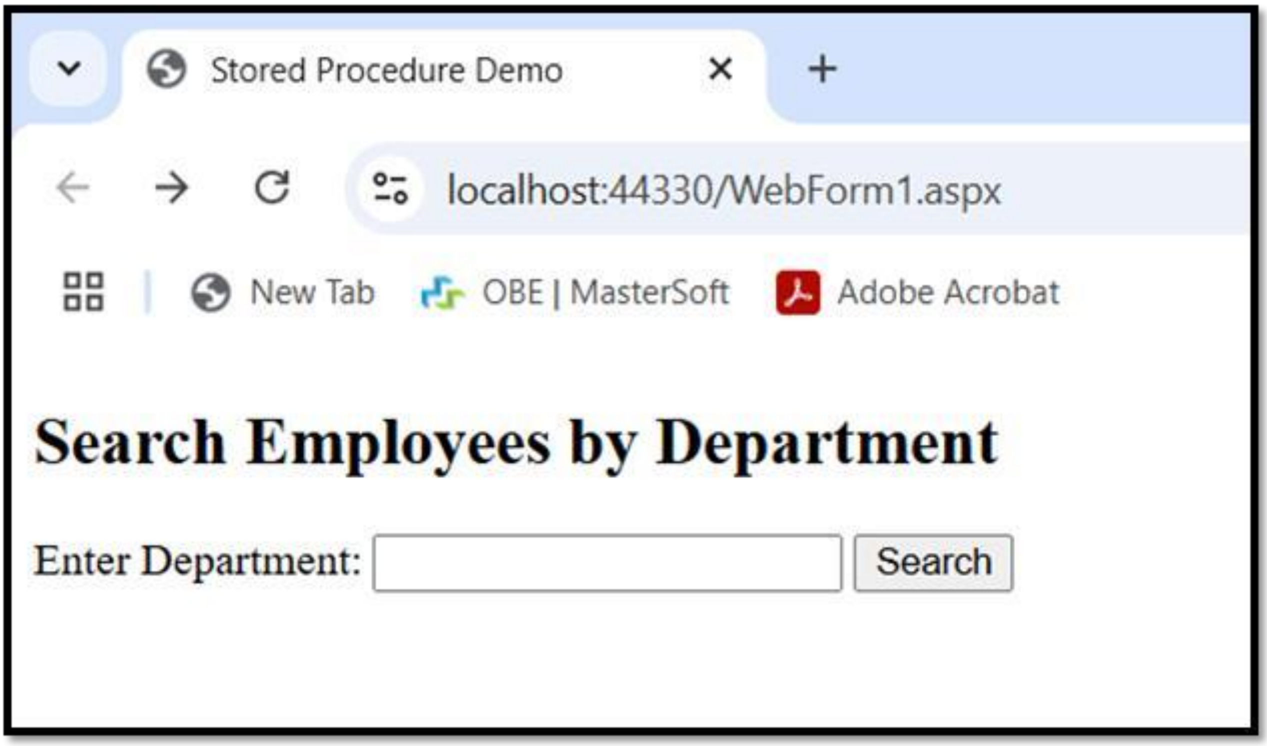
Page.aspx:

```
<!DOCTYPE html>
<html lang="en">
<head runat="server">
   <title>Stored Procedure Demo</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Search Employees by Department</h2>
         <asp:Label runat="server" Text="Enter Department:"></asp:Label>
         <asp:TextBox ID="txtDepartment" runat="server"></asp:TextBox>
         <asp:Button ID="btnSearch" runat="server" Text="Search" OnClick="btnSearch_Click"/>

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

Page.aspx.cs: using System; using

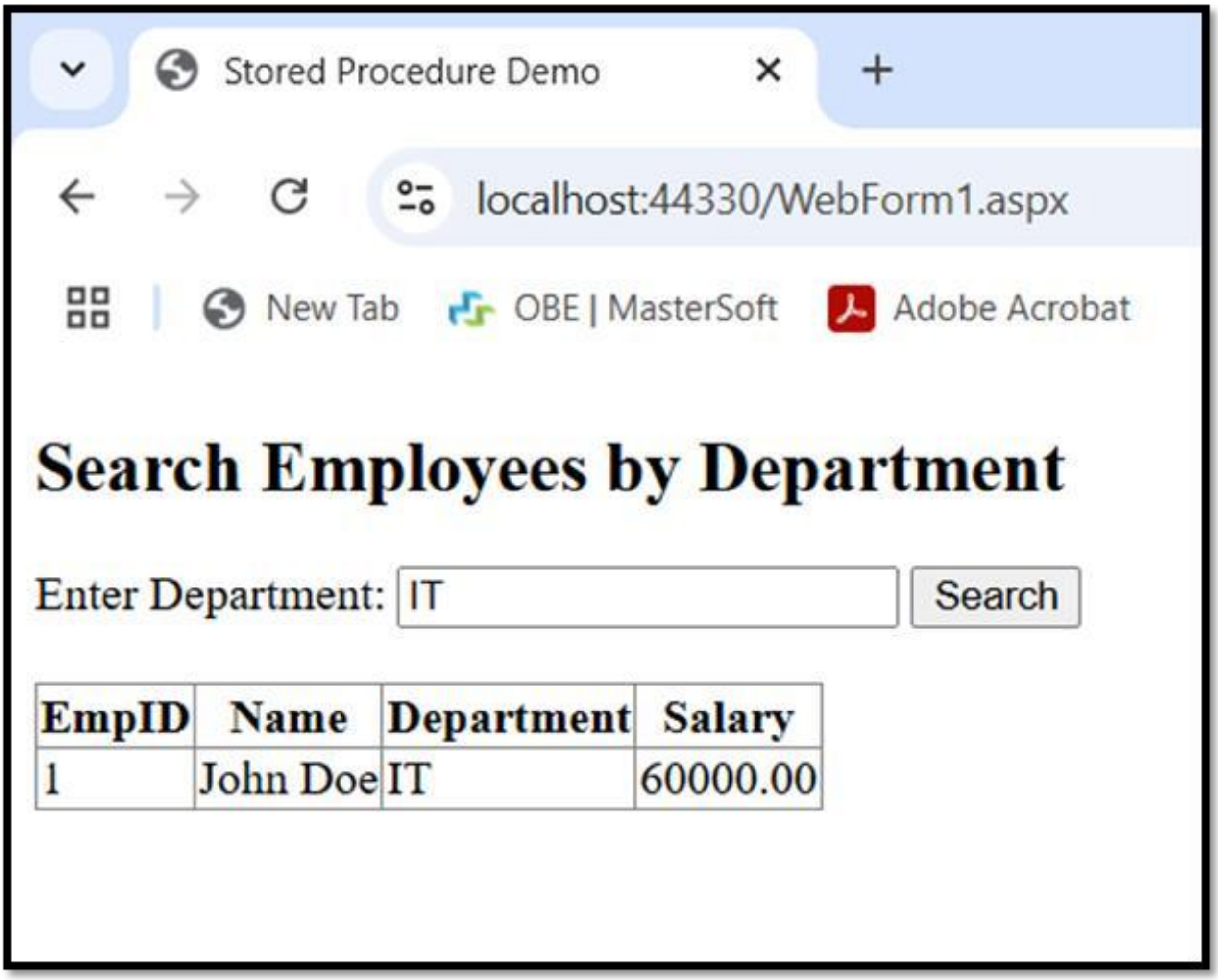
```
System.Collections.Generic;
using         System.Data.SqlClient;
using       System.Data;       using
System.Linq;  using  System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical8
{
            public partial class Page : System.Web.UI.Page
            {
                        protected void Page_Load(object sender, EventArgs e)
                        {
                }
        protected void btnSearch_Click(object sender, EventArgs e)
        {
          string connStr = "Data Source=KAPIL;Initial Catalog=dbprac;Integrated Security=True";

            {
                SqlConnection conn = new SqlConnection(connStr);
                SqlCommand    cmd    =    new    SqlCommand("GetEmployeesByDepartment",    conn);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@DepartmentName", txtDepartment.Text);

                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);

                gvEmployees.DataSource                    =                    dt;
                gvEmployees.DataBind();

            }

        }
    }
}
```

OUTPUT:



Search Employees by Department

Enter Department:



Search Employees by Department

Enter Department: IT Search

EmpID	Name	Department	Salary
1	John Doe	IT	60000.00

PRACTICAL 9: Design a webpage to display the use of LINQ.

CODE:

Page.aspx:

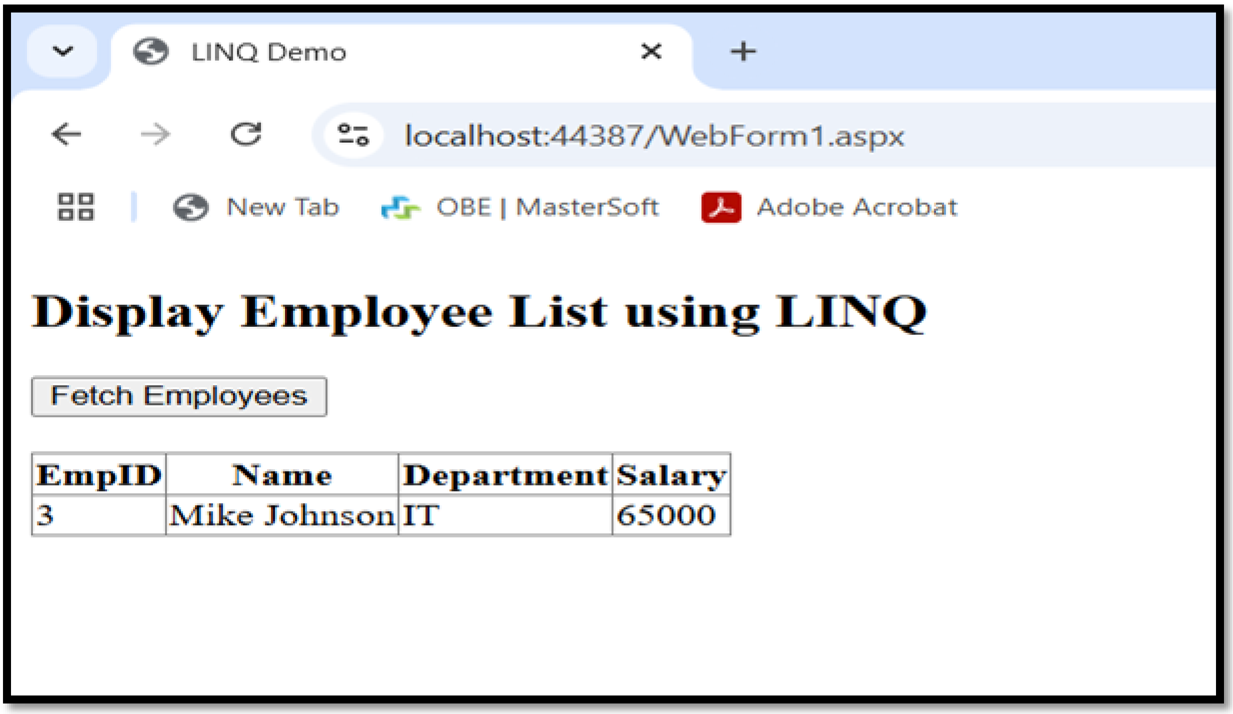
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical9.Page" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
  <title>LINQ Demo</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Display Employee List using LINQ</h2>
      <asp:Button ID="btnFetchData" runat="server" Text="Fetch Employees"
OnClick="btnFetchData_Click"/>
      <br /><br />
      <asp:GridView ID="gvEmployees" runat="server"
```

```
AutoGenerateColumns="True"></asp:GridView>
        </div>
    </form>
</body>
</html>
```

Page.aspx.cs:

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

namespace Practical9
{
            public partial class Page : System.Web.UI.Page
            {
       // Define an Employee class
       public class Employee
       {
          public int EmpID { get; set; } public
          string Name { get; set; } public string
          Department { get; set; }
          public decimal Salary { get; set; }
       }
       // Sample Employee Data (In-Memory Collection) private
       List<Employee> employees = new List<Employee> {
          new Employee { EmpID = 1, Name = "Kapil Lad", Department = "IT", Salary = 80000 }, new
Employee { EmpID = 2, Name = "Omkar Londhe", Department = "HR", Salary = 55000 }, new
Employee { EmpID = 3, Name = "Yash Parab", Department = "IT", Salary = 65000 }, new Employee {
EmpID = 4, Name = "Siddhesh Nagwekar", Department = "Finance", Salary = 70000 }
       };

       protected void Page_Load(object sender, EventArgs e)
                  {

                  }
       protected void btnFetchData_Click(object sender, EventArgs e)
       {
          //LINQ to fetch IT department employees with salary > 60,000
          var result = from emp in employees where emp.Department ==
          "IT" && emp.Salary > 60000 select emp;

          // Bind data to GridView
          gvEmployees.DataSource               =               result.ToList();
          gvEmployees.DataBind();

       }
    }
}
```

OUTPUT:



Display Employee List using LINQ

Fetch Employees

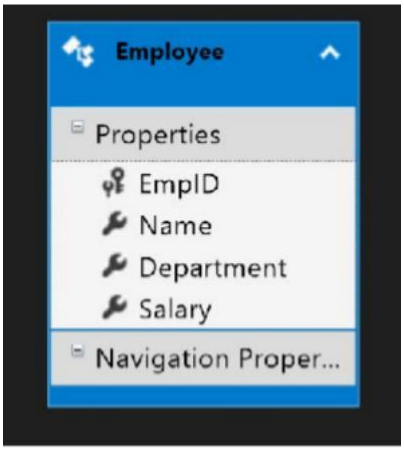
EmpID	Name	Department	Salary
3	Mike Johnson	IT	65000

PRACTICAL 10: Build websites to demonstrate the working of entity frameworks in dot net.

Steps to Implement

1. **Create a SQL Server Database & Table**
2. **Create an ASP.NET Web Application in Visual Studio**
3. **Install & Configure Entity Framework (EF) ORM**
4. **Use EF to perform CRUD operations**
5. **Display data in GridView & allow users to Add, Edit, Delete records**

ENTITY MODEL:



CODE:

WebForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Practical10.WebForm1" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
  <title>Entity Framework CRUD Demo</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Employee Management (Entity Framework)</h2>

      <!-- Add Employee Form -->
      <asp:Label runat="server" Text="Name:"></asp:Label>
      <asp:TextBox ID="txtName" runat="server" style="margin-left: 46px"></asp:TextBox>

      <br />

      <asp:Label runat="server" Text="Department:"></asp:Label>
      <asp:TextBox ID="txtDepartment" runat="server"></asp:TextBox> <br />


      <asp:Label runat="server" Text="Salary:"></asp:Label>
      <asp:TextBox ID="txtSalary" runat="server" style="margin-left: 41px"></asp:TextBox>
      <br />
      <br />
      <br />


      <asp:Button ID="btnAdd" runat="server" Text="Add Employee" OnClick="btnAdd_Click" />

      <br /><br />

      <!-- Display Employees -->
      <asp:GridView ID="gvEmployees" runat="server" AutoGenerateColumns="False"
DataKeyNames="EmpID"
        OnRowEditing="gvEmployees_RowEditing"  OnRowUpdating="gvEmployees_RowUpdating"
        OnRowCancelingEdit="gvEmployees_RowCancelingEdit"
OnRowDeleting="gvEmployees_RowDeleting" Height="233px" Width="677px">
        <Columns>
          <asp:BoundField DataField="EmpID" HeaderText="EmpID" ReadOnly="True" />
          <asp:BoundField DataField="Name" HeaderText="Name" />
          <asp:BoundField DataField="Department" HeaderText="Department" />
          <asp:BoundField DataField="Salary" HeaderText="Salary" />

          <asp:CommandField       ShowEditButton="True"       ShowDeleteButton="True"       />
        </Columns>
      </asp:GridView>
    </div>
  </form>
</body>
</html>
```

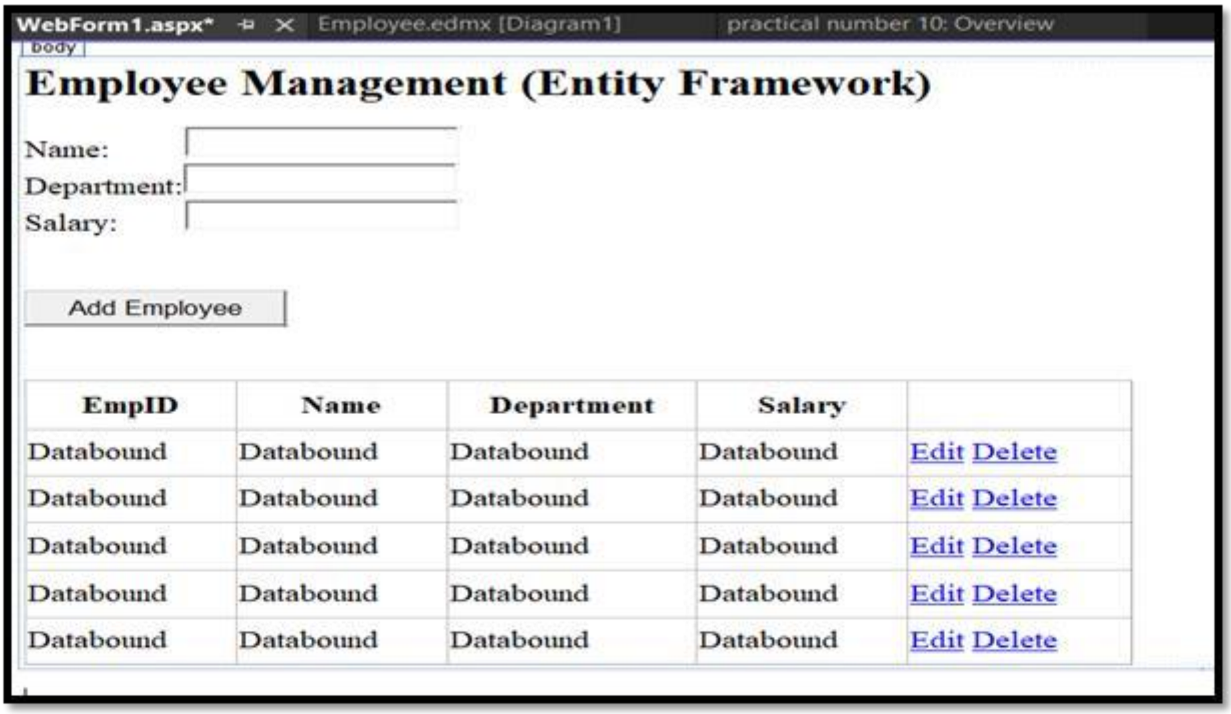
WebForm.aspx.cs:

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

namespace Practical10
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    EmployeeDBEntities db = new EmployeeDBEntities();
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        LoadEmployees();
      }

    }
    private void LoadEmployees()
    {
      gvEmployees.DataSource                    =                    db.Employees.ToList();
      gvEmployees.DataBind();
    }

    // Add Employee
    protected void btnAdd_Click(object sender, EventArgs e)
    {
```

```
    Employee emp = new Employee
    {
        Name = txtName.Text,
        Department = txtDepartment.Text,
        Salary = Convert.ToDecimal(txtSalary.Text)
    };

    db.Employees.Add(emp);
    db.SaveChanges();
    LoadEmployees();
}

// Edit Employee
protected void gvEmployees_RowEditing(object sender, GridViewEditEventArgs e)
{
    gvEmployees.EditIndex = e.NewEditIndex;
    LoadEmployees();
}

// Update Employee
protected void gvEmployees_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    int empID = Convert.ToInt32(gvEmployees.DataKeys[e.RowIndex].Value); Employee
    emp = db.Employees.Find(empID);

    TextBox txtName = (TextBox)gvEmployees.Rows[e.RowIndex].Cells[1].Controls[0];
    TextBox    txtDepartment    =    (TextBox)gvEmployees.Rows[e.RowIndex].Cells[2].Controls[0];
    TextBox txtSalary = (TextBox)gvEmployees.Rows[e.RowIndex].Cells[3].Controls[0];

    emp.Name = txtName.Text; emp.Department =
    txtDepartment.Text;          emp.Salary          =
    Convert.ToDecimal(txtSalary.Text);
    db.SaveChanges();
    gvEmployees.EditIndex = -1;
    LoadEmployees();
}

// Cancel Edit
protected void gvEmployees_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
    gvEmployees.EditIndex = -1;
    LoadEmployees();
}

// Delete Employee
protected void gvEmployees_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    int empID = Convert.ToInt32(gvEmployees.DataKeys[e.RowIndex].Value); Employee
    emp = db.Employees.Find(empID);
```

```
            db.Employees.Remove(emp);
            db.SaveChanges();
            LoadEmployees();
        }

    }
}
```

OUTPUT:



Employee Management (Entity Framework)

Name:

Department:

Salary:

Add Employee

EmpID	Name	Department	Salary	
Databound	Databound	Databound	Databound	Edit Delete
Databound	Databound	Databound	Databound	Edit Delete
Databound	Databound	Databound	Databound	Edit Delete
Databound	Databound	Databound	Databound	Edit Delete
Databound	Databound	Databound	Databound	Edit Delete

PRACTICAL 11: Design Web Applications using Client-Side Session Management

1. View State

CODE:

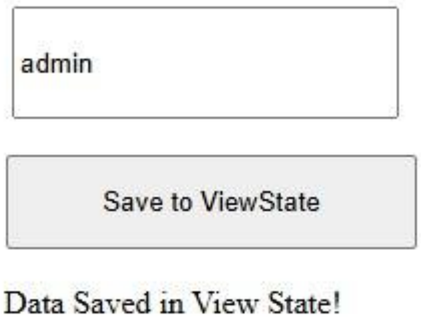
ViewPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="viewState.aspx.cs"
Inherits="Practical11.viewState" %>

<!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" Height="52px" Width="157px"></asp:TextBox>
         <br />
         <br />
         <asp:Button ID="Button1" runat="server" Height="30px" OnClick="Button1_Click" Text="Save
to ViewState" Width="164px" />
         <br />
         <br />
         <asp:Label      ID="Label1"      runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

ViewPage.aspx.cs:

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

namespace Practical11
{
   public partial class viewState : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (ViewState["UserName"] != null)
         {
                Label1.Text="Stored in View State: " + ViewState["UserName"].ToString();
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            ViewState["UserName"] = TextBox1.Text;
            Label1.Text = "Data saved in View State!";
        }
    }
}
```

OUTPUT:


admin

Data Saved in View State!

2. QueryString

CODE:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QueryString.aspx.cs"
Inherits="Practical11.QueryString" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:TextBox ID="TextBox1" runat="server" Height="65px" Width="207px"></asp:TextBox>
        <div>
            <br />
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Go to Next Page"
Width="214px" />
        </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 Practical11
{
   public partial class QueryString : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         Response.Redirect("qs2.aspx?name="+TextBox1.Text);
      }
   }
}
```

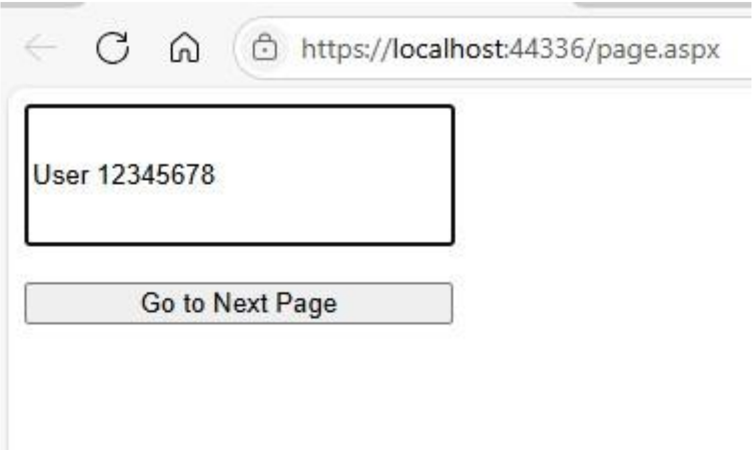
WebForm2.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="qs2.aspx.cs"
Inherits="Practical11.qs2" %>

<!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"></asp:Label>
      </div>
   </form>
</body>
</html>
```

WebForm2.aspx.cs:

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

namespace Practical11
{
   public partial class qs2 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (Request.QueryString["name"] != null)
         {
           Label1.Text = "QueryString Value: "+Request.QueryString["name"];
         }
      }
   }
}
```

OUTPUT:



```
User 12345678
```

3. Cookies:

CODE:

WebForm1.aspx:

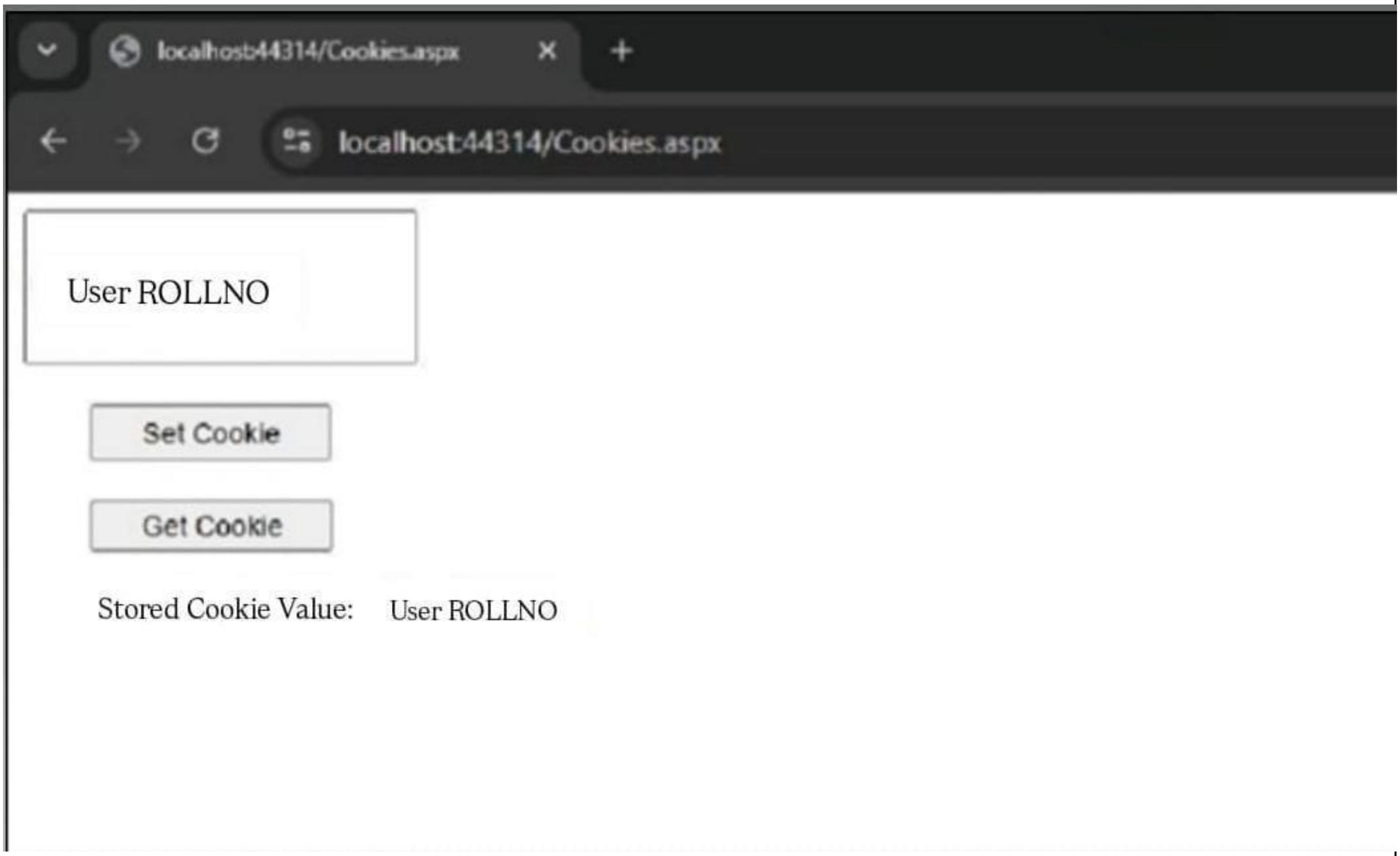
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Cookies.aspx.cs"
Inherits="Practical11.Cookies" %>

<!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" Height="68px" Width="179px"></asp:TextBox>
      <br />
      <br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
      <asp:Button ID="Button1" runat="server" Height="27px" OnClick="Button1_Click" Text="Set
Cookie" Width="114px" />
      <br />
      <br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
      <asp:Button ID="Button2" runat="server" Height="24px" OnClick="Button2_Click" Text="Get
Cookie" Width="115px" />
      <br />
      <br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
      <asp:Label    ID="Label1"    runat="server"></asp:Label>
    </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 Practical11
{
  public partial class Cookies : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
      HttpCookie cookie = new HttpCookie("UserName",TextBox1.Text);
      cookie.Expires              =              DateTime.Now.AddDays(7);
      Response.Cookies.Add(cookie);
      Label1.Text = "Cookie Set Successfuly!";
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
      HttpCookie cookie = Request.Cookies["UserName"]; if(cookie
      != null)
      {
         Label1.Text="Stored Cookie Value: "+cookie.Value;
      }
      else
      {
         Label1.Text = "No Cookie Found!";
      }
    }
  }
}
```

OUTPUT:



User ROLLNO

Stored Cookie Value: User ROLLNO

4. Hidden Fields:

CODE:

WebForm1.aspx:

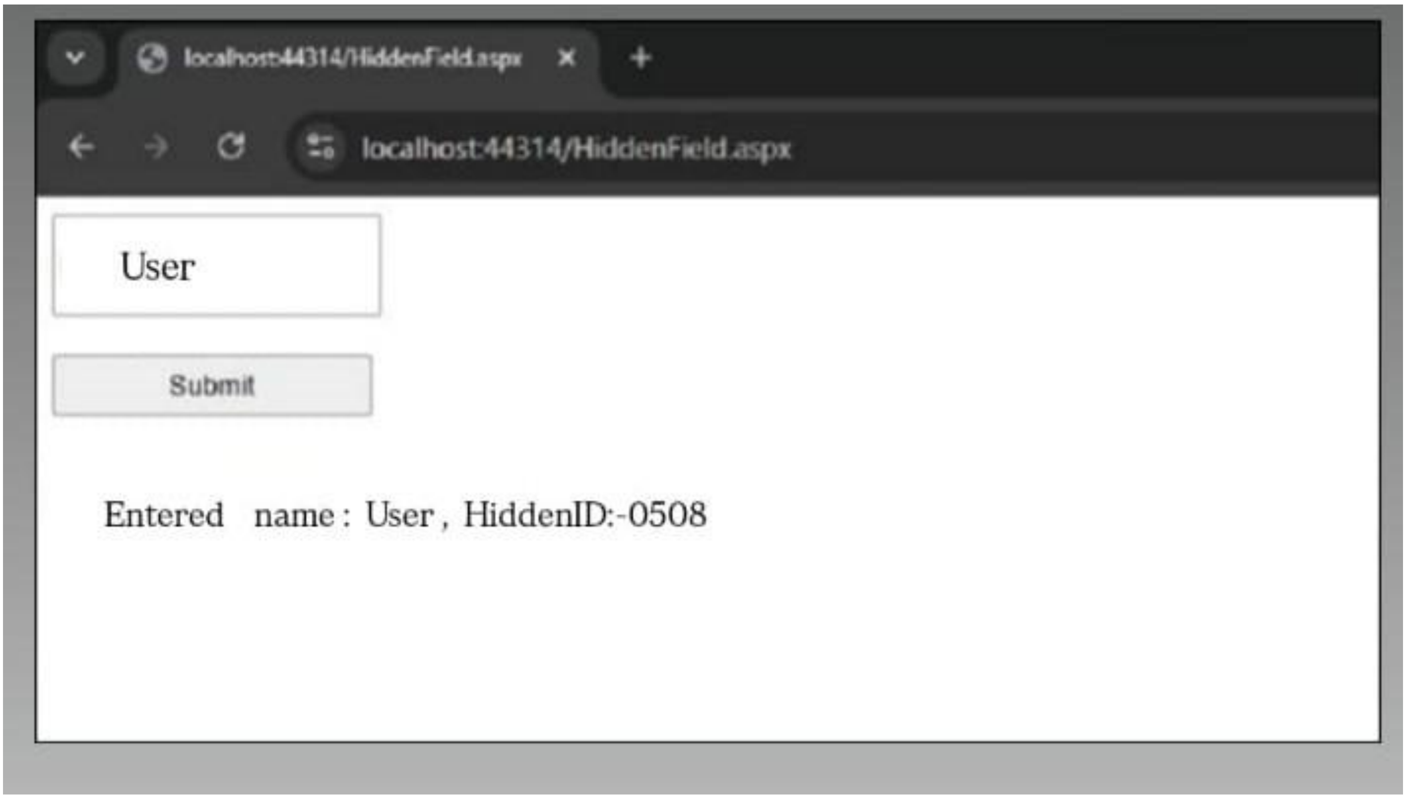
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="HiddenField.aspx.cs"
Inherits="Practical11.HiddenField" %>
```

```
<!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"  Height="44px"  Width="151px"></asp:TextBox>
      <br />
      <asp:HiddenField ID="HiddenField1" runat="server" Value="0508" /> <br
      />
      <asp:Button   ID="Button1"   runat="server"   Height="29px"   OnClick="Button1_Click"
style="margin-bottom: 32px" Text="Submit" Width="154px" />
      <br />
      <asp:Label   ID="Label1"   runat="server"></asp:Label>
    </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 Practical11
{
  public partial class HiddenField : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
      Label1.Text="Entered Name: "+TextBox1.Text+", HiddenID: "+HiddenField1.Value;
    }
  }
}
```

OUTPUT:


localhost:44314/HiddenField.aspx
localhost:44314/HiddenField.aspx
User
Submit

Entered name : User , HiddenID:-0508

PRACTICAL 12: Design a Web Application for an Organization with Registration forms and advanced controls.

Add to Web.config:

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

CODE:

Page.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical12.Page" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
  <title>Login</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Login Page</h2>
      <asp:Label  ID="lblMessage"  runat="server"  ForeColor="Red"></asp:Label>
      <br />
      Username:  <asp:TextBox  ID="txtUsername"  runat="server"></asp:TextBox>
      <br />
      Password: <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox>
      <br />
      <asp:Button  ID="btnLogin"  runat="server"  Text="Login"  OnClick="btnLogin_Click"  />
    </div>
  </form>
</body>
</html>
```

Page.aspx.cs:

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

namespace Practical12
{
        public partial class Page : System.Web.UI.Page
        {
```

```
                        protected void Page_Load(object sender, EventArgs e)
                        {
            if (Session["Username"] != null)
            {
                Response.Redirect("Dashboard.aspx");
            }
        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtUsername.Text; string
            password = txtPassword.Text;

            // Simulating user authentication
            if (username == "Kapil" && password == "24050")
            {
                Session["Username"] = username;
                Response.Redirect("Dashboard.aspx");
            }
            else
            {
                lblMessage.Text = "Invalid username or password!";
            }
        }
    }
}
```

Dashboard.aspx:

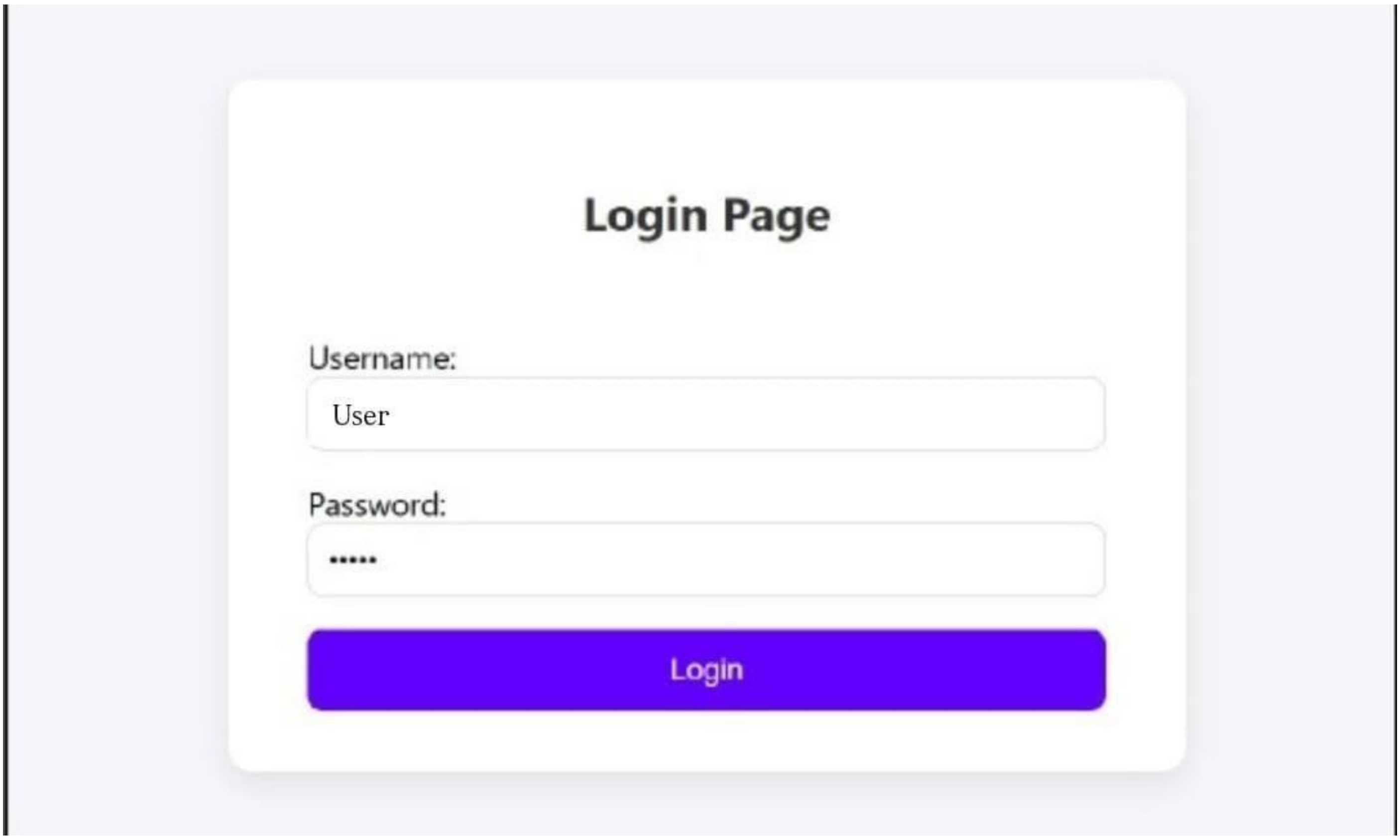
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Dashboard.aspx.cs"
Inherits="Practical12.Dashboard" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
    <title>Dashboard</title>
    <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Welcome, <asp:Label ID="lblUser" runat="server"></asp:Label>!</h2>
            <asp:Button  ID="btnLogout"  runat="server"  Text="Logout"  OnClick="btnLogout_Click"  />
        </div>
    </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 Practical12
{
            public partial class Dashboard : System.Web.UI.Page
            {
                        protected void Page_Load(object sender, EventArgs e)
                        {
          if (Session["Username"] == null)
          {
             Response.Redirect("Page.aspx");
          }
          else
          {
             lblUser.Text = Session["Username"].ToString();
          }
       }

       protected void btnLogout_Click(object sender, EventArgs e)
       {
          Session.Abandon();
          Response.Redirect("Page.aspx");
       }
    }
}
```

OUTPUT:



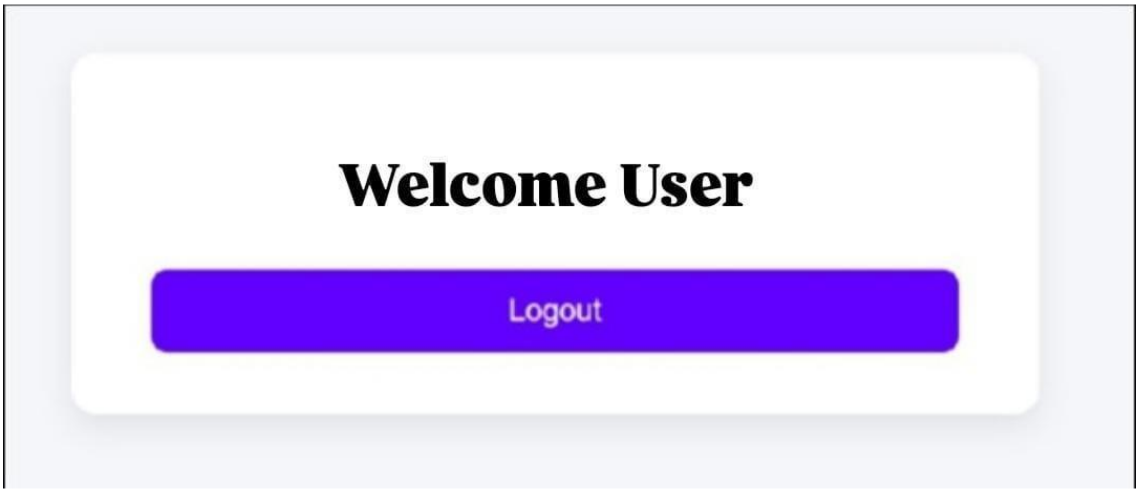
Login Page

Username:

User

Password:

•••••



Welcome User

PRACTICAL 13: Build a web page using AJAX Controls.

CODE:

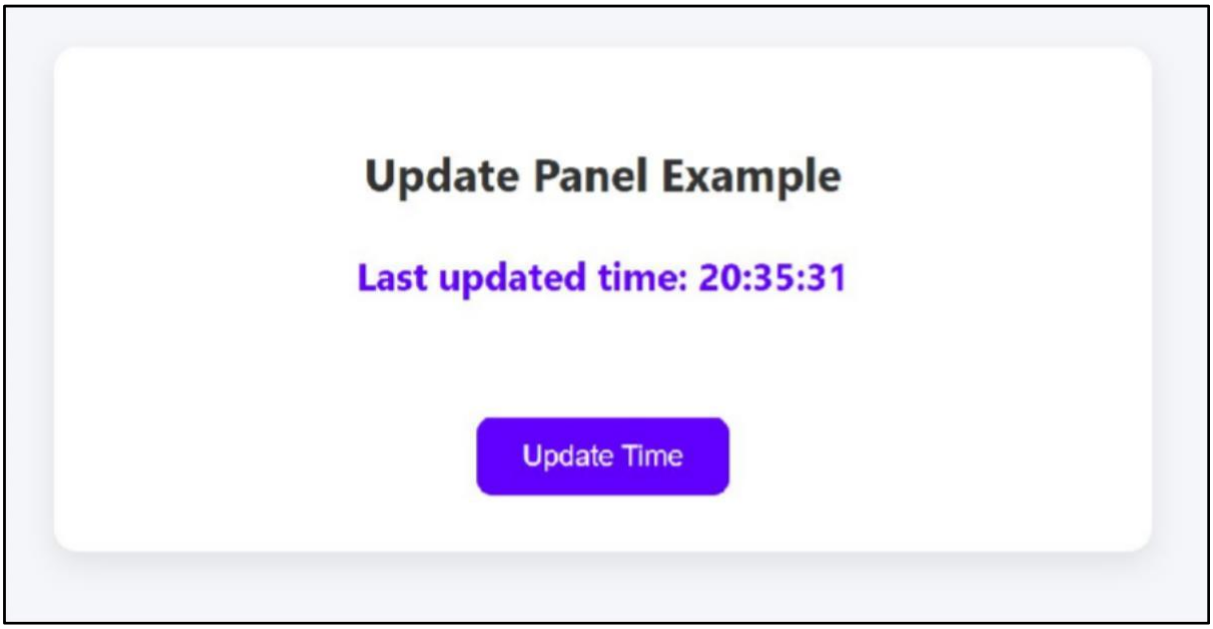
Page.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Page.aspx.cs"
Inherits="Practical13.Page" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
  <title>AJAX UpdatePanel Example</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>

    <h2>Update Panel Example</h2>

    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
      <ContentTemplate>
        <asp:Label ID="lblTime" runat="server" Font-Bold="True"></asp:Label>
        <br /><br />
        <asp:Button ID="btnUpdate" runat="server" Text="Update Time"
OnClick="btnUpdate_Click" />
      </ContentTemplate>
    </asp:UpdatePanel>
  </form>
</body>
</html>
```

Page.aspx.cs:

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

namespace Practical13
{
        public partial class Page : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {
      if (!IsPostBack)
      {
            lblTime.Text = "Last updated time: " + DateTime.Now.ToString("HH:mm:ss");
          }
        }

        protected void btnUpdate_Click(object sender, EventArgs e)
        {
          lblTime.Text = "Last updated time: " + DateTime.Now.ToString("HH:mm:ss");
        }
    }
}
```

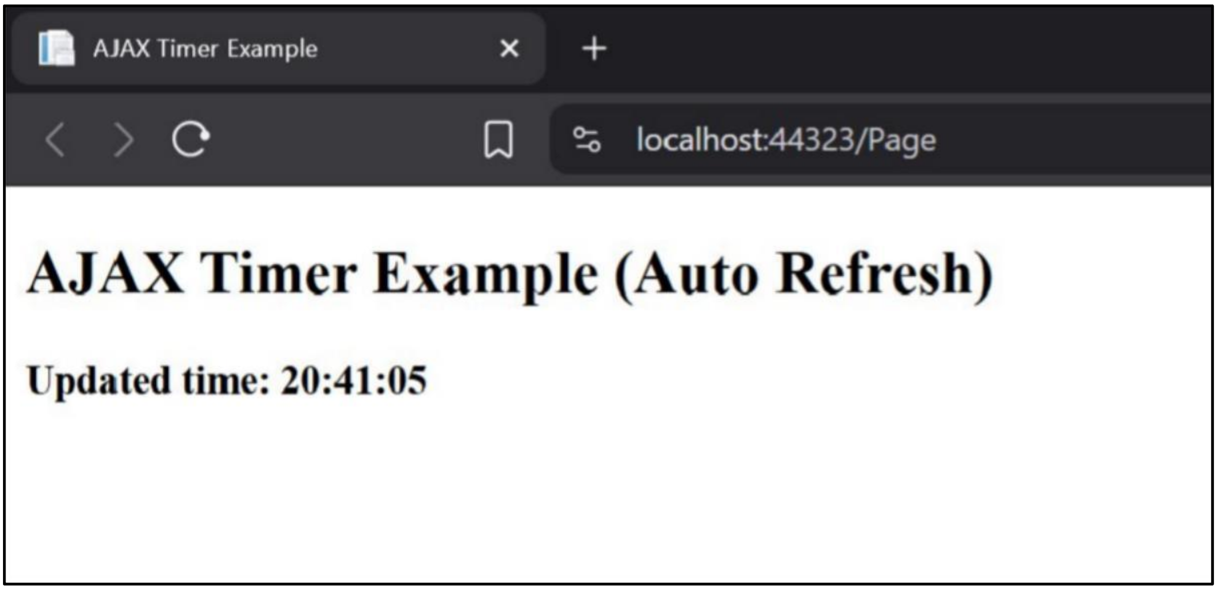
OUTPUT:



Update Panel Example

Last updated time: 20:35:31

For Auto Update:



AJAX Timer Example (Auto Refresh)

Updated time: 20:41:05

PRACTICAL 14: Build a web application to create and use web service in ASP.net

CODE:

Default.aspx:

```
<!DOCTYPE html>
<html>
<head runat="server">
   <title>Calculator Web Service Client</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Number 1: <asp:TextBox ID="txtA" runat="server" /><br />
         Number 2: <asp:TextBox ID="txtB" runat="server" /><br /><br />
         <asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd_Click" />
         <asp:Button ID="btnSub" runat="server" Text="Subtract" OnClick="btnSub_Click" /><br /><br
/>
         Result: <asp:Label ID="lblResult" runat="server" Text="" /> </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

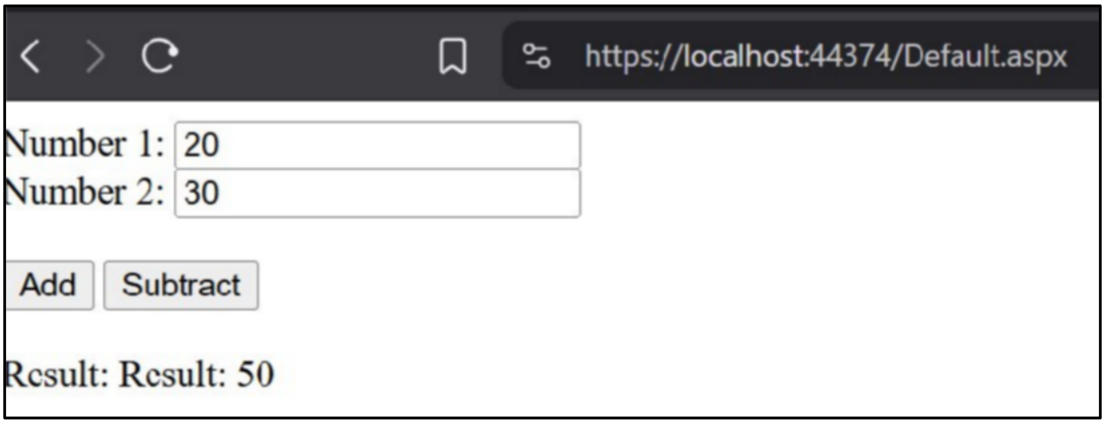
```
using System;
using CalculatorApp.CalcRef;

namespace CalculatorApp
{
   public partial class Default : System.Web.UI.Page
   {
      CalculatorServiceSoapClient client;

      protected void Page_Load(object sender, EventArgs e)
      {
         client = new CalculatorServiceSoapClient();
      }

      protected void btnAdd_Click(object sender, EventArgs e)
      {
         int a = int.Parse(txtA.Text);
         int b = int.Parse(txtB.Text);
         int result = client.Add(a, b);
         lblResult.Text = "Result: " + result;
      }

      protected void btnSub_Click(object sender, EventArgs e)
      {
         int a = int.Parse(txtA.Text); int b
         = int.Parse(txtB.Text); int result
         = client.Subtract(a, b);
```

```
            lblResult.Text = "Result: " + result;
        }
    }
}
```

CalculatorService.asmx.cs:

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

namespace CalculatorApp
{
    [WebService(Namespace          =          "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo                                    =
    WsiProfiles.BasicProfile1_1)]  public  class  CalculatorService  :
    WebService
    {
        [WebMethod]
        public int Add(int a, int b)
        {
            return a + b;
        }

        [WebMethod]
        public int Subtract(int a, int b)
        {
            return a - b;
        }
    }
}
```

OUTPUT:



Number 1: 20
Number 2: 30

Rcsult: Rcsult: 50

PRACTICAL 15: Build a web application to create and use WCF service in ASP.net

CODE:

Service.cs:

```
using System;
using    System.Collections.Generic;
using         System.Linq;         using
System.Runtime.Serialization;
using  System.ServiceModel;  using
System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name
"Service" in code, svc and config file together.
public class Service : IService
{
   public string GetData(int value)
   {
      return string.Format("You entered: {0}", value);
   }

   public double add(double a, double b)
   {
      return a + b;
   }

   public double sub(double a, double b)
   {
      return a - b;
   }

   public double mul(double a, double b)
   {
      return a * b;
   }

   public double div(double a, double b)
   {
      return a / b;
   }
   public CompositeType GetDataUsingDataContract(CompositeType composite)
   {
      if (composite == null)
      {
         throw new ArgumentNullException("composite");
      }
      if (composite.BoolValue)
      {
         composite.StringValue += "Suffix";
      }
```

```
        return composite;
    }
}
```

IService.cs:

```
using System;
using    System.Collections.Generic;
using         System.Linq;         using
System.Runtime.Serialization;
using  System.ServiceModel;  using
System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name
"IService" in both code and config file together.
[ServiceContract]
public interface IService
{

    [OperationContract]
    string GetData(int value);

    [OperationContract]
    double add(double a, double b);

    [OperationContract]
    double sub(double a, double b);

    [OperationContract]
    double mul(double a, double b);

    [OperationContract]
    double div(double a, double b);


    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);

    // TODO: Add your service operations here
}

// Use a data contract as illustrated in the sample below to add composite types to
service operations. [DataContract]
public class CompositeType
{
    bool boolValue = true; string
    stringValue     =     "Hello     ";
    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }

    [DataMember]
    public string StringValue
    {
        get { return stringValue; }
        set { stringValue = value; }
    }
}
```

Default.aspx:

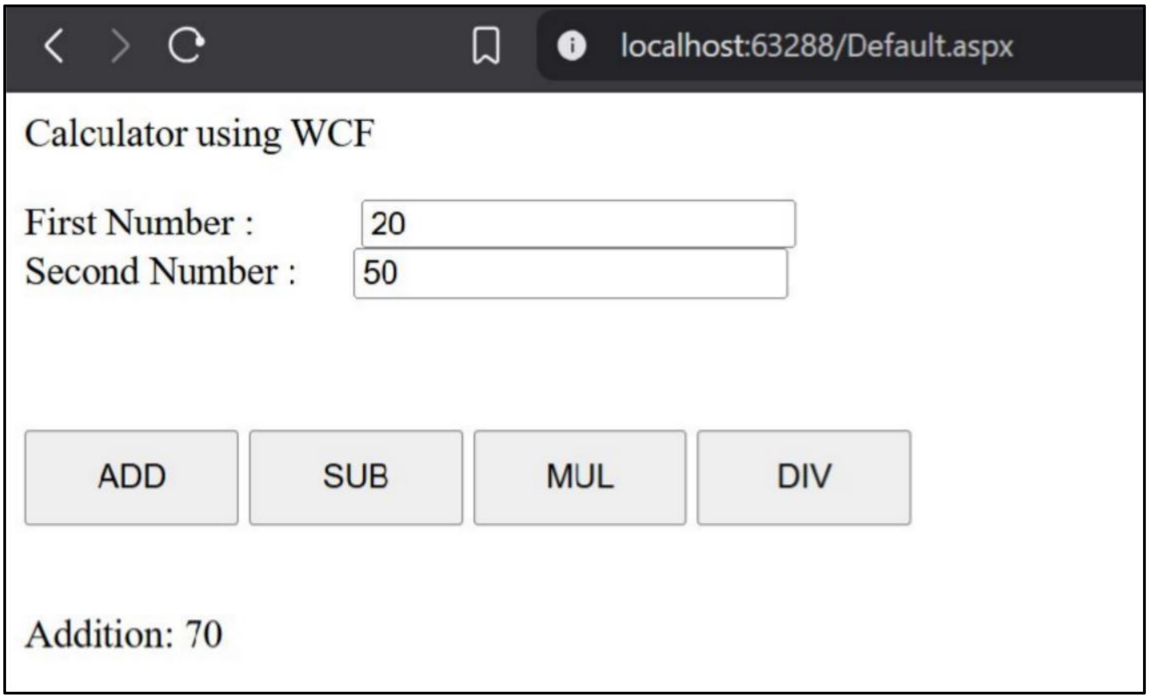
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form    id="form1"    runat="server">
        <div>
            Calculator using WCF<br />
            <br />
            First Number :<asp:TextBox ID="TextBox1" runat="server" style="margin-left: 44px"
Width="171px"></asp:TextBox> <br />
            Second Number :
            <asp:TextBox    ID="TextBox2"    runat="server"    style="margin-left:    19px"
Width="171px"></asp:TextBox>
            <br />
            <br />
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" Height="39px" OnClick="Button1_Click"
Text="ADD" Width="88px" />
            <asp:Button ID="Button2" runat="server" Height="39px" OnClick="Button2_Click" Text="SUB"
Width="88px" />
            <asp:Button ID="Button3" runat="server" Height="39px" OnClick="Button3_Click"
Text="MUL" Width="88px" />
            <asp:Button ID="Button4" runat="server" Height="39px" OnClick="Button4_Click" Text="DIV"
Width="88px" />
            <br />
            <br />
            <br />
            <asp:Label ID="Label1" runat="server" Text="RESULT :"></asp:Label>
            <br />
            <br />
            <br />
            <br />
            <br />
            <br />
            <br />
          </div>
        </form>
</body>
</html>
```

Default.aspx.cs:

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

public partial class _Default : System.Web.UI.Page
{
   ServiceReference1.ServiceClient service = new ServiceReference1.ServiceClient();

   protected void Page_Load(object sender, EventArgs e)
   {

   }

   protected void Button1_Click(object sender, EventArgs e)
   {
      double a = Convert.ToDouble(TextBox1.Text); double
      b = Convert.ToDouble(TextBox2.Text);
double result = service.add(a, b);

      Label1.Text = "Addition: " + result.ToString();
   }

   protected void Button2_Click(object sender, EventArgs e)
   {
      double a = Convert.ToDouble(TextBox1.Text); double
      b = Convert.ToDouble(TextBox2.Text);
double result = service.sub(a, b);

      Label1.Text = "Subtraction: " + result.ToString();
   }
   protected void Button3_Click(object sender, EventArgs e)
   {
      double a = Convert.ToDouble(TextBox1.Text);
      double b = Convert.ToDouble(TextBox2.Text);
```

```
        double result = service.mul(a, b);

        Label1.Text = "Multiplication: " + result.ToString();

    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        double a = Convert.ToDouble(TextBox1.Text);
        double b = Convert.ToDouble(TextBox2.Text);

        double result = service.div(a, b);

        Label1.Text = "Division: " + result.ToString();
    }
}
```

OUTPUT:

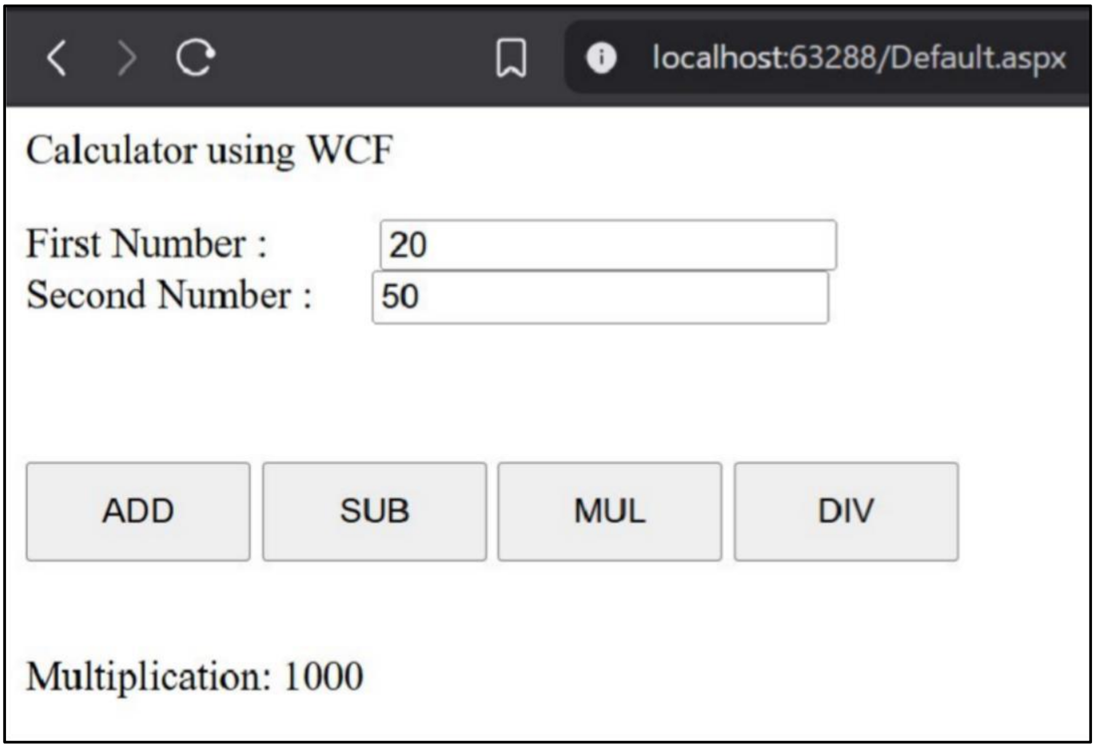


Calculator using WCF

First Number : 20

Second Number : 50

Addition: 70



Calculator using WCF

First Number : 20

Second Number : 50

Multiplication: 1000

PRACTICAL 16: MVC Application using Entity Framework

Step 1: Create a Database in SQL Server

Open SQL Server Management Studio (SSMS) or Visual Studio SQL Server Object Explorer.
Create a Database named StudentDB.
Create a Table using the following SQL: Create Database StudentDB
use StudentDB

```
CREATE TABLE Students (
   Id INT PRIMARY KEY IDENTITY,
   Name NVARCHAR(100) NOT NULL,
   Email NVARCHAR(100) NOT NULL,
   Age INT NOT NULL
);
```

Step 2: Create a New ASP.NET MVC Project

1. Open **Visual Studio**
2. Select **Create a new project**
3. Choose: **ASP.NET Web Application (.NET Framework)**
4. Name: StudentMVCApp
5. Choose **MVC** as the template Click **Create**

Step 3: Add Entity Framework Model

1. Right-click the **Models** folder → Add → New Item
2. Choose **ADO.NET Entity Data Model**
3. Name it: StudentModel.edmx
4. Choose: "EF Designer from database"
5. Select your SQL Server database (StudentDB)
6. Select the Students table

Finish to generate model classes

Step 4: Create Controller

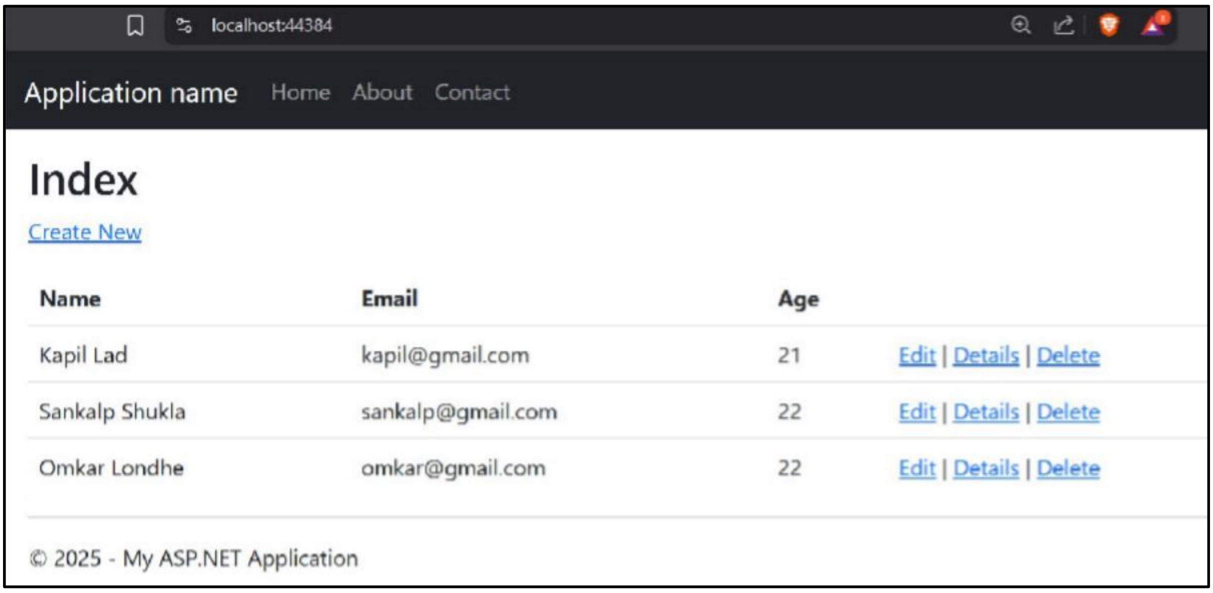
1. Right-click **Controllers** → Add → Controller
2. Choose: **MVC 5 Controller with views, using Entity Framework**
3. Model class: Student
4. Data context: StudentDBEntities (if using .edmx) or StudentDBContext Click **Add**

Step 5: Set Default Route

In App_Start/RouteConfig.cs, change default route to: csharp CopyEdit defaults: new
{ controller = "Students", action = "Index", id = UrlParameter.Optional }

RUN Application

OUTPUT:

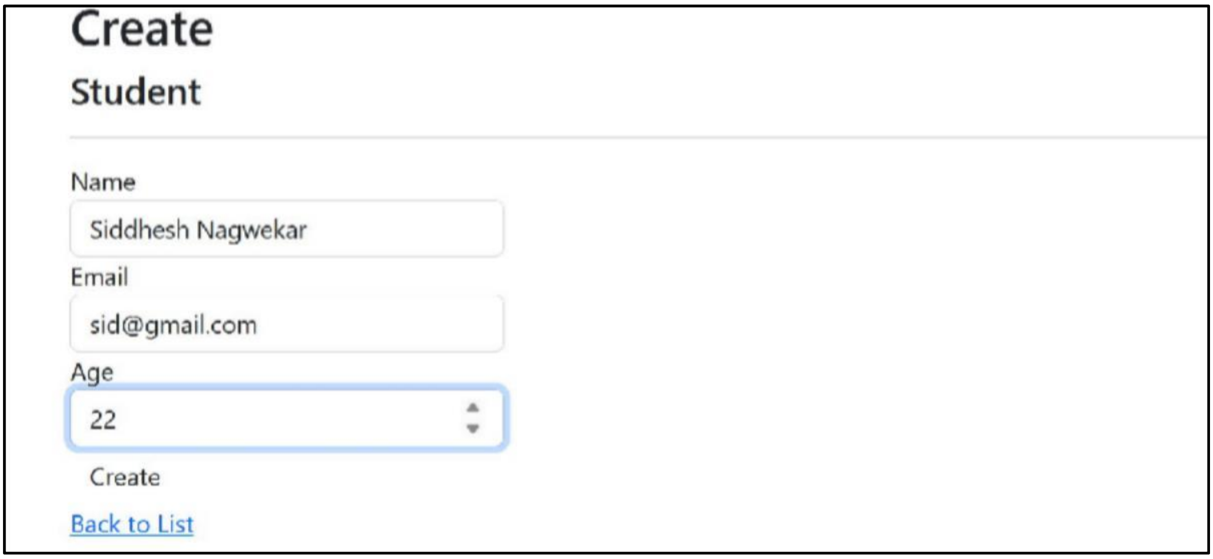


Index

Create New

Name	Email	Age	
Kapil Lad	kapil@gmail.com	21	Edit | Details | Delete
Sankalp Shukla	sankalp@gmail.com	22	Edit | Details | Delete
Omkar Londhe	omkar@gmail.com	22	Edit | Details | Delete

© 2025 - My ASP.NET Application



Create

Student

Name

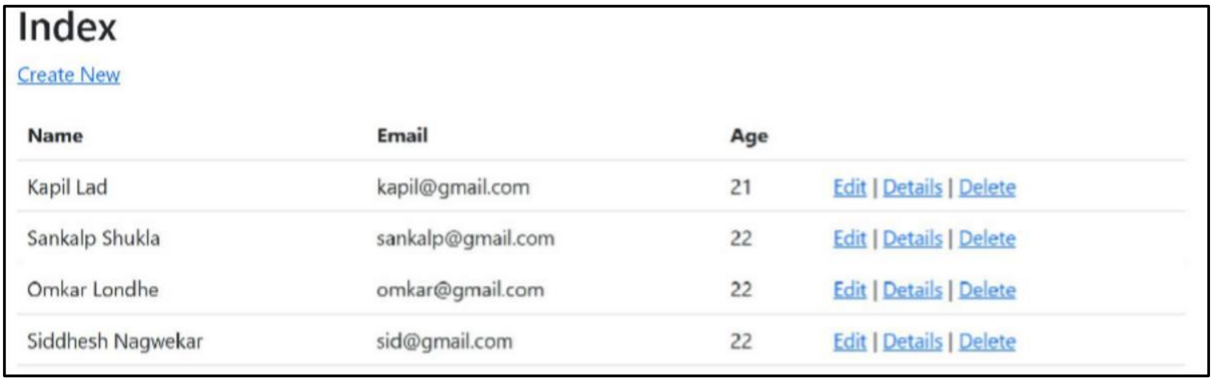
Siddhesh Nagwekar

Email

sid@gmail.com

Age

22



Index

Create New

Name	Email	Age	
Kapil Lad	kapil@gmail.com	21	Edit | Details | Delete
Sankalp Shukla	sankalp@gmail.com	22	Edit | Details | Delete
Omkar Londhe	omkar@gmail.com	22	Edit | Details | Delete
Siddhesh Nagwekar	sid@gmail.com	22	Edit | Details | Delete