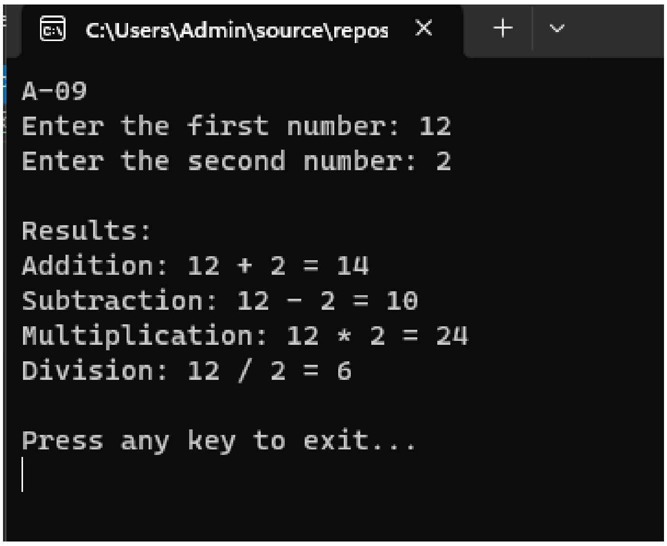
Practical-1

a) Create an application to print on screen the output of adding, subtracting, multiplying and dividing two numbers entered by the user in C#

Code:

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

namespace ConsoleApp3
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("A-09");

         Console.Write("Enter the first number: ");
         double num1 = Convert.ToDouble(Console.ReadLine());


         Console.Write("Enter the second number: ");
         double num2 = Convert.ToDouble(Console.ReadLine());


         double sum = num1 + num2;
         double difference = num1 - num2;
         double product = num1 * num2;
         double quotient = num1 / num2;


         Console.WriteLine("\nResults:");
         Console.WriteLine($"Addition: {num1} + {num2} = {sum}");
         Console.WriteLine($"Subtraction: {num1} - {num2} = {difference}");
         Console.WriteLine($"Multiplication: {num1} * {num2} = {product}");
         Console.WriteLine($"Division: {num1} / {num2} = {quotient}");


         Console.WriteLine("\nPress any key to exit...");
         Console.ReadKey();
      }
   }
}
```

Output:



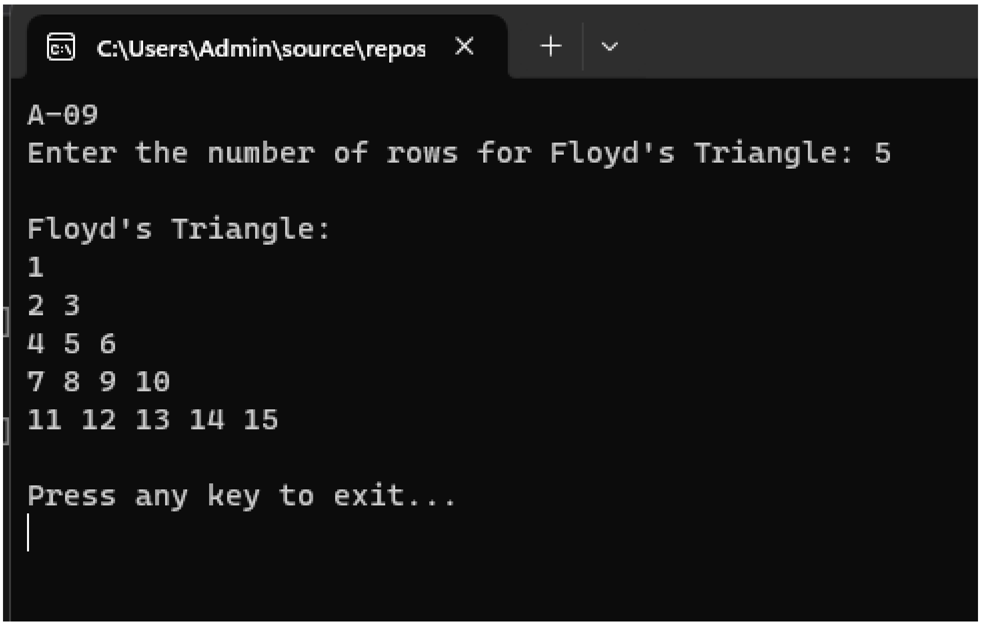
```
A-09
Enter the first number: 12
Enter the second number: 2

Results:
Addition: 12 + 2 = 14
Subtraction: 12 - 2 = 10
Multiplication: 12 * 2 = 24
Division: 12 / 2 = 6

Press any key to exit...
```

b) Create an application to print Floyd’s triangle till n rows in C#.

Code:

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

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("A-09");

            // Ask the user for the number of rows
            Console.Write("Enter the number of rows for Floyd's Triangle: ");
            int n = Convert.ToInt32(Console.ReadLine());

            int number = 1;

            Console.WriteLine("\nFloyd's Triangle:");

            // Generate Floyd's Triangle
            for (int i = 1; i <= n; i++)
            {
                for (int j = 1; j <= i; j++)
                {
                    Console.Write(number + " ");
                    number++;
                }
                Console.WriteLine();
            }

            // Wait for the user to press a key before closing the console
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```

Output:



```
A-09
Enter the number of rows for Floyd's Triangle: 5

Floyd's Triangle:
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Press any key to exit...
```

c) Create an application to demonstrate following operations i. Generate Fibonacci series. ii. Test for prime numbers.

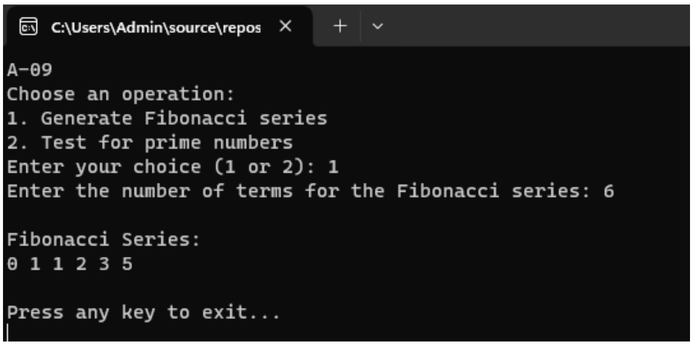
Code:

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

namespace ConsoleApp3
{
    class Program
    {
```

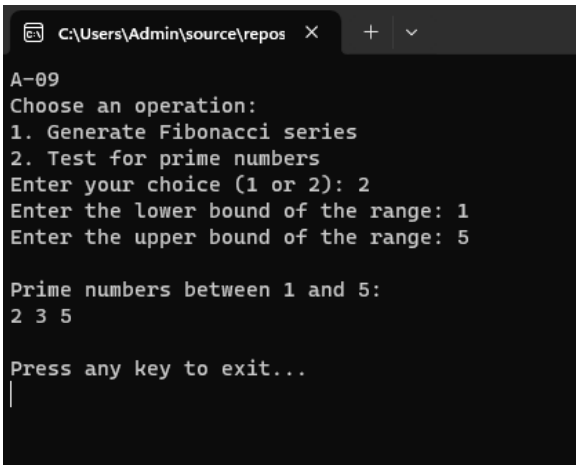
```
static void Main(string[] args)
{
    Console.WriteLine("A-09");

    // Choose the operation
    Console.WriteLine("Choose an operation:");
    Console.WriteLine("1. Generate Fibonacci series");
    Console.WriteLine("2. Test for prime numbers");
    Console.Write("Enter your choice (1 or 2): ");
    int choice = Convert.ToInt32(Console.ReadLine());

    switch (choice)
    {
        case 1:
            GenerateFibonacci();
            break;
        case 2:
            TestPrimeNumbers();
            break;
        default:
            Console.WriteLine("Invalid choice.");
            break;
    }

    // Wait for the user to press a key before closing the console
    Console.WriteLine("\nPress any key to exit...");
    Console.ReadKey();
}

// Method to generate Fibonacci series
static void GenerateFibonacci()
{
    Console.Write("Enter the number of terms for the Fibonacci series: ");
    int terms = Convert.ToInt32(Console.ReadLine());

    int first = 0, second = 1, next;

    Console.WriteLine("\nFibonacci Series:");
    for (int i = 1; i <= terms; i++)
    {
        Console.Write(first + " ");
        next = first + second;
        first = second;
        second = next;
    }
    Console.WriteLine();
}
```

```
        // Method to test for prime numbers within a range
        static void TestPrimeNumbers()
        {
            Console.Write("Enter the lower bound of the range: ");
            int lower = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter the upper bound of the range: ");
            int upper = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine($"\nPrime numbers between {lower} and {upper}:");

            for (int i = lower; i <= upper; i++)
            {
                if (IsPrime(i))
                {
                    Console.Write(i + " ");
                }
            }
            Console.WriteLine();
        }

        // Helper method to check if a number is prime
        static bool IsPrime(int number)
        {
            if (number <= 1)
                return false;
            if (number == 2)
                return true;

            for (int i = 2; i <= Math.Sqrt(number); i++)
            {
                if (number % i == 0)
                    return false;
            }

            return true;
        }
    }
}
```

Output:



```
A-09
Choose an operation:
1. Generate Fibonacci series
2. Test for prime numbers
Enter your choice (1 or 2): 1
Enter the number of terms for the Fibonacci series: 6

Fibonacci Series:
0 1 1 2 3 5

Press any key to exit...
```



```
A-09
Choose an operation:
1. Generate Fibonacci series
2. Test for prime numbers
Enter your choice (1 or 2): 2
Enter the lower bound of the range: 1
Enter the upper bound of the range: 5

Prime numbers between 1 and 5:
2 3 5

Press any key to exit...
```

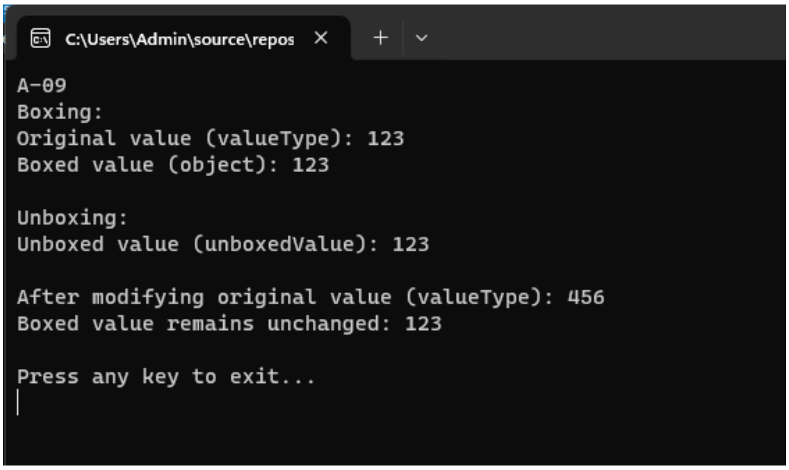
Practical-2

a) Create a simple application to demonstrate the concepts boxing and unboxing.

Code:

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

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("A-09");

            // Boxing: Converting a value type to an object type
            int valueType = 123;  // Value type
            object boxedValue = valueType;  // Boxing

            Console.WriteLine("Boxing:");
            Console.WriteLine($"Original value (valueType): {valueType}");
            Console.WriteLine($"Boxed value (object): {boxedValue}");

            // Unboxing: Converting an object type back to a value type
            int unboxedValue = (int)boxedValue;  // Unboxing

            Console.WriteLine("\nUnboxing:");
            Console.WriteLine($"Unboxed value (unboxedValue):
{unboxedValue}");

            // Modifying the original value and showing that boxed value is
unaffected
            valueType = 456;
            Console.WriteLine($"\nAfter modifying original value (valueType):
{valueType}");
            Console.WriteLine($"Boxed value remains unchanged:
{boxedValue}");

            // Wait for the user to press a key before closing the console
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```

Output:



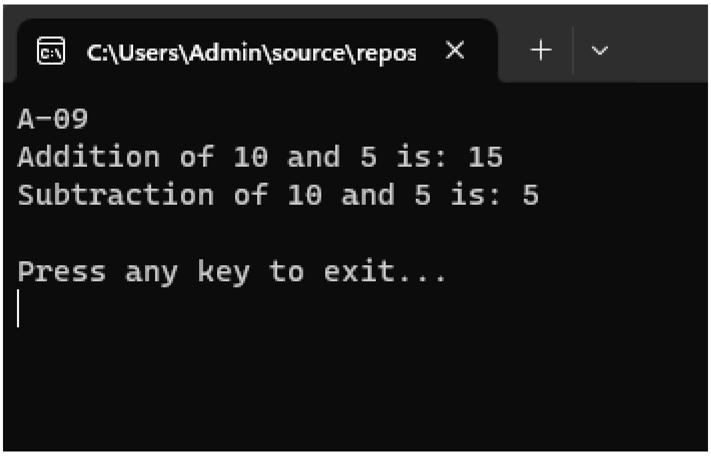
```
A-09
Boxing:
Original value (valueType): 123
Boxed value (object): 123

Unboxing:
Unboxed value (unboxedValue): 123

After modifying original value (valueType): 456
Boxed value remains unchanged: 123

Press any key to exit...
```

b) Create a simple application to perform addition and subtraction using delegate.

Code:

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

namespace ConsoleApp3
{
    class Program
    {
        // Define a delegate that takes two integers and returns an integer
        delegate int OperationDelegate(int x, int y);

        static void Main(string[] args)
        {
            Console.WriteLine("A-09");
            // Instantiate the delegate with the Addition and Subtraction methods
            OperationDelegate addDelegate = new OperationDelegate(Addition);
            OperationDelegate subtractDelegate = new
OperationDelegate(Subtraction);

            // Sample numbers to perform operations on
            int num1 = 10;
            int num2 = 5;

            // Perform addition using the delegate
            int addResult = addDelegate(num1, num2);
            Console.WriteLine($"Addition of {num1} and {num2} is: {addResult}");

            // Perform subtraction using the delegate
            int subtractResult = subtractDelegate(num1, num2);
            Console.WriteLine($"Subtraction of {num1} and {num2} is:
{subtractResult}");

            // Wait for the user to press a key before closing the console
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }

        // Method for addition
        static int Addition(int x, int y)
        {
            return x + y;
        }

        // Method for subtraction
        static int Subtraction(int x, int y)
        {
            return x - y;
        }
    }
}
```

Output:



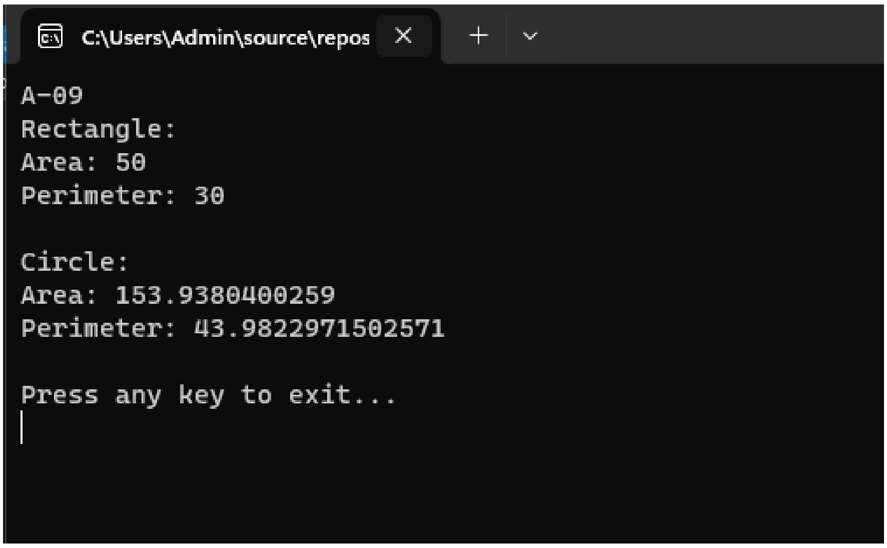
```
A-09
Addition of 10 and 5 is: 15
Subtraction of 10 and 5 is: 5

Press any key to exit...
```

c) Create a simple application to demonstrate use of the concepts of interfaces.

Code:

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

namespace ConsoleApp3
{
    // Define an interface
    interface IShape
    {
        double Area(); // Method to calculate area
        double Perimeter(); // Method to calculate perimeter
    }

    // Implement the interface in a class
    class Rectangle : IShape
    {
        private double length;
        private double width;

        // Constructor
        public Rectangle(double length, double width)
        {
            this.length = length;
            this.width = width;
        }

        // Implement the Area method
        public double Area()
        {
            return length * width;
        }

        // Implement the Perimeter method
        public double Perimeter()
        {
            return 2 * (length + width);
        }
    }

    // Implement the interface in another class
    class Circle : IShape
    {
        private double radius;
        // Constructor
        public Circle(double radius)
        {
            this.radius = radius;
        }

        // Implement the Area method
        public double Area()
        {
            return Math.PI * radius * radius;
        }

        // Implement the Perimeter method
        public double Perimeter()
        {
            return 2 * Math.PI * radius;
        }
    }
    class Program
    {

        static void Main(string[] args)
        {
            Console.WriteLine("A-09");
            // Create instances of Rectangle and Circle
            IShape rectangle = new Rectangle(10, 5);
            IShape circle = new Circle(7);

            // Display the area and perimeter of the rectangle
            Console.WriteLine("Rectangle:");
            Console.WriteLine($"Area: {rectangle.Area()}");
            Console.WriteLine($"Perimeter: {rectangle.Perimeter()}");

            // Display the area and perimeter of the circle
            Console.WriteLine("\nCircle:");
            Console.WriteLine($"Area: {circle.Area()}");
            Console.WriteLine($"Perimeter: {circle.Perimeter()}");

            // Wait for the user to press a key before closing the console
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```

Output:



```
A-09
Rectangle:
Area: 50
Perimeter: 30

Circle:
Area: 153.9380400259
Perimeter: 43.9822971502571

Press any key to exit...
```

Practical-3

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

Code:

WebForm1.aspx

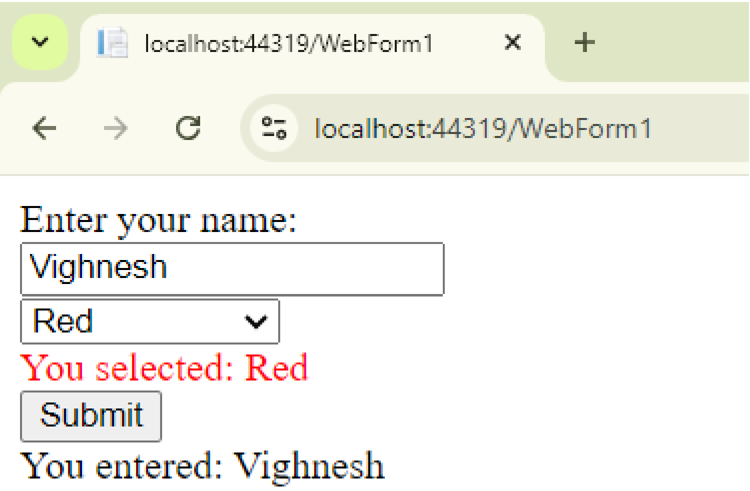
```
<<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication4.WebForm1"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <!-- Label Control -->
         <asp:Label ID="Label1" runat="server" Text="Enter your
name:"></asp:Label><br />

         <!-- TextBox Control with AutoPostBack -->
         <asp:TextBox ID="TextBox1" runat="server" AutoPostBack="True"
OnTextChanged="TextBox1_TextChanged"></asp:TextBox><br />

         <!-- DropDownList Control with AutoPostBack -->
         <asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="True"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
            <asp:ListItem>Select a color</asp:ListItem>
            <asp:ListItem>Red</asp:ListItem>
            <asp:ListItem>Green</asp:ListItem>
            <asp:ListItem>Blue</asp:ListItem>
         </asp:DropDownList><br />

         <!-- Label to display the selected value -->
         <asp:Label ID="Label2" runat="server" Text=""></asp:Label><br />

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

         <!-- Label to display the TextBox content -->
         <asp:Label ID="Label3" runat="server" Text=""></asp:Label>
```

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

WebForm1.aspx.cs

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

namespace WebApplication4
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label3.Text = "Form submitted with name: " + TextBox1.Text;
        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {
            Label3.Text = "You entered: " + TextBox1.Text;
        }

        protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            Label2.Text = "You selected: " + DropDownList1.SelectedItem.Text;
            Label2.ForeColor =
System.Drawing.Color.FromName(DropDownList1.SelectedItem.Text);
        }
    }
}
```

Output:



Enter your name:

Vighnesh

Red

You selected: Red

Submit

You entered: Vighnesh

b) Create a simple application to demonstrate your vacation using calendar control.

Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Calender.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:Calendar ID="Calendar1" runat="server"
OnDayRender="AttachHolidays"
OnSelectionChanged="Button1_Click"></asp:Calendar>

            <br />
            &nbsp;<asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />
            &nbsp;<br />
            <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
            <br />
            <asp:Label ID="Label2" runat="server"
Text="Label"></asp:Label>
            <br />
```

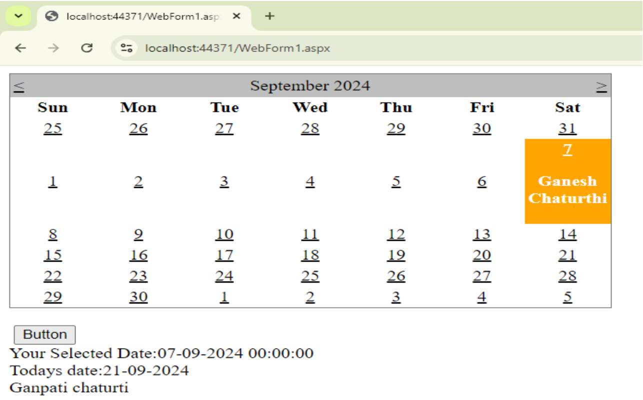
```
          <asp:Label ID="Label3" runat="server"
Text="Label"></asp:Label>

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

WebForm1.aspx.cs

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

namespace Calender
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      Dictionary<string, string> holidays = new Dictionary<string,
string>();
      protected void Page_Load(object sender, EventArgs e)
      {
         holidays.Add("7", "Ganesh </br> Chaturthi");
      }
      protected void AttachHolidays(object sender,
DayRenderEventArgs e)
      {
         if (e.Day.Date.Day == 7 && e.Day.Date.Month == 9)
         {
            e.Cell.Controls.Add(new LiteralControl("<p>" +
            holidays[e.Day.DayNumberText] + "</p>"));
            e.Cell.BackColor = System.Drawing.Color.Blue;
            e.Cell.BackColor = System.Drawing.Color.Red;
            e.Cell.BackColor = System.Drawing.Color.Orange;
            e.Cell.Font.Bold = true;
         }

      }
      protected void Button1_Click(object sender, EventArgs e)
      {
         Label1.Text = "Your Selected Date:" +
Calendar1.SelectedDate.ToString();
         Label2.Text = "Todays date:" +
Calendar1.TodaysDate.ToShortDateString();
```

```
            if (Calendar1.SelectedDate.ToShortDateString() ==
"07-09-2024")
            {
                Label3.Text = "Ganpati chaturti";
            }
        }
    }
}
```

Output:



< September 2024 >

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

Button

```
Your Selected Date:07-09-2024 00:00:00
Todays date:21-09-2024
Ganpati chaturti
```

c) Create simple application to perform following operations:

Code:

WebForm1.aspx

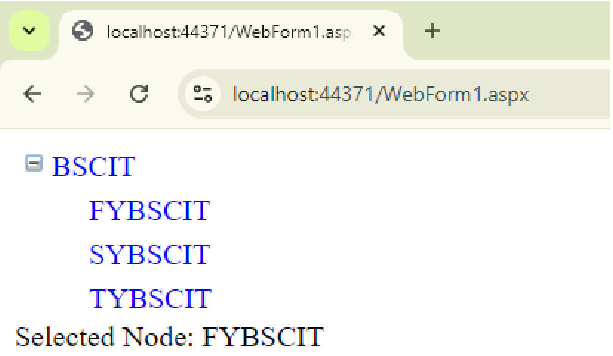
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Calender.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <asp:TreeView ID="TreeView1" runat="server"
OnSelectedNodeChanged="TreeView1_SelectedNodeChanged">

        <Nodes>
            <asp:TreeNode Text="BSCIT" Value="BSCIT">
                <asp:TreeNode Text="FYBSCIT"
Value="FYBSCIT"></asp:TreeNode>
                <asp:TreeNode Text="SYBSCIT"
Value="SYBSCIT"></asp:TreeNode>
```

```
                <asp:TreeNode Text="TYBSCIT"
Value="TYBSCIT"></asp:TreeNode>
            </asp:TreeNode>
        </Nodes>

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

WebForm1.aspx.cs

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

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

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void TreeView1_SelectedNodeChanged(object sender,
EventArgs e)
        {
            Label1.Text = "Selected Node: " +
TreeView1.SelectedNode.Text;
        }
    }
}
```

Output:



```
⊟ BSCIT
    FYBSCIT
    SYBSCIT
    TYBSCIT
Selected Node: FYBSCIT
```

Practical-4

a) Create a Registration form to demonstrate use of various Validation controls.

Code:

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical_4_Forms_.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Registration Form</h2>
            <table>
                <tr>
                    <td><asp:Label ID="Label1" runat="server"
Text="Username:"
AssociatedControlID="TextBoxUsername"></asp:Label></td>
                    <td><asp:TextBox ID="TextBoxUsername"
runat="server"></asp:TextBox></td>
                    <td><asp:RequiredFieldValidator
ID="RequiredFieldValidatorUsername" runat="server"
                        ControlToValidate="TextBoxUsername"
ErrorMessage="Username is required" ForeColor="Red" />
                    </td>
                </tr>

               <tr>
                    <td><asp:Label ID="Label2" runat="server" Text="Email:"
AssociatedControlID="TextBoxEmail"></asp:Label></td>
                    <td><asp:TextBox ID="TextBoxEmail"
runat="server"></asp:TextBox></td>
                    <td><asp:RequiredFieldValidator
ID="RequiredFieldValidatorEmail" runat="server"
                        ControlToValidate="TextBoxEmail"
ErrorMessage="Email is required" ForeColor="Red" />
                        <asp:RegularExpressionValidator
ID="RegularExpressionValidatorEmail" runat="server"
                        ControlToValidate="TextBoxEmail"
ErrorMessage="Invalid email format" ForeColor="Red"
```

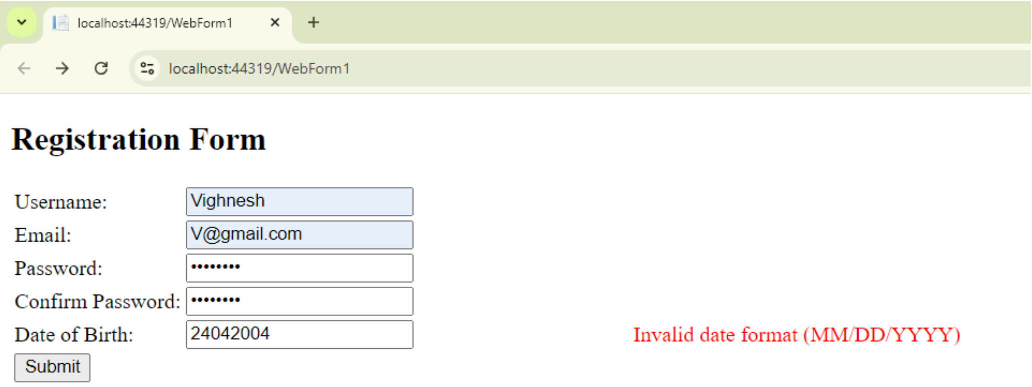
```
ValidationExpression="\w+([-+.’]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" />
          </td>
        </tr>

        <tr>
          <td><asp:Label ID="Label3" runat="server"
Text="Password:"
AssociatedControlID="TextBoxPassword"></asp:Label></td>
          <td><asp:TextBox ID="TextBoxPassword" runat="server"
TextMode="Password"></asp:TextBox></td>
          <td><asp:RequiredFieldValidator
ID="RequiredFieldValidatorPassword" runat="server"
            ControlToValidate="TextBoxPassword"
ErrorMessage="Password is required" ForeColor="Red" />
          </td>
        </tr>

        <tr>
          <td><asp:Label ID="Label4" runat="server"
Text="Confirm Password:"
AssociatedControlID="TextBoxConfirmPassword"></asp:Label></td>
          <td><asp:TextBox ID="TextBoxConfirmPassword"
runat="server" TextMode="Password"></asp:TextBox></td>
          <td><asp:RequiredFieldValidator
ID="RequiredFieldValidatorConfirmPassword" runat="server"
            ControlToValidate="TextBoxConfirmPassword"
ErrorMessage="Confirm Password is required" ForeColor="Red" />
            <asp:CompareValidator
ID="CompareValidatorPassword" runat="server"
            ControlToCompare="TextBoxPassword"
ControlToValidate="TextBoxConfirmPassword"
            ErrorMessage="Passwords do not match"
ForeColor="Red" />
          </td>
        </tr>

        <tr>
          <td><asp:Label ID="Label5" runat="server" Text="Date of
Birth:" AssociatedControlID="TextBoxDOB"></asp:Label></td>
          <td><asp:TextBox ID="TextBoxDOB"
runat="server"></asp:TextBox></td>
          <td><asp:RequiredFieldValidator
ID="RequiredFieldValidatorDOB" runat="server"
            ControlToValidate="TextBoxDOB" ErrorMessage="Date
of Birth is required" ForeColor="Red" />
            <asp:RegularExpressionValidator
ID="RegularExpressionValidatorDOB" runat="server"
```

```
      ControlToValidate="TextBoxDOB"
ErrorMessage="Invalid date format (MM/DD/YYYY)" ForeColor="Red"
      ValidationExpression="\d{2}/\d{2}/\d{4}" />
     </td>
    </tr>

    <tr>
     <td colspan="2">
      <asp:Button ID="ButtonSubmit" runat="server"
Text="Submit" OnClick="ButtonSubmit_Click" />
     </td>
    </tr>
   </table>
  </div>
 </form>
</body>
</html>
```

Default.aspx.cs:

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

namespace Practical_4_Forms_
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void ButtonSubmit_Click(object sender, EventArgs e)
        {
            Response.Write("Registration successful!");
        }
    }
}
```

Output:



Registration Form

Username: Vighnesh

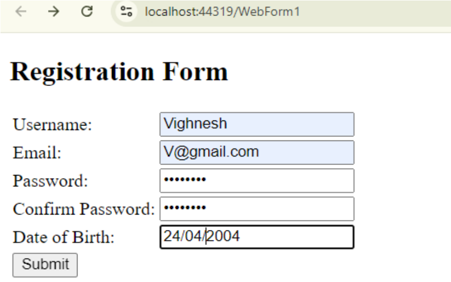
Email: V@gmail.com

Password: ••••••••

Confirm Password: ••••••••

Date of Birth: 24042004

Invalid date format (MM/DD/YYYY)



Registration Form

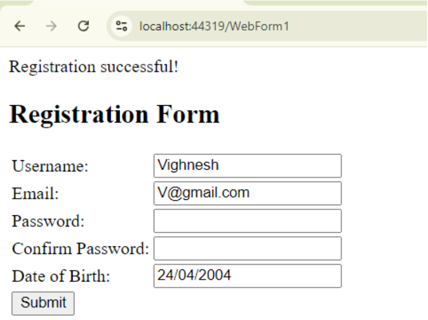
Username: Vighnesh

Email: V@gmail.com

Password: ••••••••

Confirm Password: ••••••••

Date of Birth: 24/04/2004



Registration successful!

Registration Form

Username: Vighnesh

Email: V@gmail.com

Password:

Confirm Password:

Date of Birth: 24/04/2004

b) Create Web Form to demonstrate use of Adrotator Control.

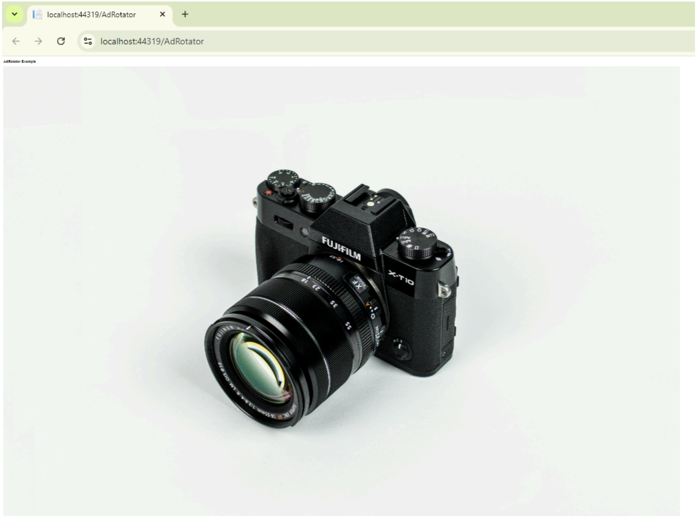
Code:
AdRotator.aspx

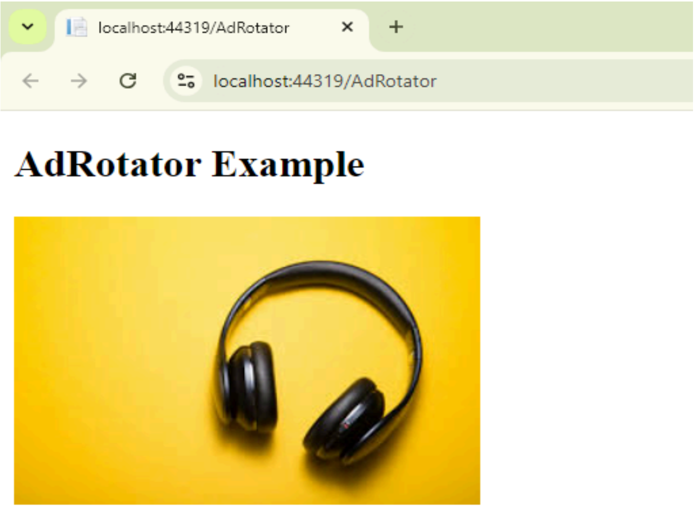
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="AdRotator.aspx.cs" Inherits="Practical_4_Forms_.AdRotator"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>AdRotator Example</h2>
            <asp:AdRotator ID="AdRotator1" runat="server"
AdvertisementFile="~/Ads.xml" />
        </div>
    </form>
</body>
</html>
```

Ads.xml

```
    <?xml version="1.0" encoding="utf-8" ?>
    <Advertisements>
        <Ad>
            <ImageUrl>~/Images/ad1.jpg</ImageUrl>

<NavigateUrl>http://www.example.com/product1</NavigateUrl>
            <AlternateText>Product 1</AlternateText>
            <Keyword>Product</Keyword>
            <Impressions>30</Impressions>
        </Ad>

        <Ad>
            <ImageUrl>~/Images/ad2.jpg</ImageUrl>

<NavigateUrl>http://www.example.com/product2</NavigateUrl>
            <AlternateText>Product 2</AlternateText>
            <Keyword>Product</Keyword>
            <Impressions>90</Impressions>
        </Ad>
    </Advertisements>
```

Output:





AdRotator Example

c) Create Web Form to demonstrate use User Controls

Code:

WebUserControl1.ascx

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="WebUserControl1.ascx.cs"
Inherits="Practical_4_Forms_.WebUserControl1" %>
<asp:Label ID="Label1" runat="server" Text="Enter Your
Name:"></asp:Label>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<p>
   <asp:Label ID="Label2" runat="server" Text="Enter Your
City:"></asp:Label>
   <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
</p>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Text="Button" />
<p>
   <asp:Label ID="Label3" runat="server"></asp:Label>
</p>
```

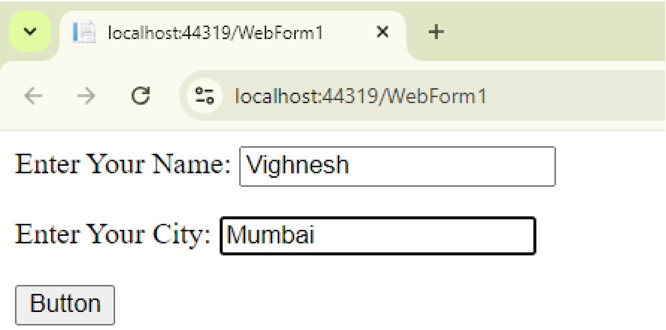
WebUserControl1.ascx.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;



namespace Practical_4_Forms_

{

   public partial class WebUserControl1 : System.Web.UI.UserControl

   {

      protected void Page_Load(object sender, EventArgs e)

      {
        }


        protected void Button1_Click(object sender, EventArgs e)
        {
            Label3.Text = "Your name is " + TextBox1.Text + " city is " +
TextBox2.Text;
        }
    }
}
```

WebForm1.aspx

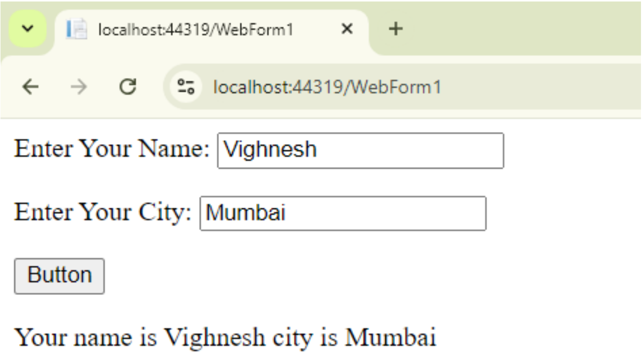
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical_4_Forms_.WebForm1" %>

<%@ Register src="WebUserControl1.ascx"
tagname="WebUserControl1" tagprefix="uc1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <uc1:WebUserControl1 ID="WebUserControl11"
runat="server" />
        </div>
    </form>
</body>
</html>
```

Output:



Enter Your Name: Vighnesh

Enter Your City: Mumbai



Enter Your Name: Vighnesh

Enter Your City: Mumbai

Your name is Vighnesh city is Mumbai

Practical-5

a) Create a Registration form to demonstrate use of various Validation controls.

Code:

Web.sitemap:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
        <siteMapNode url="WebForm2.aspx" title="Laptop"  description="">
                <siteMapNode url="WebForm3.aspx" title="Mobile" />
                <siteMapNode url="WebForm4.aspx" title="Accessories"/>
        </siteMapNode>
</siteMap>
```

Webform1.aspx:

```
        <%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication5.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:SiteMapPath ID="SiteMapPath1" runat="server">
                 </asp:SiteMapPath>
                 <br />
                 <asp:Menu ID="Menu1" runat="server"
DataSourceID="SiteMapDataSource1">
                 </asp:Menu>
                 <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server"/>
              </div>
           </form>
        </body>
        </html>
```

Webform2.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="WebApplication5.WebForm2"
%>

<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:SiteMapPath ID="SiteMapPath1" runat="server">
         </asp:SiteMapPath>
         <h1>This is Laptop</h1>
      </div>
   </form>
</body>
</html>
```

Webform3.aspx:

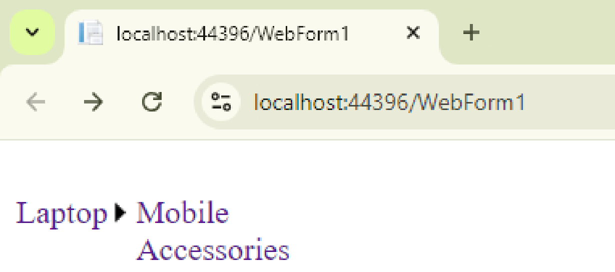
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs" Inherits="WebApplication5.WebForm3"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:SiteMapPath ID="SiteMapPath1" runat="server">
         </asp:SiteMapPath>
          <h1>This is Mobile</h1>
      </div>
   </form>
</body>
</html>
```

Webform4.aspx:

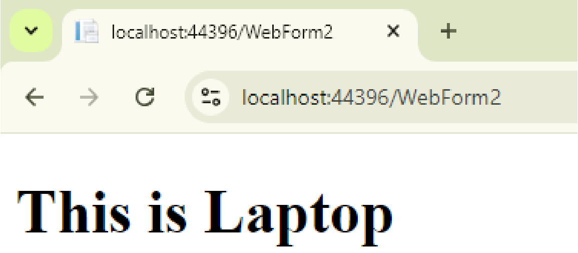
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm4.aspx.cs" Inherits="WebApplication5.WebForm4"
%>

<!DOCTYPE html>

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

```
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SiteMapPath ID="SiteMapPath1" runat="server">
            </asp:SiteMapPath>
             <h1>This is Accessories</h1>
        </div>
    </form>
</body>
</html>
```

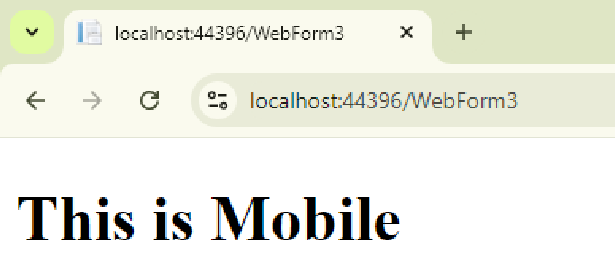
Output:



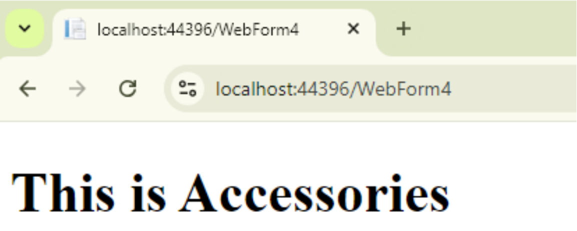
Laptop ▸ Mobile
Accessories



This is Laptop



This is Mobile



This is Accessories

b) Create a web application to demonstrate use of Master Page and content page.

Code:

Site.Master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="WebApplication4.SiteMaster" %>
<!DOCTYPE html>
<html>
<head>
   <title>My Web Application</title>
   <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <header>
            <h1>My Web Application</h1>
            <nav>
               <ul>
                  <li><a href="WebForm1.aspx">Page 1</a></li>
                  <li><a href="WebForm2.aspx">Page 2</a></li>
               </ul>
            </nav>
         </header>
         <asp:ContentPlaceHolder ID="MainContent"
runat="server"></asp:ContentPlaceHolder>
      </div>
      <footer>
         <p>&copy; 2024 My Web Application</p>
      </footer>
   </form>
</body>
</html>
```

WebForm1.aspx:

```
         <%@ Page Title="" Language="C#" MasterPageFile="~/Site.Mobile.Master"
AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication4.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
   <h2>Welcome Vighnesh</h2>
   <p>This is the content of the second page.</p>
</asp:Content>
```

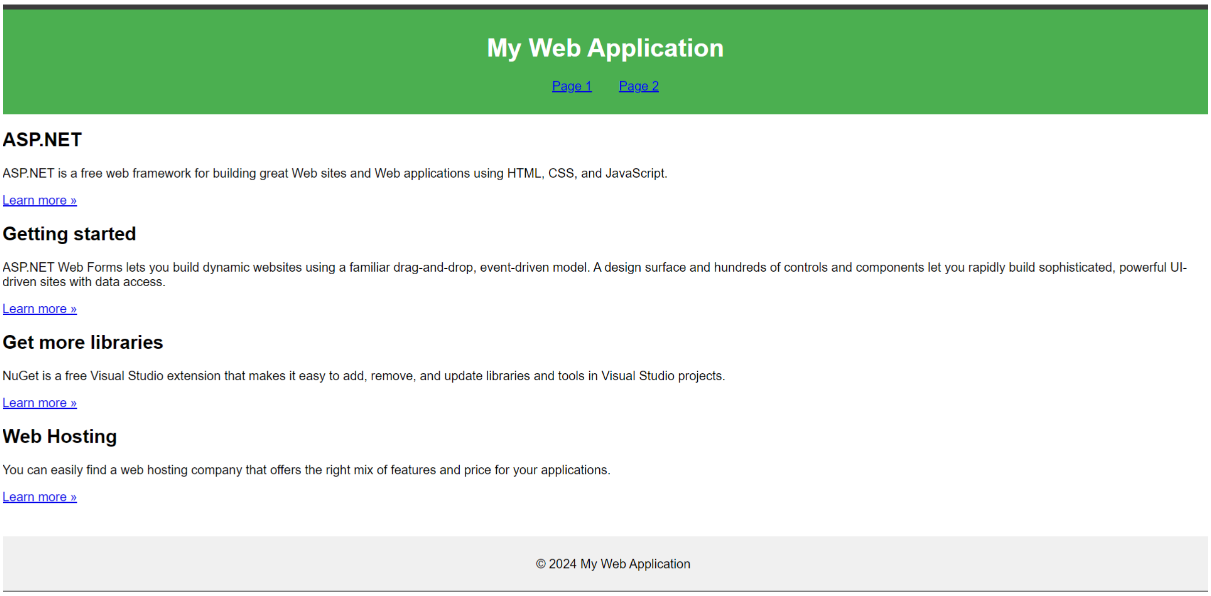
WebForm2.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication4.WebForm2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
runat="server">
  <h2>Welcome Vighnesh</h2>
  <p>This is the content of the second page.</p>
</asp:Content>
```

styles.css

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}

header {
  background-color: #4CAF50;
  color: white;
  padding: 10px;
  text-align: center;
}

nav ul {
  list-style-type: none;
  padding: 0;
}

  nav ul li {
    display: inline;
    margin: 0 15px;
  }

footer {
  background-color: #f1f1f1;
  text-align: center;
  padding: 10px;
  position: fixed;
  bottom: 0;
  width: 100%;
}
```

Output:



My Web Application

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Learn more »

Getting started

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.

Learn more »

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

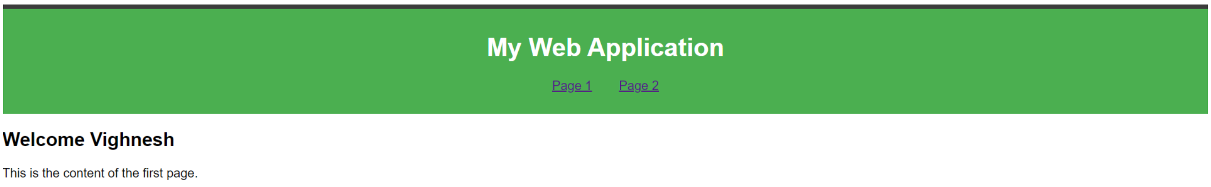
Learn more »

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

Learn more »

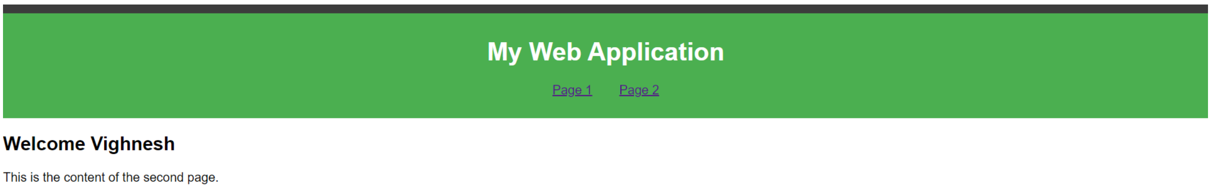
© 2024 My Web Application



My Web Application

Welcome Vighnesh

This is the content of the first page.



My Web Application

Welcome Vighnesh

This is the content of the second page.

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

Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Prac_5_c_.WebForm1" %>
<!DOCTYPE html>

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

        Practical-5[c(i)] Hidden Field<br />
        <br />
        Client-site state management<br />
        <asp:HiddenField ID="HiddenField1" runat="server" Value="3" />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            style="width: 61px" Text="Submit" />
    </div>
    </form>

</body>
</html>
```

WebForm1.aspx,cs:

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

namespace Prac_5_c_
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            if (HiddenField1.Value != null)
            {
                int val = Convert.ToInt32(HiddenField1.Value) + 1;
                HiddenField1.Value = val.ToString();
                Label1.Text = val.ToString();
            }
            int val1 = 1;
            Label2.Text = (val1 + 1).ToString();
        }
    }
}
```

WebForm2.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="Prac_5_c_.WebForm2" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      Practical-5[c(ii)] Query String<br />
      <br />
      <asp:Label ID="Label1" runat="server" Text="First Name"></asp:Label>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <br />
      <br />
      <asp:Label ID="Label2" runat="server" Text="Last Name"></asp:Label>
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      <br />
      <br />
      <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
         style="height: 26px" Text="Submit" />
   </div>
   </form>
</body>
</html>
```

WebForm2.aspx,cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Prac_5_c_
{
   public partial class WebForm2 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
         {}
      protected void Button1_Click(object sender, EventArgs e)
      {
         Response.Redirect("WebForm3.aspx?Nm=" + TextBox1.Text + "
&Nm1=" + TextBox2.Text);
      }
   }
}
```

WebForm3.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs" Inherits="Prac_5_c_.WebForm3" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
         <asp:Label ID="Label1" runat="server" Font-Bold="True"
Font-Size="XX-Large"
      ForeColor="Red" Text="Welcome" Width="700px">
      </asp:Label>
   </form>
</body>
</html>
```

WebForm3.aspx,cs:

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

namespace Prac_5_c_
{
   public partial class WebForm3 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (Request.QueryString["Nm"] != null)
         {
            Label1.Text = Label1.Text + ".........." + Request.QueryString["Nm"]+
" " + Request.QueryString["Nm1"];
         }
         else
         {
            Label1.Text = "Some problem occured";
         }

      }
   }
}
```

WebForm4.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm4.aspx.cs" Inherits="Prac_5_c_.WebForm4" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
      #bodytag
      {
         font-family: Calibri;
         }
   </style>
</head>
<body id="bodytag" runat="server">
   <form id="form1" runat="server">
   <div>
      <asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="True"
         onselectedindexchanged="DropDownList1_SelectedIndexChanged"
         ForeColor="Black" Height="60px"
         Width="120px">
         <asp:ListItem>Red</asp:ListItem>
         <asp:ListItem>Green</asp:ListItem>
         <asp:ListItem>Blue</asp:ListItem>
         <asp:ListItem>Pink</asp:ListItem>
         <asp:ListItem>yellow</asp:ListItem>
      </asp:DropDownList>
   </div>
   </form>
</body>
</html>
```

WebForm4.aspx,cs:

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

namespace Prac_5_c_
{
   public partial class WebForm4 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (Request.Cookies["BGC"] != null)
         {
            DropDownList1.SelectedValue = Request.Cookies["BGC"].Value;
            bodytag.Style["background-color"] = DropDownList1.SelectedValue;
```

```
                DropDownList1.Visible = true;
            }

        }

        protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            HttpCookie obj = new HttpCookie("BGC");
            obj.Value = DropDownList1.SelectedValue;
            obj.Expires = DateTime.Now.AddSeconds(4);
            Response.SetCookie(obj);

        }
    }
}
```

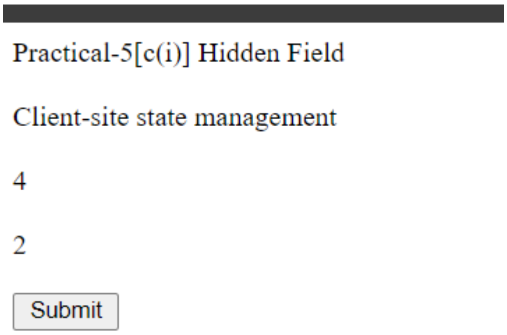
WebForm5.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm5.aspx.cs" Inherits="Prac_5_c_.WebForm5" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
     <style type="text/css">
        .style3
        {
            font-weight: normal;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div style="font-family: Calibri">
        <span class="style3">Practical5[c(d)]View State<br />
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
&nbsp;&nbsp;&nbsp;
        </span>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            Text="Get Data" style="height: 26px" />
    </div>
    </form>
</body>
</html>
```

WebForm5.aspx,cs:

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

namespace Prac_5_c_
{
    public partial class WebForm5 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                string str = "Vighnesh Chejara";
                if (ViewState["nam"] == null)
                {
                    ViewState["nam"] = str;
                }
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = ViewState["nam"].ToString();
        }
    }
}
```

Output:

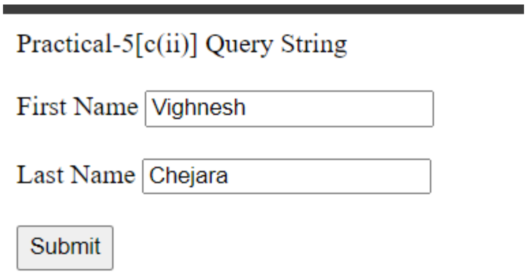


Practical-5[c(i)] Hidden Field

Client-site state management

4

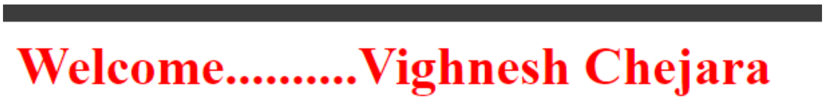
2



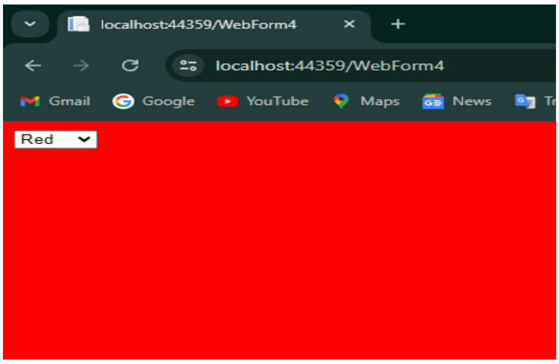
Practical-5[c(ii)] Query String

First Name Vighnesh

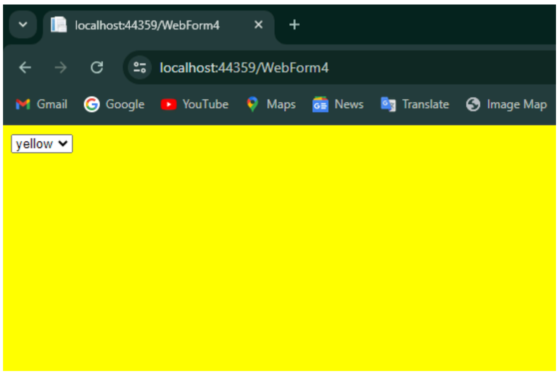
Last Name Chejara



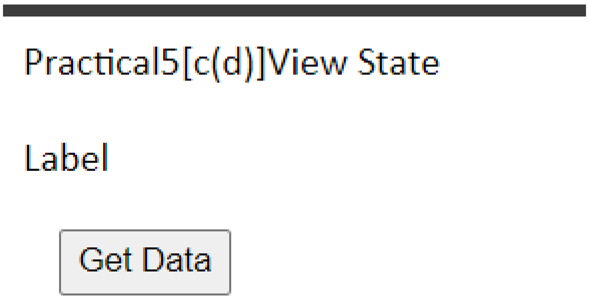
Welcome..........Vighnesh Chejara



Red

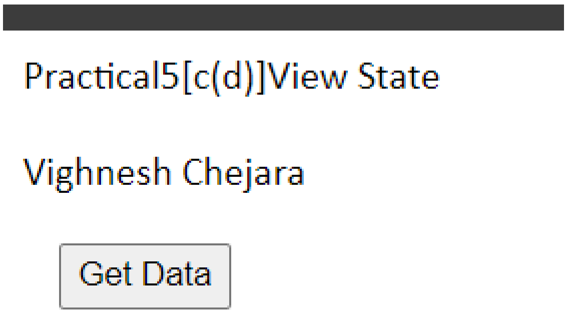


yellow



Practical5[c(d)]View State

Label



Practical5[c(d)]View State

Vighnesh Chejara

Practical-6

a) Create a web application for inserting and deleting records from a database.

Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication8.WebForm1"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
    <form id="form1" runat="server">
      <div>
         <h1>Student Details</h1>
         <asp:Label ID="Label1" runat="server" Text="Student
ID:"></asp:Label>
         <asp:TextBox ID="TextBox1" runat="server"
OnTextChanged="TextBox1_TextChanged"></asp:TextBox><br/>

         <asp:Label ID="Label2" runat="server" Text="Student
Name:"></asp:Label>
         <asp:TextBox ID="TextBox2" runat="server"
OnTextChanged="TextBox2_TextChanged"></asp:TextBox><br/>

         <asp:Label ID="Label3" runat="server" Text="Student
Class:"></asp:Label>
         <asp:TextBox ID="TextBox3" runat="server"
OnTextChanged="TextBox3_TextChanged"></asp:TextBox>
         <br />
         <br />
         <br/>
         <asp:Button ID="Button1" runat="server" Text="Insert"
OnClick="Button1_Click" />
         <asp:Button ID="Button2" runat="server" Text="Delete"
OnClick="Button2_Click" />
      </div>
   </form>
</body>
</html>
```

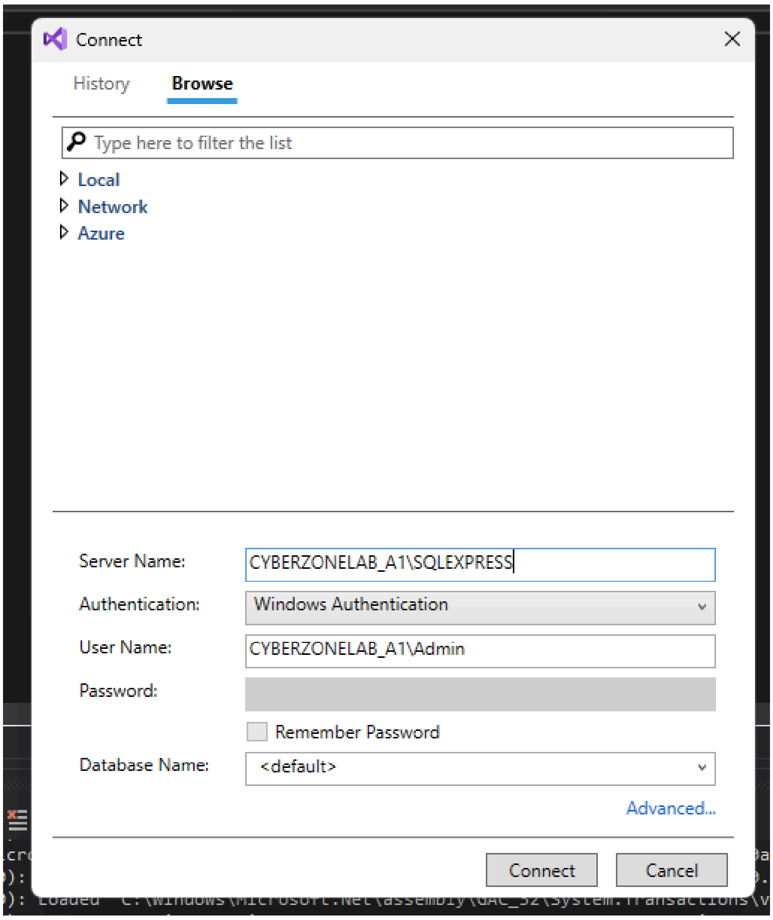
WebForm1.aspx.cs:

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

namespace WebApplication8
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data
Source=CYBERZONELAB_A1\SQLEXPRESS;Initial
Catalog=Students;Integrated
Security=True;Encrypt=True;TrustServerCertificate=True;");
            con.Open();

            // Corrected the SQL query and parameters
            SqlCommand cmd = new SqlCommand("insert into student_det
values(@std_id,@std_name,@std_class)", con);
            cmd.Parameters.AddWithValue("@std_id", int.Parse(TextBox1.Text));
            cmd.Parameters.AddWithValue("@std_name", TextBox2.Text);
            cmd.Parameters.AddWithValue("@std_class", TextBox3.Text);

            cmd.ExecuteNonQuery();

            con.Close();
            Response.Write("Data inserted successfully");
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data
Source=CYBERZONELAB_A1\SQLEXPRESS;Initial
Catalog=Students;Integrated
Security=True;Encrypt=True;TrustServerCertificate=True;");
            con.Open();
            // Delete the record where std_id matches
            SqlCommand cmd = new SqlCommand("DELETE FROM student_det
WHERE std_id = @std_id", con);
            cmd.Parameters.AddWithValue("@std_id", int.Parse(TextBox1.Text));

            cmd.ExecuteNonQuery();

            con.Close();
            Response.Write("Record deleted successfully");
        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {

        }

        protected void TextBox2_TextChanged(object sender, EventArgs e)
        {

        }

        protected void TextBox3_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

SQLQuery1.sql:

```
Create Database Students
Use Students

create table student_det(std_id INT PRIMARY KEY,std_name
VARCHAR(100),std_class VARCHAR(50));

Select * from student_det
```

Output:



Server Name: CYBERZONELAB_A1\SQLEXPRESS

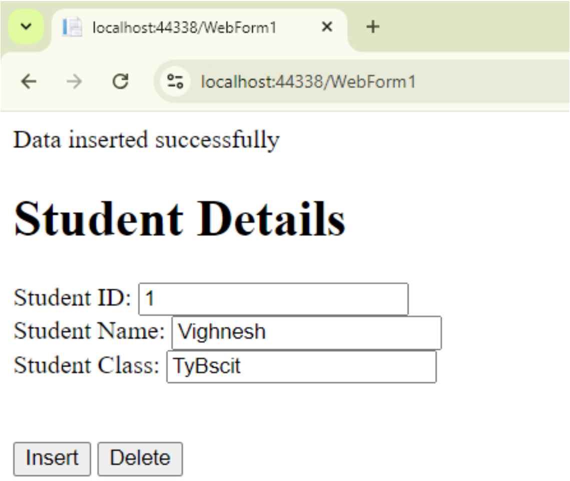
Authentication: Windows Authentication

User Name: CYBERZONELAB_A1\Admin

Password:

□ Remember Password

Database Name: <default>



localhost:44338/WebForm1

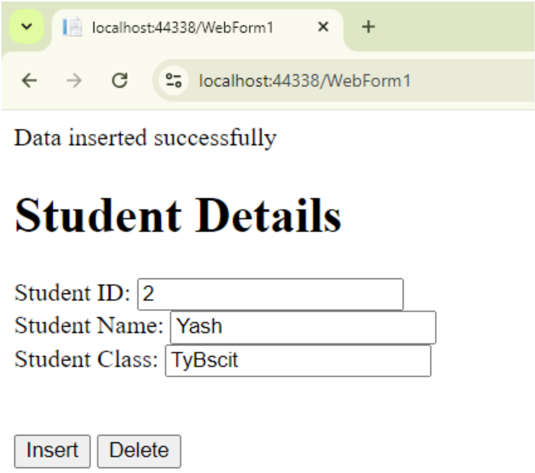
Data inserted successfully

Student Details

Student ID: 1

Student Name: Vighnesh

Student Class: TyBscit



localhost:44338/WebForm1

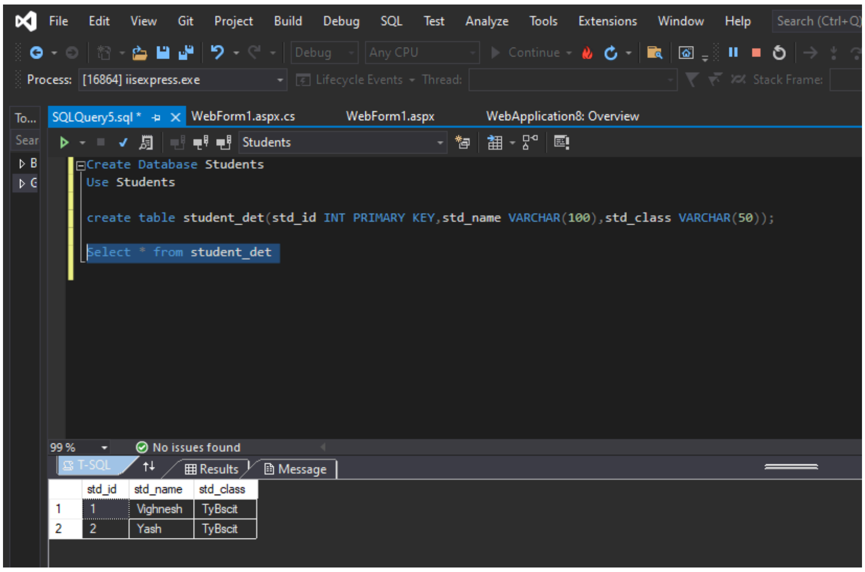
Data inserted successfully

Student Details

Student ID: 2

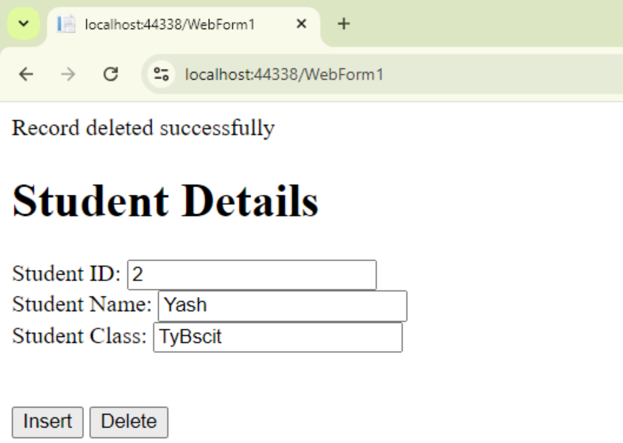
Student Name: Yash

Student Class: TyBscit



```
Create Database Students
Use Students

create table student_det(std_id INT PRIMARY KEY,std_name VARCHAR(100),std_class VARCHAR(50));

Select * from student_det
```

	std_id	std_name	std_class
1	1	Vighnesh	TyBscit
2	2	Yash	TyBscit



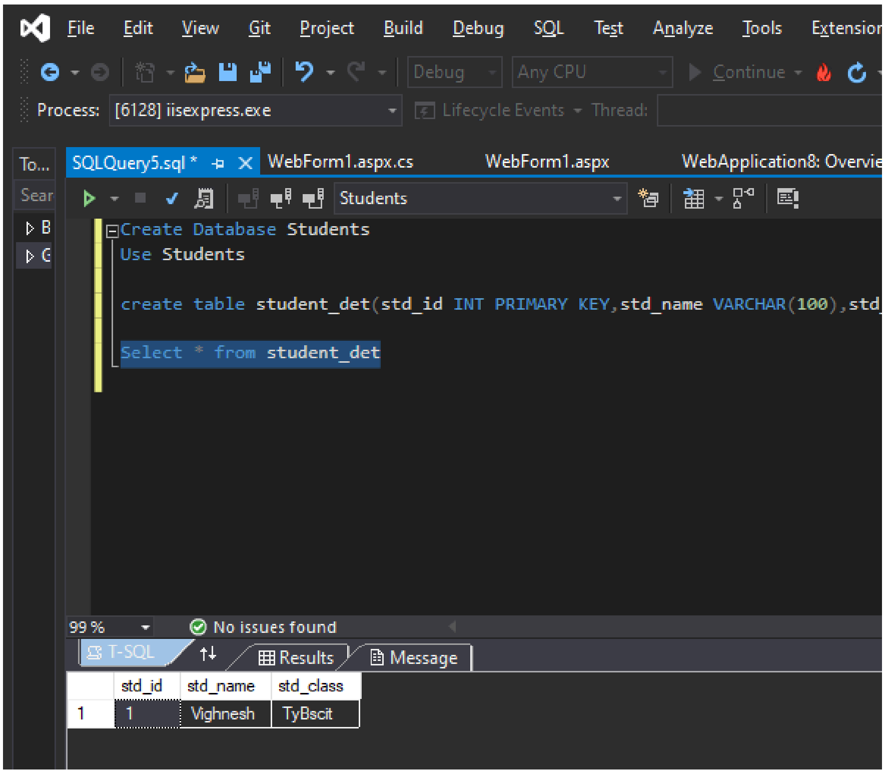
Record deleted successfully

Student Details

Student ID: 2

Student Name: Yash

Student Class: TyBscit



```
Create Database Students
Use Students

create table student_det(std_id INT PRIMARY KEY,std_name VARCHAR(100),std_

Select * from student_det
```

	std_id	std_name	std_class
1	1	Vighnesh	TyBscit

b) Create a web application to display Using Disconnected Data Access and Databinding using GridView.

Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Prac_6_b_.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <span style="color: rgb(0, 0, 0); font-family: &quot;Times New Roman&quot;; font-size: medium; font-style: normal; font-variant-ligatures: normal; font-variant-caps: normal; font-weight: 400; letter-spacing: normal; orphans: 2; text-align: start; text-indent: 0px; text-transform: none; widows: 2; word-spacing: 0px; -webkit-text-stroke-width: 0px; white-space: normal; text-decoration-thickness: initial; text-decoration-style: initial; text-decoration-color: initial; display: inline !important; float: none;">
      Vighnesh Chejara&nbsp;&nbsp;&nbsp;A-09</span><br />
      <br />
      <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
         Text="Click to see the data of employee table" />
      <br />
      <br />
      <asp:GridView ID="GridView1" runat="server">
      </asp:GridView>
   </div>
   </form>
</body>
</html>
```

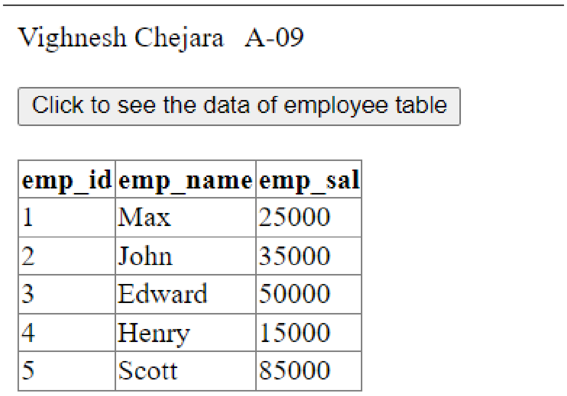
WebForm1.aspx.cs:

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

namespace Prac_6_b_
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        static string str = @"Data
Source=LAPTOP-B6OIMLRU\SQLEXPRESS10;Initial
Catalog=employee;Integrated Security=True;Pooling=False";
        SqlConnection con = new SqlConnection(str);
        SqlDataAdapter da;

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            con.Open();
            da = new SqlDataAdapter("select * from emp", con);
            DataSet ds = new DataSet();
            da.Fill(ds);
            GridView1.DataSource = ds.Tables[0];
            GridView1.DataBind();
            con.Close();

        }
    }
}
```

SQLQuery1.sql:

```
Create database employee
Use employee

create table emp(emp_id int,emp_name varchar(10),emp_sal int)
insert into emp values(1,'Max',25000)
insert into emp values(2,'John',35000)
insert into emp values(3,'Edward',50000)
insert into emp values(4,'Henry',15000)
insert into emp values(5,'Scott',85000)
select * from emp;
```

Output:



Vighnesh Chejara A-09

emp_id	emp_name	emp_sal
1	Max	25000
2	John	35000
3	Edward	50000
4	Henry	15000
5	Scott	85000

Practical-7

a) Create a web application to demonstrate the use of different types of Cookies

Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Prac_7_a_.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Cookie Demo in C#</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <h2>Cookie Management Demo</h2>
    <asp:Button ID="btnCreateSessionCookie" runat="server"
Text="Create Session Cookie" OnClick="btnCreateSessionCookie_Click" />
    <asp:Button ID="btnCreatePersistentCookie" runat="server"
Text="Create Persistent Cookie" OnClick="btnCreatePersistentCookie_Click"
/>
    <asp:Button ID="btnReadCookies" runat="server" Text="Read
Cookies" OnClick="btnReadCookies_Click" style="height: 29px" />
    <asp:Button ID="btnDeleteCookies" runat="server" Text="Delete
Cookies" OnClick="btnDeleteCookies_Click" />
    <br /><br />
    <asp:Label ID="lblMessage" runat="server" Text="" />
  </div>
  </form>
</body>
</html>
```

WebForm1.aspx.cs:

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

namespace Prac_7_a_
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }
```

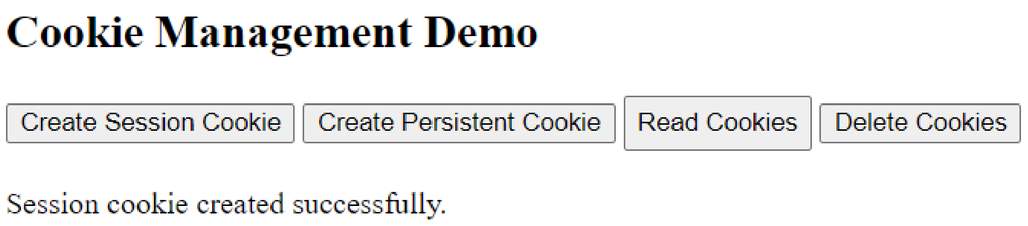
```
          protected void btnCreateSessionCookie_Click(object sender,
EventArgs e)
          {
            HttpCookie sessionCookie = new
HttpCookie("SessionCookie", "This is a session cookie");
            Response.Cookies.Add(sessionCookie);
            lblMessage.Text = "Session cookie created successfully.";
          }

          protected void btnCreatePersistentCookie_Click(object sender,
EventArgs e)
          {
            HttpCookie persistentCookie = new
HttpCookie("PersistentCookie", "This is a persistent cookie");
            persistentCookie.Expires = DateTime.Now.AddMinutes(1); //
Cookie will expire in 1 minute
            Response.Cookies.Add(persistentCookie);
            lblMessage.Text = "Persistent cookie created successfully and
will expire in 1 minute.";
          }

          protected void btnReadCookies_Click(object sender, EventArgs
e)
          {
            string message = "";

            if (Request.Cookies["SessionCookie"] != null)
            {
              message += "Session Cookie Value: " +
Request.Cookies["SessionCookie"].Value + "<br/>";
            }
            else
            {
              message += "Session Cookie does not exist.<br/>";
            }
            if (Request.Cookies["PersistentCookie"] != null)
            {
              message += "Persistent Cookie Value: " +
Request.Cookies["PersistentCookie"].Value + "<br/>";
            }
            else
            {
              message += "Persistent Cookie does not exist.<br/>";
            }
            lblMessage.Text = message;
          }
```

```
                    protected void btnDeleteCookies_Click(object sender, EventArgs
e)
                    {
                        if (Request.Cookies["SessionCookie"] != null)
                        {
                            HttpCookie sessionCookie = new
HttpCookie("SessionCookie");
                            sessionCookie.Expires = DateTime.Now.AddDays(-1); //
Setting the expiry date in the past will delete the cookie
                            Response.Cookies.Add(sessionCookie);
                        }
                        if (Request.Cookies["PersistentCookie"] != null)
                        {
                            HttpCookie persistentCookie = new
HttpCookie("PersistentCookie");
                            persistentCookie.Expires = DateTime.Now.AddDays(-1); //
Expire the cookie
                            Response.Cookies.Add(persistentCookie);
                        }
                        lblMessage.Text = "Both session and persistent cookies have
been deleted.";

                    }
                }
            }
```

Web1.config:

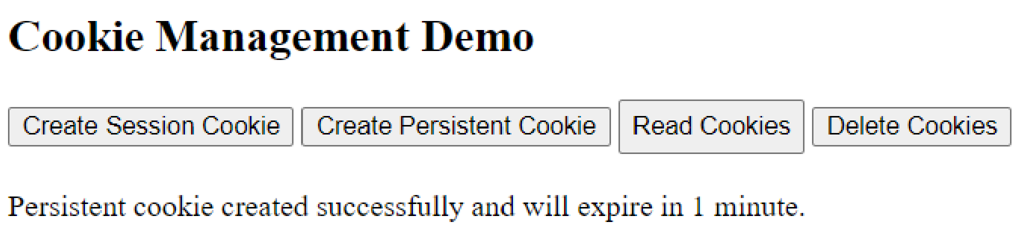
```
            <?xml version="1.0"?>
            <configuration>

                        <system.web>
                                    <compilation debug="false" targetFramework="4.0" />
                        </system.web>

            </configuration>
```

Output:



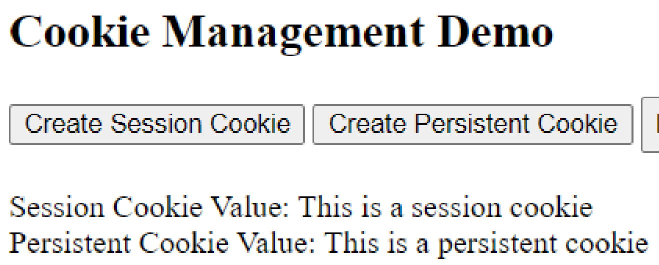
Cookie Management Demo

Session cookie created successfully.



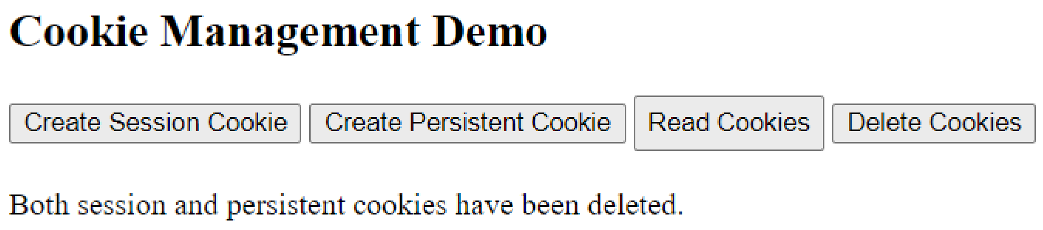
Cookie Management Demo

Persistent cookie created successfully and will expire in 1 minute.



Cookie Management Demo

Session Cookie Value: This is a session cookie
Persistent Cookie Value: This is a persistent cookie



Cookie Management Demo

Both session and persistent cookies have been deleted.

b) Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization properties

Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Prac_7_b_.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
```

```
                Vighnesh&nbsp;&nbsp;&nbsp; 09<br />
            <br />
            user name:
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <br />
            password:
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="login" />
            <br />
            <asp:CheckBox ID="CheckBox1" runat="server"
OnCheckedChanged="CheckBox1_CheckedChanged" />
            Check here if this is not&nbsp; a public computer<br />
            <br />
            <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
            </div>
         </form>
       </body>
       </html>
```

WebForm1.aspx.cs:

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


       namespace Prac_7_b_
       {
          public partial class WebForm1 : System.Web.UI.Page
          {
             protected void Page_Load(object sender, EventArgs e)
             {

             }
             protected bool authenticate(String uname, String pass)
             {
                if (uname == "Vighnesh")
                {
                   if (pass == "Chejara")
```

```
                    return true;
                }
                if (uname == "Yash")
                {
                    if (pass == "Shah")
                        return true;
                }
                if (uname == "Aman")
                {
                    if (pass == "Singh")
                        return true;
                }
                return false;
            }

            protected void CheckBox1_CheckedChanged(object sender,
EventArgs e)
            {

            }

            protected void Button1_Click(object sender, EventArgs e)
            {
                if (authenticate(TextBox1.Text, TextBox2.Text))
                {

FormsAuthentication.RedirectFromLoginPage(TextBox1.Text,
CheckBox1.Checked);
                    Session["Username"] = TextBox1.Text;
                    Response.Redirect("WebForm2.aspx");
                }
                else
                {
                    Response.Write("Invalid user name or password");
                }

            }
        }
    }
```

WebForm2.aspx:

```
        <%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="Prac_7_b_.WebForm2" %>

        <!DOCTYPE html>

        <html xmlns="http://www.w3.org/1999/xhtml">
        <head runat="server">
            <title></title>
        </head>
        <body>
            <form id="form1" runat="server">
                <div>
                      Welcome....&nbsp;&nbsp;&nbsp;
                <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
                </div>
            </form>
        </body>
        </html>
```

WebForm2.aspx.cs:

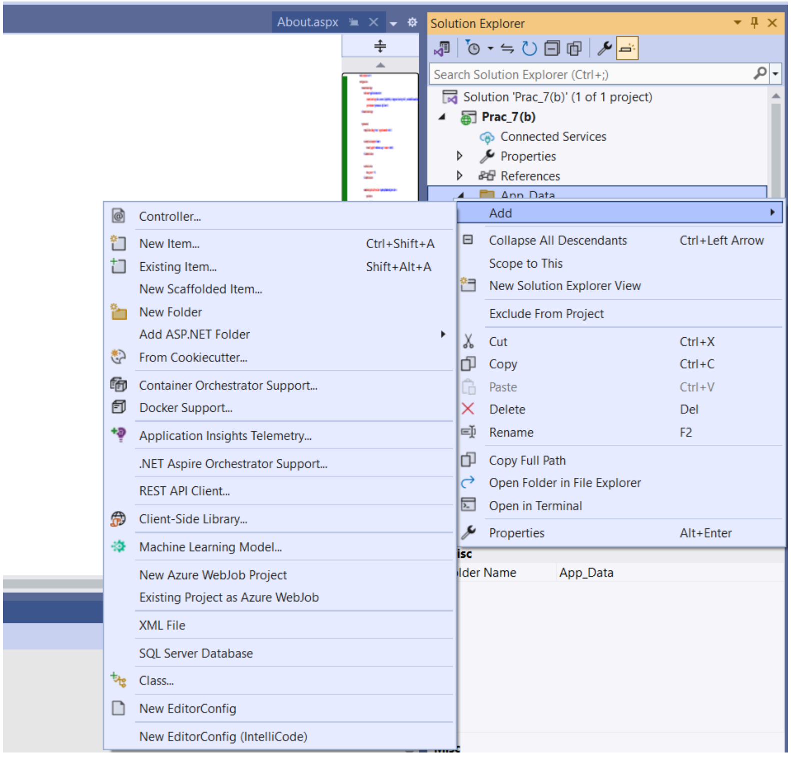
```
        using System;
        using System.Collections.Generic;
        using System.Linq;
        using System.Web;
        using System.Web.UI;
        using System.Web.UI.WebControls;

        namespace Prac_7_b_
        {
            public partial class WebForm2 : System.Web.UI.Page
            {
                protected void Page_Load(object sender, EventArgs e)
                {
                    if (Session["username"] != null)
                    {
                        Label1.Text = Session["username"].ToString();
                    }

                }
            }
        }
```

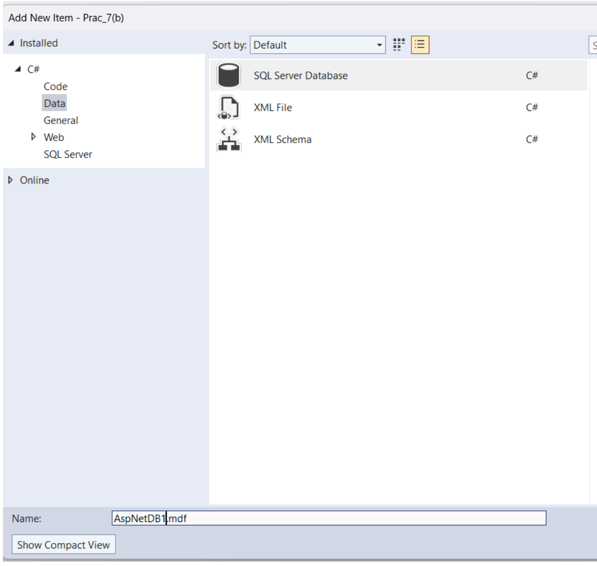
Web1.config:

```
<?xml version="1.0"?>
<configuration>
        <connectionStrings>
                <add name="ApplicationServices"
                            connectionString="data source=.\SQLEXPRESS;
Integrated Security=SSPI; AttachDbFilename=|DataDirectory|\AspNetDB1.mdf"
                            providerName="System.Data.SqlClient"/>
        </connectionStrings>
        <system.web>
                <compilation debug="true" targetFramework="4.0"/>

                <authentication mode="Forms">
                    <forms loginUrl="WebForm1.aspx" timeout="2880"/>
                </authentication>

                <authorization>
                    <deny users="?"/>
                </authorization>

                <membership
defaultProvider="AspNetSqlMembershipProvider">
                    <providers>
                        <clear/>
                        <add name="AspNetSqlMembershipProvider"

type="System.Web.Security.SqlMembershipProvider"

connectionStringName="ApplicationServices"
                        applicationName="/"
        />
                    </providers>
                </membership>

                <profile defaultProvider="AspNetSqlProfileProvider">
                    <providers>
                        <clear/>
                        <add name="AspNetSqlProfileProvider"

type="System.Web.Profile.SqlProfileProvider"

connectionStringName="ApplicationServices"
                        applicationName="/"
        />
                    </providers>
                </profile>

                <roleManager enabled="false">
                    <providers>
                        <clear/>
                        <add name="AspNetSqlRoleProvider"

type="System.Web.Security.SqlRoleProvider"

connectionStringName="ApplicationServices"
                        applicationName="/"
        />
```

```
            </providers>
        </roleManager>
    </system.web>
</configuration>
```

Output:





Sort by: Default

Name: AspNetDB1.mdf

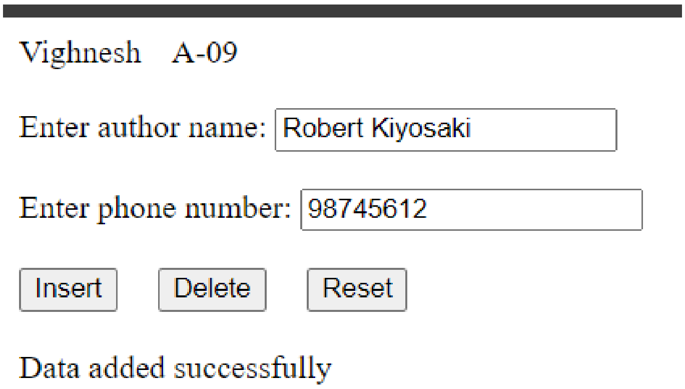
Practical-8

a) Create a web application for inserting and deleting records from a database. (Using Execute-Non Query).

Code:

Default.aspx:

Output:

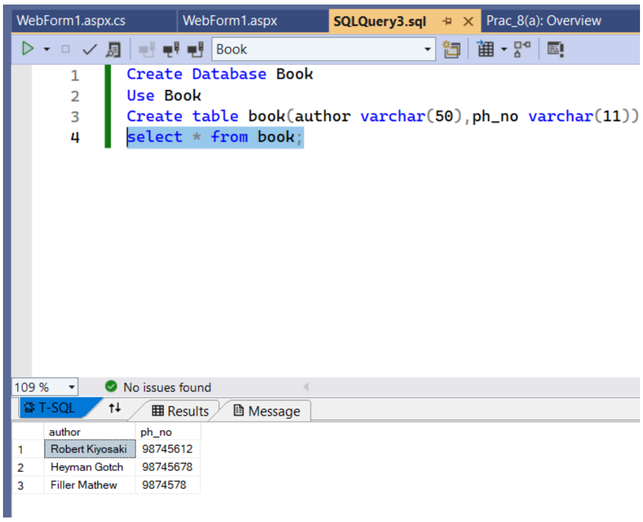


Vighnesh A-09

Enter author name: Robert Kiyosaki

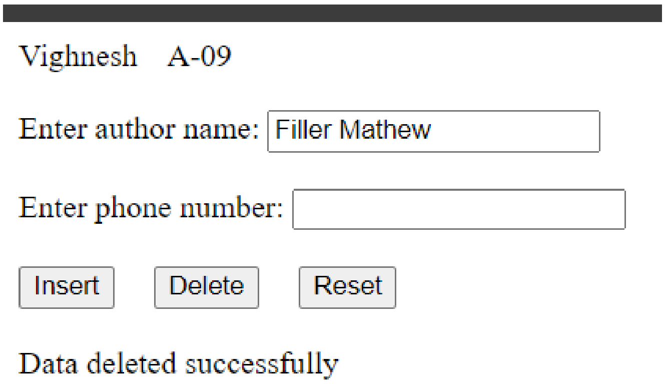
Enter phone number: 98745612

Data added successfully



```
Create Database Book
Use Book
Create table book(author varchar(50),ph_no varchar(11))
select * from book;
```

	author	ph_no
1	Robert Kiyosaki	98745612
2	Heyman Gotch	98745678
3	Filler Mathew	9874578

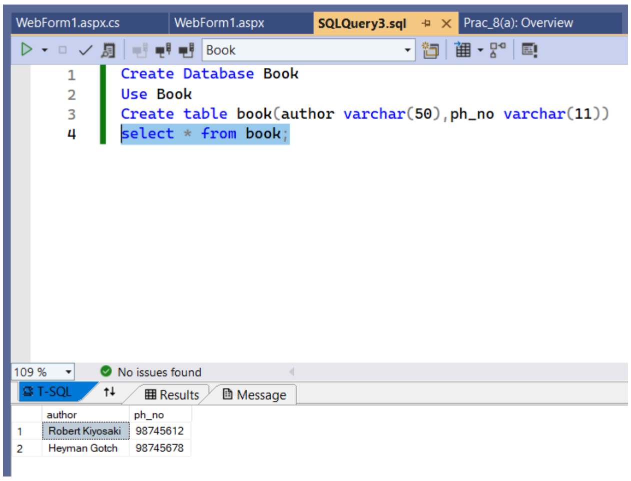


Vighnesh A-09

Enter author name: Filler Mathew

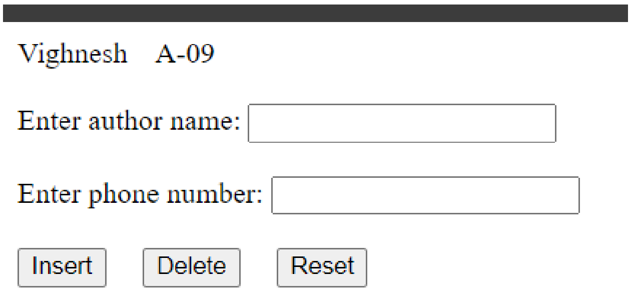
Enter phone number:

Data deleted successfully



```
Create Database Book
Use Book
Create table book(author varchar(50),ph_no varchar(11))
select * from book;
```

	author	ph_no
1	Robert Kiyosaki	98745612
2	Heyman Gotch	98745678



Vighnesh A-09

Enter author name:

Enter phone number:

b) Create a web application for user defined exception handling.

Code:

WebForm1.aspx:

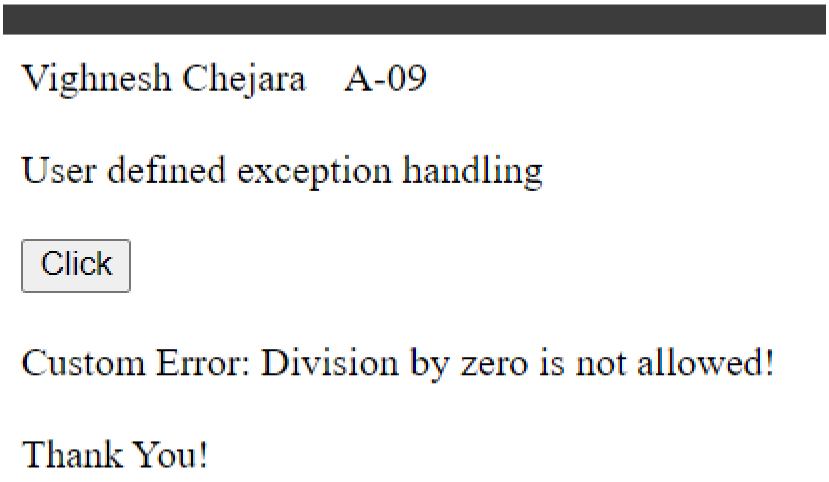
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Prac_8_b_.WebForm1"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
      <!--8b. Create a web application for user defined exception
handling.-->
      Vighnesh Chejara&nbsp;&nbsp;&nbsp; A-09<br />
      <br />
      User defined exception handling<br />
      <br />
      <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Click" />
      <br />
      <br />
      <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label2" runat="server"
Text="Label"></asp:Label>

   </div>

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

WebForm1.aspx.cs:

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

```
namespace Prac_8_b_
{
    public class CustomException : Exception
    {
        public CustomException(string message) : base(message)
        {
        }
    }

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

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            try
            {
                int a, b, c;
                a = 10;
                b = 0;
                if (b == 0)
                {
                    throw new CustomException("Custom Error: Division by
zero is not allowed!");
                }
                c = a / b;
                Label1.Text = c.ToString();
            }
            catch (CustomException ex)
            {
                Label1.Text = ex.Message;
            }
            catch (System.Exception ex)
            {
                Label1.Text = "System Exception: " + ex.Message;
            }
            finally
            {
                Label2.Text = "Thank You!";
            }

        }
    }
}
```

Output:



Vighnesh Chejara   A-09

User defined exception handling

Custom Error: Division by zero is not allowed!

Thank You!