

Deccan Education Society's
Navinchandra Mehta Institute of
Technology and Development

C E R T I F I C A T E

This is to certify that Mr. **Dipali Patel** of M.C.A. Semester II with Roll No. **C22092** has completed practicals of **Advanced Web Technologies** under my supervision in this college during the year 2022 -2023.

CO	R1 (Attendance)	R2 (Performance during lab session)	R3 (Innovation in problem solving technique)	R4 (Mock Viva)	R5 (Variation in implementation of learnt topics on projects)
CO1					
CO2					
CO3					
CO4					

Practical-in-charge

Head of Department
MCA Department
(NMITD)

AWT INDEX		DATE	SIGN
Windows Form Application			
I	A. Design a calculator UI based applications using basic Windows forms Controls.	11/04	
C# Console			
II	A. Design Applications using Classes and Objects B. Design Applications using Inheritance and Abstract Classes	13/04	
ASP.NET			
III	A. Design a Web Application for an Organization with Registration forms and advanced controls(Validation) B. Create website using master page and theme concept.	24/04	
ADO.NET			
IV	A. Create a webpage that demonstrates the use of data bound controls of ASP.NET B. Design a webpage to demonstrate a connection oriented architecture. C. Design a webpage to demonstrate a disconnected architecture. D. Design a webpage to demonstrate use of stored procedure.	19/05	
State Management Techniques			
V	A. Design Web Applications using Client Side Session Managements Techniques B. Design Web Applications using Server Side Session Management Techniques	23/05	
Web Services and WCF			
VI	A. Design Web Application to produce and Consume a web Service B. Design Web Application to produce and Consume a WCF Service	01/06	
ASP.NET MVC			
VII	A. Design MVC based Web applications.	06/06	
LINQ			
VIII	A. Design a webpage to display the use of LINQ.	09/06	

I] Windows Form Application**A. Design a calculator UI based applications using basic Windows forms Controls.**

Source 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;

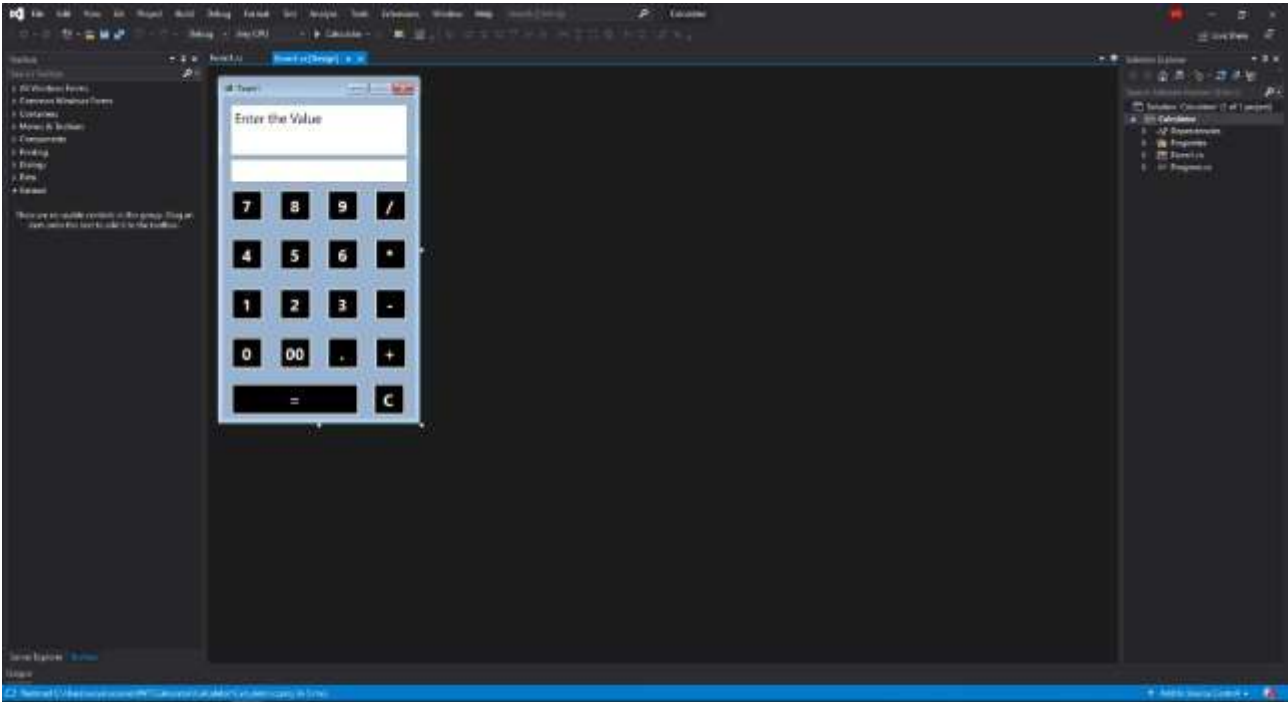
namespace Calculator
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        float num1, ans;
        int count;

        private void key1_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 1;
        }
        private void key2_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 2;
        }
        private void key3_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 3;
        }
        private void key4_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 4;
        }
        private void key5_Click(object sender, EventArgs e)
        {
            label1.Text = null;
            textBox1.Text = textBox1.Text + 5;
        }
    }
}
```

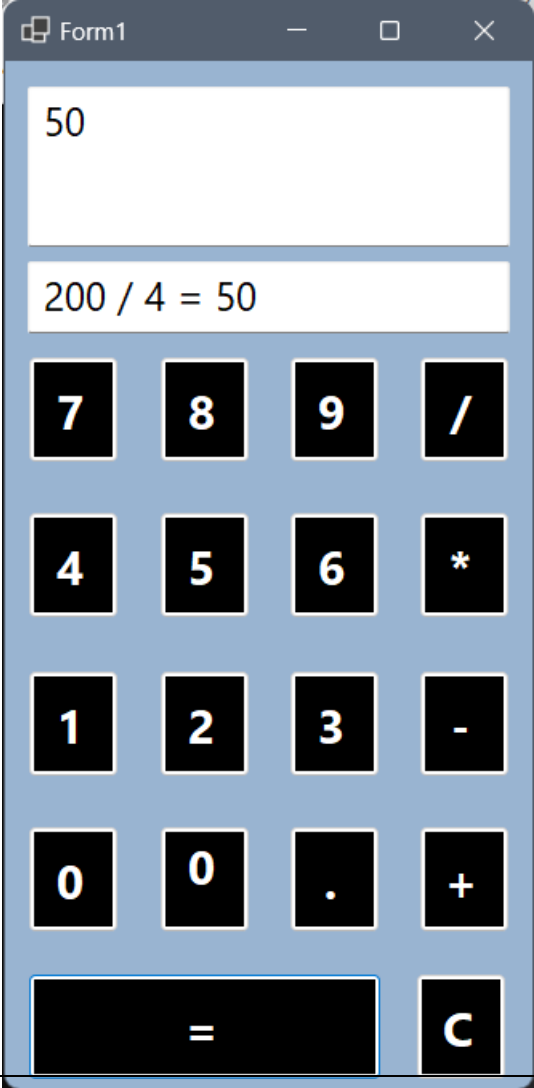
```
private void key6_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 6;
}
private void key7_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 7;
}
private void key8_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 8;
}
private void key9_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 9;
}
private void key00_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 0+0;
}
private void key0_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + 0;
}
private void keyDot_Click(object sender, EventArgs e)
{
    label1.Text = null;
    textBox1.Text = textBox1.Text + ".";
}
private void keyDivide_Click(object sender, EventArgs e)
{
    num1 = float.Parse(textBox1.Text);
    textBox2.Text = num1 + " / ";
    textBox1.Clear();
    textBox1.Focus();
    count = 4;
}
private void keyMul_Click(object sender, EventArgs e)
{
    num1 = float.Parse(textBox1.Text);
    textBox2.Text = num1 + " * ";
    textBox1.Clear();
    textBox1.Focus();
    count = 3;
}
private void keySub_Click(object sender, EventArgs e)
```

```
{
    num1 = float.Parse(textBox1.Text);
    textBox2.Text = num1 + " - ";
    textBox1.Clear();
    textBox1.Focus();
    count = 2;
}
private void keyAdd_Click(object sender, EventArgs e)
{
    num1 = float.Parse(textBox1.Text);
    textBox2.Text = num1 + " + ";
    textBox1.Clear();
    textBox1.Focus();
    count = 1;
}
private void keyClear_Click(object sender, EventArgs e)
{
    label1.Text = "Enter your input";
    textBox1.Text = null;
    textBox2.Text = null;
}
private void keyEqual_Click(object sender, EventArgs e)
{
    compute(count);
}
public void compute(int count)
{
    switch (count)
    {
        case 1:
            ans = num1 + float.Parse(textBox1.Text);
            textBox2.Text = textBox2.Text + textBox1.Text + " = " + ans;
            textBox1.Text = ans.ToString();
            break;
        case 2:
            ans = num1 - float.Parse(textBox1.Text);
            textBox2.Text = textBox2.Text + textBox1.Text + " = " + ans;
            textBox1.Text = ans.ToString();
            break;
        case 3:
            ans = num1 * float.Parse(textBox1.Text);
            textBox2.Text = textBox2.Text + textBox1.Text + " = " + ans;
            textBox1.Text = ans.ToString();
            break;
        case 4:
            ans = num1 / float.Parse(textBox1.Text);
            textBox2.Text = textBox2.Text + textBox1.Text + " = " + ans;
            textBox1.Text = ans.ToString();
            break;
        default:
            break;
    }
}
```

}
}
}
}



Output:



II] C# Console**A. Design Applications using Classes and Objects**Source code:

```
using System;

namespace ClassandObject
{
    public class Student
    {

        // Instance Variables
        String name;
        int rollNo;
        int std;
        String div;

        // Constructor Declaration of Class
        public Student(String name, int rollNo,
            int std, String div)
        {
            this.name = name;
            this.rollNo = rollNo;
            this.std = std;
            this.div = div;
        }

        // Property 1
        public String GetName()
        {
            return name;
        }

        // Property 2
        public int GetRollNo()
        {
            return rollNo;
        }

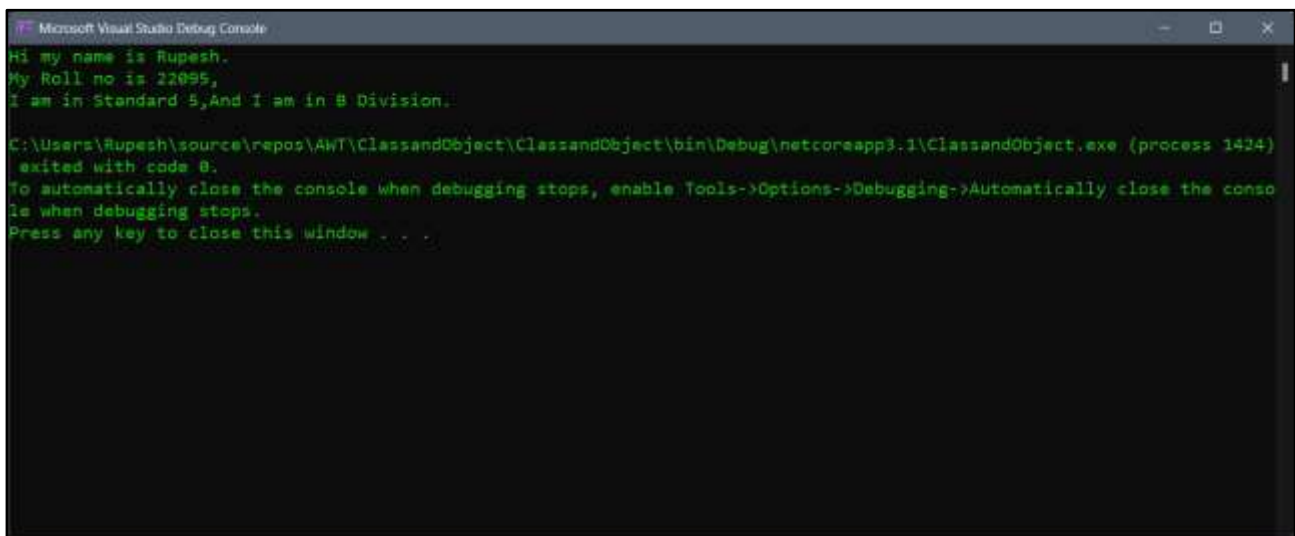
        // Property 3
        public int GetStd()
        {
            return std;
        }

        // Property 4
        public String GetDiv()
        {
            return div;
        }
    }
}
```

```
// Method 1
public String ToString()
{
    return ("Hi my name is " + this.GetName()
        + ".\nMy Roll no is " + this.GetRollNo()
        + ", \nI am in Standard " + this.GetStd() + ",And I am in " + this.GetDiv() + " Division.");
}

// Main Method
public static void Main(String[] args)
{
    // Creating object
    Student rupesh = new Student("Rupesh", 22095, 5, "B");
    Console.WriteLine(rupesh.ToString());
}
}
```

Output:



```
Microsoft Visual Studio Debug Console
Hi my name is Rupesh.
My Roll no is 22095,
I am in Standard 5,And I am in B Division.

C:\Users\Rupesh\source\repos\AWT\ClassandObject\ClassandObject\bin\Debug\netcoreapp3.1\ClassandObject.exe (process 1424)
exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```


B. Design Applications using Inheritance and Abstract ClassesSource code:

AbstractClass.cs:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Q_3_Inheritance
{
    public abstract class AbstractClass
    {
        public abstract string occupation(string occuapation);
        public string location(string loc)
        {
            return loc;
        }
    }
}
```

ChildClass.cs:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Q_3_Inheritance
{
    class ChildClass : AbstractClass, Interface1
    {
        public string name(string name)
        {
            Console.WriteLine("My name is " + name);
            return name;
        }
        public int info(int age, int height)
        {
            Console.WriteLine("I am " + age + " year old .");
            Console.WriteLine("My height is " + height + " cm .");
            return 1;
        }
        public override string occupation(string occuapation)
        {
            return occuapation;
        }
    }
    class main
    {
        public static void Main(string[] args)
```

```
{  
  
    Interface1 i;  
    i = new ChildClass();  
    i.name("Rupesh .");  
    i.info(21, 168);  
  
    AbstractClass cc = new ChildClass();  
    Console.WriteLine(cc.occupation("I am student."));  
    Console.WriteLine(cc.location("At dadar"));  
  
    Console.ReadKey();  
  
}  
}  
}
```

ChildClass.cs

```
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Q_3_Inheritance  
{  
    class ChildClass : AbstractClass, Interface1  
    {  
        public string name(string name)  
        {  
            Console.WriteLine("My name is " + name);  
            return name;  
        }  
        public int info(int age, int height)  
        {  
            Console.WriteLine("I am " + age + " year old .");  
            Console.WriteLine("My height is " + height + " cm .");  
            return 1;  
        }  
        public override string occupation(string occupation)  
        {  
            return occupation;  
        }  
    }  
    class main  
    {  
        public static void Main(string[] args)  
        {  
  
            Interface1 i;  
            i = new ChildClass();
```

```
i.name("Rupesh .");  
i.info(21, 168);
```

```
AbstractClass cc = new ChildClass();  
Console.WriteLine(cc.occupation("I am student."));  
Console.WriteLine(cc.location("At dadar"));
```

```
Console.ReadKey();
```

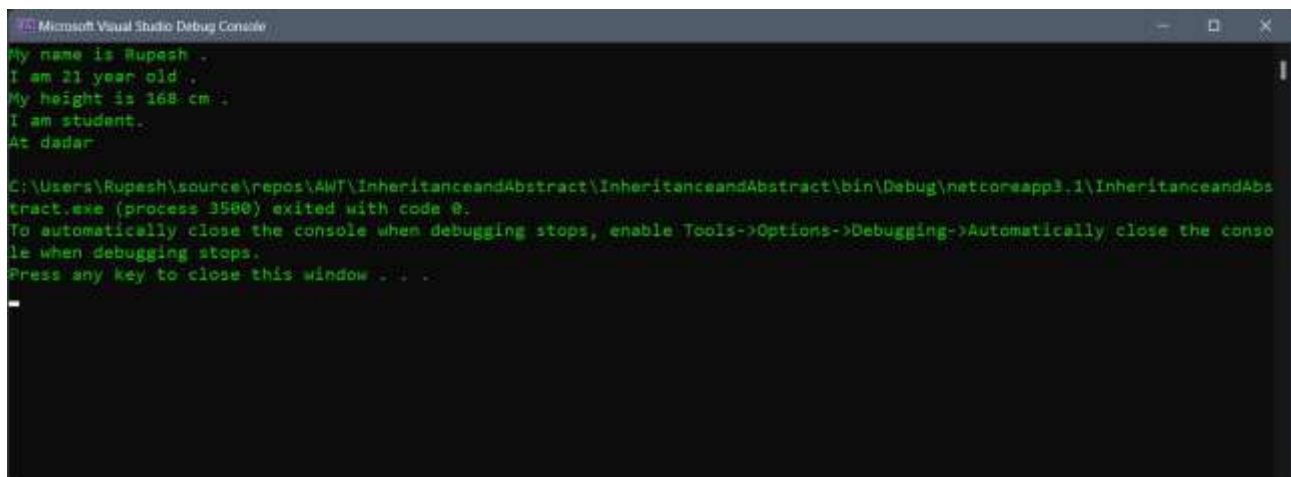
```
    }  
}  
}
```

Interface1.cs

```
using System;  
using System.Collections.Generic;  
using System.Text;
```

```
namespace Q_3_Inheritance  
{  
    interface Interface1  
    {  
        string name(string name);  
        int info(int age, int height);  
    }  
}
```

Output:



```
Microsoft Visual Studio Debug Console  
My name is Rupesh .  
I am 21 year old .  
My height is 168 cm .  
I am student .  
At dadar  
C:\Users\Rupesh\source\repos\AWT\InheritanceandAbstract\InheritanceandAbstract\bin\Debug\netcoreapp3.1\InheritanceandAbstract.exe (process 3590) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .  
-
```

III] ASP.NET**A. Design a Web Application for an Organization with Registration forms and advanced controls(Validation).**Source code:

RegistrationPage.aspx:

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

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
  <title></title>
```

```
</head>
```

```
<body>
```

```
  <form id="form1" runat="server">
```

```
    <div>
```

```
      <table>
```

```
        <tr>
```

```
          <td>
```

```
            <asp:Label ID="Label1" runat="server" Text="Username"></asp:Label></td>
```

```
          <td>
```

```
            <asp:TextBox ID="username" runat="server"></asp:TextBox></td>
```

```
          <td>
```

```
            <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
```

```
ControlToValidate="username" runat="server" ErrorMessage="Enter
```

```
username"></asp:RequiredFieldValidator>
```

```
          </td>
```

```
        </tr>
```

```
        <tr>
```

```
          <td>
```

```
            <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label></td>
```

```
          <td>
```

```
            <asp:TextBox ID="password" TextMode="Password"
```

```
runat="server"></asp:TextBox></td>
```

```
          <td>
```

```
            <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
```

```
ControlToValidate="password" runat="server" ErrorMessage="Enter
```

```
password"></asp:RequiredFieldValidator>
```

```
          </td>
```

```
        </tr>
```

```
        <tr>
```

```
          <td>
```

```
            <asp:Label ID="Label3" runat="server" Text="Confirm Password"></asp:Label></td>
```

```
          <td>
```

```
            <asp:TextBox ID="confirm" TextMode="Password"
```

```
runat="server"></asp:TextBox></td>
```

```
          <td>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator3"
ControlToValidate="confirm" runat="server"
ErrorMessage="*required"></asp:RequiredFieldValidator><asp:CompareValidator
ID="CompareValidator1" runat="server" ControlToValidate="confirm"
ControlToCompare="password" ErrorMessage="Please enter correct
password"></asp:CompareValidator>
</td>
</tr>
<tr>
<td>
<asp:Label ID="Label4" runat="server" Text="Email"></asp:Label></td>
<td>
<asp:TextBox ID="email" TextMode="Email" runat="server"></asp:TextBox></td>
<td>
<asp:RequiredFieldValidator ID="RequiredFieldValidator4"
ControlToValidate="email" runat="server"
ErrorMessage="*required"></asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td>
<asp:Label ID="Label5" runat="server" Text="Gender"></asp:Label></td>
<td>
<asp:RadioButton ID="RadioButton1" GroupName="gender" Text="Male"
runat="server" Checked="True" /><asp:RadioButton ID="RadioButton2" GroupName="gender"
Checked="false" Text="Female" runat="server" /></td>
</tr>
<tr>
<td>
<asp:Label ID="Label6" runat="server" Text="Courses"></asp:Label></td>
<td>
<asp:CheckBox ID="CheckBox1" runat="server" Text="C++" />
<asp:CheckBox ID="CheckBox2" runat="server" Text="Java"/>
<asp:CheckBox ID="CheckBox3" runat="server" Text="Python"/></td>
</tr>
<tr>
<td></td>
<td>
<asp:Button ID="Button1" runat="server" Text="Register"
OnClick="Button1_Click"/>
</td>
</tr>
</table>
<asp:Label ID="message" runat="server" ></asp:Label>
<table>
<tr>
<td>
<asp:Label ID="ShowUsernameLabel" runat="server" ></asp:Label></td>
<td>
<asp:Label ID="ShowUsername" runat="server"></asp:Label></td>
</tr>
<tr>
```

```
<td>
    <asp:Label ID="ShowEmailLabel" runat="server" ></asp:Label></td>
<td>
    <asp:Label ID="ShowEmail" runat="server" ></asp:Label></td>
<tr>
    <td>
        <asp:Label ID="ShowGenderLabel" runat="server" ></asp:Label></td>
    <td>
        <asp:Label ID="ShowGender" runat="server" ></asp:Label></td>
</tr>
<tr>
    <td>
        <asp:Label ID="ShowCourseLabel" runat="server" ></asp:Label></td>
    <td>
        <asp:Label ID="ShowCourse" runat="server" ></asp:Label></td>
</tr>
</tr>
</table>
</div>
</form>
</body>
</html>
```

RegistrationPage.cs

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

namespace RegistrationValidation
{
    public partial class RegistrationPage : System.Web.UI.Page
    {
        protected System.Web.UI.HtmlControls.HtmlInputFile File1;

        protected System.Web.UI.HtmlControls.HtmlInputButton Submit1;

        protected void Page_Load(object sender, EventArgs e)
        {

        }

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

```
message.Text = "Hello " + username.Text + " ! ";
```

```
message.Text = message.Text +  
" <br/> You have successfully Registered with the following details.";
```

```
ShowUsername.Text = username.Text;
```

```
ShowEmail.Text = email.Text;
```

```
if (RadioButton1.Checked)
```

```
{
```

```
    ShowGender.Text = RadioButton1.Text;
```

```
}
```

```
else ShowGender.Text = RadioButton2.Text;
```

```
var courses = "";
```

```
if (CheckBox1.Checked)
```

```
{
```

```
    courses = CheckBox1.Text + " ";
```

```
}
```

```
if (CheckBox2.Checked)
```

```
{
```

```
    courses += CheckBox2.Text + " ";
```

```
}
```

```
if (CheckBox3.Checked)
```

```
{
```

```
    courses += CheckBox3.Text;
```

```
}
```

```
ShowCourse.Text = courses;
```

```
ShowUsernameLabel.Text = "User Name";
```

```
ShowEmailLabel.Text = "Email ID";
```

```
ShowGenderLabel.Text = "Gender";
```

```
ShowCourseLabel.Text = "Courses";

username.Text = "";

email.Text = "";

RadioButton1.Checked = false;

RadioButton2.Checked = false;

CheckBox1.Checked = false;

CheckBox2.Checked = false;

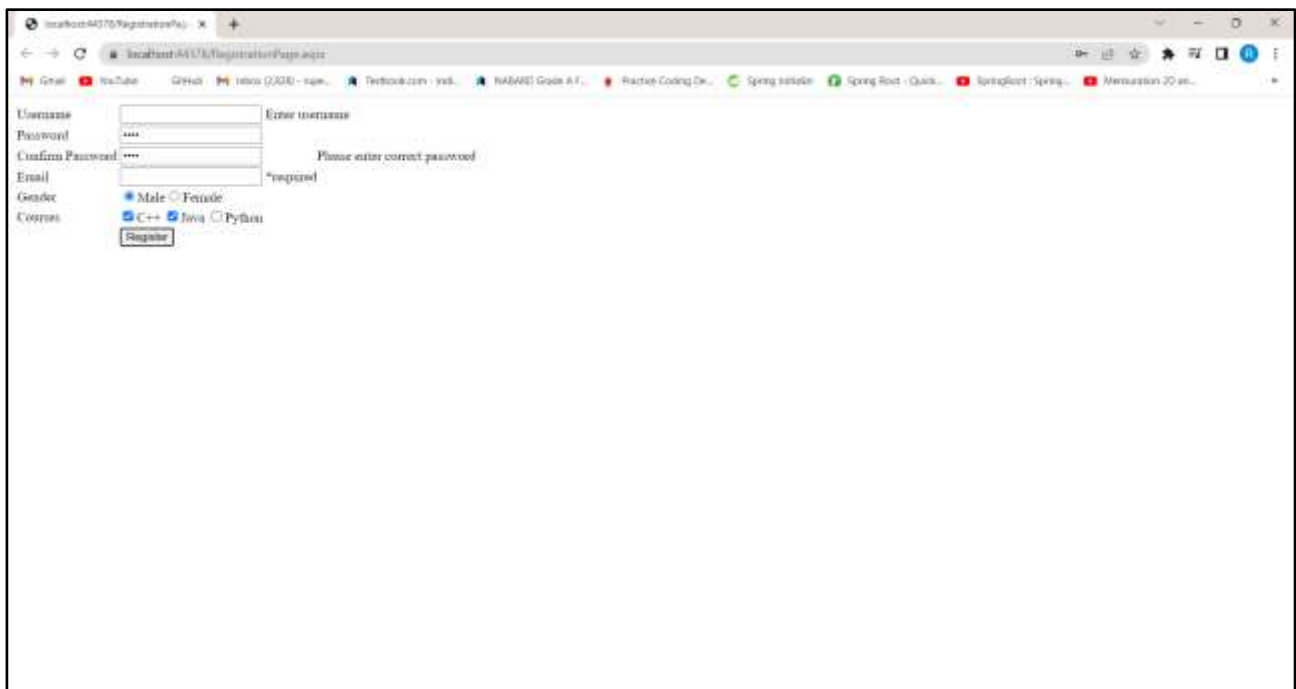
CheckBox3.Checked = false;

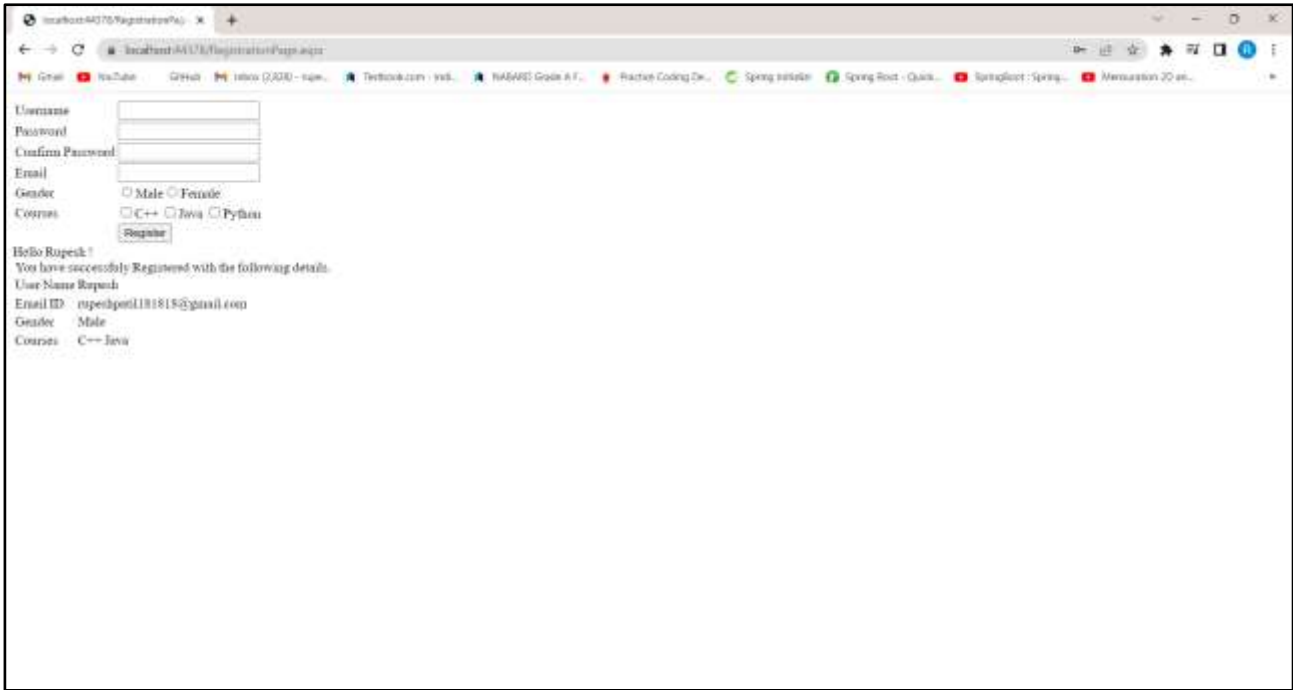
}

}

}
```

Output:





B. Create website using master page and theme concept.Source code:

Site1.Master:

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

<!DOCTYPE html>

<html>
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
    <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.4/dist/jquery.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <!-- Grey with black text -->
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
    <ul class="navbar-nav">
        <li class="nav-item ">
            <a class="nav-link" href="WebForm1.aspx">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="Sport.aspx">Sports</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="Entertainment.aspx">Entertainment</a>
        </li>
        <li class="nav-item">
            <a class="nav-link disabled " href="#">Politics</a>
        </li>
    </ul>
</nav>

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

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="PracMaster.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <style type="text/css">
        .auto-style1 {
            width: 1623px;
            height: 785px;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    
</asp:Content>
```

Entertainment.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="Entertainment.aspx.cs" Inherits="PracMaster.Entertainment" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        
        <h3>Controversial Adipurush dialogues altered after public pressure, 'jalegi tere baap ki'
changed to 'jalegi bhi teri Lanka'</h3>
    </center>
    <p style="text-align:left">
```

Days after its debut, the objectionable dialogues in the film Adipurush have been altered. The new version of the movie is now being screened in theatres, as per sources. A section of the film's core audience objected to certain lines of dialogue in the film, based on the Hindu epic Ramayana. It was suggested that some lines, penned by lyricist-writer Manoj Muntashir, had a pedestrian quality that didn't respect the sanctity of the source material.

The Central Board of Film Certification, which had previously given Adipurush a U certificate, approved of the changes on June 19, according to a report on Filminformation.com. The official CBFC website also lists Adipurush as having been re-certified on June 19, after being initially cleared on June 12. The film was released on June 16. Here's a list of the original lines, and the new versions:

1. Tu andar kaise ghusa... tu jaanta bhi hai kaun hoon main, has been replaced by, Tum andar kaise ghuse... tum jaante bhi ho kaun hoon main.

2. Kapda tere baap ka... toh jalegi bhi tere baap ki, has been modified to, Kapda teri Lanka ka... toh jalegi bhi teri Lanka.

3. Jo hamari behno... unki Lanka laga denge, has been replaced by, Jo hamari behno... unki Lanka mein aag laga denge.

4. Mere ek sapole ne tumhare iss sheshnaag ko lamba kar diya... bhara pada hai, has been changed to, Mere ek sapole ne tumhare iss sheshnaag ko samapt kar diya... bhara pada hai.

Backlash against the dialogues was observed on the first day of release itself, with many critics and audience members noting their jarring quality. A day after release, Muntashir was asked about this in an interview with the Republic, and he claimed to have grown up listening to versions of the Ramayana told in colloquial language. A day after that, he told Aaj Tak that Adipurush isn't an adaptation of the Ramayana after all, but is 'inspired' by it. Later that same day, he announced that the objectionable lines will be altered to honour audience sentiment.

Earlier this week, Muntashir said that the studio's decision to change these lines is 'brave', and they did it despite knowing what a 'logistical nightmare' it would present. Adipurush is directed by Om Raut, who said in the Republic interview that he is more concerned about audience response than reviews. But there seem to be few takers for the movie, which has experienced historical drops at the box office after delivering big numbers over the opening weekend. After five days, the film's worldwide gross collection stands at Rs 395 crore. It was reportedly produced on a budget of at least Rs 500 crore.

</p>

</asp:Content>

Sport.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="Sport.aspx.cs" Inherits="PracMaster.Sport" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
```

</asp:Content>

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
```

<center>

<p>

</p>

<h3>Ashes 2023: Australians seal edge-of-the-seat thriller to surge ahead in series</h3>

<p>

The ball was veering onto his body. Pat Cummins shuffled across and paused and opened the face of his bat to guide it to square third man. The ball was trickling to the fence, the outfield made heavy by the rain. Zak Crawley was galloping from his fielding station to stop the ball. But in the middle, Cummins and Nathan Lyon were running for their lives, one eye on the ball, one ear on their partner's call, the mind but a scrambled blur of thoughts.

</p>

<p>

rope, wrapping up a two-wicket win in the crimson twilight of Edgbaston, to conclude a match of epic stature, one that was subject to wicked twists and turns, till the match could no longer twist and turn. Cummins, who scored the most precious 44 runs of his career and stitched a match-winning 55-run stand with Lyon, threw his helmet into the distance; Lyon climbed on his captain. Cummins ran in circles, flexed his biceps and smiled as broad as he could. In the glassed dressing room, their teammates went berserk, before they ran onto the field, past scattered and devastated England counterparts. Some stopped and commiserated, but when the euphoria and devastation sank in, they would reflect on a truly ageless classic.

But how did it get this far! When Joe Root clung onto a return catch from Alex Carey — he had spilled a couple earlier — the match seemed to drift beyond Australia's grasp. A few overs ago, Ben Stokes had consumed Usman Khawaja to lift England's hopes of a straightforward victory.

</p>

<p>

When Lyon united with Cummins, their team was still 54 runs away from victory. England celebrated the fall of Carey as though they had won the match. They surely were on match point. Or was it Ashes Edgbaston Part 2? The narrative unfolded in an eerily similar way.

But Cummins and Lyon are made of steelier and sterner stuff to give up the game. Adversity has often channelled the best out of them. Cummins, whose batting is often understated, decided to counterpunch. He has the requisites of a competent lower-order batsman, a robust technique, ability to swing those arms down the ground, and beyond all these physical attributes, a serenity, a perspective about him. He thundered Root for a pair of sixes in the space of three balls. These two shots tore through the morale of England's cricketers.

The long overdue new ball was flung to Stuart Broad. But Cummins flayed him in front of square. In Broad's next over, Lyon laced a glorious four down the ground. As the pair whittled down the target, England's nerves clutched and tightened. A mid-pitch discussion broke after every ball. Cummins and Lyon would just nod and flick a thumbs up. Sometimes with a grin, often with an assuringly sober face. England would try everything they could to force another twist — short-ball therapy, yorker barrage. But Cummins and Lyon weathered the storm phlegmatically.

Scenario-plotting

The what ifs would hurt and haunt England. What if Root had clung onto Cummins' catch when he was on seven? What if Stokes had clung onto a Lyon skier when he was on one? Did England take the new ball a bit too late? Should they have taken it at all? There was ample purchase for Root's off-breaks. The hard new ball travelled faster to the rope. Besides, there was little assistance for the seamers with the new ball. And the biggest debate of them all — did England blunder when declaring on 393/8 on the first evening? Was it daring or arrogance? Did they recklessly throw their wickets away in the second innings? Did they make a mistake in not hurling the new ball to James Anderson? He had looked gingery throughout the game, but Anderson is Anderson.

Therein lies the enduring charm of an epic Test match — the what ifs and what nots that would stamp themselves to the mind of the audience, lurk as invisible notes in the scorecards. It's a Test that would be told and retold several times in different parts of the world.

But such twists and turns did not seem to arrive at half time.

The teams parted for tea after two hours of slow see-sawing, the game still not tipping to either side. Both sides could perceive the scenario as a glass half-full or half-empty. England heckled only a pair of wickets, one of them night- watchman Scott Boland for a frustrating 20 and the other a flaky Travis Head. But England, sticking to diligent lengths and sometimes left-of-left-field field settings, had a noose on the scoring for most of the time. In 29 overs, Australia mustered only 76 runs, a crawl by England's new dizzying standards. But the pace of scoring hardly bothered the visitors. Not for them the keeping-up-with-the-Joneses temptation. They would grit, graft and grind the old-fashioned way, a no-nonsense, no-frills approach, that fidgeted rather than thrilled.

Typically, Khawaja embodied their approach, batting as if in an invisible astronaut's helmet, pressure- and temptation-proof. He hit just one four, courtesy a Moeen Ali full toss. Then, Australia themselves fetched just five fours in a session that seemed an antithesis to England's brave new leitmotif. Singles and twos, not the stolen types but straightforward ones, kept the scoreboard ticking along. Apart from a stray instance of Head backing away to Stuart Broad and trying to sledgehammer him through cover, Australia's batsmen exhibited hardly any portent of counterpunching. Head's 24-ball 16 was the only phase in the game where excitement brimmed. He slashed, slapped and swished before Moeen Ali snaffled him with a ripping off-break.

Ali, inhibited by his battered fingers, would soon lose his control and was replaced with Root's part-time all-sorts. Ali was sorely missed on a deck that conspired with the spinners, against a left-hand-heavy batting firm.

Endgame

There was precious little assistance for the seamers on a surface slower than Bengaluru traffic, apart from the cloud cover upon resumption, which soon disappeared. Once the conventional tricks proved ineffective, England resorted to cross-seam deliveries and back-of-the hand knuckle balls, but without reward, as Khawaja and Cameron Green knuckled down.

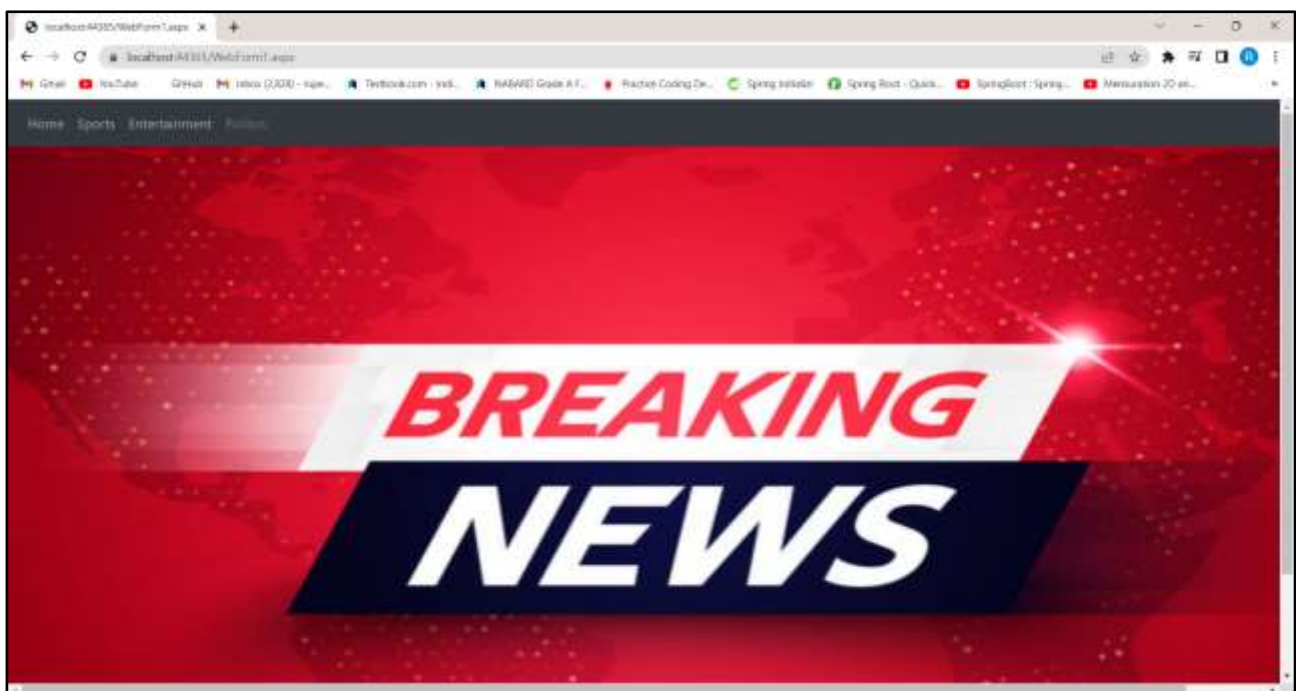
The kiss of life for England arrived in fifth over of the final session, when Green hacked an Ollie Robinson in-ducker to the base of his off-stump, thus ending a 49-run alliance that seemed to steer Australia closer to the shores of victory. An unusual stroke of indecisiveness seized Green, who was hitherto clear-headed. Maybe, he was readying to wear a bouncer after Robinson had deputed a fielder at deep square-leg. He hung on his back-foot, shuffled across, and looked to dab to third-man, but was suffocated for room. The moment was ripe to ratchet up the aggression. Robinson, as he had all through the day, was all fire and brimstone. He did not strain the speed-gun but in-your-face machismo rattled Australia for the first time in the day. At the other end, Root spat one past Khawaja's defensive thrust and maintained the pressure. Just three runs were accrued in the next four overs. Tension brimmed over. Jonny Bairstow amped up his chirp behind the stumps. The close-in field cordon swooped in like vultures over their prey. A momentum shift was perceptible. After adding 17 more, Australia lost Khawaja and then Carey. But in the end, the purpose of the collapse was to make Cummins and Lyon heroes. Timeless Ashes heroes.

</p>

</center>

</asp:Content>

Output:



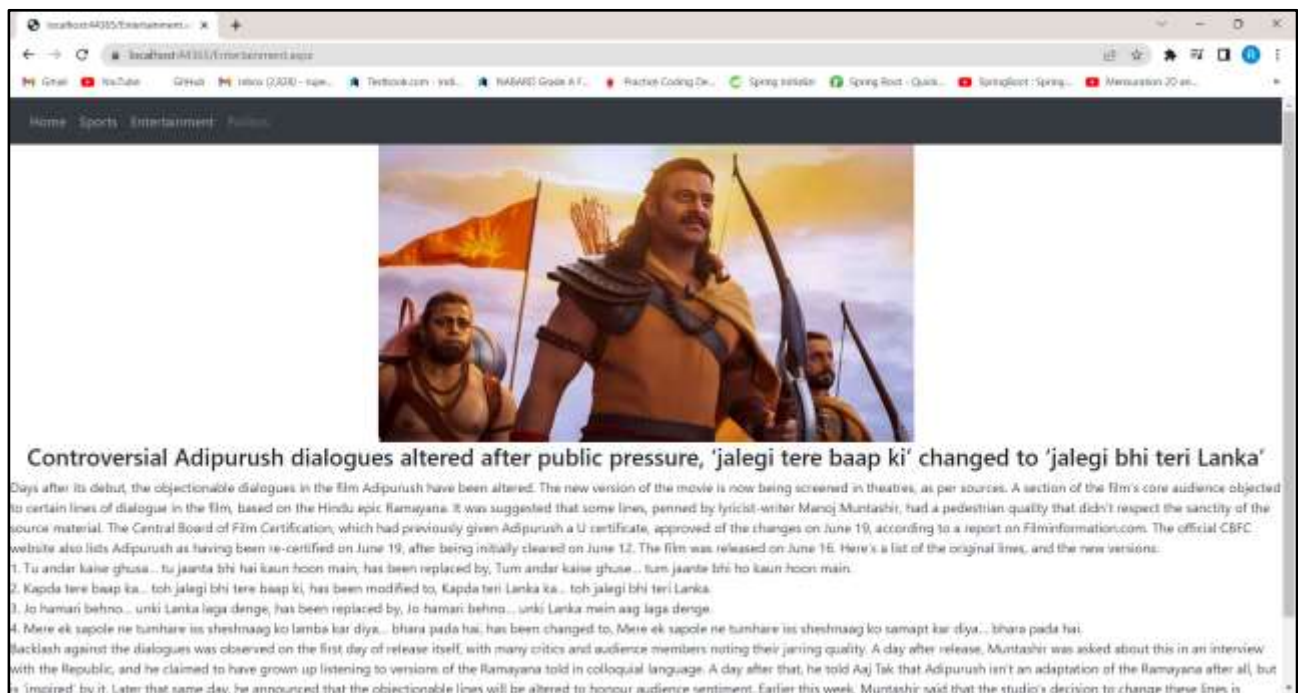


Ashes 2023: Australians seal edge-of-the-seat thriller to surge ahead in series

The ball was veering onto his body. Pat Cummins shuffled across and paused and opened the face of his bat to guide it to square third man. The ball was trickling to the fence; the outfield made heavy by the rain. Zak Crawley was galloping from his fielding station to stop the ball. But in the middle, Cummins and Nathan Lyon were running for their lives, one eye on the ball, one ear on their partner's call, the mind but a scrambled blur of thoughts.

rope, wrapping up a two-wicket win in the crimson twilight of Edgbaston, to conclude a match of epic stature, one that was subject to wicked twists and turns, till the match could no longer twist and turn. Cummins, who scored the most precious 44 runs of his career and stitched a match-winning 55-run stand with Lyon, threw his helmet into the distance; Lyon climbed on his captain. Cummins ran in circles, flexed his biceps and smiled as broad as he could. In the glassed dressing room, their teammates went berserk, before they ran onto the field, past scattered and devastated England counterparts. Some stopped and commiserated, but when the euphoria and devastation sank in, they would reflect on a truly ageless classic. But how did it get this far? When Joe Root clung onto a return catch from Ales Carey — he had spilled a couple earlier — the match seemed to drift beyond Australia's grasp. A few overs ago, Ben Stokes had consumed Uthman Khawaja to left England's hopes of a straightforward victory.

When Lyon united with Cummins, their team was still 54 runs away from victory. England celebrated the fall of Carey as though they had won the match. They surely were on match point. Or was it Ashes Edgbaston?



Controversial Adipurush dialogues altered after public pressure, 'jalegi tere baap ki' changed to 'jalegi bhi teri Lanka'

Days after its debut, the objectionable dialogues in the film Adipurush have been altered. The new version of the movie is now being screened in theatres, as per sources. A section of the film's core audience objected to certain lines of dialogue in the film, based on the Hindu epic Ramayana. It was suggested that some lines, penned by lyricist-writer Manoj Muntashir, had a pedestrian quality that didn't respect the sanctity of the source material. The Central Board of Film Certification, which had previously given Adipurush a U certificate, approved of the changes on June 19, according to a report on Filminformation.com. The official CBFC website also lists Adipurush as having been re-certified on June 19, after being initially cleared on June 12. The film was released on June 16. Here's a list of the original lines, and the new versions:

1. Tu andar kaise ghuse... tu jaanta bhi hai kaun hoon main, has been replaced by: Tum andar kaise ghuse... tum jaante bhi ho kaun hoon main.
2. Kapda tere baap ka... toh jalegi bhi tere baap ki, has been modified to, Kapda teri Lanka ka... toh jalegi bhi teri Lanka.
3. Jo hamari behno... unki Lanka laga denge, has been replaced by, Jo hamari behno... unki Lanka mein aag laga denge.
4. Mere ek sapne ne tumhare iss sheshaag ko lamba kar diya... bhara pada hai, has been changed to, Mere ek sapne ne tumhare iss sheshaag ko samapt kar diya... bhara pada hai.

Backlash against the dialogues was observed on the first day of release itself, with many critics and audience members noting their jarring quality. A day after release, Muntashir was asked about this in an interview with the Republic, and he claimed to have grown up listening to versions of the Ramayana told in colloquial language. A day after that, he told Aaj Tak that Adipurush isn't an adaptation of the Ramayana after all, but is 'inspired' by it. Later that same day, he announced that the objectionable lines will be altered to honour audience sentiment. Earlier this week, Muntashir said that the studio's decision to change these lines is

IV] ADO.NET**A. Create a webpage that demonstrates the use of data bound controls of ASP.NET**

Source code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="DataBound.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:GridView ID="GridView1" runat="server" AllowPaging="True"
AutoGenerateColumns="False" BackColor="#DEBA84" BorderColor="#DEBA84"
BorderStyle="None" BorderWidth="1px" CellPadding="3" CellSpacing="2" DataKeyNames="Id"
DataSourceID="SqlDataSource1">
                <Columns>
                    <asp:BoundField DataField="Id" HeaderText="Id" ReadOnly="True"
SortExpression="Id" />
                    <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
                    <asp:BoundField DataField="City" HeaderText="City" SortExpression="City" />
                </Columns>
                <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
                <HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
                <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
                <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
                <SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
                <SortedAscendingCellStyle BackColor="#FFF1D4" />
                <SortedAscendingHeaderStyle BackColor="#B95C30" />
                <SortedDescendingCellStyle BackColor="#F1E5CE" />
                <SortedDescendingHeaderStyle BackColor="#93451F" />
            </asp:GridView>
            <asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
DataSourceID="SqlDataSource1">
                <AlternatingItemTemplate>
                    <li style="background-color: #FAFAD2;color: #284775;">Id:
                        <asp:Label ID="IdLabel" runat="server" Text='<%= Eval("Id") %>' />
                        <br />
                        Name:
                        <asp:Label ID="NameLabel" runat="server" Text='<%= Eval("Name") %>' />
                        <br />
                        City:
                        <asp:Label ID="CityLabel" runat="server" Text='<%= Eval("City") %>' />
                        <br />
                    </li>
                </AlternatingItemTemplate>
            </asp:ListView>
        </div>
    </form>
</body>
</html>
```



```

</AlternatingItemTemplate>
<EditItemTemplate>
  <li style="background-color: #FFCC66;color: #000080;">Id:
    <asp:Label ID="IdLabel1" runat="server" Text='<%=# Eval("Id") %>' />
    <br />
    Name:
    <asp:TextBox ID="NameTextBox" runat="server" Text='<%=# Bind("Name") %>' />
    <br />
    City:
    <asp:TextBox ID="CityTextBox" runat="server" Text='<%=# Bind("City") %>' />
    <br />
    <asp:Button ID="UpdateButton" runat="server" CommandName="Update"
Text="Update" />
    <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
Text="Cancel" />
  </li>
</EditItemTemplate>
<EmptyDataTemplate>
  No data was returned.
</EmptyDataTemplate>
<InsertItemTemplate>
  <li style="">Id:
    <asp:TextBox ID="IdTextBox" runat="server" Text='<%=# Bind("Id") %>' />
    <br />Name:
    <asp:TextBox ID="NameTextBox" runat="server" Text='<%=# Bind("Name") %>' />
    <br />City:
    <asp:TextBox ID="CityTextBox" runat="server" Text='<%=# Bind("City") %>' />
    <br />
    <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
Text="Insert" />
    <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
Text="Clear" />
  </li>
</InsertItemTemplate>
<ItemSeparatorTemplate>
<br />
</ItemSeparatorTemplate>
<ItemTemplate>
  <li style="background-color: #FFFBD6;color: #333333;">Id:
    <asp:Label ID="IdLabel" runat="server" Text='<%=# Eval("Id") %>' />
    <br />
    Name:
    <asp:Label ID="NameLabel" runat="server" Text='<%=# Eval("Name") %>' />
    <br />
    City:
    <asp:Label ID="CityLabel" runat="server" Text='<%=# Eval("City") %>' />
    <br />
  </li>
</ItemTemplate>
<LayoutTemplate>
  <ul id="itemPlaceholderContainer" runat="server" style="font-family: Verdana, Arial,
Helvetica, sans-serif;">
    <li runat="server" id="itemPlaceholder" />
  </ul>
  <div style="text-align: center;background-color: #FFCC66;font-family: Verdana,
Arial, Helvetica, sans-serif;color: #333333;">

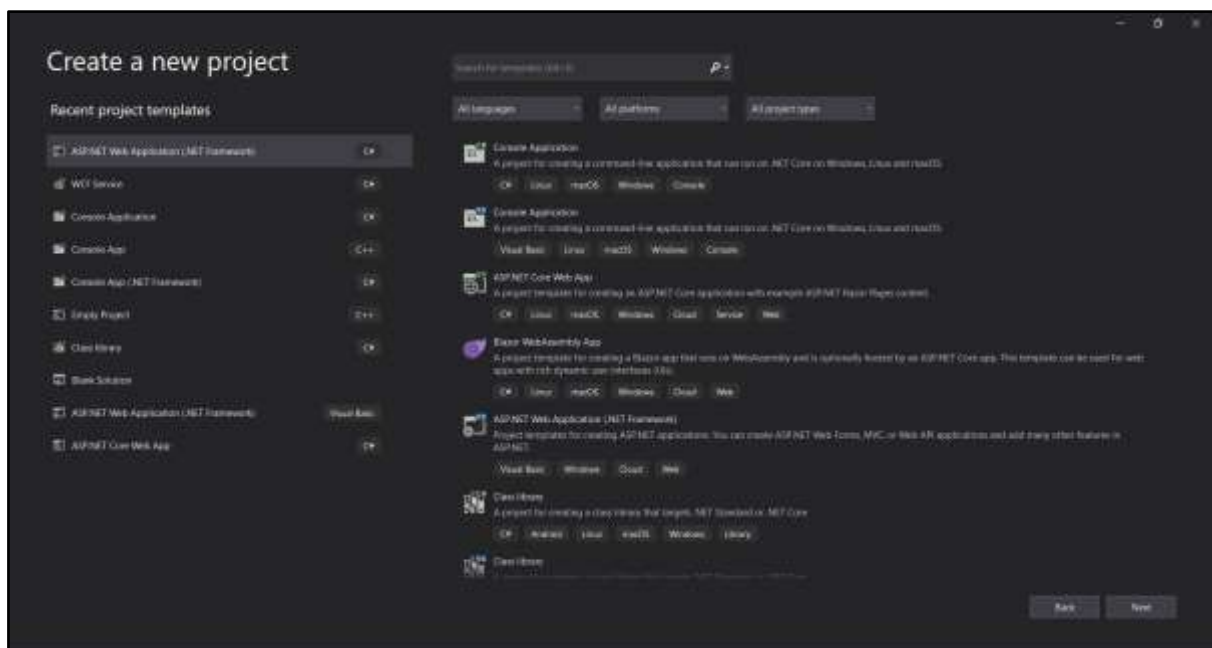
```

```

</div>
</LayoutTemplate>
<SelectedItemTemplate>
  <li style="background-color: #FFCC66;font-weight: bold;color: #000080;">Id:
    <asp:Label ID="IdLabel" runat="server" Text='<%= Eval("Id") %>' />
    <br />
    Name:
    <asp:Label ID="NameLabel" runat="server" Text='<%= Eval("Name") %>' />
    <br />
    City:
    <asp:Label ID="CityLabel" runat="server" Text='<%= Eval("City") %>' />
    <br />
  </li>
</SelectedItemTemplate>
</asp:ListView>
<br />
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%= $ ConnectionStrings:rupeshConnectionString %>"
SelectCommand="SELECT * FROM [employee]"></asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

Screenshots of steps:



Configure your new project

ASP.NET Web Application (.NET Framework) OK WebAPI Check Web

Project name

Location


Solution name


☐ Place solution and project in the same directory


Framework


Back Create


Create a new ASP.NET Web Application

**Empty**
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

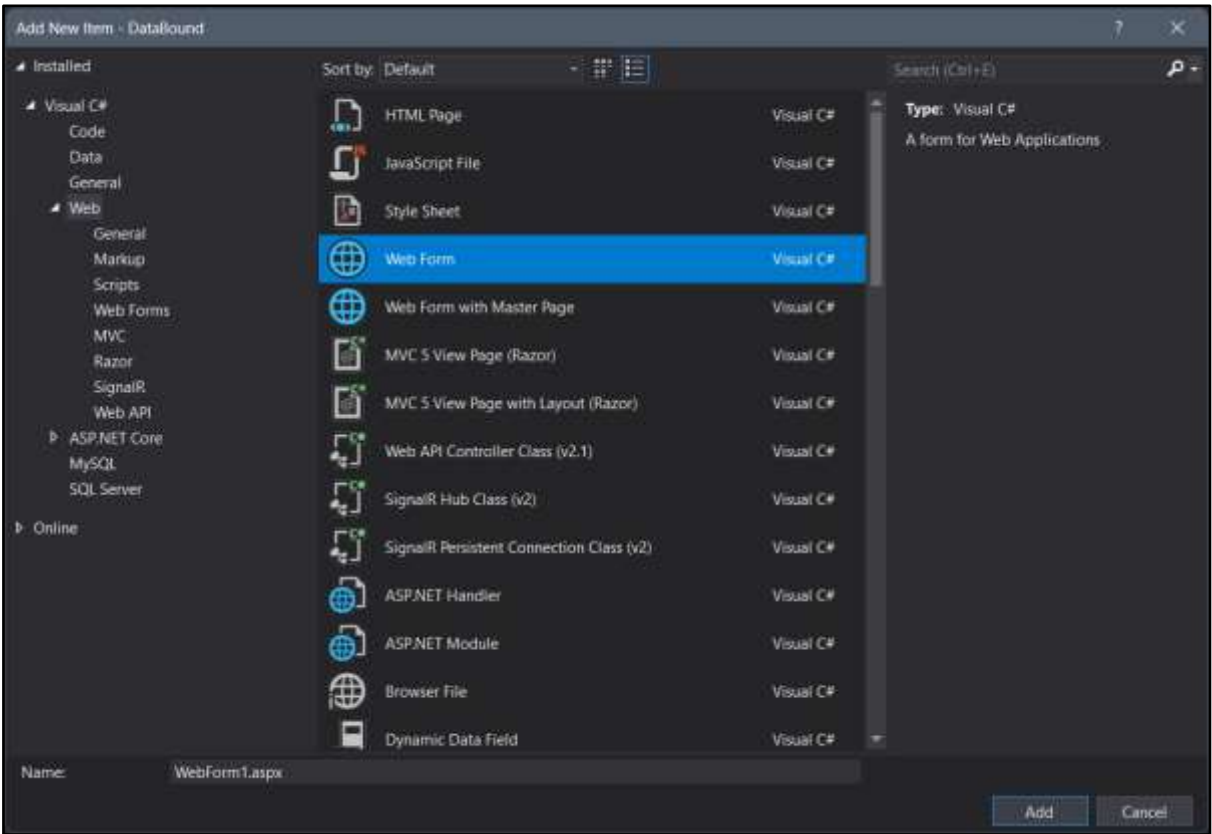
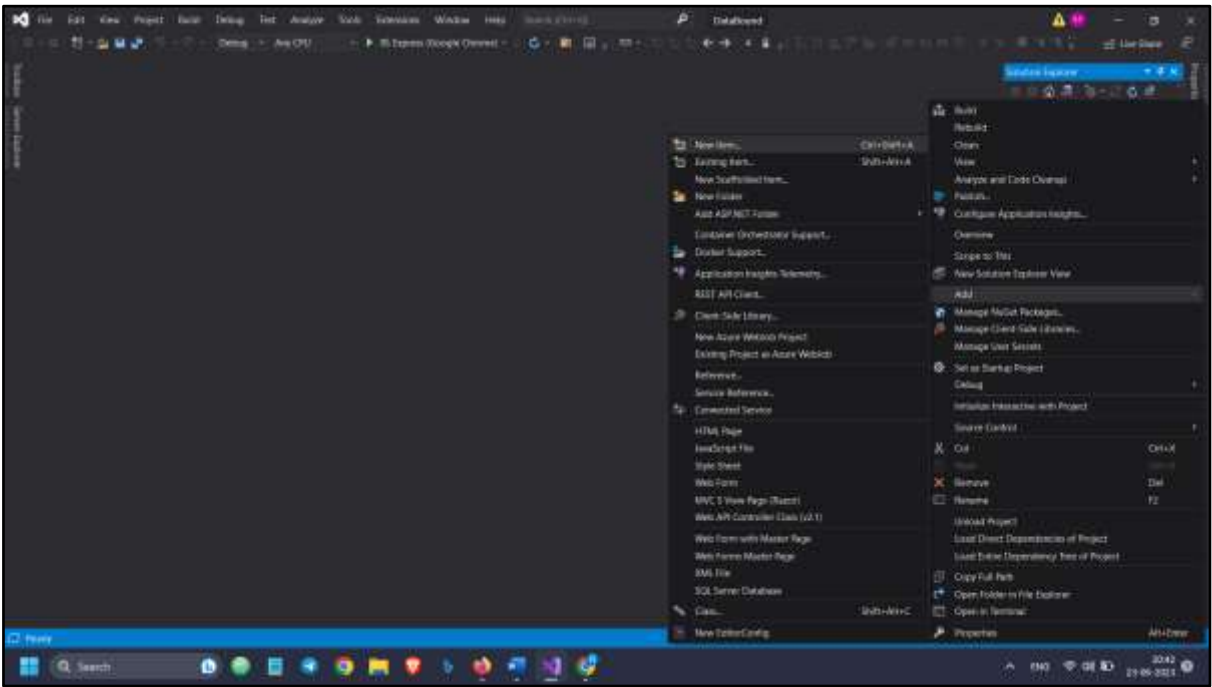
**Single Page Application**
A project template for creating rich client-side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

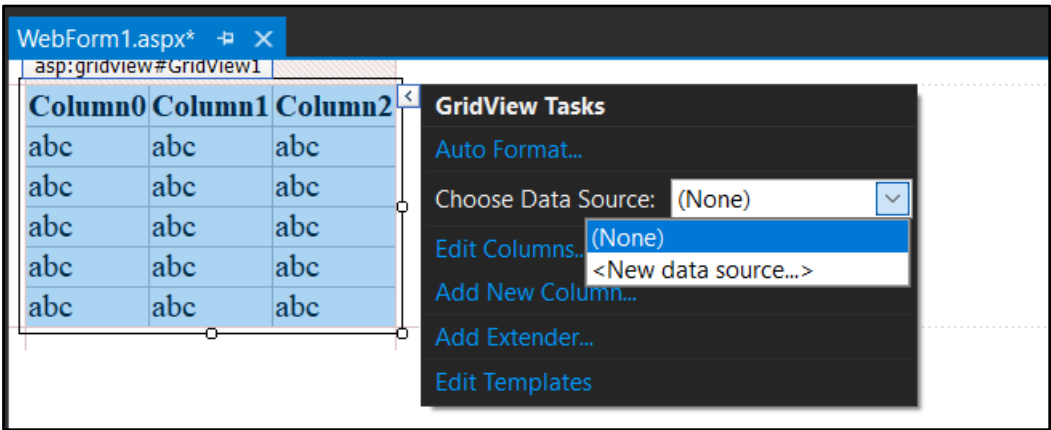
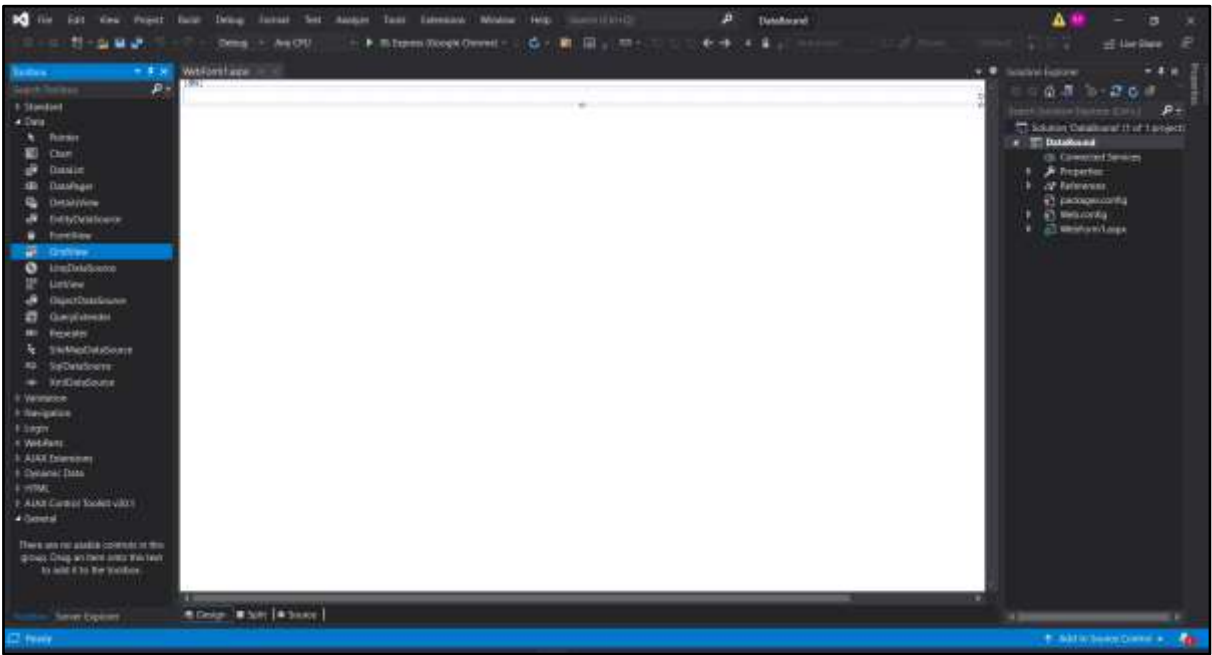
Authentication
No Authentication
[Change](#)

Add folders & core references
☐ Web Forms
☐ MVC
☐ Web API


Advanced
☒ Configure for HTTPS
☐ Docker support
(Requires Docker Desktop)
☐ Also create a project for unit tests
[Download tests](#)

Back Create











Data Source Configuration Wizard

 **Choose a Data Source Type**

Where will the application get data from?

 **Database**  Entity  LINQ  Object  Site Map  XML File


Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.

Specify an ID for the data source:

SqlDataSource1

OK Cancel

Configure Data Source - SqlDataSource1

 **Choose Your Data Connection**

Which data connection should your application use to connect to the database?

rupesh\sqlexpress.rupesh.dbo New Connection...

☒ Connection string

Data Source=RUPESH\SQLEXPRESS;Initial Catalog=rupesh;Integrated Security=True

< Previous Next > Finish Cancel

Configure Data Source - SqlDataSource1

Configure the Select Statement

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name: employee

Columns:

☒ *
☐ Id
☐ Name
☐ City

☐ Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
SELECT * FROM [employee]

< Previous Next > Finish Cancel

Configure Data Source - SqlDataSource1

Save the Connection String to the Application Configuration File

Storing connection strings in the application configuration file simplifies maintenance and deployment. To save the connection string in application configuration file, enter a name in the text box and then click Next. If you choose not to do this, the connection string is saved in the page as a property of the data source control.


Do you want to save the connection in the application configuration file?

☒ Yes, save this connection as:

rupeshConnectionString

< Previous Next > Finish Cancel

Configure Data Source - SqlDataSource1



Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

Id	Name	City
1	Rupesh	Vikhroli
2	Siddhesh	Dadar
3	Vinayak	Ghatkopar
4	Saurabh	Ghatkopar
5	Sumit	Dadar

Test Query

SELECT statement:

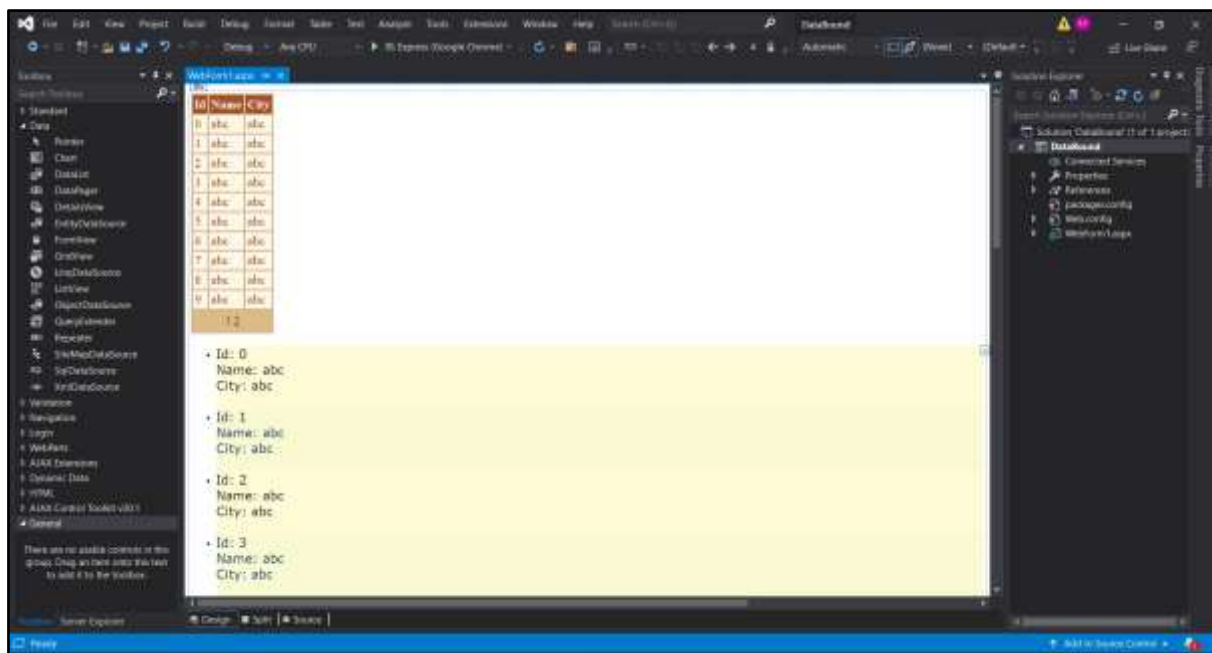
SELECT * FROM [employee]

< Previous

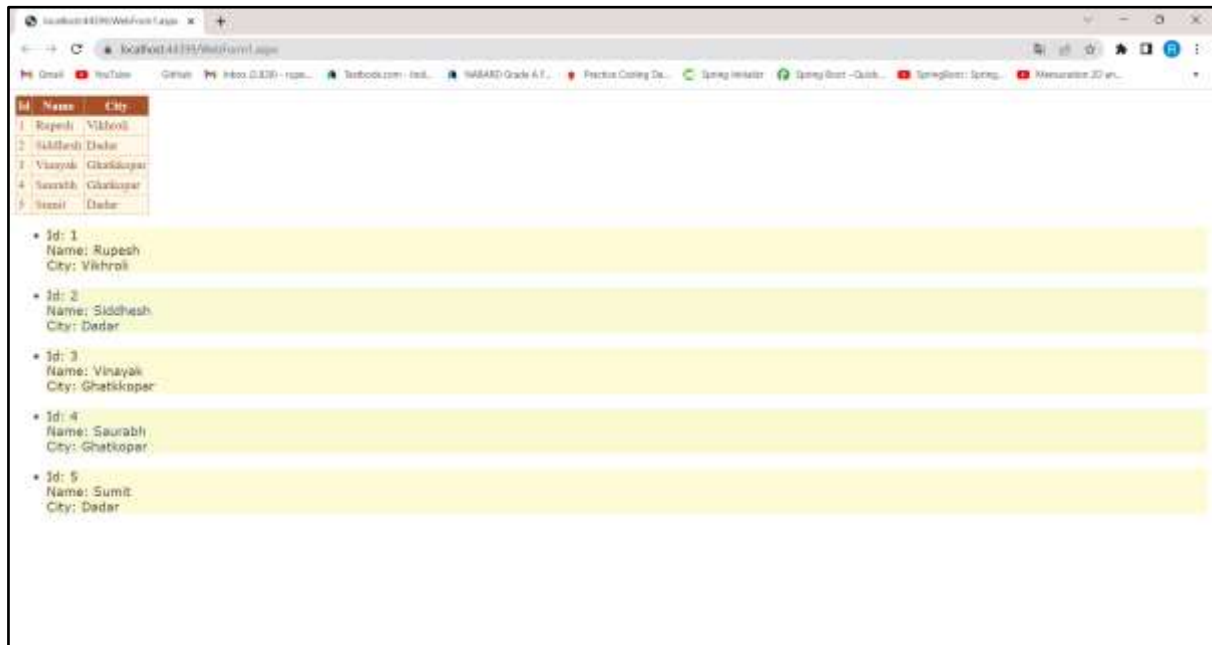
Next >

Finish

Cancel



Output:



Id	Name	City
1	Rupesh	Vikhrol
2	Siddhesh	Dadar
3	Vinayak	Ghatkoper
4	Saurabh	Ghatkoper
5	Sumit	Dadar

- Id: 1
Name: Rupesh
City: Vikhrol
- Id: 2
Name: Siddhesh
City: Dadar
- Id: 3
Name: Vinayak
City: Ghatkoper
- Id: 4
Name: Saurabh
City: Ghatkoper
- Id: 5
Name: Sumit
City: Dadar

B. Design a webpage to demonstrate a connection oriented architecture.Source Code:

WebForm1.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">

    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <center>
                <table class="auto-style1">
                    <tr>
                        <th>
                            <asp:Label ID="Label1" runat="server" Text="Student ID"></asp:Label>
                        </th>
                        <th>
                            <asp:Label ID="Label2" runat="server" Text="Student Name"></asp:Label>
                        </th>
                    </tr>
                    <tr>
                        <td>
                            <asp:TextBox ID="TextBox1" TextMode="MultiLine" runat="server"
Height="172px"></asp:TextBox>
                        </td>
                        <td>
                            <asp:TextBox ID="TextBox2" TextMode="MultiLine" runat="server"
Height="171px"></asp:TextBox>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <asp:Button ID="Button1" runat="server" Text="Show Student Data"
OnClick="Button1_Click" /></td>
                        </tr>
                </table>
            </center>
        </div>
    </form>
</body>
</html>
```

WebForm1.cs:

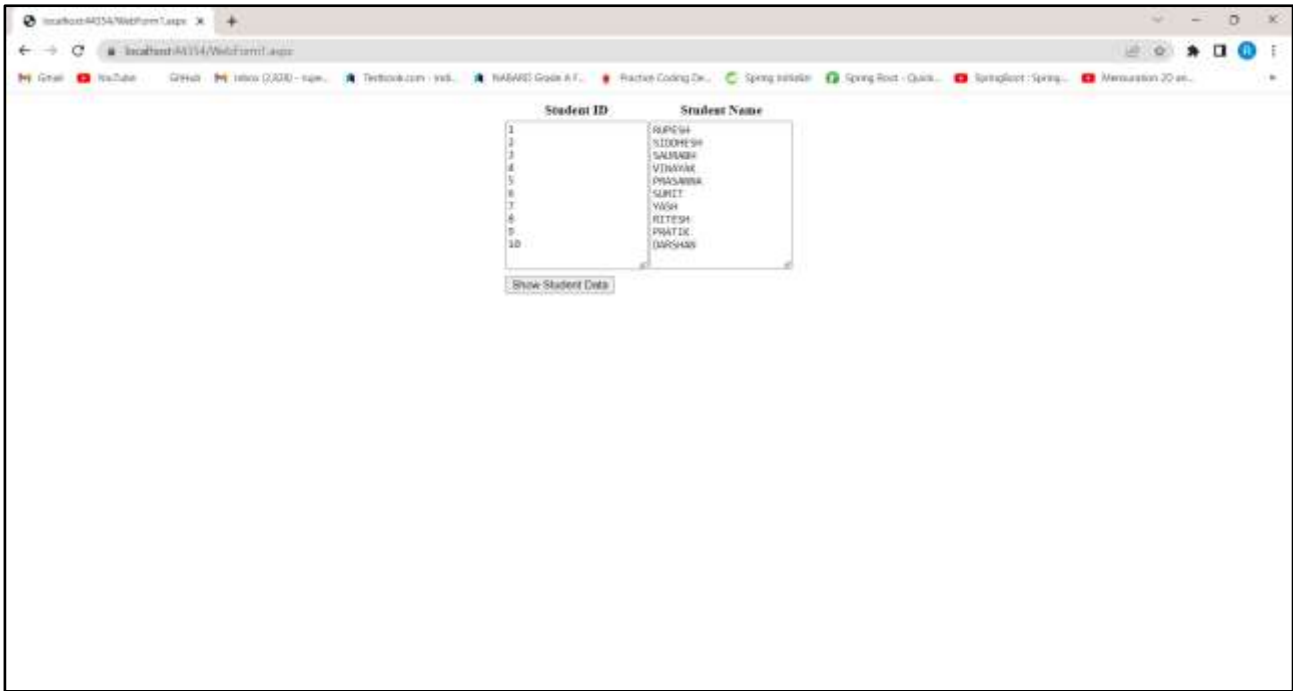
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace ConnectedArch
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader reader;
        protected void Page_Load(object sender, EventArgs e)
        {
            con = new SqlConnection();
            con.ConnectionString = @"Data Source=RUPESH\SQLEXPRESS;Initial
Catalog=rupesh;Integrated Security=True";
            con.Open();

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            TextBox1.Text = null;
            TextBox2.Text = null;
            cmd = new SqlCommand("select * from student", con);
            reader = cmd.ExecuteReader();
            while(reader.Read())
            {
                TextBox1.Text+=reader[0].ToString() + " "+Environment.NewLine;
                TextBox2.Text+=reader[1].ToString()+" "+Environment.NewLine;
            }
            con.Close();
        }
    }
}
```

Output:



C. Design a webpage to demonstrate a disconnected architecture.Source Code:

WebForm1.aspx:

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

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
<title></title>
```

```
<style type="text/css">
```

```
.center-table {
```

```
margin: 0 auto; /* Set left and right margins to auto */
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<center>
```

```
<div>
```

```
<asp:Label ID="Label1" runat="server" Text="Disconnected Architecture"></asp:Label>
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<table align="center" class="center-table auto-style1">
```

```
<tr>
```

```
<td>
```

```
<asp:Label ID="Label2" runat="server" Text="Id"></asp:Label>
```

```
</td>
```

```
<td>
```

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

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td class="auto-style2">
```

```
<asp:Label ID="Label3" runat="server" Text="Name"></asp:Label>
```

```
</td>
```

```
<td class="auto-style2">
```

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

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
<asp:Label ID="Label4" runat="server" Text="City"></asp:Label>
```

```
</td>
```

```
<td>
```



```
myAdapter = new SqlDataAdapter(query, con);
DataSet ds = new DataSet();
myAdapter.Fill(ds);
GridView1.DataSource = ds;
GridView1.DataBind();

}

protected void Insert_Click(object sender, EventArgs e)
{
    string query = "select * from employee";
    myAdapter = new SqlDataAdapter(query, con);
    DataSet ds = new DataSet();
    myAdapter.Fill(ds);
    SqlCommandBuilder cmb = new SqlCommandBuilder(myAdapter);
    DataRow dr = ds.Tables[0].NewRow();
    dr["Id"] = TextBox1.Text;
    dr["Name"] = TextBox2.Text;
    dr["City"] = TextBox3.Text;
    ds.Tables[0].Rows.Add(dr);
    myAdapter.Update(ds);
    GridView1.DataSource = ds;
    GridView1.DataBind();
}

protected void Update_Click(object sender, EventArgs e)
{
    string query = "select * from employee";
    myAdapter = new SqlDataAdapter(query, con);
    DataSet ds = new DataSet();
    myAdapter.Fill(ds);
    SqlCommandBuilder cmb = new SqlCommandBuilder(myAdapter);
    DataRow dr = ds.Tables[0].Rows[sinindex];
    dr["Id"] = Convert.ToInt32(TextBox1.Text);
    dr["Name"] = TextBox2.Text;
    dr["City"] = TextBox3.Text;

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

protected void Delete_Click(object sender, EventArgs e)
{
    string query = "select * from employee";
    myAdapter = new SqlDataAdapter(query, con);
    DataSet ds = new DataSet();
    myAdapter.Fill(ds);
    SqlCommandBuilder cmb = new SqlCommandBuilder(myAdapter);
    ds.Tables[0].Rows[sinindex].Delete();

    myAdapter.Update(ds);
    GridView1.DataSource = ds;
    GridView1.DataBind();
}
```

```
}

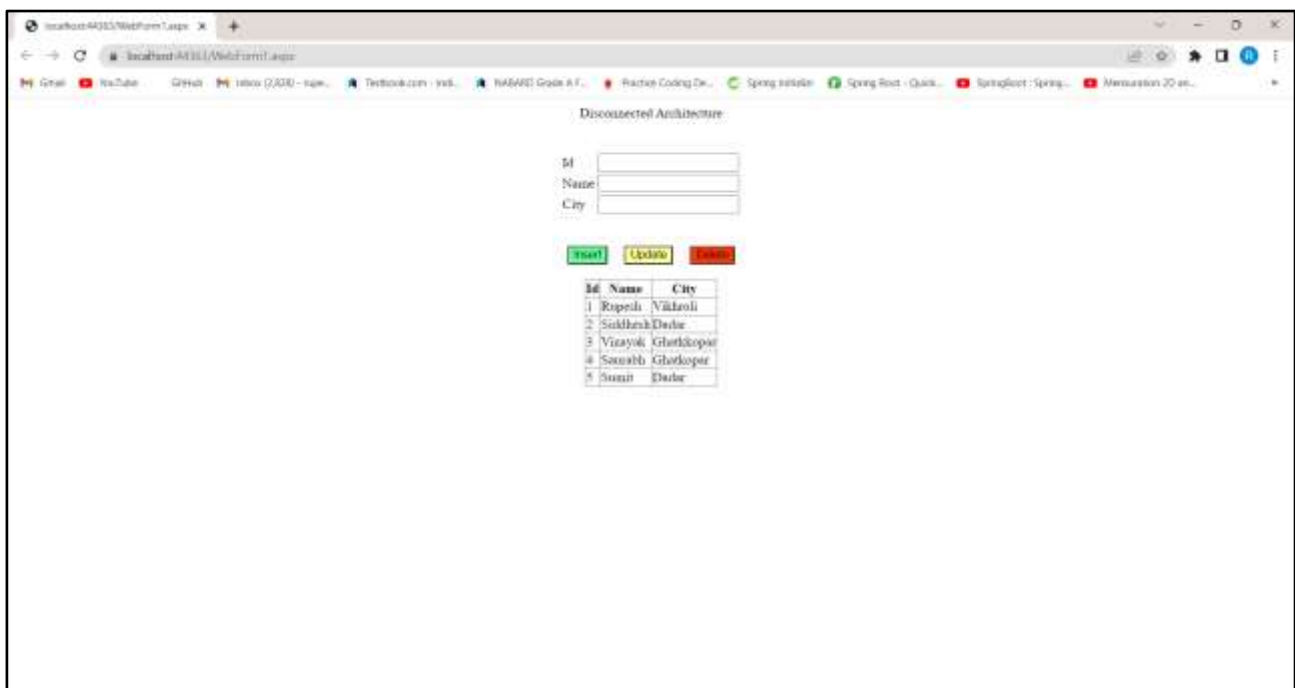
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBox1.Text = GridView1.SelectedRow.Cells[0].Text;
    TextBox2.Text = GridView1.SelectedRow.Cells[1].Text;
    TextBox3.Text = GridView1.SelectedRow.Cells[2].Text;

    sinindex =GridView1.SelectedIndex;

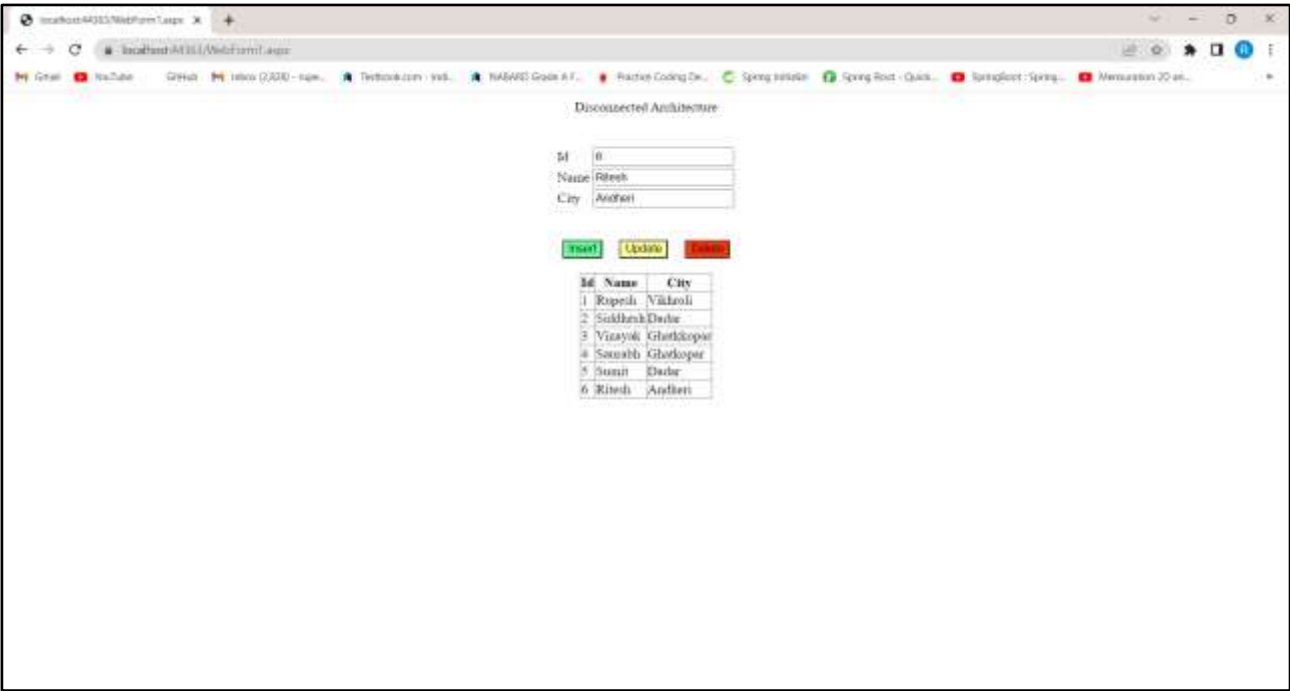
}

protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow) {
        e.Row.Attributes["onclick"] =
Page.ClientScript.GetPostBackClientHyperlink(GridView1, "select$" + e.Row.RowIndex);
        e.Row.ToolTip = "Click to select this row";
    }
}
}
```

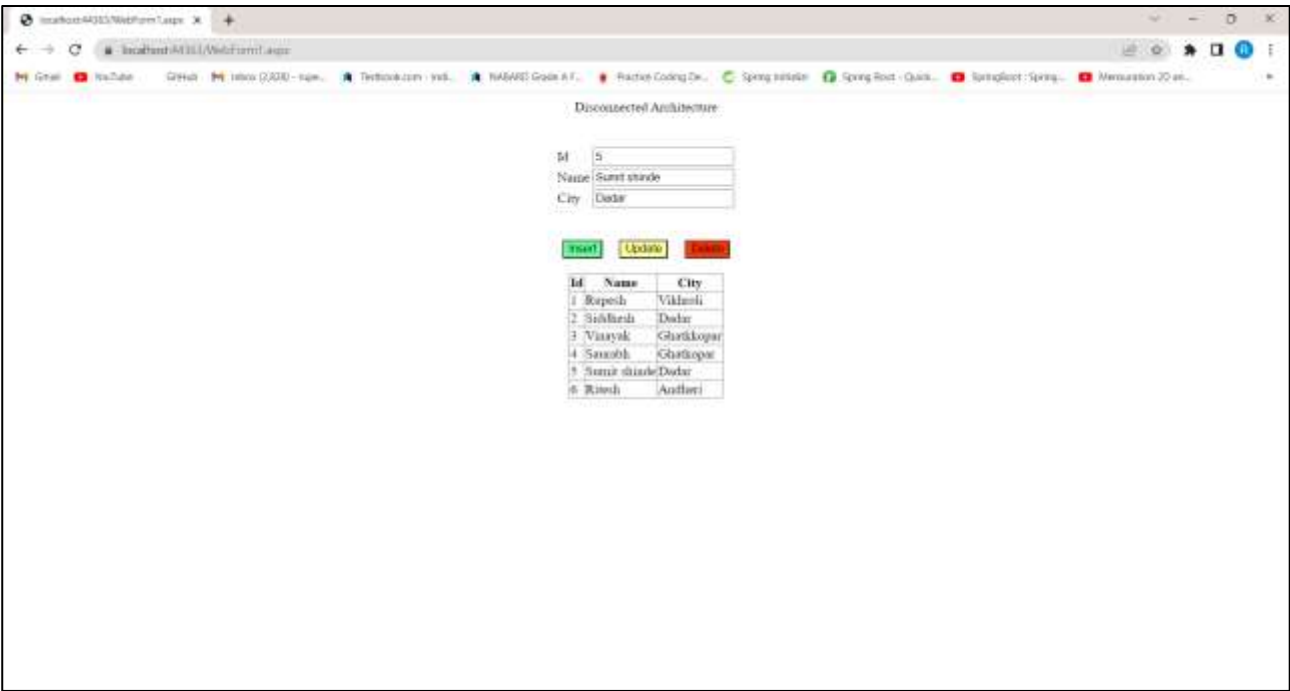
Output:



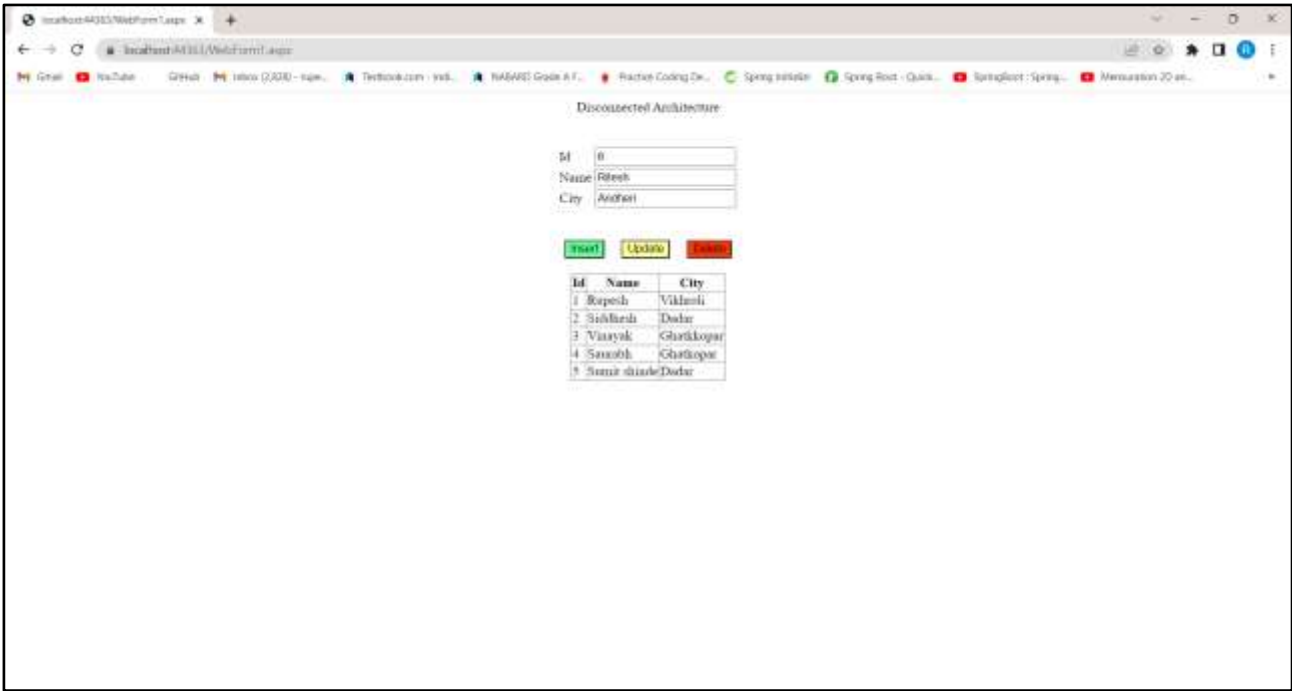
Insert:

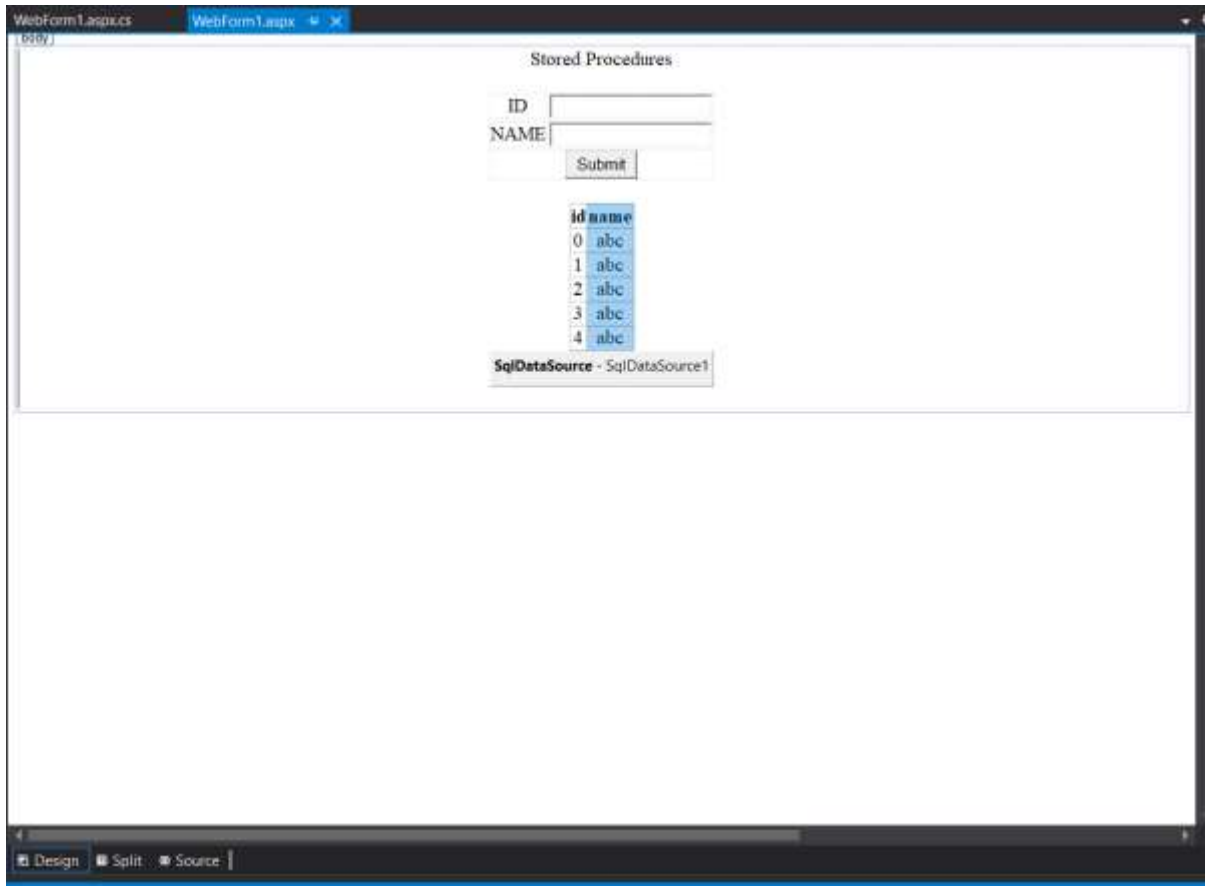


Update:



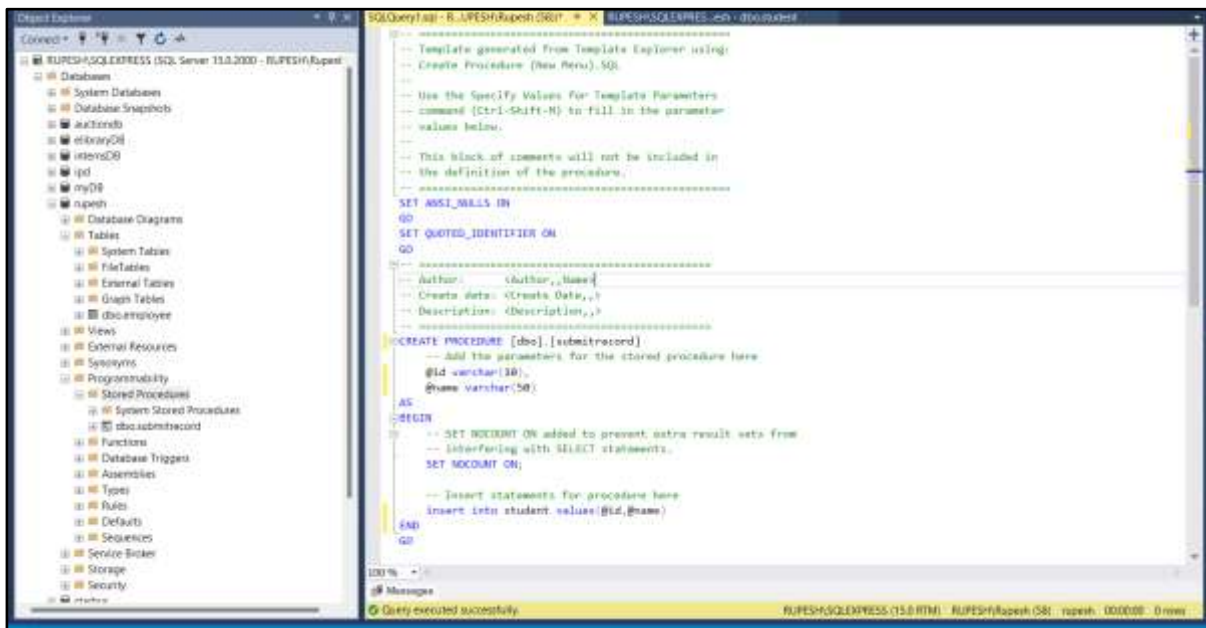
Delete:



D. Design a webpage to demonstrate use of stored procedure.Sql code:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[submitrecord]
    -- Add the parameters for the stored procedure here
    @id varchar(10),
    @name varchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    insert into student values(@id,@name)
END
GO
```



Cs file:

```
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 StoredProcedures
```

```
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;

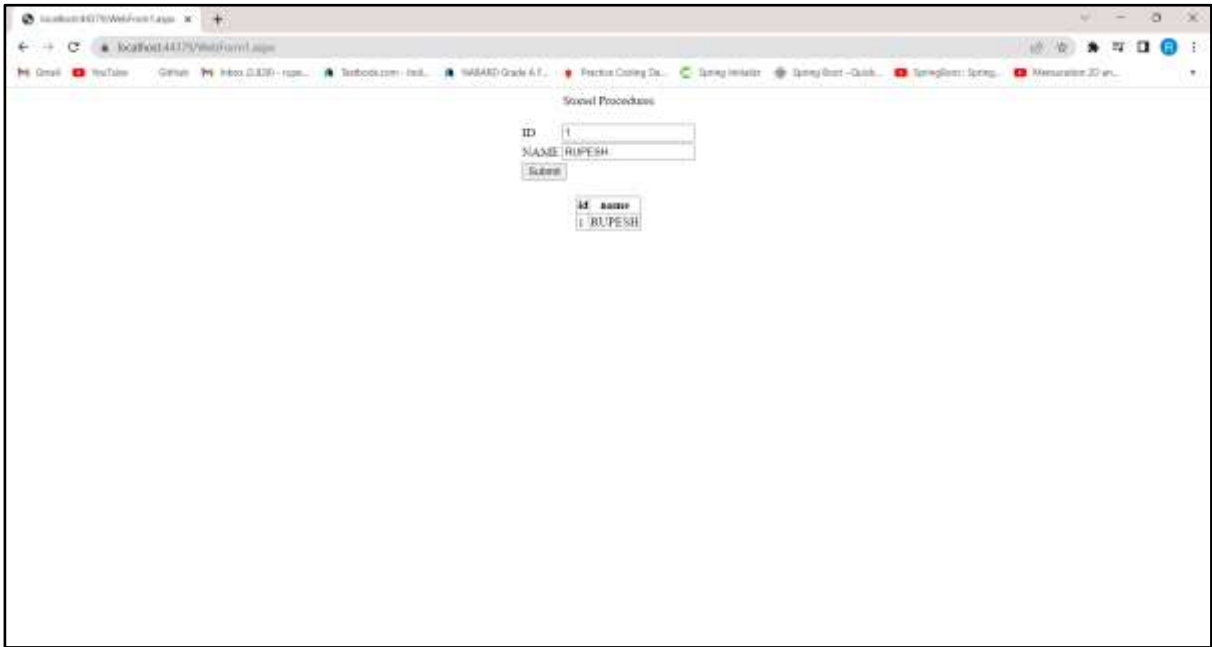
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            con = new SqlConnection(@"Data Source=RUPESH\SQLEXPRESS;Initial
Catalog=rupesh;Integrated Security=True");
            con.Open();
            cmd = new SqlCommand("submitrecord", con);
            cmd.CommandType = CommandType.StoredProcedure;
            SqlParameter param = cmd.Parameters.Add("@id", SqlDbType.VarChar);
            param.Value = TextBox1.Text;
            SqlParameter param1 = cmd.Parameters.Add("@name", SqlDbType.VarChar);
            param1.Value = TextBox2.Text;
            cmd.ExecuteNonQuery();
            GridView1.DataBind();
            con.Close();
        }
    }
}
```

```
}  
}  
}
```

Output:



V] State Management Techniques

A. Design Web Applications using Client Side Session Managements Techniques.

Source Code:

CookiesPage1.aspx:

```
<% @ Page Language="C#" AutoEventWireup="true" CodeBehind="CookiesPage1.aspx.cs"
Inherits="StateManagement2.CookiesPage1" %>
```

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
<title></title>
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAwIGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DkItkvYIK3UENzmM7KCKR/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<center>
```

```
<div class="container">
```

```
<div class="card, card-body border border-warning rounded">
```

```
<h1>ASP.NET State Control Management</h1>
```

```
<h3>Cookies</h3>
```

```
<table class="w-100">
```

```
<tr>
```

```
<td>Username</td>
```

```
<td>
```

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

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Password</td>
```

```
<td>
```

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

```
</td>
```

```

        </tr>
        <tr>
            <td>
                <asp:Button ID="Button1" runat="server" Text="Create Cookie"
OnClick="Button1_Click" />
            </td>
            <td>
                <asp:Button ID="Button2" runat="server" Text="Clear Cookie"
OnClick="Button2_Click" />
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <asp:Button ID="Button3" runat="server" Text="Show Cookies"
OnClick="Button3_Click" />
            </td>
        </tr>
    </table>

    <asp:Table ID="Table2" runat="server" Visible="false">
        <asp:TableRow>
            <asp:TableCell>Cookie Username :</asp:TableCell>
            <asp:TableCell>
                <asp:Label ID="Label1" runat="server"></asp:Label>
            </asp:TableCell>
        </asp:TableRow>
        <asp:TableRow>
            <asp:TableCell>Cookie Password :</asp:TableCell>
            <asp:TableCell>
                <asp:Label ID="Label2" runat="server"></asp:Label>
            </asp:TableCell>
        </asp:TableRow>

    </asp:Table>
    <asp:Label ID="Label3" runat="server"></asp:Label>
</div>

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

```

CookiesPage1.cs:

```

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

```

```

namespace StateManagement2
{

```

```

    public partial class CookiesPage1 : System.Web.UI.Page

```

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        HttpCookie cookie = new HttpCookie("info");
        cookie["username"] = TextBox1.Text.ToString();
        cookie["password"] = TextBox2.Text.ToString();
        Response.Cookies.Add(cookie);
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        HttpCookie cookie = Request.Cookies["info"];
        if (cookie != null)
        {
            Label1.Text = cookie["username"].ToString();
            Label2.Text = cookie["password"].ToString();
            Table2.Visible = true;
            Label3.Text = "";
        }
        else
        {
            Label3.Text = "Cookies are empty";
        }
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        if(Request.Cookies["info"] != null)
        {
            Response.Cookies["info"].Expires = DateTime.Now.AddDays(-1);
            Table2.Visible = false;
        }
    }
}
```

HiddenFieldPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="HiddenFieldPage.aspx.cs"
Inherits="StateManagement2.HiddenFieldPage" %>
```

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
<title></title>
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```



```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DkItIkVYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"
crossorigin="anonymous"></script>
</head>
<body>
  <form id="form1" runat="server">
    <div >
      <center>
        <div class="container" >

          <div class="card, card-body border border-warning rounded">
            <h1>ASP.NET State Control Management</h1>
            <h3>Hidden Field</h3>
            <ul>
              <li>Hidden Field is a state control management provided by ASP.NET, Which is use
to store small amount of data on the client (Browser).</li>
              <li>Hidden Field Control is not rendered to the browser and it is not visible on the
browser.</li>
            </ul>
            <asp:HiddenField ID="HiddenField1" runat="server" Value="5" />
            <asp:Label ID="Label1" runat="server" Text="" Visible="false" class="font-italic
"></asp:Label>
            <asp:Button ID="Button1" class="btn btn-primary btn-lg" runat="server" Text="Show
Hidden Field Value" OnClick="Button1_Click" />
          </div>

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

QueryStringPage1.aspx:

```
<% @ Page Language="C#" AutoEventWireup="true" CodeBehind="QueryStringPage1.aspx.cs"
Inherits="StateManagement2.QueryStringPage1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<center>
```

```
<div class="container">
```

```
<div class="card, card-body border border-dark rounded">
```

```
<h1>ASP.NET State Control Management</h1>
```

```
<h3>Query String</h3>
```

```
<ul style="text-align: left">
```

```
<li>Query String is a collection of character input to a computer or a browser.
```

```
</li>
```

```
</ul>
```

```
<table class="w-100">
```

```
<tr>
```

```
<td style="text-align: right">Enter First Name : </td>
```

```
<td style="text-align: left">
```

```
<div class="form-group">
```

```
<asp:TextBox Class="form-control" ID="TextBox1"
```

```
runat="server"></asp:TextBox></div>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="text-align: right">Enter Last Name : </td>
```

```
<td style="text-align: left">
```

```
<div class="form-group">
```

```
<asp:TextBox Class="form-control" ID="TextBox2"
```

```
runat="server"></asp:TextBox></div>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="text-align: center" colspan="2">
```

```
<asp:Button ID="submit" runat="server" Text="Submit"
```

```
OnClick="submit_Click" /></td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</center>
```

```
</div>
```

```
</form>
```

```
</body>
```

</html>

HiddenFieldPage.cs:

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

namespace StateManagement2
{
    public partial class HiddenFieldPage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (HiddenField1.Value != null)
            {
                int val = Int32.Parse(HiddenField1.Value.ToString());
                val = val + 5;
                Label1.Text = "The value of Hidden Field is incremented by 5 and its current value is
<b>" + val + "</b>";
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Visible = true;
        }
    }
}
```

ViewStatePage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ViewStatePage.aspx.cs"
Inherits="StateManagement2.ViewStatePage" %>
```

<!DOCTYPE html>

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAwIGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DkTlIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtV3Rn7W3mgPxxU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-
```

JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"

crossorigin="anonymous"></script>

</head>

<body>

<form id="form1" runat="server">

<div>

<center>

<div class="container">

<div class="card, card-body border border-dark rounded">

<h1>ASP.NET State Control Management</h1>

<h3>View State</h3>

<ul style="text-align: left">

View State is another client side state management control/technique, which is used to store user's data

View State provide page level State Management.

View State can store any type of data.

<table class="w-100">

<tr>

<td style="text-align: right">Enter Your Name : </td>

<td style="text-align: left">

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

</td>

</tr>

<tr>

<td style="text-align: right">

<asp:Button ID="save" runat="server" Text="Save" OnClick="save_Click"

class="btn btn-primary" />

</td>

<td style="text-align: left">

<asp:Button ID="display" runat="server" Text="Display"

OnClick="display_Click" class="btn btn-danger" />

</td>

</tr>

<tr>

<td style="text-align: right">Data in View State : </td>

<td style="text-align: left">

<asp:Label ID="Label1" runat="server" Text=""></asp:Label></td>

</tr>

<tr>

<td style="text-align: right">Number ofPostBack: </td>

<td style="text-align: left">

<asp:Label ID="Label2" runat="server" Text=""></asp:Label></td>

</tr>

</table>

</div>

</div>

</center>

</div>

</form>

</body>

</html>

ViewStatePage.cs:

using System;

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

```
namespace StateManagement2
{
    public partial class ViewStatePage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (IsPostBack)
            {
                if (ViewState["Counter"] != null)
                {
                    int counter = Int32.Parse(ViewState["Counter"].ToString());
                    counter = counter + 1;
                    Label2.Text = counter.ToString();
                    ViewState["Counter"] = counter.ToString();
                }
                else
                {
                    ViewState["Counter"] = "1";
                }
            }
        }

        protected void save_Click(object sender, EventArgs e)
        {
            ViewState["userName"] = TextBox1.Text;
            TextBox1.Text = "";
            Label2.Text = ViewState["Counter"].ToString();
        }

        protected void display_Click(object sender, EventArgs e)
        {
            Label1.Text = ViewState["userName"].ToString();
            Label2.Text = ViewState["Counter"].ToString();
        }
    }
}
```

Welcome.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Welcome.aspx.cs"
Inherits="StateManagement2.Welcome" %>
```

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
<title></title>
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-
```

Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"

crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2Dk1IkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</head>

<body>

<form id="form1" runat="server">

<div>

<center>

<div class="container">

<div class="card, card-body border border-dark rounded">

<h1>Welcome Page</h1>

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

</div>

</div>

</center>

</div>

</form>

</body>

</html>

Welcome.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace StateManagement2

{

public partial class Welcome : System.Web.UI.Page

{

protected void Page_Load(object sender, EventArgs e)

{

string fname = Request.QueryString["fname"].ToString();

string lname = Request.QueryString["lname"].ToString();

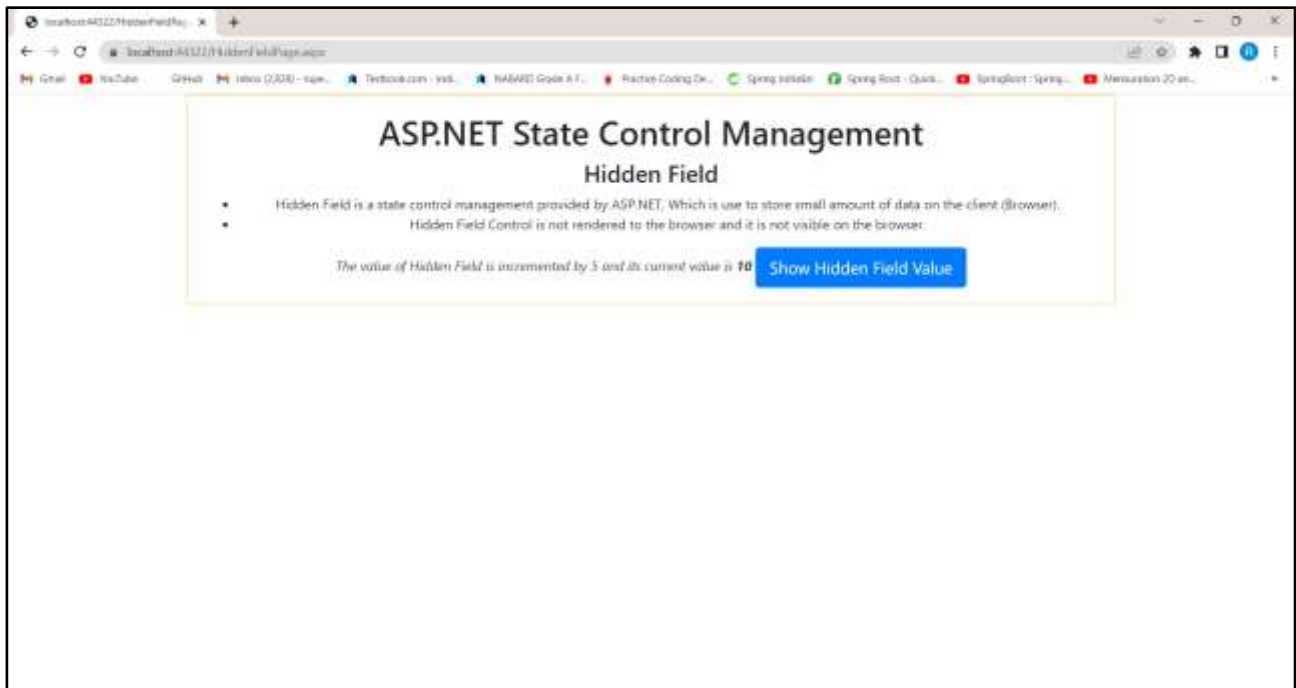
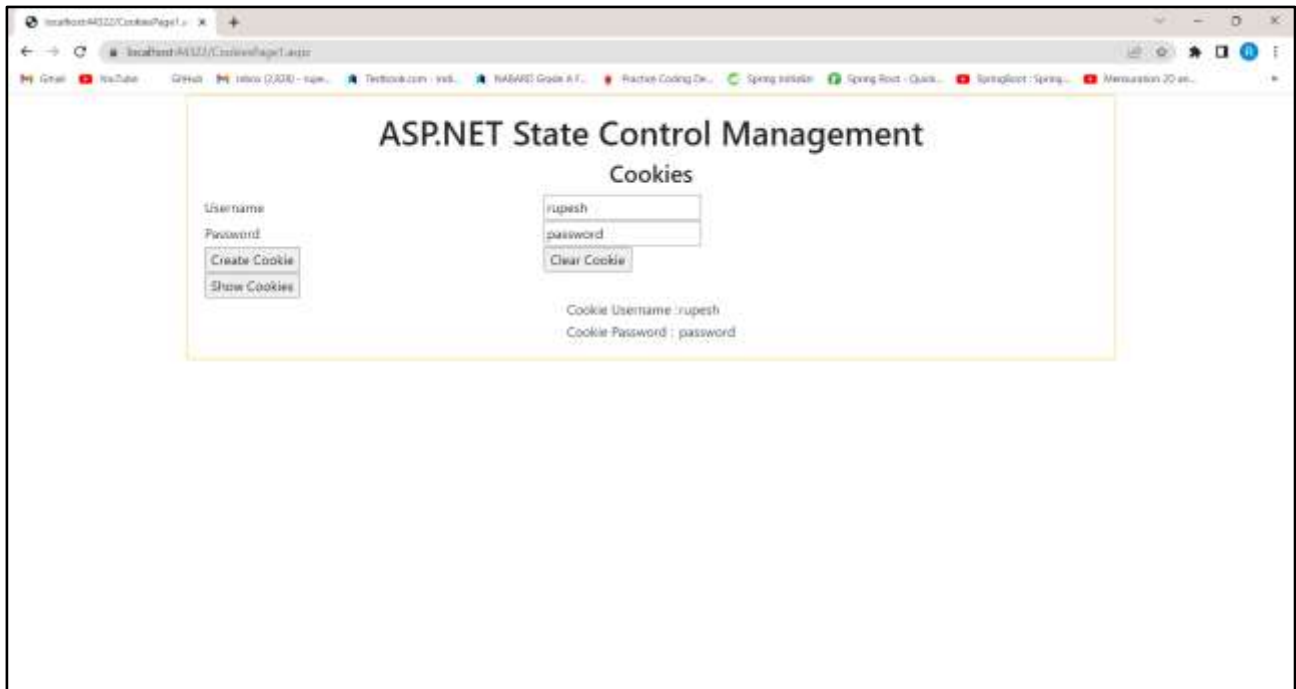
Label1.Text = "Welcome " + fname + " " + lname + " .";

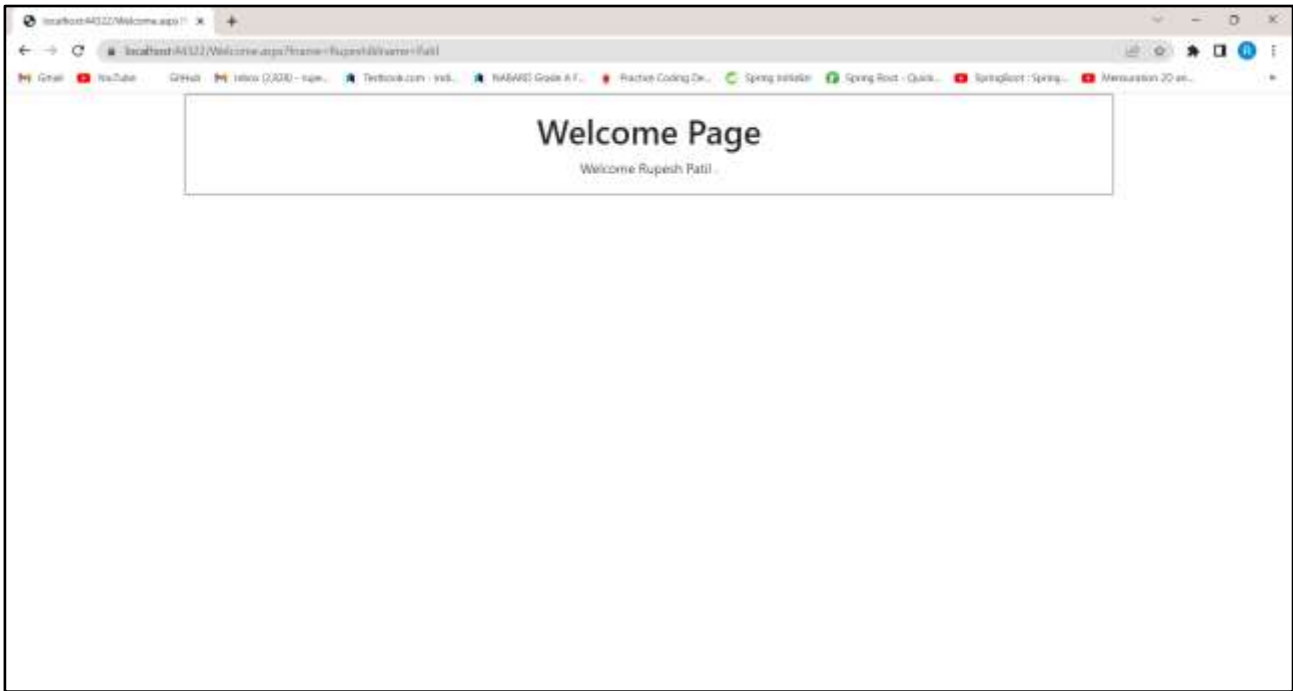
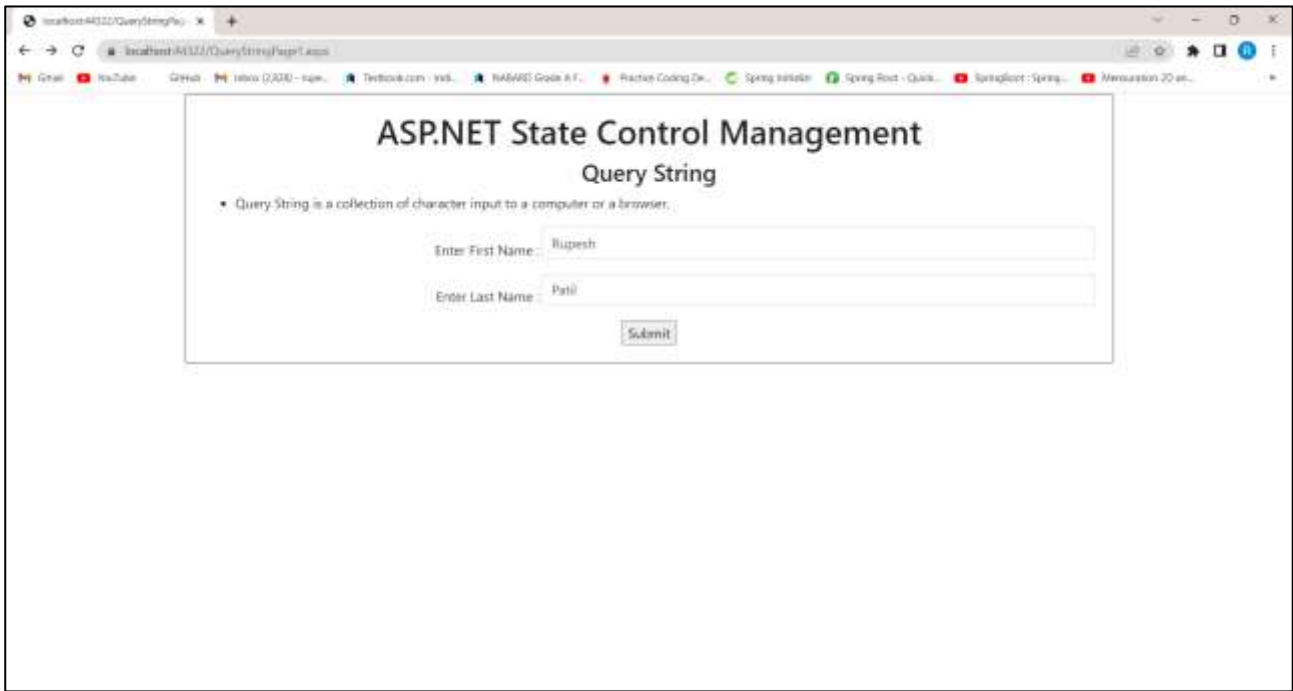
}

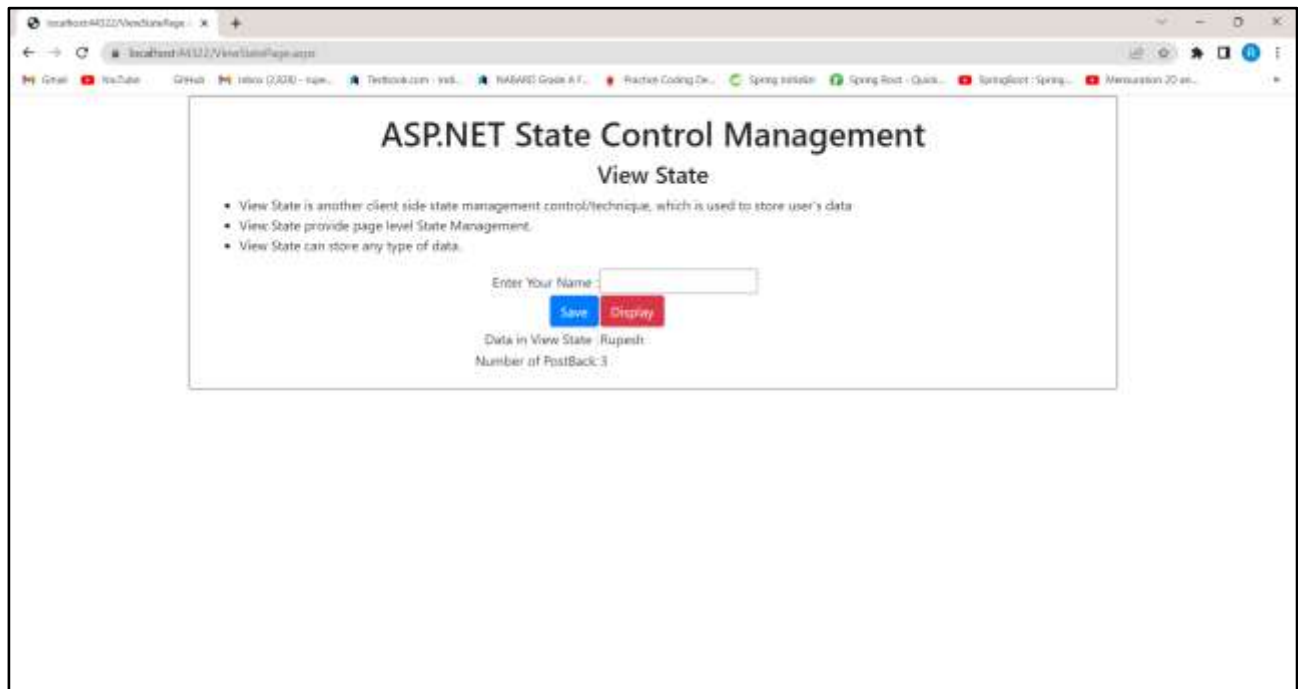
}

}

Output:







B. Design Web Applications using Server Side Session Management TechniquesSource 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");
        }
    }
}
```

```
<%@ 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:

VI] Web Services and WCF**A. Design Web Application to produce and Consume a web Service.**Source Code:

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

```
<!DOCTYPE html>
```

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

```
<head runat="server">
```

```
<title></title>
```

```
<style type="text/css">
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<center>
```

```
<table class="auto-style1">
```

```
<tr>
```

```
<td>Enter value of Principal</td>
```

```
<td>
```

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

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Enter value of Years</td>
```

```
<td>
```

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

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Enter value of ROI</td>
```

```
<td>
```

```
<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2">
```

```
<asp:Button ID="Button1" runat="server" Text="Calculate Simple Interest"
```

```
OnClick="Button1_Click" />
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td colspan="2">&nbsp;  </td>
```

```
</tr>
```

```
</table>
```

```
<div>
```

```
<asp:Label ID="Label1" runat="server" Text=""></asp:Label>
```

```
</div>
</center>
</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 SimpleInterestWebService
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            localhost.WebServiceSI s=new localhost.WebServiceSI();
            double p=Convert.ToDouble(TextBox1.Text);
            double n = Convert.ToDouble(TextBox2.Text);
            double r = Convert.ToDouble(TextBox3.Text);
            double result=s.SI(p, n, r);
            Label1.Text = "Simple Interest: "+result.ToString();
        }
    }
}
```

Output:



B. Design Web Application to produce and Consume a WCF Service.Source Code:

```
<%@ 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>
            Enter principle:
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            Enter number of years:
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            Enter rate of intrest:
            <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Calculate" />
            <br />
            <asp:Label ID="Label1" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.cs:

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

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        ServiceReference1.ServiceClient srv = new ServiceReference1.ServiceClient();
```

```
double p = Convert.ToDouble(TextBox1.Text);
double n = Convert.ToDouble(TextBox2.Text);
double r = Convert.ToDouble(TextBox3.Text);
double ans = srv.simpleIntrest(p, n, r);
Label1.Text = ans.ToString();
```

```
    }
}
```

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 simpleIntrest(double p, double n, double r);
```

```
    [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; }
```

```
}
```


}

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

    public double simpleIntrest(double p, double n, double r)
    {
        return ((p * n * r) / 100);
    }
}
```

Output:



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" data-
target=".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 = "navbar-
brand" })
            </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>
```

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>

</div>

</div>

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

<h2>Fantasy Novels</h2>

<div>

</div>

</div>

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

<h2>Marathi Books</h2>

<div>

</div>

</div>

</div>

StoryBooks.cshtml:

@{

ViewBag.Title = "StoryBooks";

}

@section Styles

{

}

<h1>StoryBooks</h1>

<div class="row">

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

<div>

</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>

</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>

</div>

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

</div>

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

<div>

</div>

<h3>The Name of The Wind.</h3>

</div>

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

<div>

</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>

</div>

<h3>मृत्युंजय</h3>

</div>

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

<div>

<h3>श्रीमान योगी</h3>

</div>

</div>

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

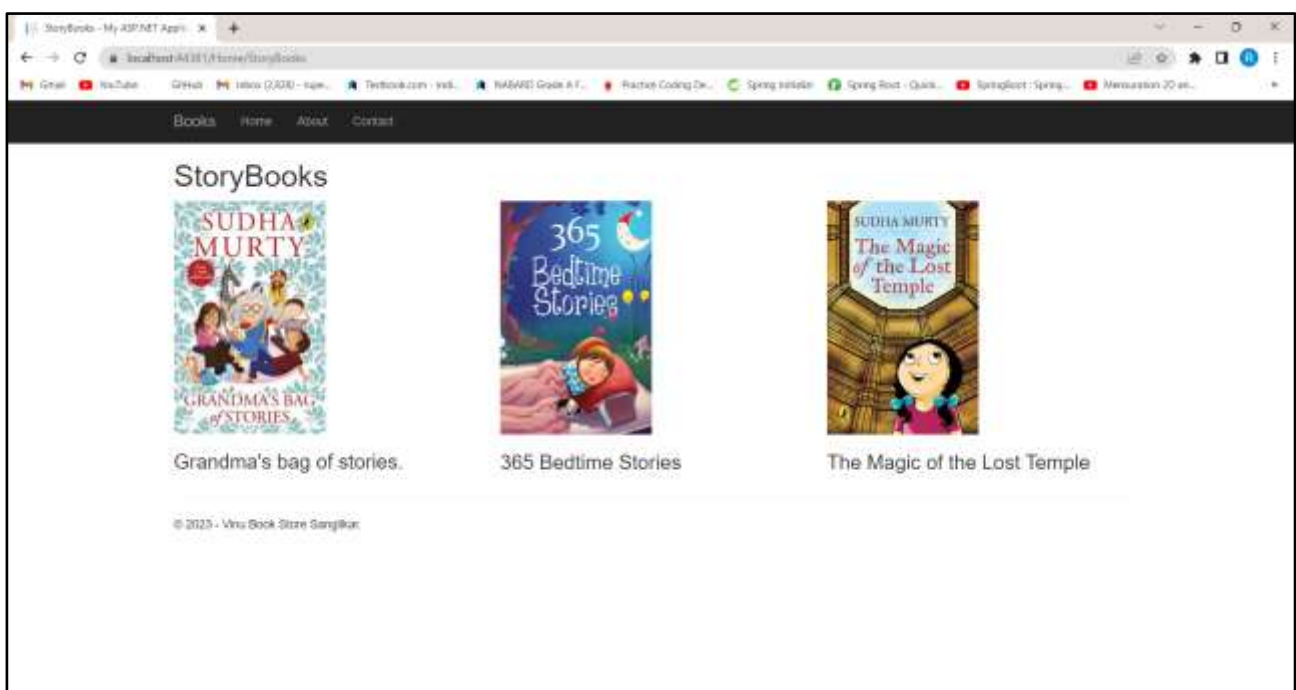
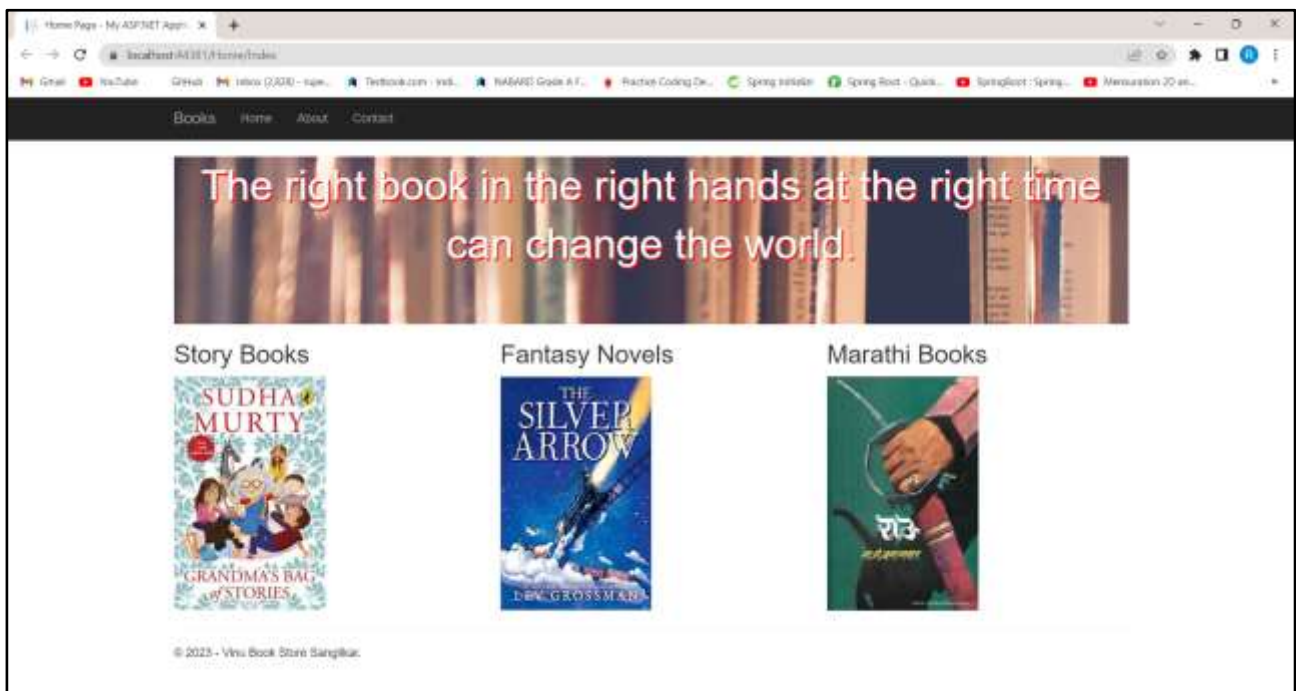
<div>

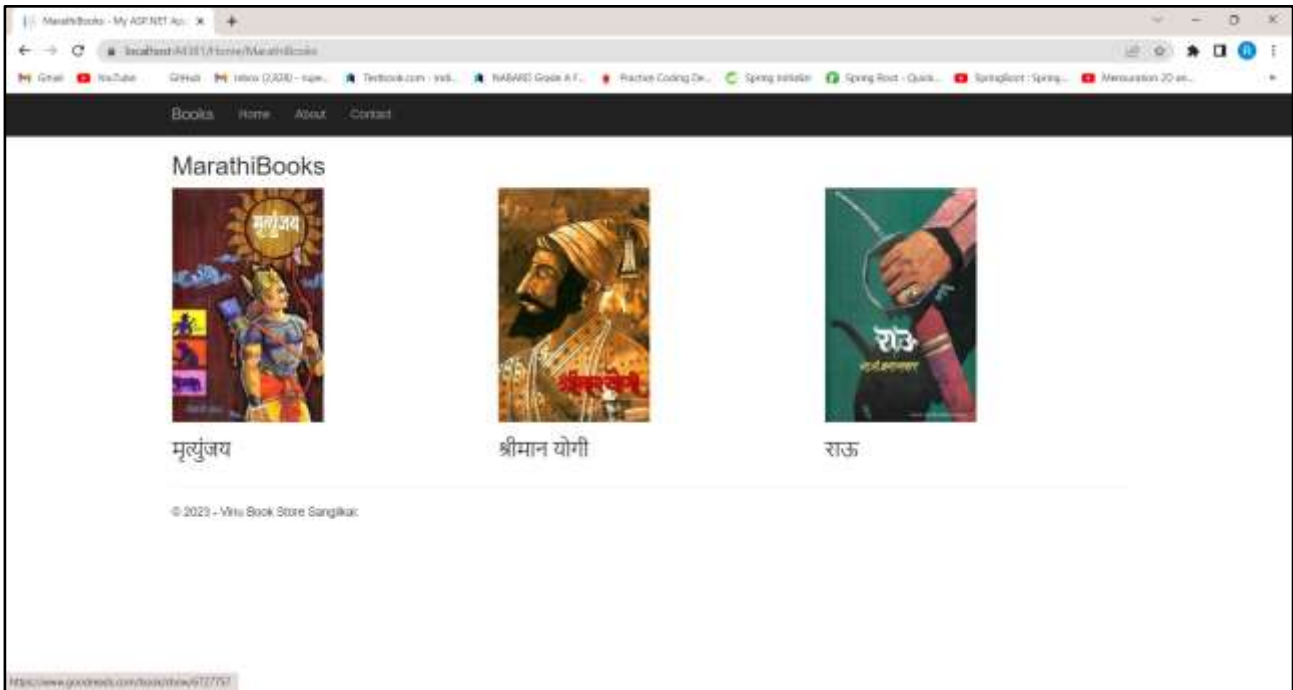
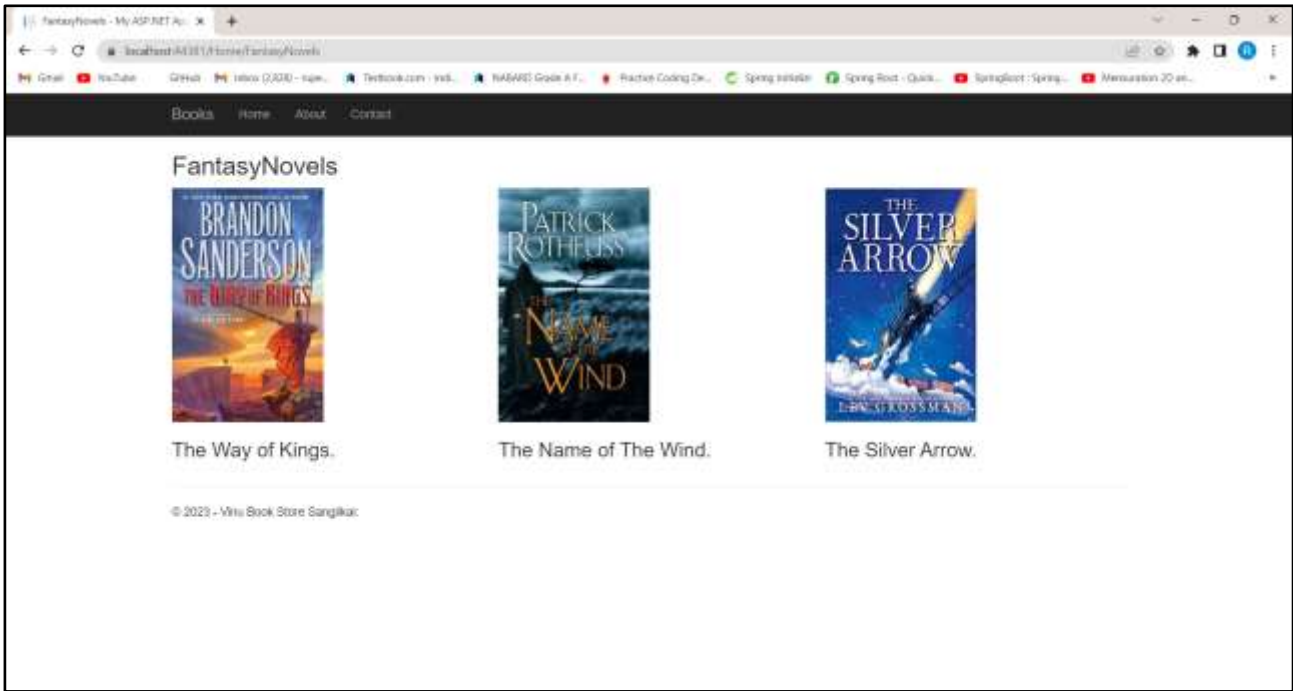
राऊ

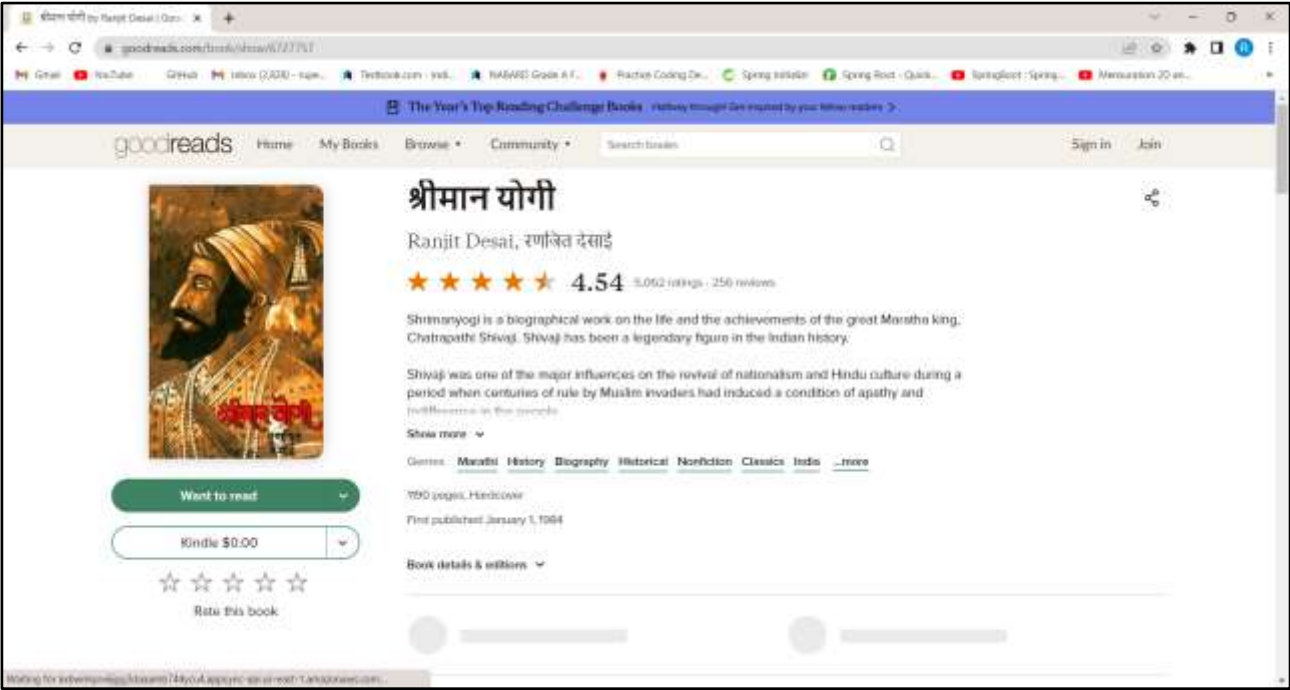
</div>

</div>

</div>

Output:





VIII] LINQ**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;
        }
    }
}
```

Output:

