# Practical-1

1. 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}");

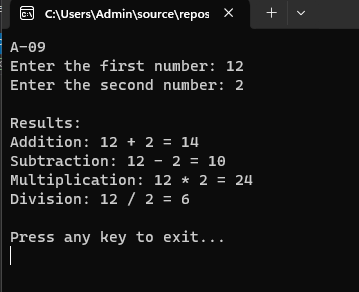
Console.WriteLine("\nPress any key to exit..."); Console.ReadKey();

}

}

}

## Output:



1. 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:

1. 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;

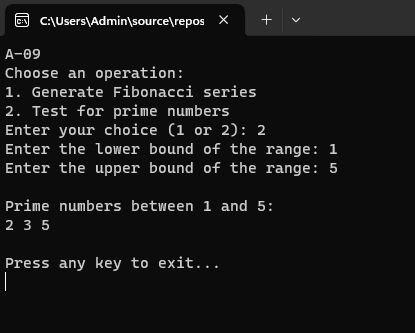
}

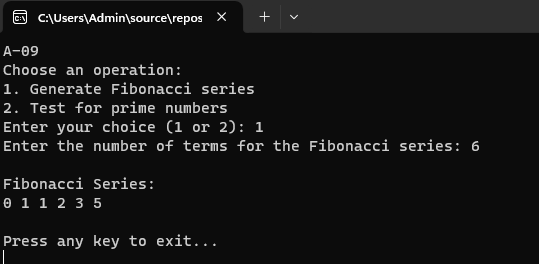
}

}

## Output:

return true;

}



# Practical-2

1. 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}");

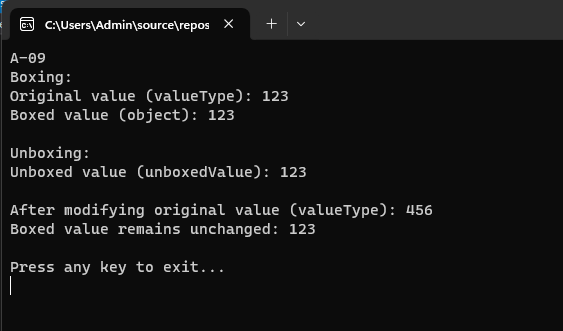
// Wait for the user to press a key before closing the console Console.WriteLine("\nPress any key to exit..."); Console.ReadKey();

}

}

}

## Output:



1. 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)

{

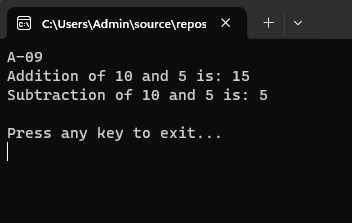
return x - y;

}

}

}

## Output:



1. 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()}");

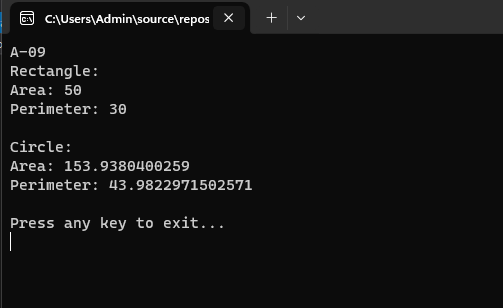
// Wait for the user to press a key before closing the console Console.WriteLine("\nPress any key to exit..."); Console.ReadKey();

}

}

}

**Output:**



# Practical-3

1. 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 =

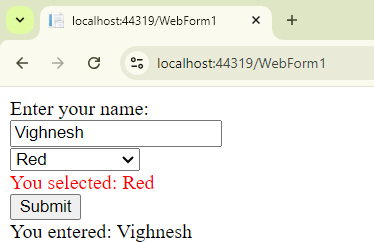
System.Drawing.Color.FromName(DropDownList1.SelectedItem.Text);

}

}

}

## Output:



1. 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";

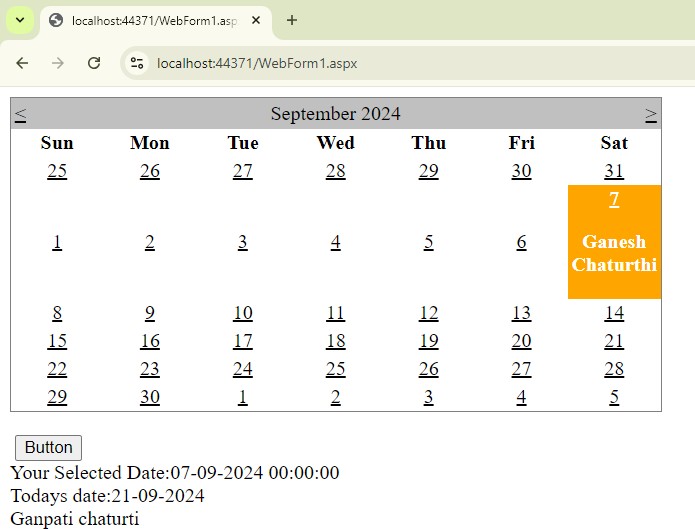
}

}

}

}

## Output:



1. 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)

{

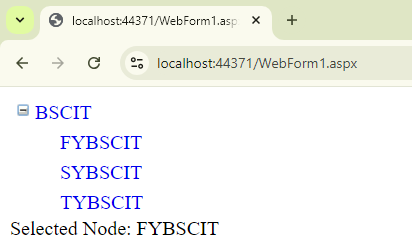
Label1.Text = "Selected Node: " + TreeView1.SelectedNode.Text;

}

}

}

**Output:**



# Practical-4

1. 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)

{

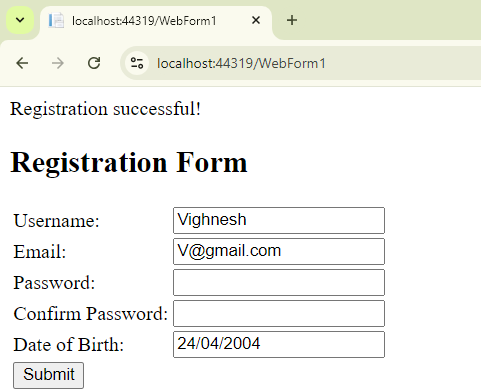
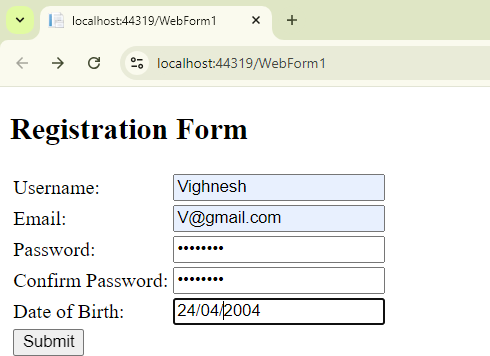
Response.Write("Registration successful!");

}

}

}

## Output:



1. 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>

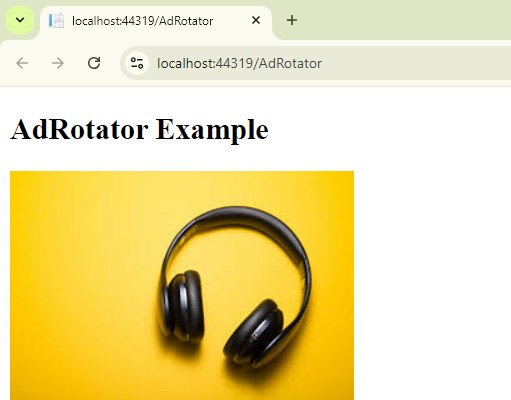
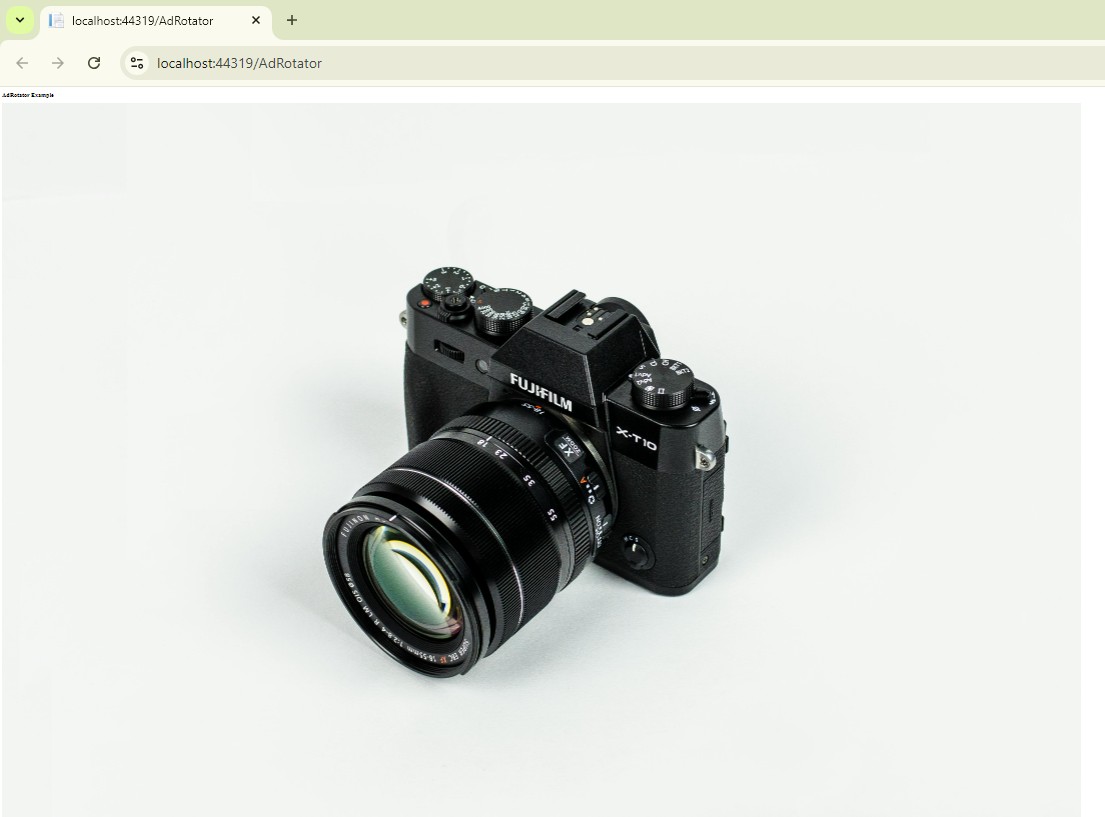
<Keyword>Product</Keyword>

<Impressions>90</Impressions>

</Ad>

</Advertisements>

## Output:



1. 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>

## Output:

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

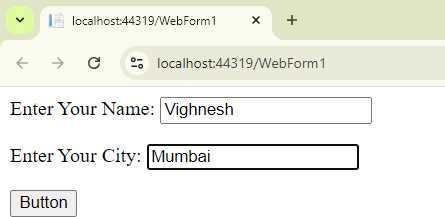
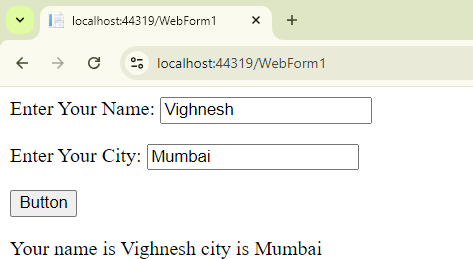
<uc1:WebUserControl1 ID="WebUserControl11" runat="server" />

</div>

</form>

</body>

</html>

# Practical-5

1. 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>

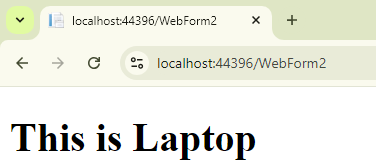
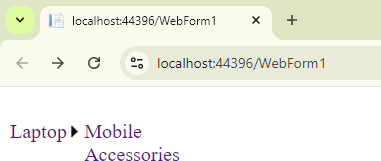
</div>

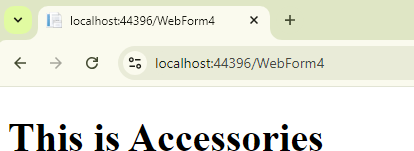
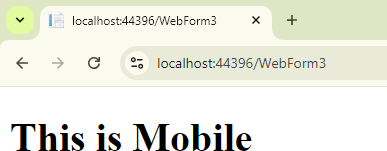
</form>

</body>

</html>

## Output:





1. 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 {

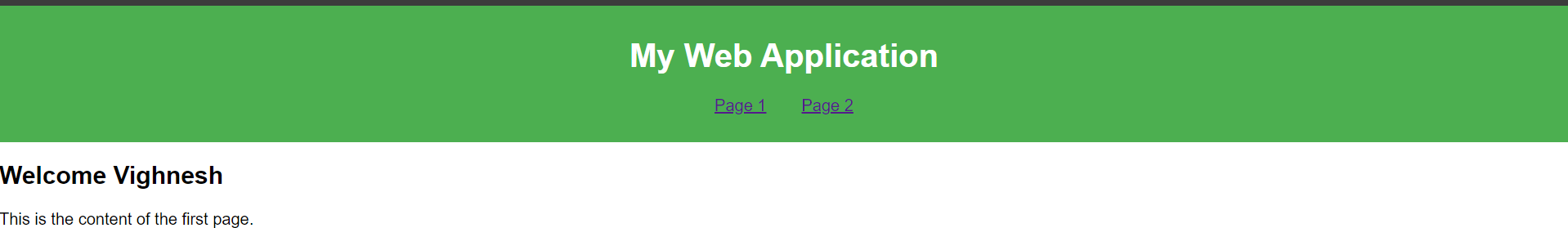
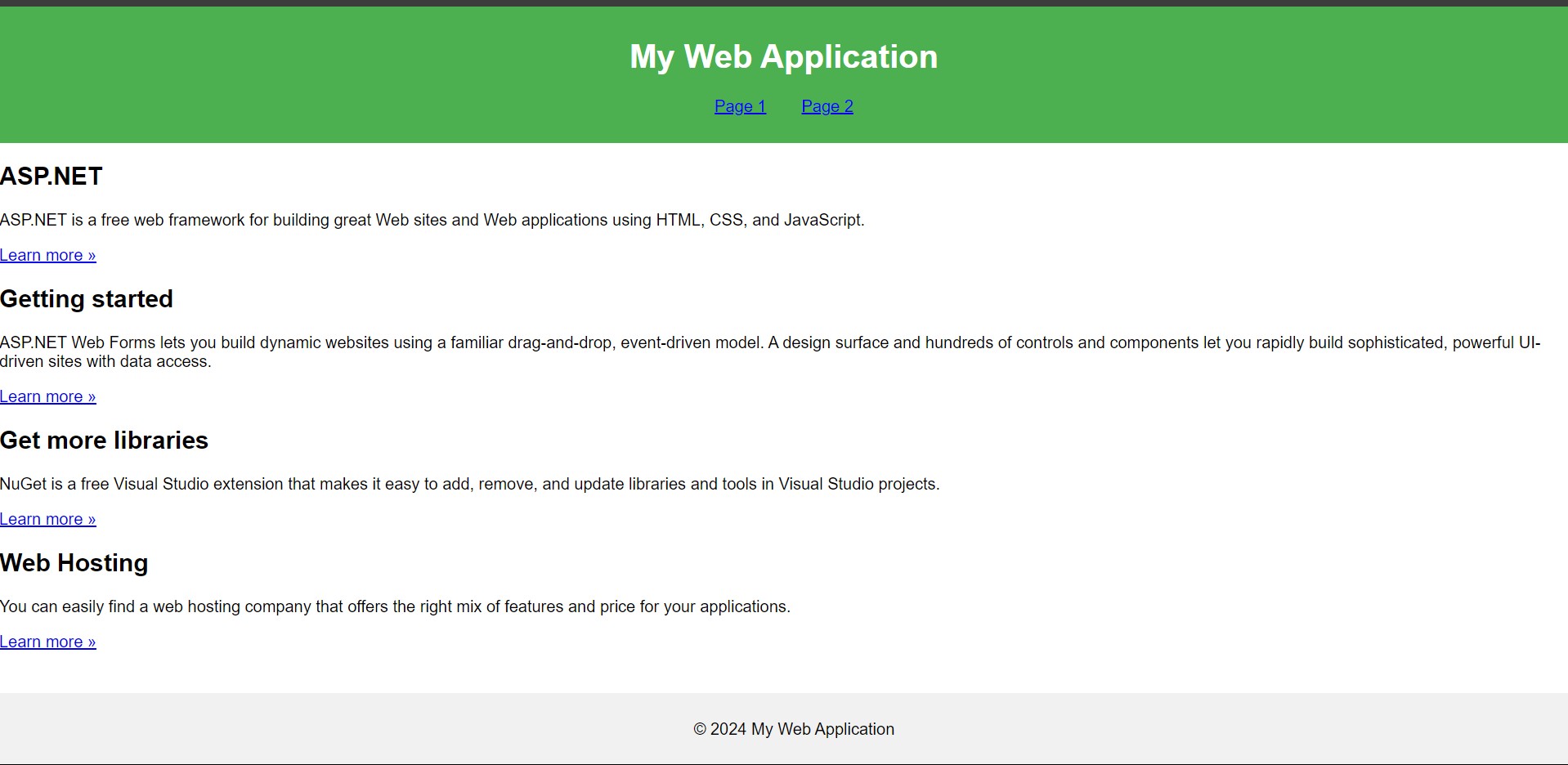
background-color: #f1f1f1; text-align: center; padding: 10px;

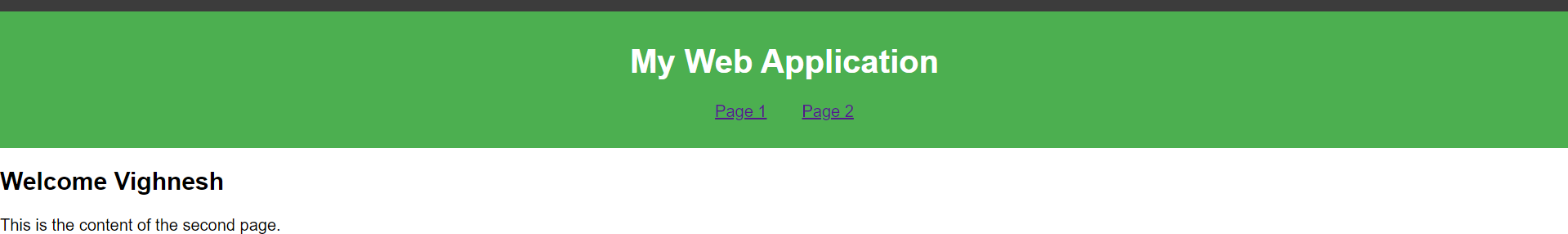
position: fixed; bottom: 0;

width: 100%;

}

## Output:





1. 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)

{

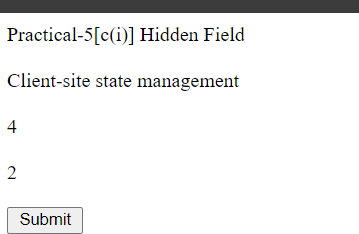
}

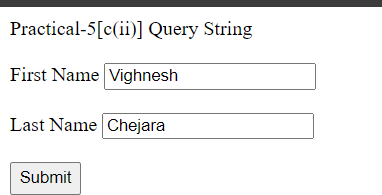
}

## Output:

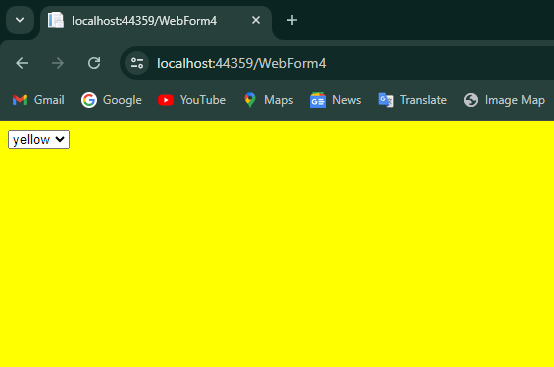
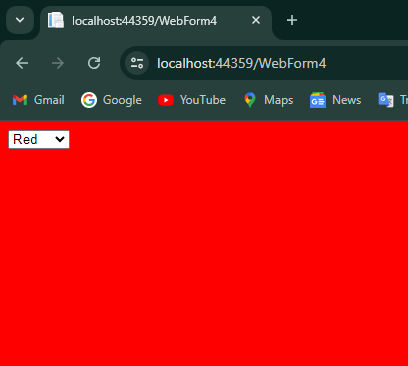
Label1.Text = ViewState["nam"].ToString();

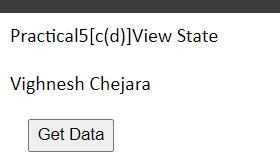
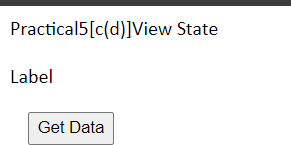
}











# Practical-6

1. 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)

{

}

}

}

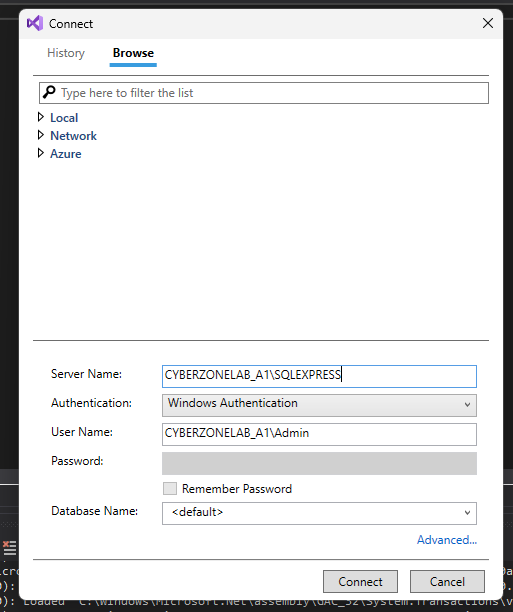
## SQLQuery1.sql:

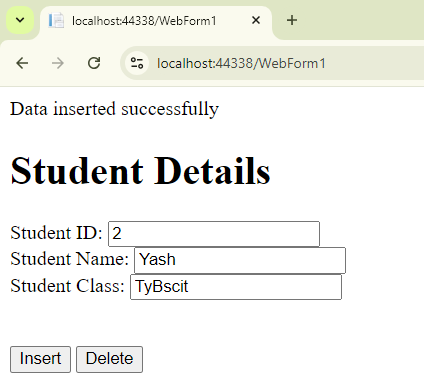
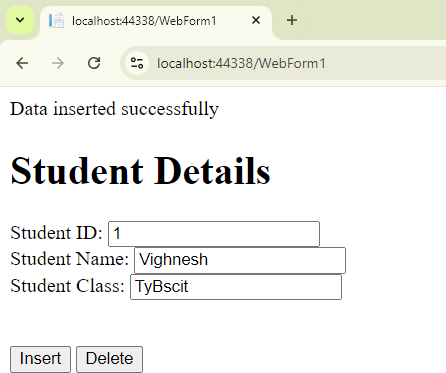
Create Database Students Use Students

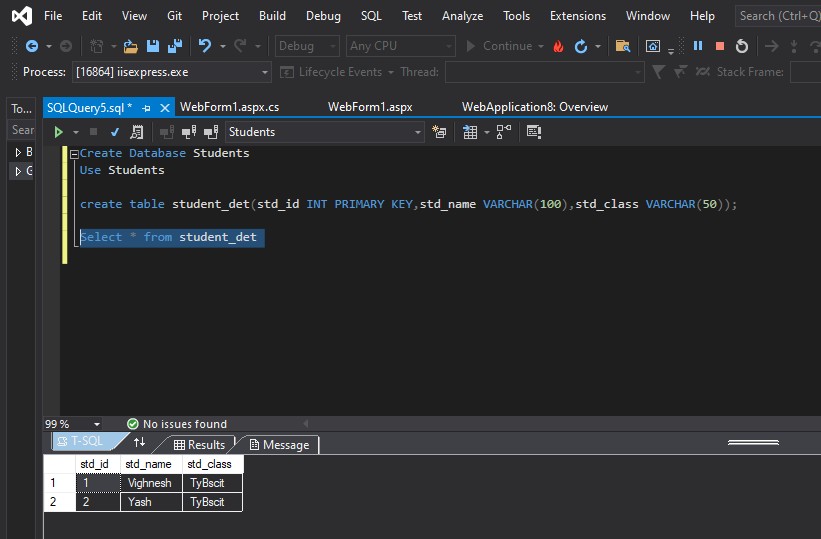
create table student\_det(std\_id INT PRIMARY KEY,std\_name VARCHAR(100),std\_class VARCHAR(50));

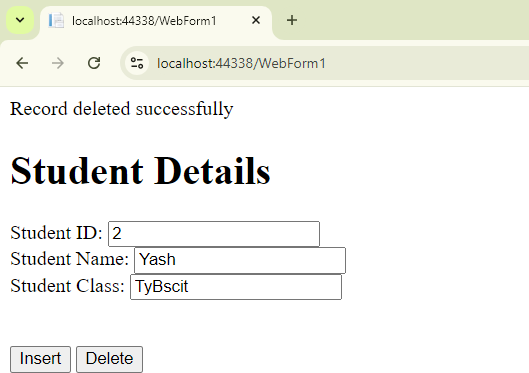
Select \* from student\_det

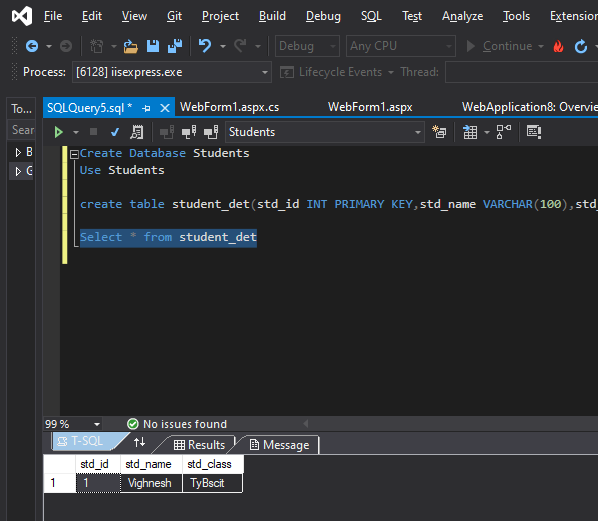
## Output:











1. 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();

}

}

}

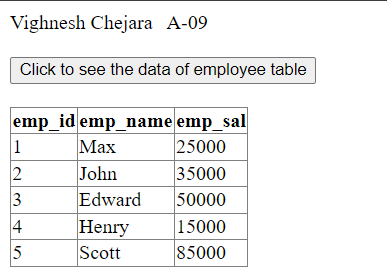
## SQLQuery1.sql:

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:**



# Practical-7

1. 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>

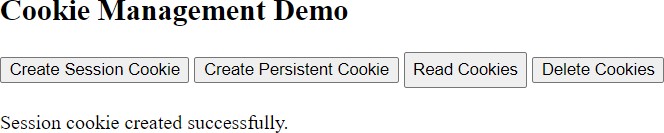
<system.web>

<compilation debug="false" targetFramework="4.0" />

</system.web>

</configuration>

## Output:









1. 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="/"

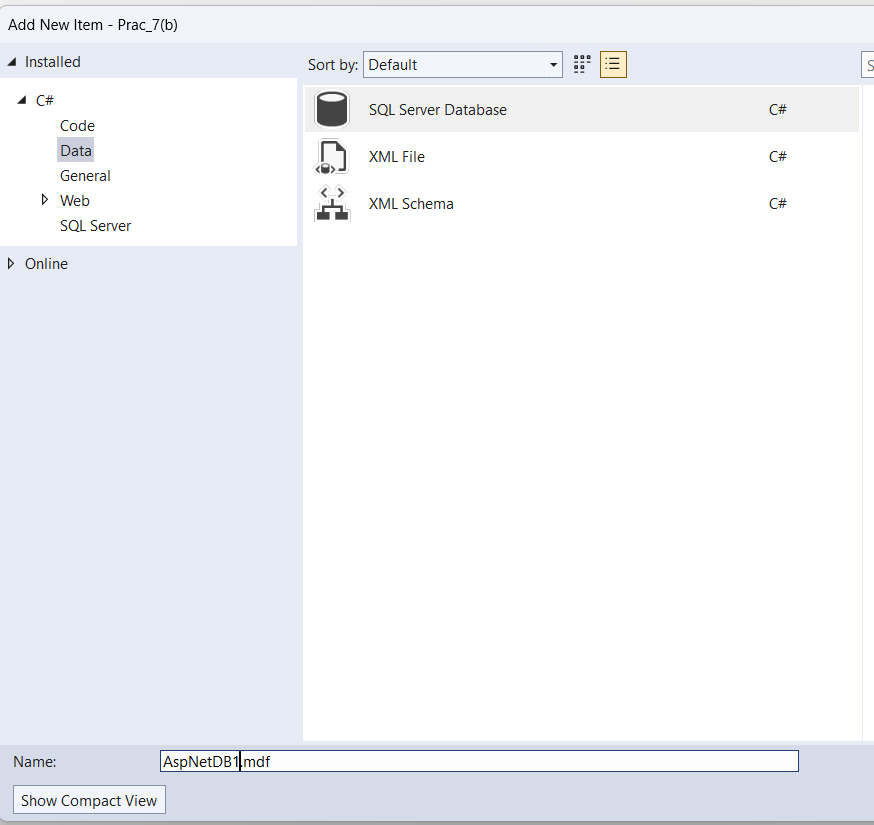
/>

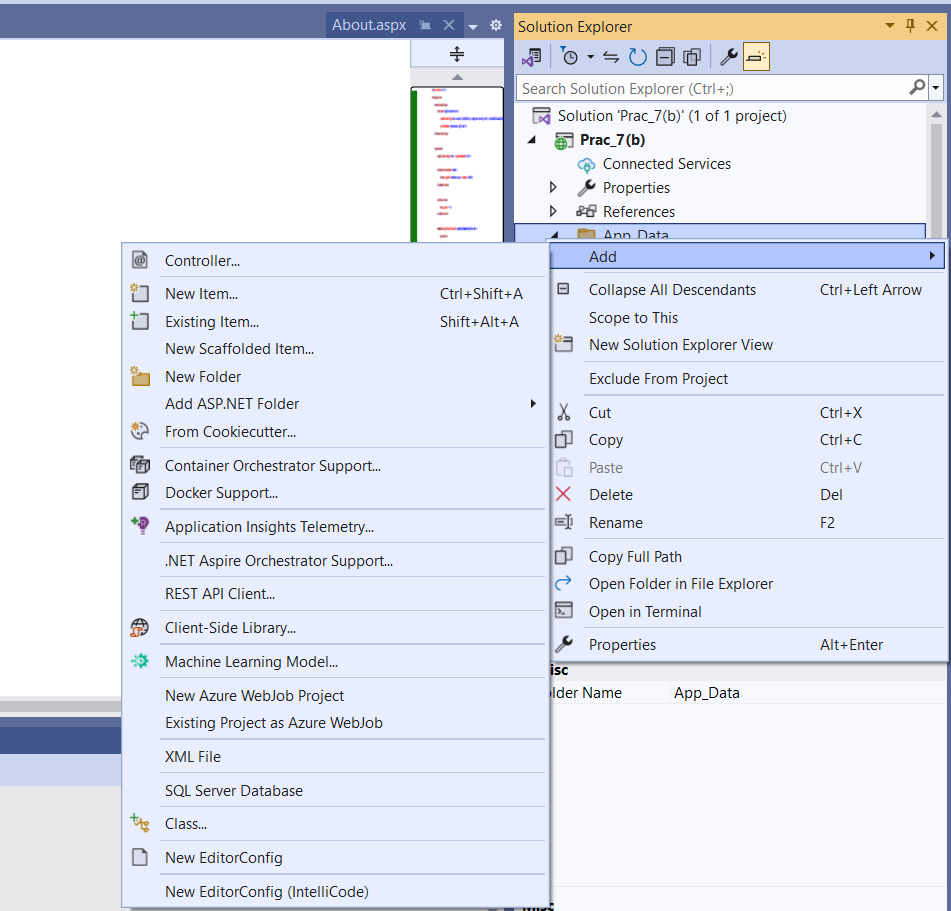
</providers>

</roleManager>

</system.web>

</configuration>

**Output:**

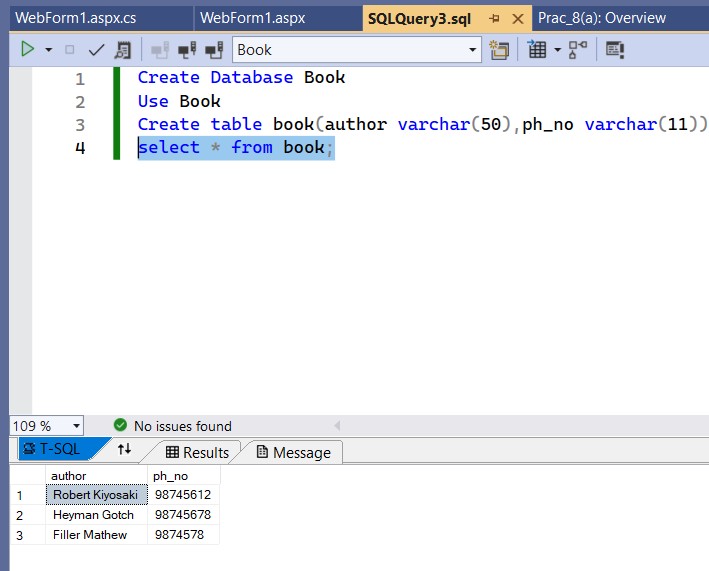
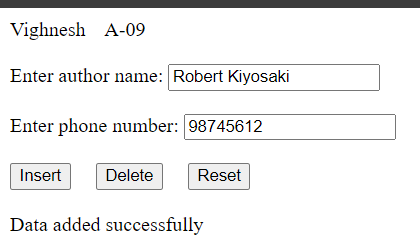


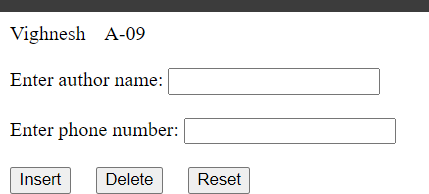
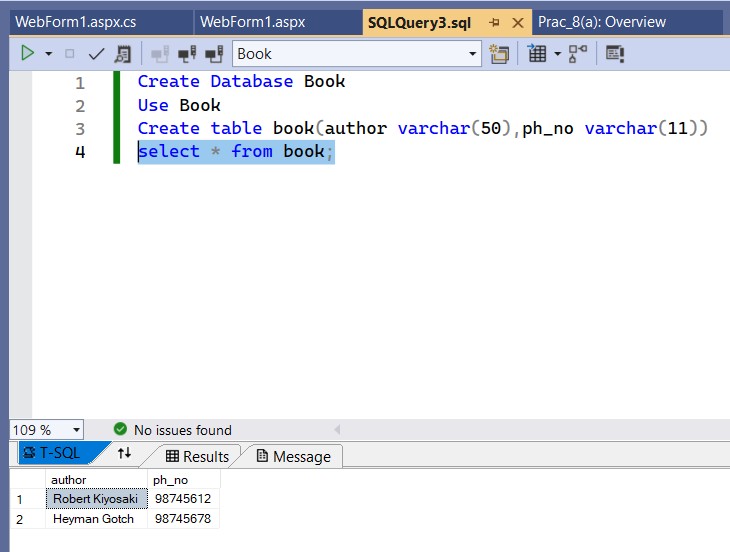
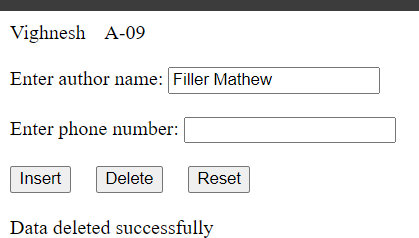
# Practical-8

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

## Code:

**Default.aspx: Output:**





1. 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:

