WINDOWS FORM APPLICATION

A. Design a calculator UI based application using basic window forms controls.

Code:-

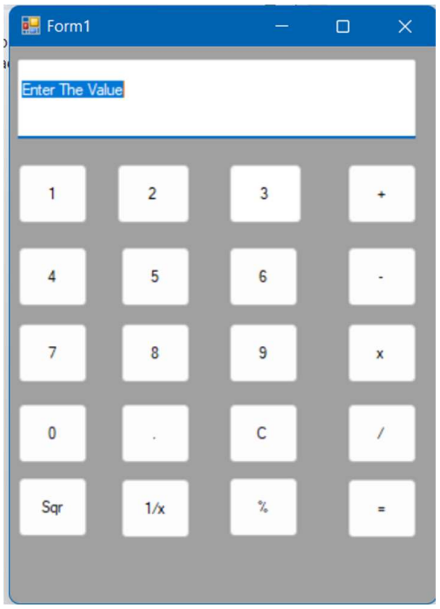
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace calcu
{
    public partial class C22093 : Form
    {
        public C22093()
        {
            InitializeComponent();
        }

        float num1;
        double ans;
        int count;


        private void button1_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 1;

        }

        private void button2_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 2;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 3;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 4;
        }

        private void button5_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 5;
```

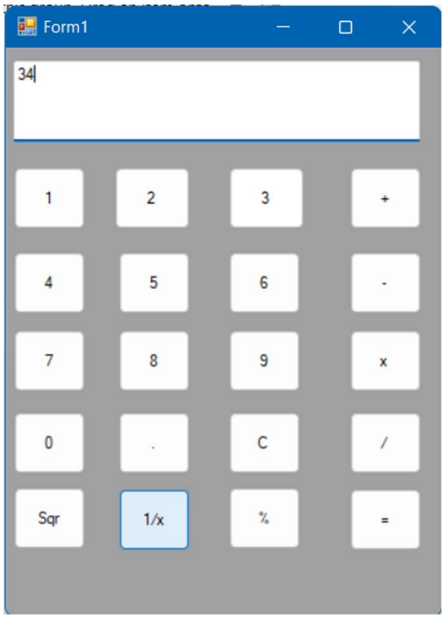
```
        }


 private void button6_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 6;
        }

        private void button7_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 7;
        }

        private void button8_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 8;
        }

        private void button9_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 9;
        }

        private void button10_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 0;
        }

        private void button11_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + ".";
        }

        private void button12_Click(object sender, EventArgs e)
        {
            label1.Text = "Enter Your Value";
            textBox1.Text = null;
        }

        private void button17_Click(object sender, EventArgs e)
        {
            ans = float.Parse(textBox1.Text);
            ans = ans*ans;
            textBox1.Text = ans.ToString();


        }

        private void button20_Click(object sender, EventArgs e)
        {
            ans = float.Parse(textBox1.Text);
            ans = 1/ans;
            textBox1.Text = ans.ToString();

        }

        private void button18_Click(object sender, EventArgs e)
        {
            ans = float.Parse(textBox1.Text);
```

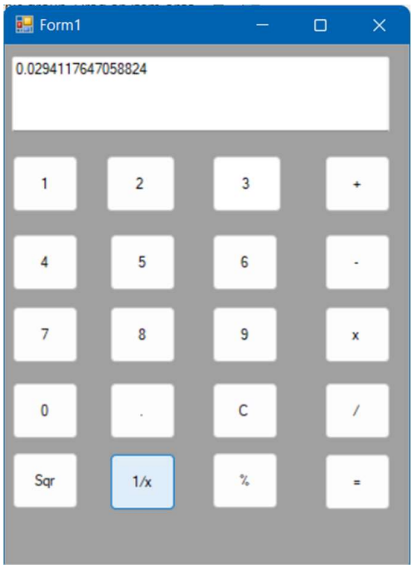
```
            ans = ans/100;
            textBox1.Text = ans.ToString();

        }


        private void button13_Click(object sender, EventArgs e)
        {
            num1 = float.Parse(textBox1.Text);
            textBox1.Clear();
            textBox1.Focus();
            count = 1;

        }

        private void button14_Click(object sender, EventArgs e)
        {
            num1 = float.Parse(textBox1.Text);
            textBox1.Clear();
            textBox1.Focus();
            count = 2;

        }

        private void button15_Click(object sender, EventArgs e)
        {
            num1 = float.Parse(textBox1.Text);
            textBox1.Clear();
            textBox1.Focus();
            count = 3;

        }

        private void button16_Click(object sender, EventArgs e)
        {
            num1 = float.Parse(textBox1.Text);
            textBox1.Text = "0";
            textBox1.Clear();
            textBox1.Focus();
            count = 4;

        }
        private void button19_Click(object sender, EventArgs e)
        {
            compute(count);
        }

        public void compute (int count)
        {
            float num2 =float.Parse(textBox1.Text);
            switch (count)
            {
                case 1:
                    ans = num1+num2;
                    textBox1.Text = ans.ToString();
                    break;
                case 2:
                    ans = num1 - num2;
                    textBox1.Text = ans.ToString();
                    break;

                case 3:
                    ans = num1 * num2;
                    textBox1.Text = ans.ToString();
                    break;
```

```
                case 4:
                    ans = num1 / num2;
                    textBox1.Text = ans.ToString();
                    break;
            }
        }
    }
}
```

OUTPUT:-







C# CONSOLE

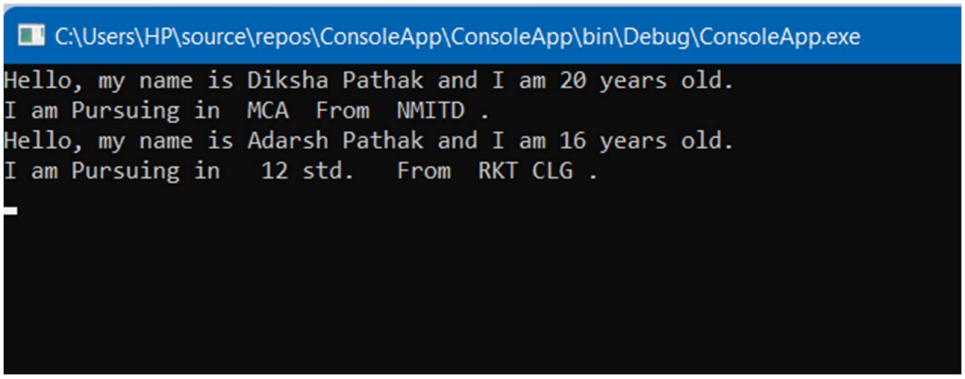
A. Design Application using class and objects.

CODE:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp
{
    // Define a class called Person
    class Person
    {
        // Properties
        public string Name { get; set; }
        public int Age { get; set; }
        public string Degree { get; set; }
        public string Clg { get; set; }

        // Method to introduce the person
        public void Introduce()
        {
            Console.WriteLine("Hello, my name is " + Name + " and I am " + Age + "
years old.");
            Console.WriteLine("I am Pursuing in " + Degree + " From " + Clg + ".");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Create a new instance of the Person class
            Person person = new Person();

            // Set the properties of the person object
            person.Name = "Diksha Pathak";
            person.Age = 20;
            person.Degree = " MCA ";
            person.Clg = " NMITD ";

            // Call the Introduce method of the person object
             person.Introduce();

            // Create another instance of the Person class
            Person anotherPerson = new Person();

            // Set the properties of the anotherPerson object
            anotherPerson.Name = "Adarsh Pathak";
            anotherPerson.Age = 16;

            anotherPerson.Degree = "  12 std.  ";
            anotherPerson.Clg = " RKT CLG ";
```

```
            // Call the Introduce method of the anotherPerson object

            anotherPerson.Introduce();

            // Wait for user input before closing the console window
            Console.ReadLine();
        }
    }
}
```

OUTPUT:-



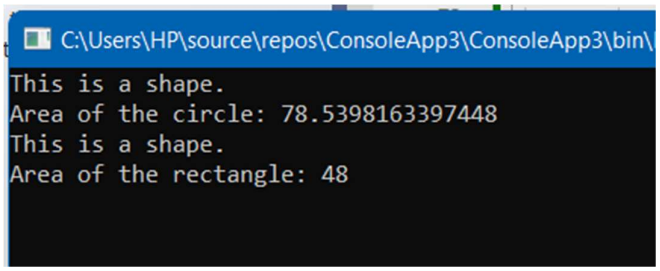
B. Design an application using inheritance and abstract class.

Code:-

```
using System;

namespace ConsoleApp
{
    // Define an abstract class called Shape
    abstract class Shape
    {
        // Abstract method to calculate area
        public abstract double CalculateArea();

        // Method to display shape information
        public void display()
            {
                Console.WriteLine("This is a shape.");

        }

            //public void Display()
            //{
            // Console.WriteLine("This is a shape.",+);
            //}

        }

        // Define a derived class called Circle
        class Circle : Shape
        {
            // Properties
            public double Radius { get; set; }

            // Implementation of CalculateArea method for Circle
            public override double CalculateArea()
            {
                return Math.PI * Radius * Radius;
            }
```

```
            // Override Display method to provide specific information for Circle
            /* public override void display()
             {
                 Console.WriteLine("This is a circle with radius " + Radius);
             }*/
        }

        // Define another derived class called Rectangle
        class Rectangle : Shape
        {
            // Properties
            public double Width { get; set; }
            public double Height { get; set; }

            // Implementation of CalculateArea method for Rectangle
            public override double CalculateArea()
            {
                return Width * Height;
            }

            // Override Display method to provide specific information for Rectangle
            /* public override void display()
             {
                 Console.WriteLine("This is a rectangle with width " + Width + " and
height " + Height);
             }*/
        }

        class Program
        {
            static void Main(string[] args)
            {
                // Create an instance of Circle
                Circle = new Circle();
                circle.Radius = 5;


                // Call CalculateArea method on the circle object
                double circleArea = circle.CalculateArea();

                // Call Display method on the circle object
                circle.display();
                Console.WriteLine("Area of the circle: " + circleArea);

                // Create an instance of Rectangle
                Rectangle rectangle = new Rectangle();
                rectangle.Width = 6;
                rectangle.Height = 8;

                // Call CalculateArea method on the rectangle object
                double rectangleArea = rectangle.CalculateArea();

                // Call Display method on the rectangle object
                rectangle.display();
                Console.WriteLine("Area of the rectangle: " + rectangleArea);

                // Wait for user input before closing the console window
                Console.ReadLine();
            }
        }
    }
```

Output:-



ASP.NET

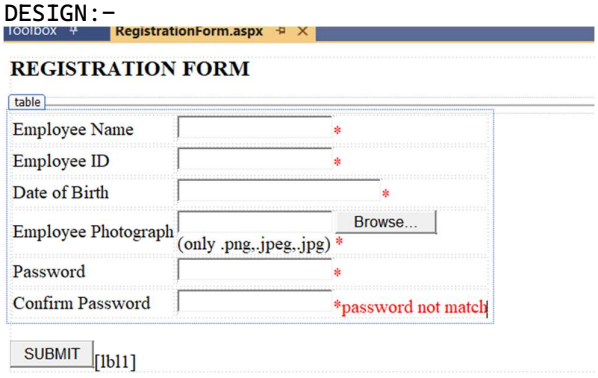
A. **Design a web application for an organisation with registration form and advanced controls(validation).**

Code:-

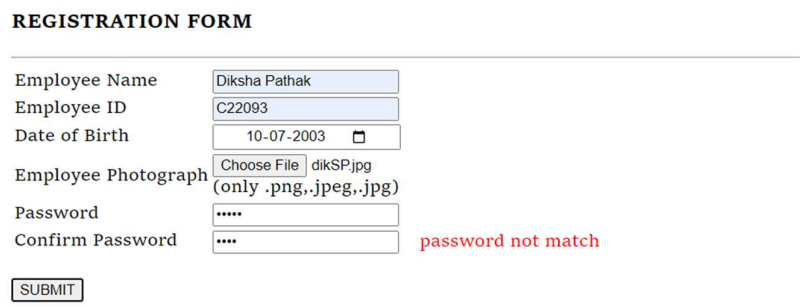
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="Registration.RegistrationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h3>REGISTRATION FORM</h3><hr />
            <table border="0">
                <tr>
                    <td>Employee Name</td>
                    <td>
                        <asp:TextBox runat="server" ID="txtempname" ></asp:TextBox>
                        <asp:RequiredFieldValidator ID="rvf1" ErrorMessage="*"
ControlToValidate="txtempname" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td>Employee ID</td>
                    <td>
                        <asp:TextBox runat="server" ID="txtempid" ></asp:TextBox>
                        <asp:RequiredFieldValidator ID="rfv2" ErrorMessage="*"
ControlToValidate="txtempid" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td>Date of Birth</td>
                    <td>
                        <asp:TextBox runat="server" ID="txtdob" TextMode="Date"
Style="padding: 1px 28px"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="rvf3" ErrorMessage="*"
ControlToValidate="txtdob" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td>Employee Photograph</td>
                    <td>
```

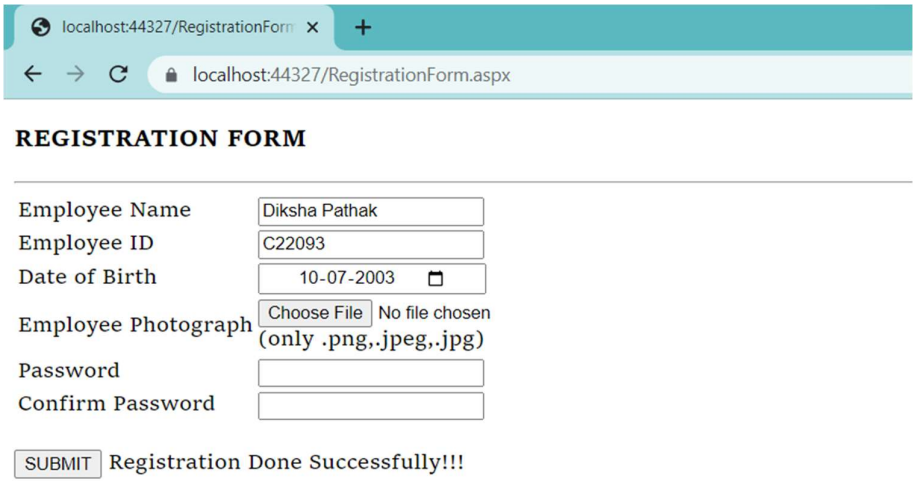
```
                        <asp:FileUpload runat="server" ID="fileupload"
accept=".png,.jpg,.jpeg"/>
                        <br />
                        (only .png,.jpeg,.jpg)
                        <asp:RequiredFieldValidator ID="rvf4" ErrorMessage="*"
ControlToValidate="fileupload" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td>Password</td>
                    <td>
                        <asp:TextBox runat="server" ID="txtpass"
TextMode="Password"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="rvf5" ErrorMessage="*"
ControlToValidate="txtpass" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td>Confirm Password</td>
                    <td>
                        <asp:TextBox runat="server" ID="txtconpass"
TextMode="Password"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="rvf6" ErrorMessage="*"
ControlToValidate="txtconpass" runat="server"
ForeColor="Red"></asp:RequiredFieldValidator>
                        <asp:CompareValidator ID="cv1" ErrorMessage="password not
match" ControlToCompare="txtpass" ControlToValidate="txtconpass" runat="server"
ForeColor="Red"></asp:CompareValidator>
                    </td>
                </tr>
            </table>
            <br />
            <asp:Button runat="server" ID="btnsubmit" Text="SUBMIT"
OnClick="btnsubmit_Click" />
            <asp:Label ID="lbl1" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

DESIGN:-



Output:-





B. Create website using master page and theme concept.

Code:- master pg

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs"
Inherits="WebApplication2.Site1" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>Tourist App</title>
        <linkrel="stylesheet" type="text/css" href="style.css">
    <asp:ContentPlaceHolder ID="head" runat="server">
```

```
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
            </asp:ContentPlaceHolder>

    <!-- Master Page Content -->
    <header>
        <nav>

                <a href="index.html">Home </a> 
                <a href="destination.html">Destinations </a> 
                <a href="activities.html">Activities </a> 
                <a href="contact.html">Contact </a> 

        </nav>
    </header>

    <div class="hero-image">
        <div class="hero-content">
            <h1>Welcome to the Tourist App</h1>
            <p>Explore new destinations, discover exciting activities, and plan your
next adventure!</p>
            <a class="cta-button" href="destination.html">Get Started</a>
        </div>
    </div>

    <section>
        <h2>About Us</h2>
        <p>Tourism is a social, cultural and economic phenomenon which entails the
movement of people to countries or places outside their usual environment for personal
or business/professional purposes.
</p>

    </section>

    <footer>
        <p>&copy; 2023 Tourist App. All rights reserved.</p>
    </footer>


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

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tourist App</title>
    <link rel="stylesheet" type="text/css" href="style.css">

</head>
<body>
    <!-- Master Page Content -->
    <header>
        <nav>

            <a href="index.html">Home </a> 
            <a href="destinations.html">Destinations </a> 
            <a href="activities.html">Activities </a> 
            <a href="contact.html">Contact </a> 
```

```
        </nav>
    </header>

    <div class="hero-image">
        <div class="hero-content">
            <h1>Welcome to the Tourist App</h1>
            <p>Explore new destinations, discover exciting activities, and plan your
next adventure!</p>
            <a class="cta-button" href="destination.html">Get Started</a>
        </div>
    </div>

    <section>
        <h2>About Us</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc at turpis vel
ligula congue tincidunt. Curabitur mollis diam in ligula commodo, ac aliquam dolor
fringilla.</p>
        <p>Mauris ac nisi in est semper gravida. Sed scelerisque mi sit amet libero
euismod finibus. Sed malesuada tortor ac sem euismod vestibulum.</p>
    </section>



    <footer>
        <p>&copy; 2023 Tourist App. All rights reserved.</p>
    </footer>
</body>
</html>
```

destinations.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tourist App - Destinations</title>
    <link rel="stylesheet" type="text/css" href="Style.css">

</head>
<body>
    <!-- Master Page Content -->
    <header>
        <nav>
            <a href="index.html">Home </a> 
            <a href="destination.html">Destinations </a> 
            <a href="activities.html">Activities </a> 
            <a href="contact.html">Contact </a> 
        </nav>
    </header>

    <h1>Explore Exciting Destinations</h1>

    <div class="destination-card">
        <img src="det1.jpg" alt="Destination 1">
        <h3>Mahabaleshwar</h3>
        <p>Mahabaleshwar is a hill station in India's forested Western Ghats range,
south of Mumbai. It features several elevated viewing points, such as Arthur’s Seat.
West of here is centuries-old Pratapgad Fort, perched atop a mountain spur. East,
Lingmala Waterfall tumbles off a sheer cliff.</p>
        <a href="destination1.html">Learn More</a>
    </div>

    <div class="destination-card">
        <img src="dest2.jpg" alt="Destination 2">
        <h3>Manali</h3>
        <p>Manali is a high-altitude Himalayan resort town in India’s northern
Himachal Pradesh state. It has a reputation as a backpacking center and honeymoon
```

```
destination. Set on the Beas River, it’s a gateway for skiing in the Solang Valley and
trekking in Parvati Valley.</p>
        <a href="destination2.html">Learn More</a>
    </div>

    <div class="destination-card">
        <img src="dest3.jpg" alt="Destination 3">
        <h3>kashmir</h3>
        <p>Dal is a lake in Srinagar, the summer capital of Jammu and Kashmir, India.
It is an urban lake, the second largest lake in Jammu and Kashmir, and the most
visited place in Srinagar by tourists and locals.</p>
        <a href="destination3.html">Learn More</a>
    </div>

    <footer>
        <p>&copy; 2023 Tourist App. All rights reserved.</p>
    </footer>
</body>
</html>
```

activities.html

```
    <!DOCTYPE html>
    <html>
    <head>
        <title>Tourist App - Activities</title>
        <link rel="stylesheet" type="text/css" href="Style.css">
    </head>
    <body>
        <!-- Master Page Content -->
        <header>
            <nav>
                <a href="index.html">Home </a> 
                <a href="destination.html">Destinations </a> 
                <a href="activities.html">Activities </a> 
                <a href="contact.html">Contact </a> 

            </nav>
        </header>

        <h1>Exciting Activities to Try</h1>

        <div class="activity-card">
            <img src="act1.jpg" alt="Activity 1">
            <h3>Adventurous sports in mahabaleshwar</h3>
            <p>Mahabaleshwar is a hill station in India's forested Western Ghats
range, south of Mumbai. It features several elevated viewing points, such as Arthur’s
Seat. West of here is centuries-old Pratapgad Fort, perched atop a mountain spur.
East, Lingmala Waterfall tumbles off a sheer cliff..</p>
            <a href="activity1.html">Learn More</a>
        </div>

        <div class="activity-card">
            <img src="act2.jpg" alt="Activity 2">
            <h3>20 Adventure sports in manali</h3>
            <p>Manali is a high-altitude Himalayan resort town in India’s northern
Himachal Pradesh state. It has a reputation as a backpacking center and honeymoon
destination. Set on the Beas River, it’s a gateway for skiing in the Solang Valley and
trekking in Parvati Valley.</p>
            <a href="activity2.html">Learn More</a>
        </div>

        <div class="activity-card">
            <img src="act3.jpg" alt="Activity 3">
            <h3>Adventure  in kashmir</h3>
```

```
            <p>Jammu and Kashmir is a region administered by India as a union
territory and consists of the southern portion of the larger Kashmir region, which has
been the subject of a dispute between India and Pakistan since 1947 and between India
and China since 1959.</p>
            <a href="activity3.html">Learn More</a>
        </div>

        <footer>
            <p>&copy; 2023 Tourist App. All rights reserved.</p>
        </footer>
    </body>
</html>
```

Contact.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tourist App - Contact</title>
    <link rel="stylesheet" type="text/css" href="style.css">

</head>
<body>
    <!-- Master Page Content -->
    <header>
        <nav>
            <a href="index.html">Home </a> 
            <a href="destination.html">Destinations </a> 
            <a href="activities.html">Activities </a> 
            <a href="contact.html">Contact </a> 
        </nav>
    </header>

    <h1>Contact Us</h1>

    <div class="contact-form">
        <form>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>

            <label for="email">Email:</label>
            <input type="text" id="email" name="email" required>

            <label for="message">Message:</label>
            <textarea id="message" name="message" rows="4" required></textarea>

            <button type="submit">Send Message</button>
        </form>
    </div>

    <footer>
        <p>&copy; 2023 Tourist App. All rights reserved.</p>
    </footer>
</body>
</html>
```

Style.css

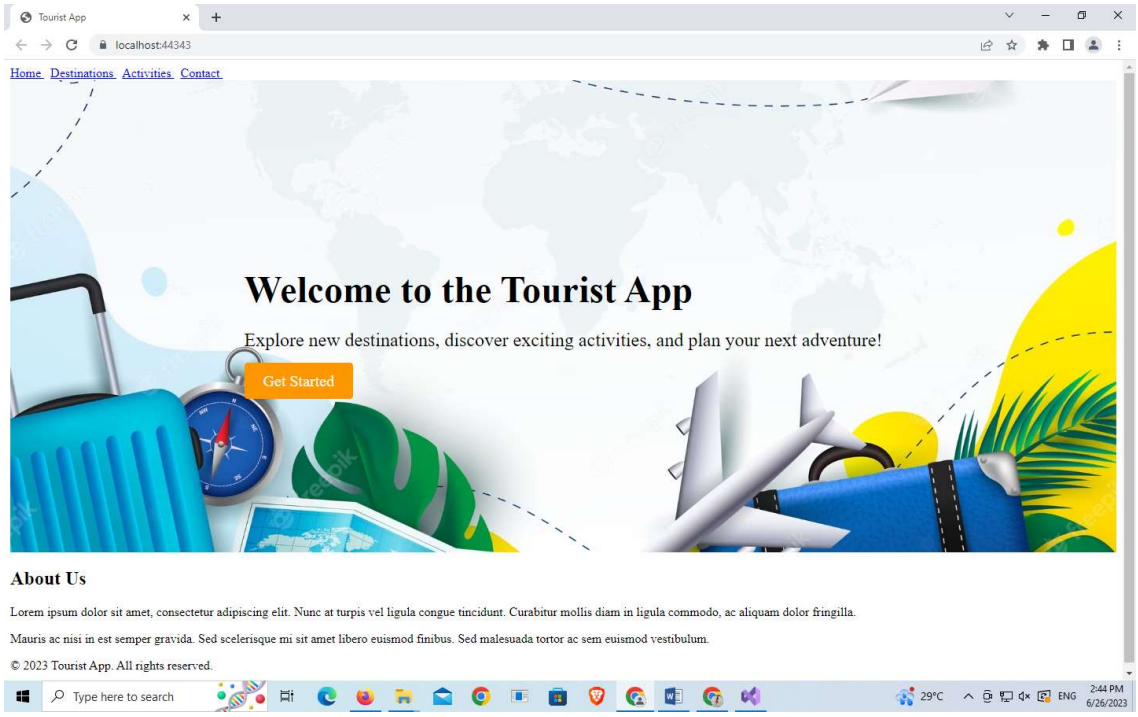
```
/* Additional styles specific to the home page */
```

```
    header nav {
        display: inline;
    }

        header nav  {
            padding-right: 20px; /* Adds 20 pixels of space within each item */
        }
.hero-image {
    background-image: url("background.jpg");
    background-size: cover;
    background-position: center;
    height: 600px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-size: 24px;
}

.hero-image h1 {
    font-size: 48px;
    margin-bottom: 20px;
}

.cta-button {
    background-color: #ff9800;
    color: white;
    padding: 12px 24px;
    border-radius: 4px;
    font-size: 20px;
    text-decoration: none;
}

    .cta-button:hover {
        background-color: #ffac33;
    }


.destination-card {
    border: 1px solid #ccc;
    border-radius: 4px;
    padding: 20px;
    margin-bottom: 20px;
}

    .destination-card img {
        width: 20%;
        border-radius: 4px;
        margin-bottom: 10px;
    }

    .destination-card h3 {
        margin: 0;
        font-size: 24px;
    }

    .destination-card p {
        margin: 10px 0;
    }
/* Additional styles specific to the activities page */
.activity-card {
    border: 1px solid #ccc;
    border-radius: 4px;
    padding: 20px;
    margin-bottom: 20px;
```

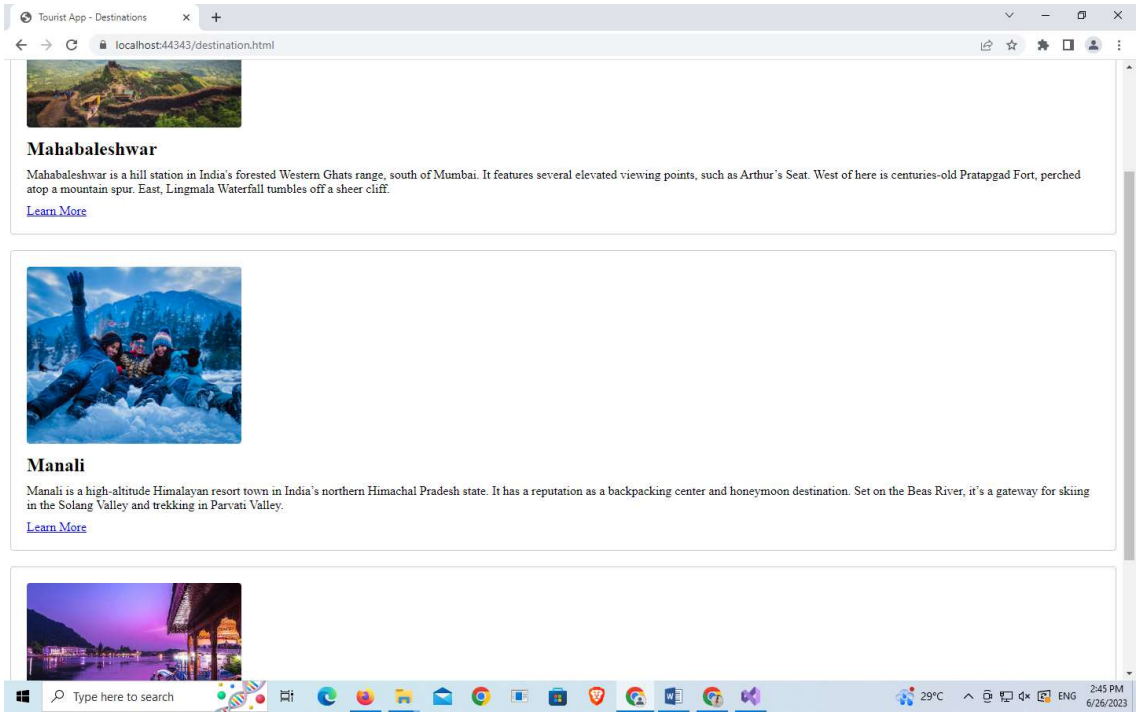
```
}

    .activity-card img {
        width: 20%;
        border-radius: 4px;
        margin-bottom: 10px;
    }

    .activity-card h3 {
        margin: 0;
        font-size: 24px;
    }

    .activity-card p {
        margin: 10px 0;
    }
.contact-form {
    max-width: 500px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

    .contact-form label {
        display: block;
        margin-bottom: 10px;
        font-weight: bold;
    }

    .contact-form input[type="text"],
    .contact-form textarea {
        width: 100%;
        padding: 10px;
        border-radius: 4px;
        border: 1px solid #ccc;
        margin-bottom: 20px;
    }

    .contact-form button {
        background-color: #ff9800;
        color: white;
        padding: 10px 20px;
        border-radius: 4px;
        font-size: 16px;
        border: none;
        cursor: pointer;
    }

        .contact-form button:hover {
            background-color: #ffac33;
        }
.hero-image {
    background-image: url("background.jpg");
    background-size: cover;
    background-position: center;
    height: 600px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: black;
    font-size: 24px;
}

    .hero-image h1 {
```

```
        font-size: 48px;
        margin-bottom: 20px;
    }

.cta-button {
    background-color: #ff9800;
    color: white;
    padding: 12px 24px;
    border-radius: 4px;
    font-size: 20px;
    text-decoration: none;
}

    .cta-button:hover {
        background-color: #ffac33;
       }
```

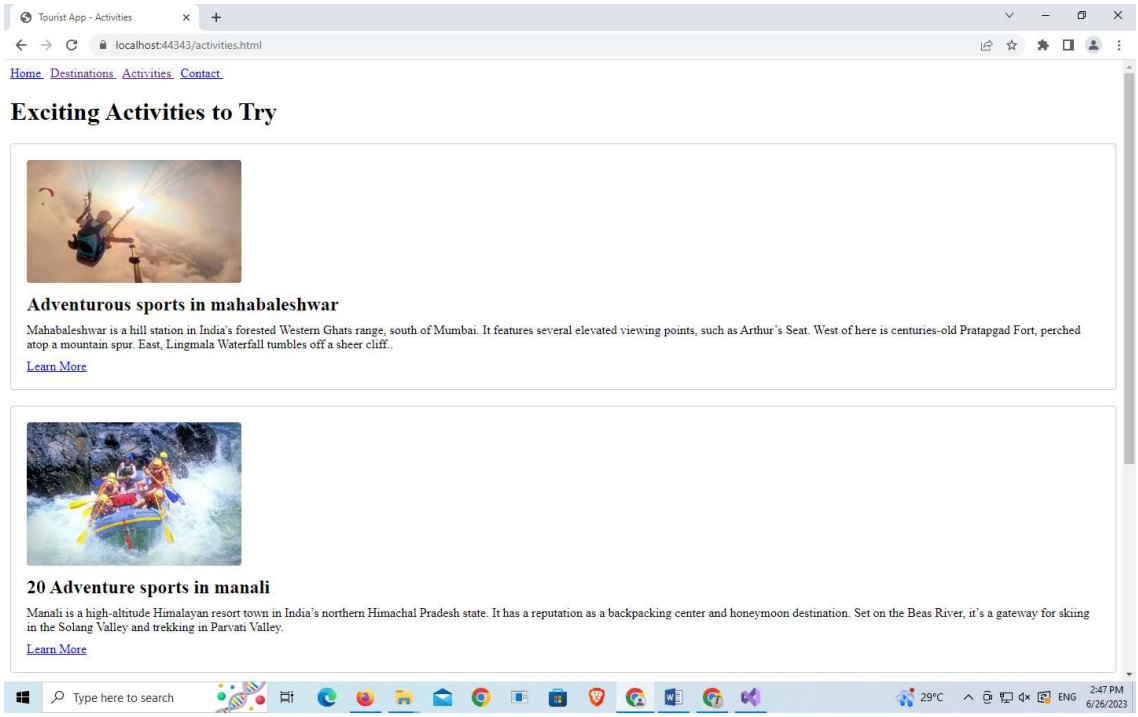
Output:-



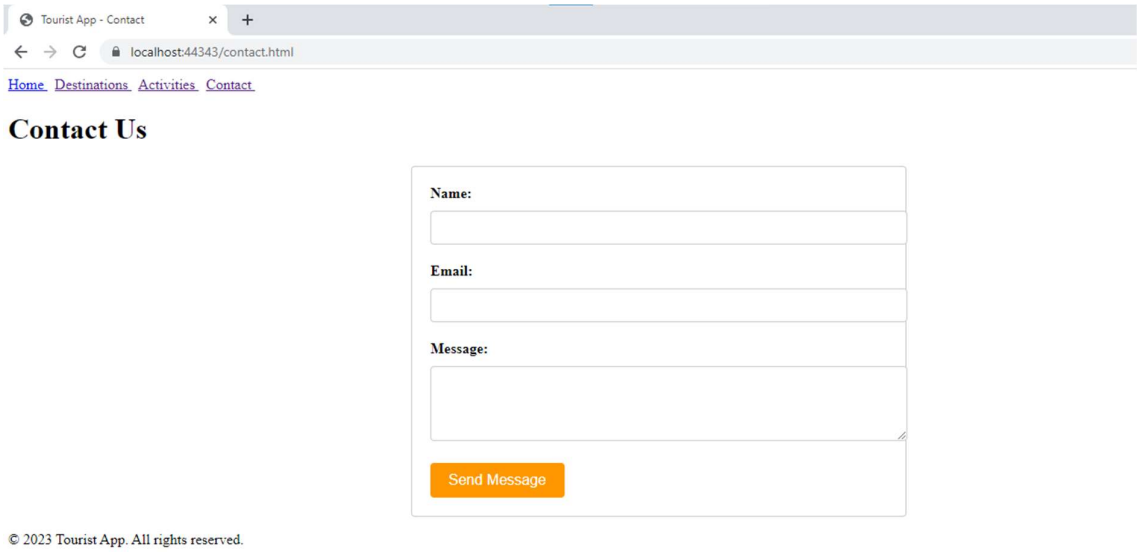
Destination



Activities



Contact



ADO.NET PRACTICAL

4(A) AIM: Create a webpage that demonstrates the use of data bound controls of ASP.NET

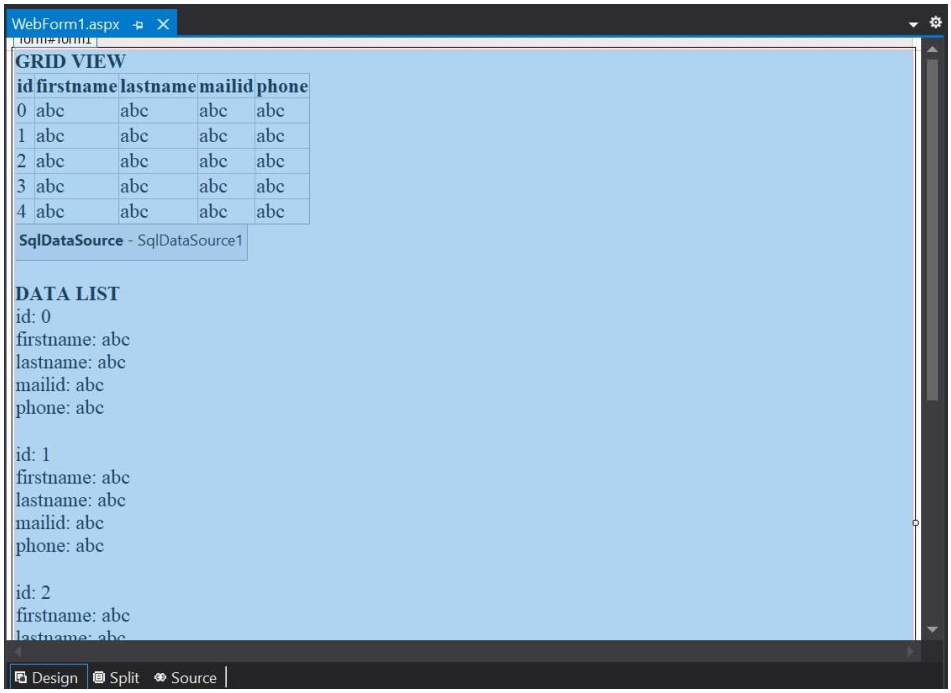
Code:-

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="DataBound_Controls.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>GRID VIEW</strong>
            <asp:GridView ID="GridView1" runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" DataKeyNames="id">
                <Columns>
                    <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
ReadOnly="True" SortExpression="id" />
                    <asp:BoundField DataField="firstname" HeaderText="firstname"
SortExpression="firstname" />
                    <asp:BoundField DataField="lastname" HeaderText="lastname"
SortExpression="lastname" />
                    <asp:BoundField DataField="mailid" HeaderText="mailid"
SortExpression="mailid" />
```

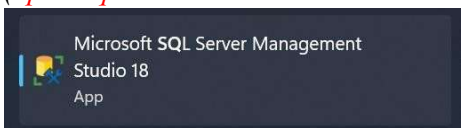
```
                    <asp:BoundField DataField="phone" HeaderText="phone"
SortExpression="phone" />
                </Columns>
            </asp:GridView>             <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:TestConnectionString %>" SelectCommand="SELECT * FROM
[tbllogin]"></asp:SqlDataSource>
            <strong>
            <br />
            DATA LIST</strong><br />
            <asp:DataList ID="DataList1" runat="server" DataKeyField="id"
DataSourceID="SqlDataSource1">                 <ItemTemplate>
id:
                    <asp:Label ID="idLabel" runat="server" Text='<%# Eval("id") %>' />
                    <br />
firstname:
                    <asp:Label ID="firstnameLabel" runat="server" Text='<%#
Eval("firstname") %>' />
<br />
lastname:
                    <asp:Label ID="lastnameLabel" runat="server" Text='<%#
Eval("lastname") %>' />                     <br />
mailid:
                    <asp:Label ID="mailidLabel" runat="server" Text='<%# Eval("mailid")
%>' />
                    <br />
phone:
                    <asp:Label ID="phoneLabel" runat="server" Text='<%# Eval("phone") %>'
/>
                    <br />
<br />
                </ItemTemplate>
            </asp:DataList>
        </div>
    </form>
</body>
</html>
```

OUTPUT

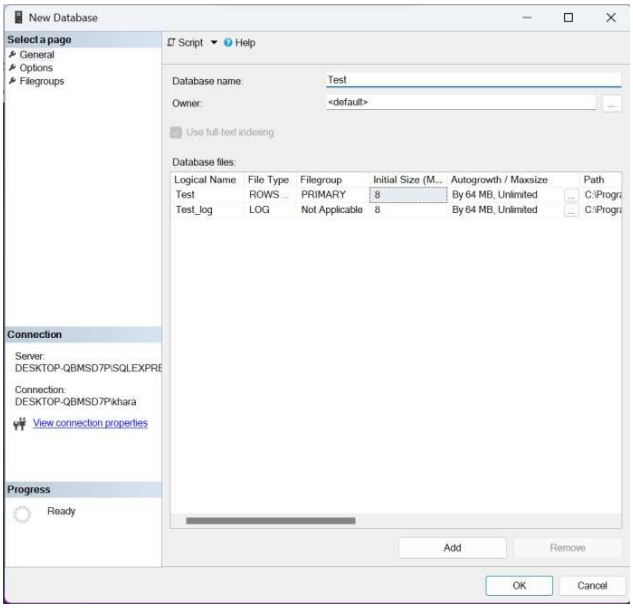


OUTPUT

B. AIM: ***Design a webpage to demonstrate a connection-oriented architecture.***
(open Sql Server >> then connect to server)



(create a new database – right click on database >> new database(name a database "Test"))



(In Database (Test) expand options ---right click on tables >> new >> table(create a table "tbllogin"))

(Insert records into tbllogin)

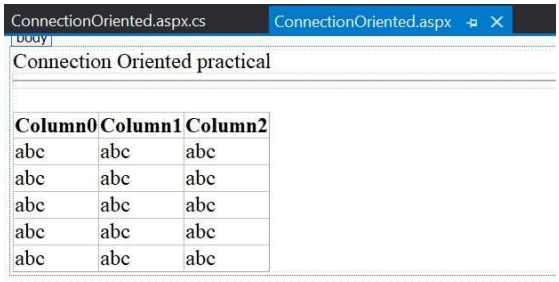
(create a new asp.net framework project with empty forms >> then add a new webform)

(then in visual Studio click on tools from menubar >> connect to database)

1. *Provide server name*
2. *Then select your database and test connection*
3. *Now you will have connection string to connect through*

Form.aspx

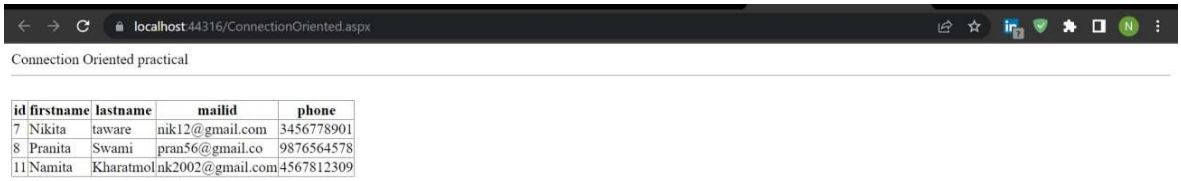
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ConnectionOriented.aspx.cs"
Inherits="DataBound_Controls.ConnectionOriented" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Connection Oriented practical<hr /><br />
<asp:GridView ID="gvdtls" runat="server"></asp:GridView>
</div>
    </form>
</body>
</html>
```



C# Code

```
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 DataBound_Controls
{
    public partial class ConnectionOriented : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection(@"Data
Source=DESKTOPQBMSD7P\SQLEXPRESS;Initial Catalog=Test;Integrated
Security=True");             conn.Open();
            SqlCommand cmd = new SqlCommand("select * from tbllogin", conn);
            SqlDataAdapter sda = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
sda.Fill(ds);             gvdtls.DataSource
= ds;             gvdtls.DataBind();
        }
}
}
```

OUTPUT



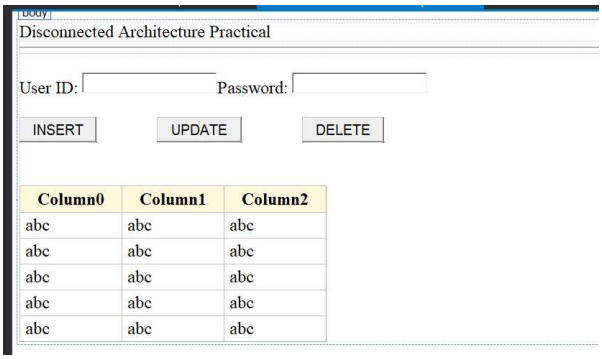
id	firstname	lastname	mailid	phone
7	Nikita	taware	nik12@gmail.com	3456778901
8	Pranita	Swami	pran56@gmail.co	9876564578
11	Namita	Kharatmol	nk2002@gmail.com	4567812309

C. AIM: ***Design a webpage to demonstrate a disconnected architecture.***

Form.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="DisconnectedArchitecture.aspx.cs"
Inherits="DataBound_Controls.DisconnectedArchitecture" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style>
        .gridview1 th{
            padding: 5px 9px 5px 9px;
background-color: cornsilk;
        }
        .gridview1 td{
            padding: 4px 100px 4px 7px;
```

```
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Disconnected Architecture Practical<hr /><br />
            User ID: <asp:TextBox runat="server" ID="txtid" ></asp:TextBox>
            Password: <asp:TextBox runat="server" ID="txtpass"></asp:TextBox><br /><br />
            <asp:Button ID="btnInsert" runat="server" Text="INSERT"
OnClick="btnInsert_Click"
/>             
            <asp:Button ID="btnUpdate" runat="server" Text="UPDATE"
OnClick="btnUpdate_Click"
/>             
            <asp:Button ID="btnDelete" runat="server" Text="DELETE"
onclick="btnDelete_Click"/><br /><br /><br />
            <asp:GridView ID="gvdtls" runat="server" CssClass="gridview1"></asp:GridView>
        </div>
    </form>
</body>
</html>
```



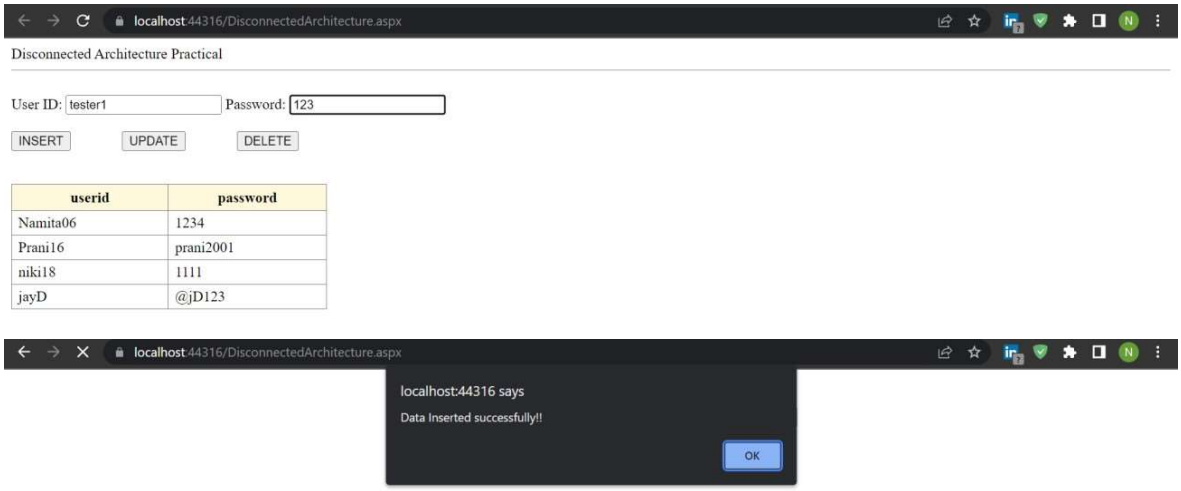
C# Code

```
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 DataBound_Controls
{
    public partial class DisconnectedArchitecture : System.Web.UI.Page
    {
        SqlConnection conn = new SqlConnection(@"Data
Source=DESKTOPQBMSD7P\SQLEXPRESS;Initial Catalog=Test;Integrated
Security=True");         protected void Page_Load(object sender, EventArgs e)
        {
            conn.Open();
            SqlCommand cmd = new SqlCommand("select * from tblCRUD1", conn);
            SqlDataAdapter sda = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
sda.Fill(ds);             gvdtls.DataSource
= ds;             gvdtls.DataBind();
        }
        protected void btnInsert_Click(object sender, EventArgs e)
        {
```

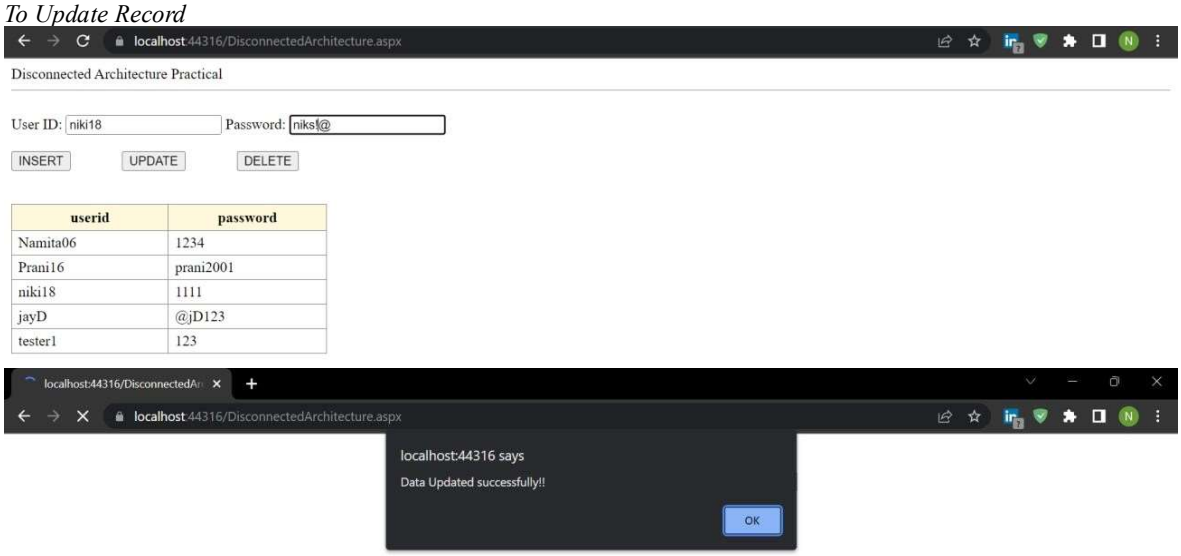
```
            DataTable dt = new DataTable();
            SqlCommand cmd = new SqlCommand("insert into tblCRUD1(userid,password)
values('" + txtid.Text + "','" + txtpass.Text + "')",conn);
SqlDataAdapter sda = new SqlDataAdapter(cmd);             sda.Fill(dt);
            Response.Write("<script> alert('Data Inserted successfully!!') </script>");

        }
        protected void btnUpdate_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("update tblCRUD1 set
password='"+txtpass.Text+"' where userid='"+txtid.Text+"'", conn);
cmd.ExecuteNonQuery();
            Response.Write("<script> alert('Data Updated successfully!!') </script>");
        }
        protected void btnDelete_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("delete from tblCRUD1 where userid='" +
txtid.Text + "'", conn);             cmd.ExecuteNonQuery();
            Response.Write("<script> alert('Data Deletion done successfully!!')
</script>");
        }
    } }
```

OUTPUT

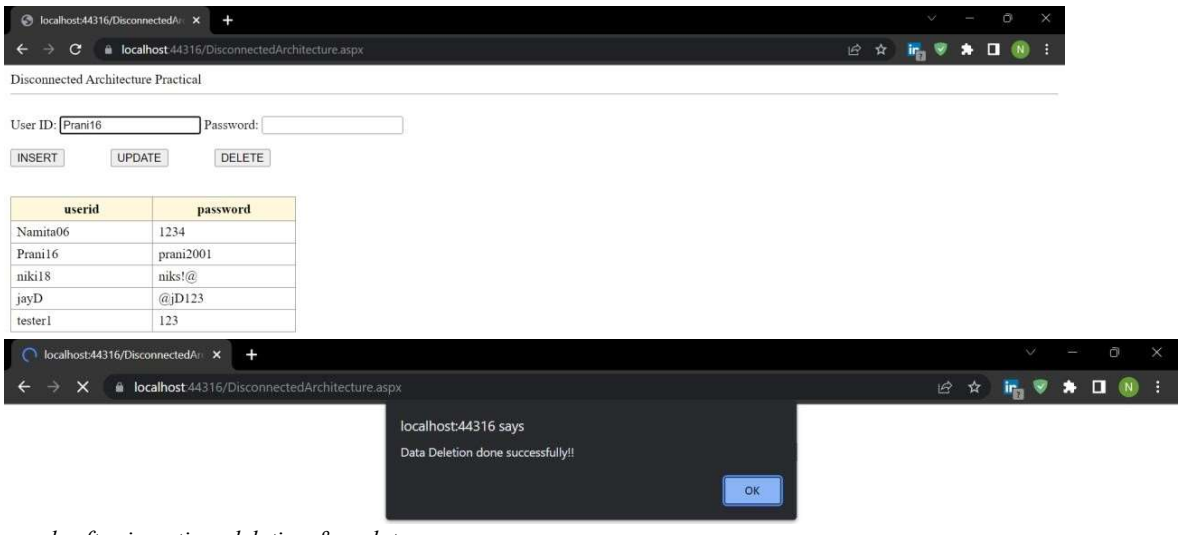
To insert record



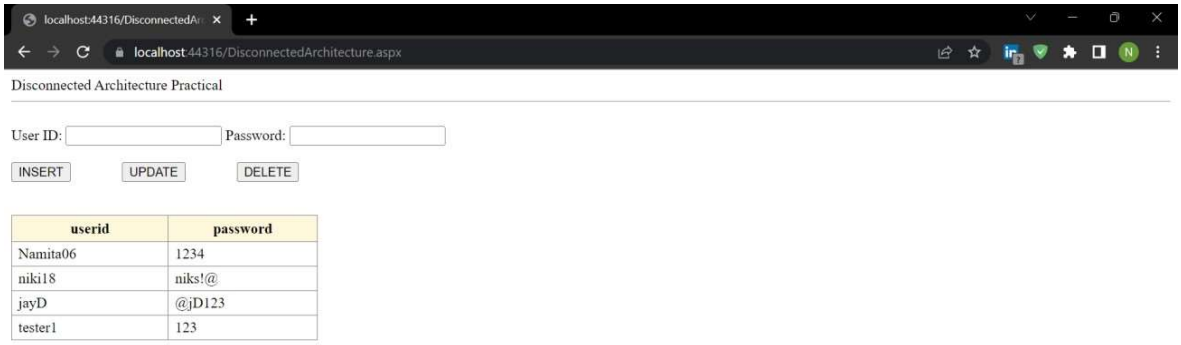
To Update Record



To Delete record



Records after insertion, deletion & update



D.AIM: ***Design a webpage to demonstrate use of stored procedure.***

Stored Procedure

(steps to create stored procedure

1. *Create database and table (table_name "tbllogin")*
2. *Expand databases option>>your_database(Test)>>Programmability>>stored procedures*
3. *Create a new stored procedure)*

Table that we used for this practical

Results | Messages

	id	firstname	lastname	mailid	phone
1	7	Nikita	taware	nik12@gmail.com	3456778901
2	8	Pranita	Swami	pran56@gmail.co	9876564578
3	11	Namita	Kharatmol	nk2002@gmail.com	4567812309

```
USE [Test]
GO
/****** Object:  StoredProcedure [dbo].[USP_CRUDoperation]    Script Date: 6/22/2023
1:34:53 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/* ============================================
-- Author:           <Author,,Namita K>
-- Create date: <Create Date,,>
-- Description:      <Description,,>
select * from tbllogin
EXEC USP_CRUDoperation 'update','','','','',''
-- ============================================*/
ALTER PROCEDURE [dbo].[USP_CRUDoperation]
@action varchar(10),
@id int,
@fname varchar(15),
@lname varchar(15),
@mail varchar(max),
@phone varchar(10)
AS
BEGIN

	if(@action = 'insert')
	begin
		insert into tbllogin(firstname,lastname,mailid,phone)
values(@fname,@lname,@mail,@phone)
	end
	else if(@action = 'update')
	begin
		if exists(select * from tbllogin where id=@id)
		begin
			update tbllogin set
firstname=@fname,lastname=@lname,mailid=@mail,phone=@phone where id=@id
		end
	end
	else if(@action = 'delete')
	begin

			delete from tbllogin where id=@id
```

```
        end
        else if(@action = 'show')
        begin
                select * from tbllogin
        end
END
```

Home.aspx

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="text-align:center">
            <h2>CRUD OPERATION</h2>
            <hr />
            <asp:TextBox runat="server" ID="txtid" placeholder="User ID"
OnTextChanged="txtid_TextChanged" cssstyle="margin-left: 313px;"></asp:TextBox>           <i
style="color:red">*id can be only use to update and delete record! Can not use for to insert
record*</i>
            <br /><br />
            <asp:TextBox runat="server" ID="txtfname" placeholder="First
Name"></asp:TextBox>
            <br /><br />
            <asp:TextBox runat="server" ID="txtlname" placeholder="Last
Name"></asp:TextBox>
            <br /><br />
            <asp:TextBox runat="server" ID="txtmail" placeholder="E-mail
ID"></asp:TextBox>
            <br /><br />
            <asp:TextBox runat="server" ID="txtcontact" placeholder="Contact
No."></asp:TextBox>
            <br /><br />
            <span>
                <asp:Button Text="Insert" ID="btninsert" runat="server"
OnClick="btninsert_Click" />
                <asp:Button Text="Delete" ID="btndelete" runat="server"
OnClick="btndelete_Click" />
                <asp:Button Text="Update" ID="btnupdate" runat="server"
OnClick="btnupdate_Click" />
                <asp:Button Text="SHOW" ID="btnshow" runat="server"
OnClick="btnshow_Click" />
            </span>
        </div>
        <div>
            <asp:GridView ID="gvdtls" runat="server">

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

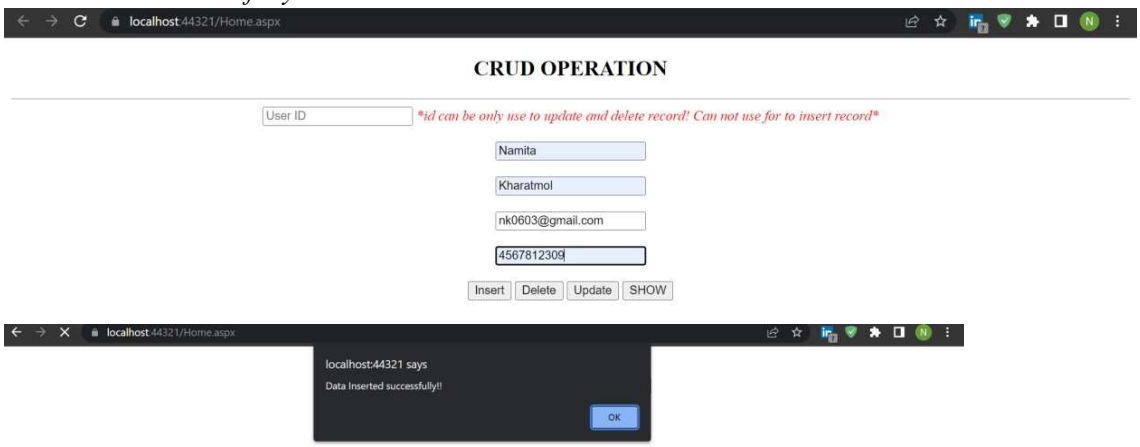
C# Code

```
using System;
using System.Collections.Generic;
using System.Configuration; using
System.Data; using
System.Data.SqlClient; using
System.Linq; using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
 namespace
CRUDoperation
{     public partial class Home :
System.Web.UI.Page     {
        SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
        protected void Page_Load(object sender, EventArgs e)
        {
            conn.Open();

}
        protected void btninsert_Click(object sender, EventArgs e)
        {
            DataTable dt = new DataTable();
            SqlCommand cmd = new SqlCommand("USP_CRUDoperation", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@id", "");
cmd.Parameters.AddWithValue("@action", "insert");
cmd.Parameters.AddWithValue("@fname", txtfname.Text);
cmd.Parameters.AddWithValue("@lname", txtlname.Text);
cmd.Parameters.AddWithValue("@mail", txtmail.Text);
cmd.Parameters.AddWithValue("@phone", txtcontact.Text);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
            Response.Write("<script> alert('Data Inserted successfully!!') </script>");
        }
        protected void btndelete_Click(object sender, EventArgs e)
        {
            DataTable dt = new DataTable();
            SqlCommand cmd = new SqlCommand("USP_CRUDoperation", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@id", txtid.Text);
cmd.Parameters.AddWithValue("@action", "delete");
cmd.Parameters.AddWithValue("@fname", "");
cmd.Parameters.AddWithValue("@lname", "");
cmd.Parameters.AddWithValue("@mail", "");
cmd.Parameters.AddWithValue("@phone", "");             SqlDataAdapter
da = new SqlDataAdapter(cmd);             da.Fill(dt);
            int rowsAffected = cmd.ExecuteNonQuery();
        }
        protected void btnupdate_Click(object sender, EventArgs e)
        {
            DataTable dt = new DataTable();
            SqlCommand cmd = new SqlCommand("USP_CRUDoperation", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@id", txtid.Text);
cmd.Parameters.AddWithValue("@action","update");
cmd.Parameters.AddWithValue("@fname", txtfname.Text);
cmd.Parameters.AddWithValue("@lname", txtlname.Text);
cmd.Parameters.AddWithValue("@mail", txtmail.Text);
cmd.Parameters.AddWithValue("@phone", txtcontact.Text);
```

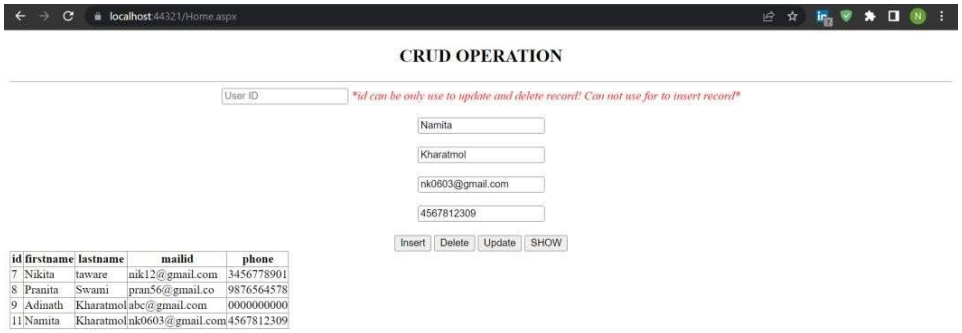
```
            SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
            int rowsAffected = cmd.ExecuteNonQuery();
}
        protected void btnshow_Click(object sender, EventArgs e)
        {
try
            {
                SqlCommand cmd = new SqlCommand("USP_CRUDoperation", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@action", "show");
cmd.Parameters.AddWithValue("@id", "");
cmd.Parameters.AddWithValue("@fname", "");
cmd.Parameters.AddWithValue("@lname", "");
cmd.Parameters.AddWithValue("@mail", "");
cmd.Parameters.AddWithValue("@phone", "");                  SqlDataAdapter sda = new
SqlDataAdapter(cmd);
                DataSet ds = new DataSet();
sda.Fill(ds);                  gvdtls.DataSource
= ds;                 gvdtls.DataBind();
            }
            catch(Exception)
            {
                Response.Write("<Script>alert('No Record found!')</script>");
            }
finally             {
                conn.Close();
            }
        }
        protected void txtid_TextChanged(object sender, EventArgs e)
        {
            btninsert.Enabled = false;
        }
    } }
```

OUTPUT

Record inserted successfully



show records



id	firstname	lastname	mailid	phone
7	Nikita	taware	nik12@gmail.com	3456778901
8	Pranita	Swami	pran56@gmail.co	9876564578
9	Adinath	Kharatmol	abc@gmail.com	0000000000
11	Namita	Kharatmol	nk0603@gmail.com	4567812309

after updating record

id	firstname	lastname	mailid	phone
7	Nikita	taware	nik12@gmail.com	3456778901
8	Pranita	Swami	pran56@gmail.co	9876564578
9	Adinath	Kharatmol	abc@gmail.com	0000000000
11	Namita	Kharatmol	nk2002@gmail.com	4567812309

To delete record

id	firstname	lastname	mailid	phone
7	Nikita	taware	nik12@gmail.com	3456778901
8	Pranita	Swami	pran56@gmail.co	9876564578
9	Adinath	Kharatmol	abc@gmail.com	0000000000
11	Namita	Kharatmol	nk2002@gmail.com	4567812309

Deleted record

id	firstname	lastname	mailid	phone
7	Nikita	taware	nik12@gmail.com	3456778901
8	Pranita	Swami	pran56@gmail.co	9876564578
11	Namita	Kharatmol	nk2002@gmail.com	4567812309

State Management Techniques

PRACTICAL 5(A)

AIM: ***Design Web Applications using Client Side Session Managements Techniques***

ClientSide.aspx

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <b>COOKIES</b><hr />
           
        <asp:TextBox runat="server" ID="txtusername" placeholder="username"></asp:TextBox>
```

```
            
        <asp:TextBox runat="server" ID="txtpass" placeholder="Password"
TextMode="Password"></asp:TextBox>
        <br />
        <br />
        <asp:Button runat="server" ID="btnCreate" Text="Create Cookie"
OnClick="btnCreate_Click" />
           
        <asp:Button runat="server" ID="btnDelete" Text="Delete Cookie"
OnClick="btnDelete_Click" />
           
        <asp:HyperLink runat="server" ID="btnShow" Text="Show Cookies"
NavigateUrl="~/StoredCookie.aspx"></asp:HyperLink>
    </form>
</body>
</html>
```

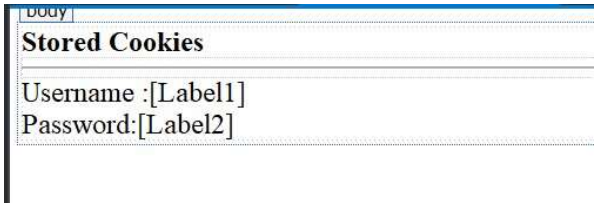
C# code

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

namespace StateManagement
{
    public partial class ClientSide : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void btnCreate_Click(object sender, EventArgs e)
        {
            HttpCookie cookie1 = new HttpCookie("Info");
cookie1["Username"] = txtusername.Text;             cookie1["Password"]
= txtpass.Text;
            Response.Cookies.Add(cookie1);
            Response.Write("<script>alert('Created a Cookie')</script>");

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

            HttpContext.Current.Session.Abandon();
            Response.Write("<script>alert('Cookie Deleted!')</script>");
        }
    }
}
```

StoredCookie.aspx

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

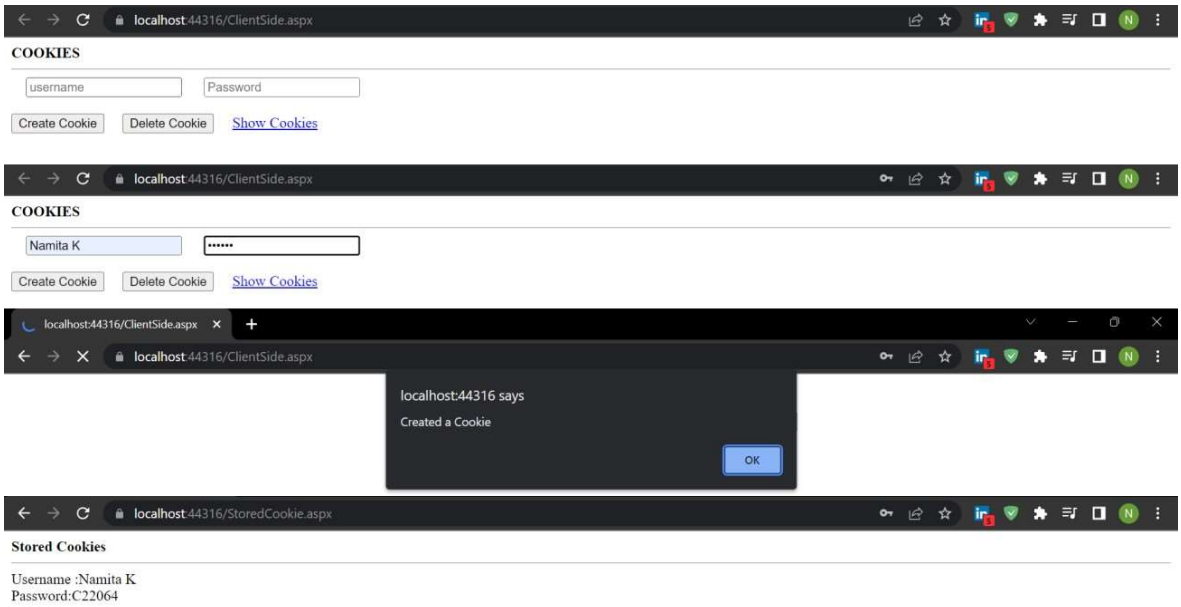
```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <b>Stored Cookies</b>
            <hr />
            Username :<asp:Label ID="Label1" runat="server"></asp:Label>
            <br />
            Password:<asp:Label ID="Label2" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```



C# code

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

namespace StateManagement
{
    public partial class StoredCookie : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie cookie1 = Request.Cookies["Info"];
if (cookie1 != null)
            {
                Label1.Text = cookie1["Username"];
                Label2.Text = cookie1["Password"];
            }

        }
    } }
```

OUTPUT



B. Design Web Applications using Server Side Session Management Techniques

Source Code:

Home.aspx:

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

<!DOCTYPE html>

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

```
        <center>
            <asp:Label ID="Label1" runat="server" Text="Enter your Name:"></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />
            <asp:Button ID="Button1" runat="server" Text="Submit" OnClick="Button1_Click" />
        </center>
      </div>
   </form>
</body>
</html>
```

Home.cs:

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

namespace sessionandapplication
{
   public partial class Home : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         Session["User"] = TextBox1.Text;
      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         Response.Redirect("WebForm1.aspx");
      }
   }
}
```

WebForm1.aspx:

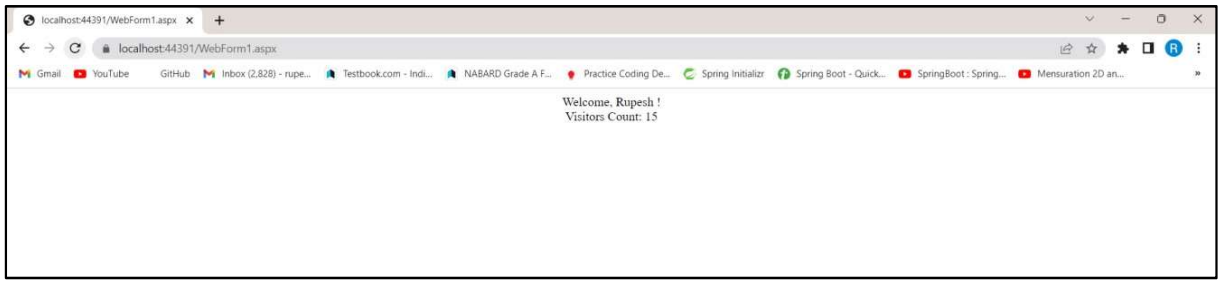
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="sessionandapplication.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div><center>
         <asp:Label ID="Label3" runat="server" Text=""></asp:Label><br />
         <asp:Label ID="Label1" runat="server" Text="Visitors Count: "></asp:Label><asp:Label
ID="Label2" runat="server" Text=""></asp:Label>

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

WebForm1.cs:

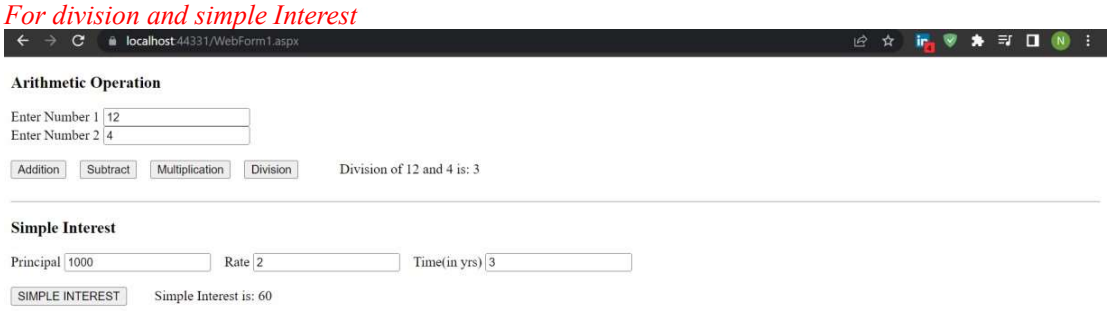
```
using System;
using System.Collections.Generic;
using System.Linq; using
System.Web; using System.Web.UI;
using System.Web.UI.WebControls;

namespace sessionandapplication
{    public partial class WebForm1 : System.Web.UI.Page
   {
```

```
    protected void Page_Load(object sender, EventArgs e)
    {
        Label3.Text = "Welcome, "+Session["User"].ToString()+" !";
        Application["NoOfVisitors"] = (int)Application["NoOfVisitors"] + 1;          Label2.Text
=Application["NoOfVisitors"].ToString();


    }
  }
}
```

Output:



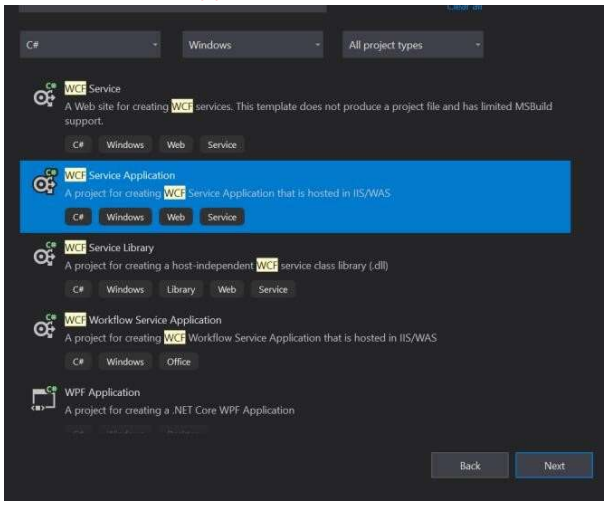
For division and simple Interest



PRACTICAL 6(B)

AIM: ***Design Web Application to produce and Consume a WCF Service***

1. Create a new WCF application



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;

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]
string GetData(int value);

        [OperationContract]
        CompositeType GetDataUsingDataContract(CompositeType composite);
        // TODO: Add your service operations here

        [OperationContract]
        double Calculator(double rate, double time, double Principal);


    }
```

```
    // 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; }
        }
    }
}
```

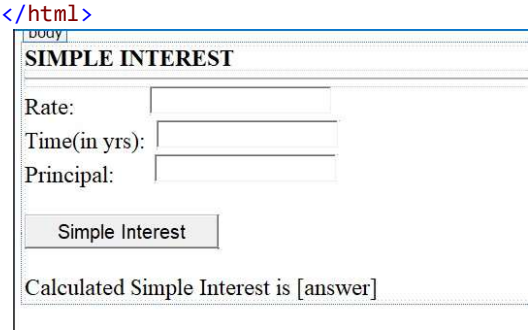
Service1.svc.cs

```
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 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 string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }
        public double Calculator(double rate, double time, double Principal)
        {
            return (rate * time * Principal) / 100;
        }

        public CompositeType GetDataUsingDataContract(CompositeType composite)
        {
            if (composite == null)
            {
                throw new ArgumentNullException("composite");
            }
            if (composite.BoolValue)
            {
                composite.StringValue += "Suffix";
            }
            return composite;
```

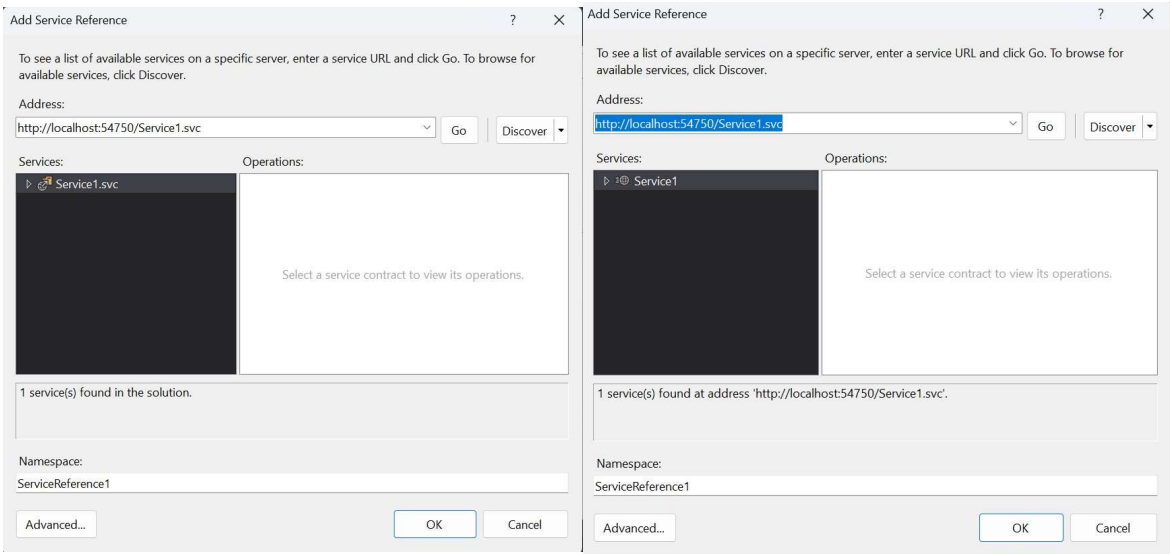
```
        }
    } }
```

Then create a webForm

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WcfService1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <b style="text-align:center">SIMPLE INTEREST</b><hr />
Rate:            
        <asp:TextBox ID="txtrate" runat="server"></asp:TextBox>
        <br />
        Time(in yrs): 
        <asp:TextBox ID="txttime" runat="server"></asp:TextBox>
        <br />
        Principal:      
        <asp:TextBox ID="txtprincipal" runat="server"></asp:TextBox>
        <br />
        <br />
        <asp:Button ID="BtnInterest" runat="server" Text="Simple Interest"
OnClick="BtnInterest_Click"  />
        <br />
        <br />
        Calculated Simple Interest is
        <asp:Label ID="answer" runat="server" Text=""></asp:Label>
    </form>
</body>
</html>
```



After this right click on project in solution explorer (add >> service references). Click on discover then go and ok


Add Service Reference
To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.
Address:
http://localhost:54750/Service1.svc
Go
Discover
Services:
Operations:
Service1.svc
Select a service contract to view its operations.
1 service(s) found in the solution.
Namespace:
ServiceReference1
Advanced...
OK
Cancel
Add Service Reference
To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.
Address:
http://localhost:54750/Service1.svc
Go
Discover
Services:
Operations:
Service1
Select a service contract to view its operations.
1 service(s) found at address 'http://localhost:54750/Service1.svc'.
Namespace:
ServiceReference1
Advanced...
OK
Cancel

VII] ASP.NET MVC

A. Design MVC based Web applications.

Source Code:
HomeController.cs:

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

namespace mvcproject.Controllers
{
   public class HomeController : Controller
   {
      public ActionResult Index()
      {
         return View();
      }

      public ActionResult About()
      {
         ViewBag.Message = "Your application description page.";

         return View();
      }

      public ActionResult Contact()
      {
         ViewBag.Message = "Your contact page.";

         return View();
      }
      public ActionResult StoryBooks()
      {
         ViewBag.Message = "Your contact page.";

         return View();
      }

      public ActionResult FantasyNovels()
      {
         ViewBag.Message = "Your contact page.";
```

```
    return View();
}
public ActionResult MarathiBooks()
{
    ViewBag.Message = "Your contact page.";
```

```
        return View();
      }
    }
}
```

_Layout.cshtml:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
  @RenderSection("Styles", required: true)
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" datatarget=".navbar-
collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        @Html.ActionLink("Books", "Index", "Home", new { area = "" }, new { @class = "navbarbrand" })
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li>@Html.ActionLink("Home", "Index", "Home")</li>
          <li>@Html.ActionLink("About", "About", "Home")</li>
          <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
        </ul>
      </div>
    </div>
  </div>
  <div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
      <p>&copy; @DateTime.Now.Year - Vinu Book Store Sanglikar.</p>      </footer>
  </div>

  @Scripts.Render("~/bundles/jquery")
  @Scripts.Render("~/bundles/bootstrap")
  @RenderSection("scripts", required: false) </body>
```

```
</html>
```

Index.cshtml:

```
@{
   ViewBag.Title = "Home Page";
}

@section Styles
{
   <style type="text/css">
      .maindiv {
         background-image:url('../../img/wallhavenBook.jpg');
height: 200px;          width: 100%;          color:azure;
         align-content:center;
      }
      .lala {
         text-align: center;          font-
size: 50px;          text-shadow:
inherit;
         text-shadow: 2px 2px #FF0000;
      }
   </style>
}

<div class="maindiv">

   <h1 class="lead  lala">
      The right book in the right hands at the right time        can
change the world.
   </h1>

</div>

<div class="row">
   <div class="col-md-4">
      <h2>Story Books</h2>
      <div>
         <a href="~/Home/StoryBooks">
            <img src="~/img/storybooks.jpg" />
         </a>

      </div>

   </div>
   <div class="col-md-4">
      <h2>Fantasy Novels</h2>
      <div>
```

```
<a href="~/Home/FantasyNovels">
  <img width="181" height="279" src="~/img/fantasy.jpg" />
```

```
        </a>

      </div>
    </div>
    <div class="col-md-4">
      <h2>Marathi Books</h2>
      <div>
        <a href="~/Home/MarathiBooks">
          <img src="~/img/rau.jpg" />
        </a>


      </div>

    </div>
</div>
```

StoryBooks.cshtml:

```
@{
   ViewBag.Title = "StoryBooks";
}
@section Styles
{

}

<h1>StoryBooks</h1>

<div class="row">
   <div class="col-md-4">

      <div>

        <a href="https://www.goodreads.com/en/book/show/13510813"><img src="~/img/storybooks.jpg"
/></a>


      </div>
      <h3>Grandma's bag of stories.</h3>



   </div>
   <div class="col-md-4">
      <div>

        <a
```

href="https://www.goodreads.com/book/show/523663.Disney_365_Bedtime_Stories?ref=nav_sb
_ss_2_11">

```
      </div>
      <h3>365 Bedtime Stories</h3>
   </div>
   <div class="col-md-4">
      <div>

         <a  href="https://www.goodreads.com/book/show/27558257-the-magic-of-the-
losttemple?ref=nav_sb_ss_1_23"> <img width="181" height="279" src="~/img/themagic.jpg" /></a>


      </div>
      <h3>The Magic of the Lost Temple</h3>

   </div>
</div>
```

FantasyNovels.cshtml:

```
@{
   ViewBag.Title = "FantasyNovels";
}
@section Styles
{

}

<h2>FantasyNovels</h2>

<div class="row">
   <div class="col-md-4">

      <div>

         <a href="https://www.goodreads.com/book/show/7235533-the-way-of-kings"><img width="181"
height="279" src="~/img/wayofking.jpg" /></a>


      </div>
      <h3>The Way of Kings.</h3>



   </div>
   <div class="col-md-4">
      <div>

         <a href="https://www.goodreads.com/book/show/186074.The_Name_of_the_Wind">
<img width="181" height="279" src="~/img/nameofwind.jpg" /></a>
```

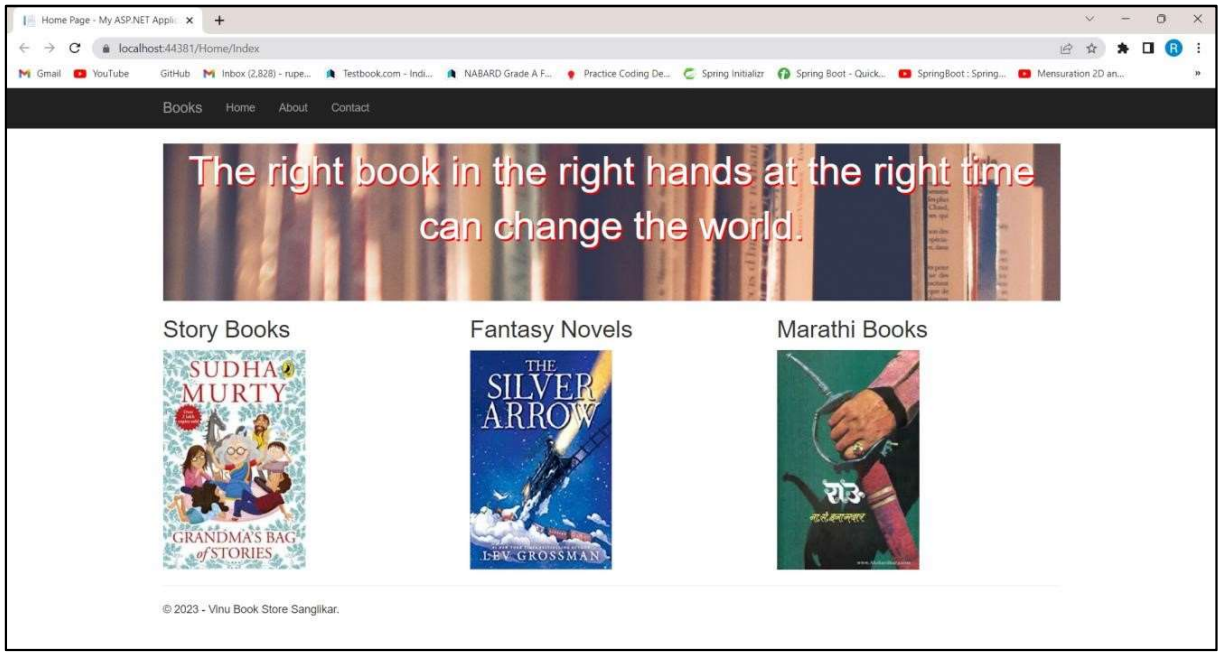
```
      </div>
      <h3>The Name of The Wind.</h3>
    </div>
    <div class="col-md-4">
      <div>

        <a href="https://www.goodreads.com/book/show/50358143-the-silver-
arrow?ref=nav_sb_ss_2_13"> <img width="181" height="279" src="~/img/fantasy.jpg" /></a>



      </div>
      <h3>The Silver Arrow.</h3>

    </div>
</div>
```

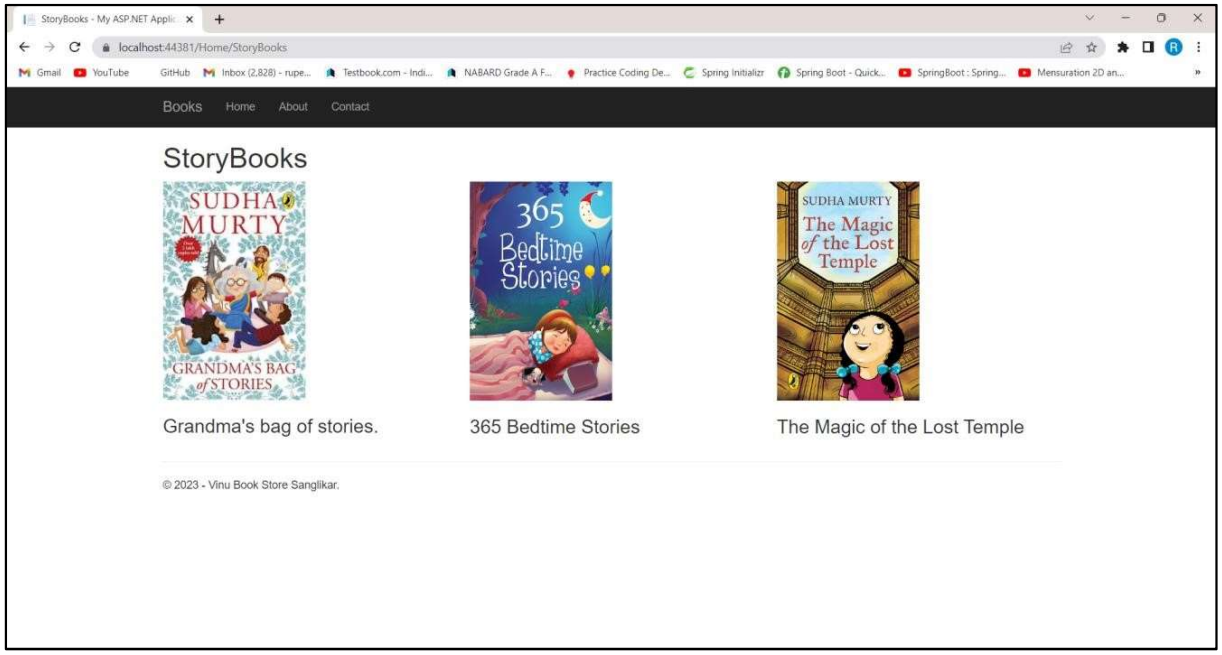
MarathiBooks.cshtml:

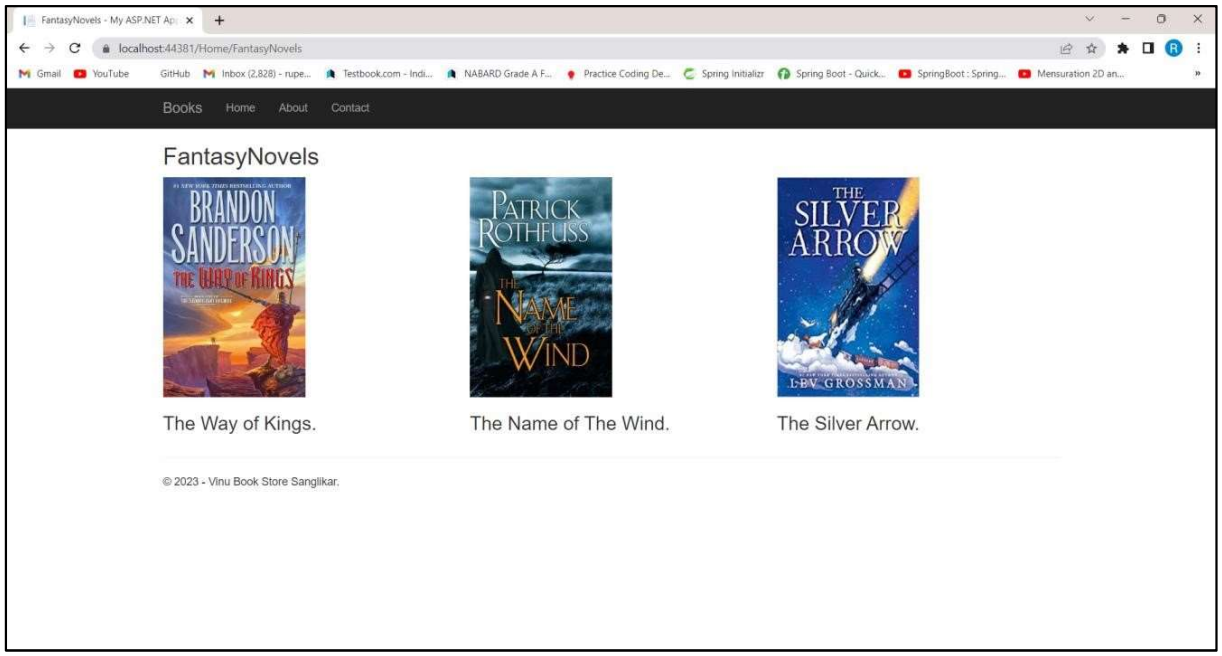
```
@{
  ViewBag.Title = "MarathiBooks";
}

<h2>MarathiBooks</h2>

<div class="row">
  <div class="col-md-4">

    <div>
      <a href="https://www.goodreads.com/book/show/6369447">
        <img width="181" height="279" src="~/img/mrutyunjay.jpg" />
      </a>
    </div>
    <h3>मृत्युंजय</h3>

  </div>
  <div class="col-md-4">

    <div>
      <a href="https://www.goodreads.com/book/show/6727757">
        <img width="181" height="279" src="~/img/shrimanyogi.jpg" />
      </a>
      <h3>श्रीमान योगी</h3>

    </div>
  </div>
  <div class="col-md-4">

    <div>
      <a href="https://www.goodreads.com/en/book/show/10370459">
```

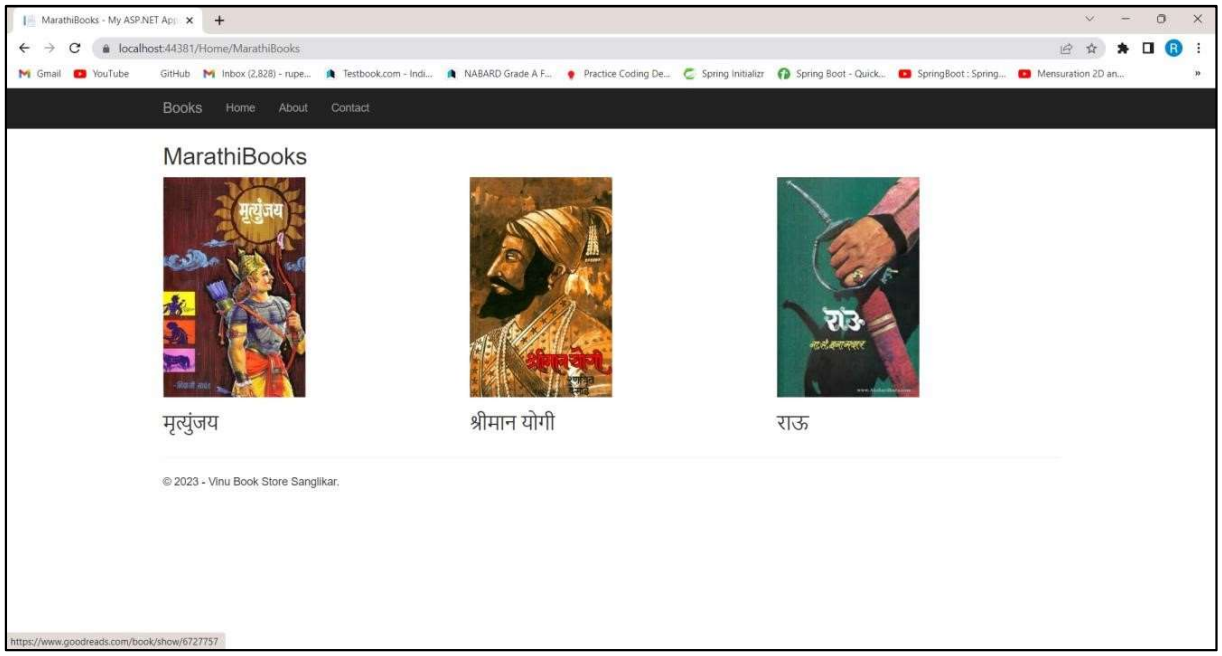
```
            <img width="181" height="279" src="~/img/rau.jpg" />
        </a>
        <h3>राऊ</h3>

    </div>

  </div>
</div>
```

Output:




StoryBooks - My ASP.NET Applic
localhost:44381/Home/StoryBooks
Books
Home
About
Contact
StoryBooks
Grandma's bag of stories.
365 Bedtime Stories
The Magic of the Lost Temple
© 2023 - Vinu Book Store Sanglikar.


FantasyNovels - My ASP.NET Ap
localhost:44381/Home/FantasyNovels
Books
Home
About
Contact
FantasyNovels
The Way of Kings.
The Name of The Wind.
The Silver Arrow.
© 2023 - Vinu Book Store Sanglikar.


MarathiBooks - My ASP.NET Ap
localhost:44381/Home/MarathiBooks
Gmail
YouTube
GitHub
Inbox (2,828) - rupe...
Testbook.com - Indi...
NABARD Grade A F...
Practice Coding De...
Spring Initializr
Spring Boot - Quick...
SpringBoot : Spring...
Mensuration 2D an...
Books
Home
About
Contact
MarathiBooks
मृत्युंजय
श्रीमान योगी
राऊ
© 2023 - Vinu Book Store Sanglikar.
https://www.goodreads.com/book/show/6727757

A. Design a webpage to display the use of LINQ.

Source Code:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="linq.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=""></asp:Label>        </div>
   </form>
</body> </html>
```

WebForm1.cs:

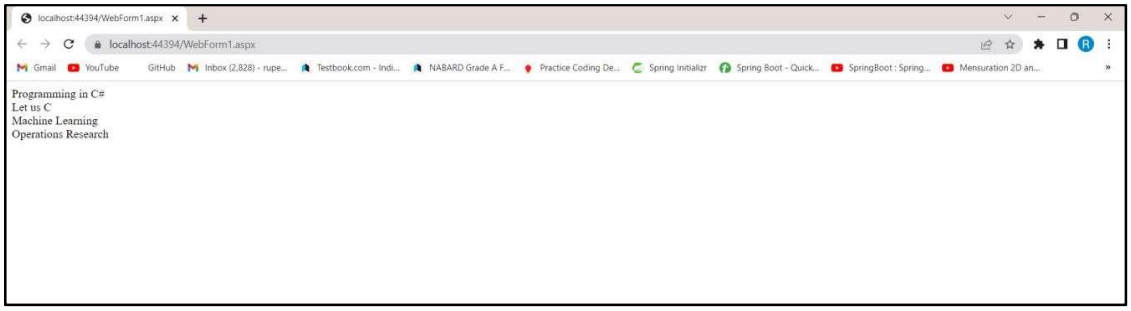
```
using System;
using System.Collections.Generic;
using System.Linq; using
System.Web; using System.Web.UI;
using System.Web.UI.WebControls;

namespace linq
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         List<Class1> books = Class1.GetBooks();            var
booktitles = from b in books select b.title;
foreach(var title in booktitles)
         {
```

```
            Label1.Text += String.Format("{0}<br/>", title);

        }


    }
  }
}
```

Class1.cs:

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

namespace linq
{
   public class Class1
   {
      public string id { get; set; }        public
string title { get; set; }        public decimal
price { get; set; }
      public DateTime dateOfRelease { get; set; }

      public static List<Class1> GetBooks()
      {
         List<Class1> list = new List<Class1>();           list.Add(new
Class1
         {
            id = "001",
            title = "Programming in C#",
price = 600.14m,
            dateOfRelease=Convert.ToDateTime("2018-05-07")
         });
         list.Add(new Class1
         {
            id = "002",
title = "Let us C",
price = 340.00m,
            dateOfRelease = Convert.ToDateTime("2010-01-20")
         });
         list.Add(new Class1
         {
            id = "003",
            title = "Machine Learning",
price = 1200m,
            dateOfRelease = Convert.ToDateTime("2018-12-14")
         });
         list.Add(new Class1
         {
            id = "004",
            title = "Operations Research",
price = 475m,
            dateOfRelease = Convert.ToDateTime("2013-05-30")
```

```
            });
            return list;
        }
    }
}
```


localhost:44394/WebForm1.aspx
localhost:44394/WebForm1.aspx
Gmail
YouTube
GitHub
Inbox (2,828) - rupe...
Testbook.com - Indi...
NABARD Grade A F...
Practice Coding De...
Spring Initializr
Spring Boot - Quick...
SpringBoot : Spring...
Mensuration 2D an...
Programming in C#
Let us C
Machine Learning
Operations Research