# ADVANCED WEB TECHNOLOGY & DOT NET

# INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **Prac. No.** | **AIM** | **Date**  **of Submission** | **Sign** |
| **1** | Create a simple window form application for calculator. | 10th June 2021 |  |
| **2** | Design account class to handle different operations such as open, deposit, withdrawn, check balance. Design GUI to demonstrate the use of this class. | 10th June 2021 |  |
| **3** | Design a GUI to demonstrate Single Inheritance. | 10th June 2021 |  |
| **4** | Design a GUI to demonstrate Multiple Inheritance. | 10th June 2021 |  |
| **5** | Design a GUI to demonstrate runtime polymorphism using abstract class. | 10th June 2021 |  |
| **6** | Create a simple Web application to Display your Details entered in registration form using advanced controls. | 1st July 2021 |  |
| **7** | Develop a windows application to create StudentInfo (Name, RollNo, Sem) table & add, delete, modify, search records(Connected Architecture) | 1st July 2021 |  |
| **8** | Design a program for multiple choice quiz application. Store at least 3 questions with their 4 optional answers, the correct answer & marks assigned in sql database table. Calculate total score & display result in message box on click of submit button.(Disconnected Arcitechture) | 1st July 2021 |  |
| **9** | Write a program on LinqToSQLClass. Show select, selectWith, insert, update, delete command on Course/Product Table. | 1st July 2021 |  |
| **10** | Create ASP.NET program to demonstrate binding of different controls using LINQ fetch all the records from database and display that records on grid view. | 1st July 2021 |  |
| **11** | Create a windows application to implement Simple & Parameterized Stored Procedure. | 11th July 2021 |  |
| **12** | Create a windows applications on managing State: Client side (view state) | 11th July 2021 |  |
| **13** | Create a windows applications on managing State: Client side (Hidden field) | 11th July 2021 |  |
| **14** | Create a windows applications on managing State: Client side (Persistent & Non Persistent Cookies) | 11th July 2021 |  |
| **15** | Create a windows applications on managing State: Server Side(Session Management) | 11th July 2021 |  |
| **16** | Display ASP.NET web page to demonstrate postback and crosspage posting with all web controls. | 21st July 2021 |  |
| **17** | Create ASP.NET program using master page & themes and skins. | 21st July 2021 |  |
| **18** | Create ASP.NET program based on validation controls. | 21st July 2021 |  |
| **19** | Display digital clock using Ajax. | 21st July 2021 |  |
| **20** | Design a registration form with current time as one field, update the time using Ajax while you are entering details in registration form | 21st July 2021 |  |
| **21** | Design a web service to access the method of BankAccount class, consume this web service using web client. | 21st July 2021 |  |
| **22** | Create an web service that returns all student details from student table. Write windows application that user this service to display student details in a DataGridView control. | 21st July 2021 |  |
| **23** | Design WCF service for a simple arithmetic calculator; consume the service using a web client. | 21st July 2021 |  |
| **24** | Design a simple MVC application to demonstrate use of ActionResult & ViewResult Method, ViewBag Object. | 21st July 2021 |  |
| **25** | Design a simple Data-Entry Application( for Customer) with MVC using following:   * Creating & accessing strongly typed View & model * Automatically implemented properties, * Html helper Methods, * Validations, * Style sheet for highlighting Invalid fields * Bootstrap functionality | 21st July 2021 |  |

**Prof. Pragati Mestry**

*(Subject-In-Charge)*

Practical 1

**Aim: Create a simple window form application for calculator.**

**CODE**:

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace mca {

public partial class Form1:Form {

int num1, num2;

int op;

public Form1() {

InitializeComponent();

}

private void label1\_Click(object sender,EventArgs e) {

}

private void Form1\_Load(object sender,EventArgs e) {

}

private void button1\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "1";

}

private void button2\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "2";

}

private void button3\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "3";

}

private void button8\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "4";

}

private void button7\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "5";

}

private void button6\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "6";

}

private void button12\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "7";

}

private void button11\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "8";

}

private void button15\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "0";

}

private void button10\_Click(object sender,EventArgs e) {

textBox1.Text = textBox1.Text + "9";

}

private void button4\_Click(object sender,EventArgs e) {

num1 = int.Parse(textBox1.Text);

textBox1.Text = "";

op = '+';

}

private void button5\_Click(object sender,EventArgs e) {

num1 = int.Parse(textBox1.Text);

textBox1.Text = "";

op = '-';

}

private void button9\_Click(object sender,EventArgs e) {

num1 = int.Parse(textBox1.Text);

textBox1.Text = "";

op = '\*';

}

private void button13\_Click(object sender,EventArgs e) {

num1 = int.Parse(textBox1.Text);

textBox1.Text = "";

op = '/';

}

private void button14\_Click(object sender,EventArgs e)

{

num2 = int.Parse(textBox1.Text);

switch (op) {

case '+':

int result = num1 + num2;

textBox1.Text = result.ToString();

break;

case '-':

int result1 = num1 - num2;

textBox1.Text = result1.ToString();

break;

case '\*':

int result2 = num1 \* num2;

textBox1.Text = result2.ToString();

break;

case '/':

int result3 = num1 / num2;

textBox1.Text = result3.ToString();

break;

default:

textBox1.Text = "";

break;

}

}

private void button16\_Click(object sender,EventArgs e)

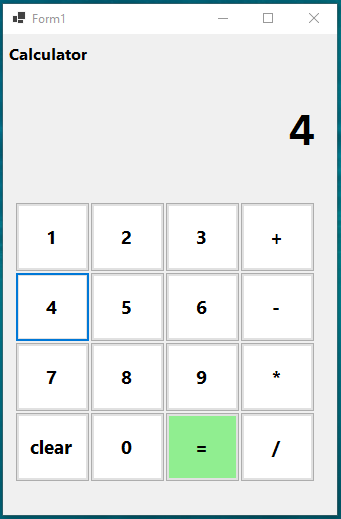
{

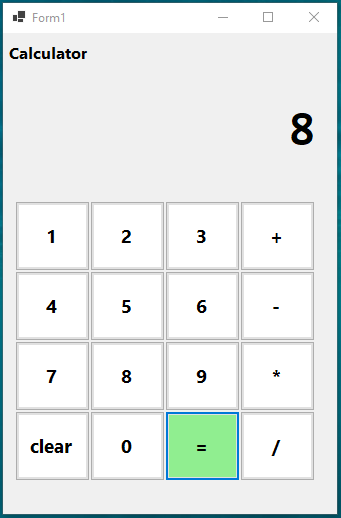
textBox1.Text = "";

}

}

}





Practical 2

Aim: Design account class to handle different operations such as open, deposit, withdrawn, check balance. Design GUI to demonstrate the use of this class.

**CODE**:

**amount.cs**

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinFormsApp2 {

class amount {

int balance = 0;

public void open(int amount) {

int amt = amount;

balance += amt;

MessageBox.Show("Successfully open account");

}

public void withdraw(int amount) {

balance -= amount;

MessageBox.Show("Successfully withdraw, amount remaining:" + balance);

}

public void deposite(int amount) {

balance += amount;

MessageBox.Show("Successfully deposited, balance amount:" + balance);

}

public void checkbalance() {

MessageBox.Show("Available balance is" + balance);

}

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinFormsApp2 {

public partial class Form1:Form {

amount obj = new amount();

public Form1() {

InitializeComponent();

}

private void label2\_Click(object sender,EventArgs e) {

}

private void Form1\_Load(object sender,EventArgs e) {

button2.Visible = false;

button3.Visible = false;

button4.Visible = false;

label5.Visible = false;

textBox4.Visible = false;

}

private void label1\_Click(object sender,EventArgs e) {

}

private void button3\_Click(object sender,EventArgs e) {

obj.deposite(int.Parse(textBox4.Text));

textBox4.Text = "";

}

private void textBox4\_TextChanged(object sender,EventArgs e) {

}

private void button1\_Click(object sender,EventArgs e) {

obj.open(int.Parse(textBox3.Text));

button2.Visible = true;

button3.Visible = true;

button4.Visible = true;

label5.Visible = true;

textBox4.Visible = true;

button1.Visible = false;

label4.Visible = false;

textBox3.Visible = false;

}

private void button2\_Click(object sender,EventArgs e) {

obj.withdraw(int.Parse(textBox4.Text));

textBox4.Text = "";

}

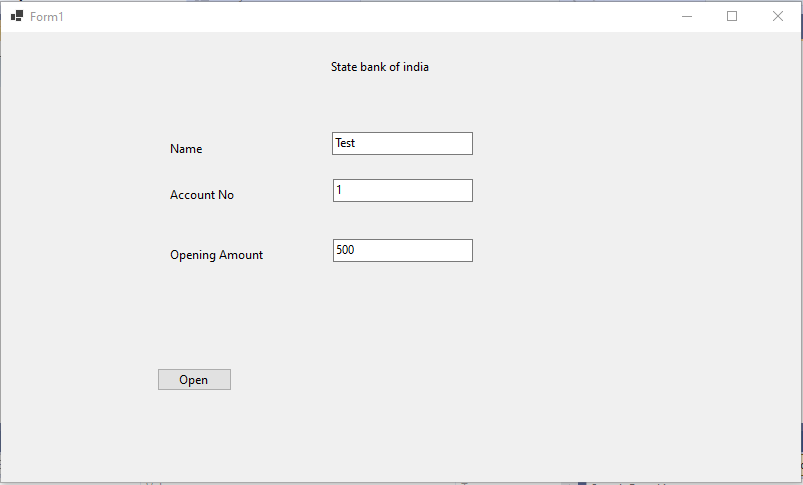
private void button4\_Click(object sender,EventArgs e) {

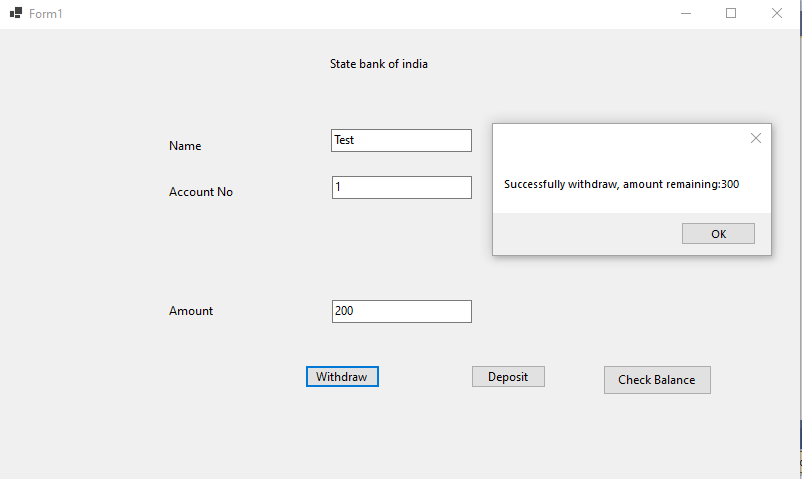
obj.checkbalance();

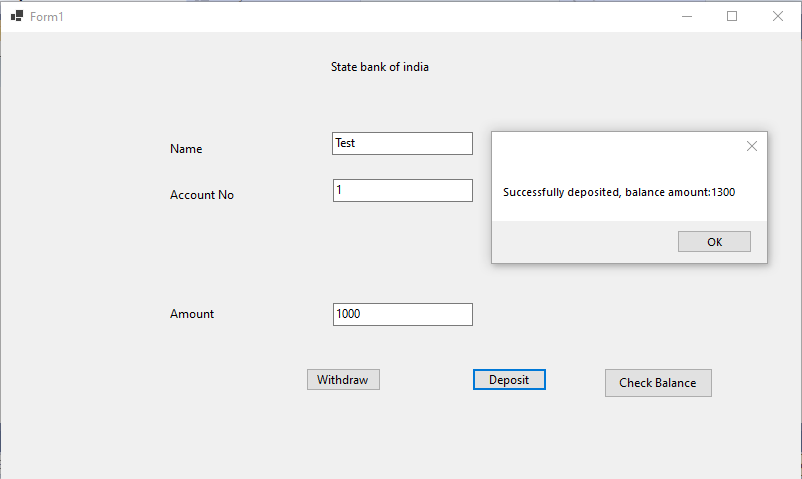
}

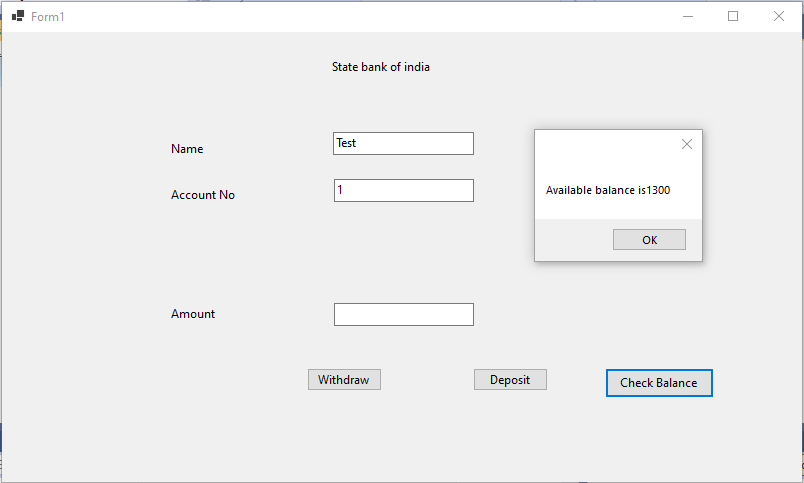
}

}









Practical 3

Aim: Design a GUI to demonstrate Single Inheritance.

**CODE**

**basicinfo.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SingleInheritance {

public class basicinfo {

public string fn, mn, ln;

public basicinfo() { }

public basicinfo(string f,string m,string l) {

fn = f;

mn = m;

ln = l;

}

public string processbasic() {

return "Welcome " + fn + " " + mn + " " + ln;

}

}

}

**addinfo.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SingleInheritance

{

public class addinfo : basicinfo

{

string course;

int year, sem;

public addinfo(string fnm, string mnm, string lnm, string cr, int yr, int s)

{

fn = fnm;

mn = mnm;

ln = lnm;

course = cr;

year = yr;

sem = s;

}

public string processadd()

{ return "\n" + course + " Course " + year + " Year " + sem + " Sem "; }

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SingleInheritance

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

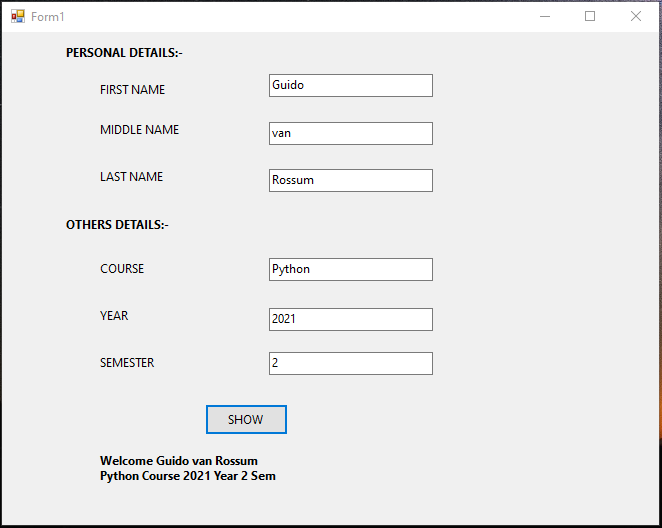
addinfo inf = new addinfo(textBox1.Text, textBox2.Text, textBox3.Text, textBox4.Text, int.Parse(textBox5.Text), int.Parse(textBox6.Text));

label9.Text = inf.processbasic() + inf.processadd();

}

}

}



Practical 4

**Aim:** Design a GUI to demonstrate Multiple Inheritance.

**CODE:**

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Multiple\_Inheritance

{

public partial class Form1 : Form

{

int ar;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

rec a = new rec(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text));

ar = a.getarea();

label4.Text = "Area is :" + ar;

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

rec a = new rec(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text));

ar = a.getarea();

int cost = a.getcost(ar);

label3.Text+= "\n Cost is: " + cost;

}

private void label3\_Click(object sender, EventArgs e)

{

}

}

}

**Cost.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Multiple\_Inheritance

{

public interface Cost

{

int getcost(int a);

}

}

**Rec.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Multiple\_Inheritance

{

public class rec : Shape, Cost

{

public rec(int a, int b) : base(a, b) { }

public int getarea()

{ return height \* width; }

public int getcost(int a)

{ return a \* 30; }

}

}

**Shape.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Multiple\_Inheritance

{

public class Shape

{

public int height, width;

public Shape(int m, int n)

{

height = m;

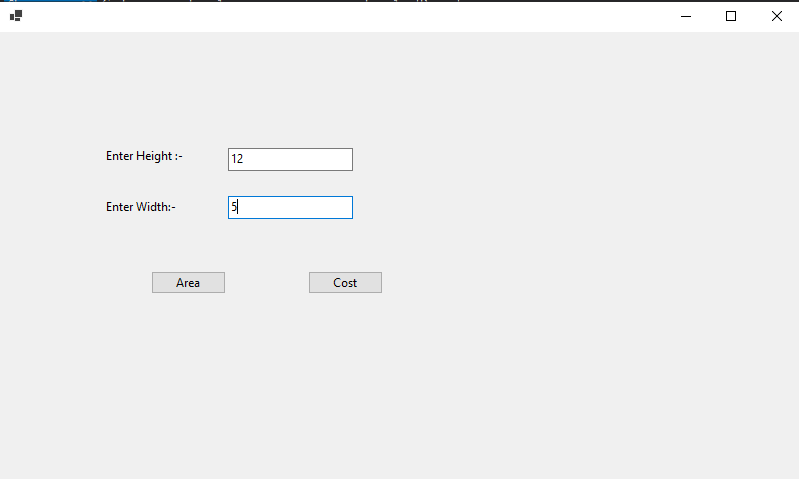
width = n;

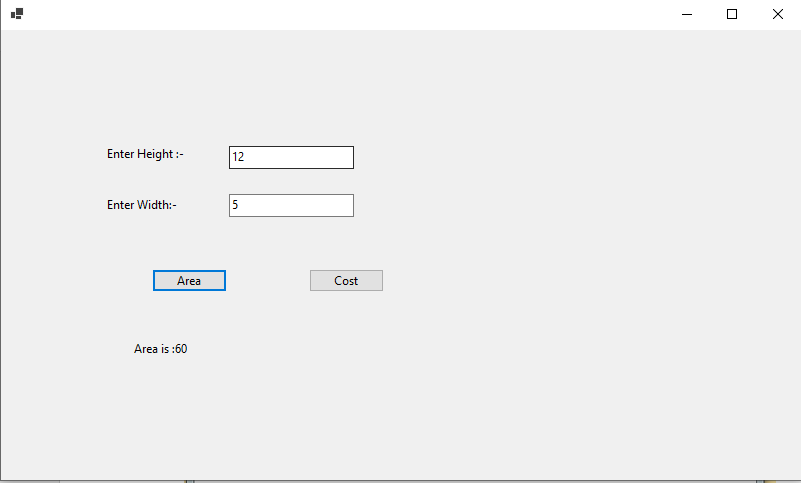
}

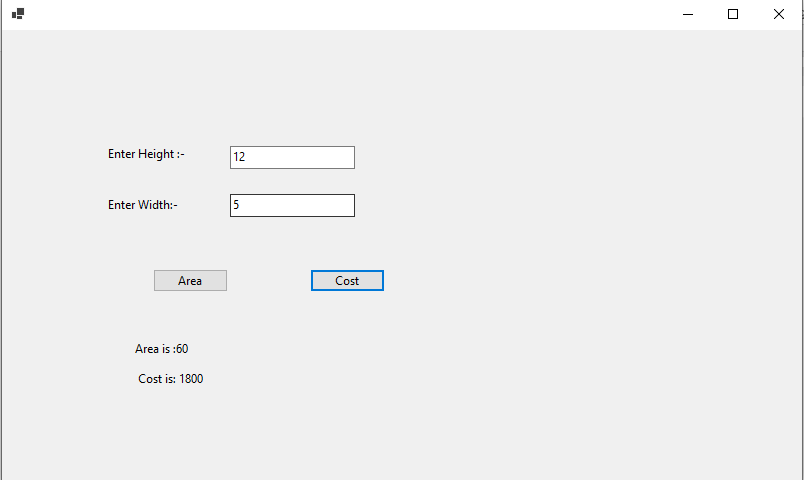
}

}

**Output:-**







Practical 5

**Aim:** Design a GUI to demonstrate runtime polymorphism using abstract class.

**CODE:**

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Abstract\_Class

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int a = Convert.ToInt32(textBox1.Text);

int b = Convert.ToInt32(textBox2.Text);

rectangle r = new rectangle();

r.set(a, b);

double c = r.area();

label3.Text = "Area is :" + Convert.ToString(c);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

**Shape.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace Abstract\_Class

{

abstract class Shape

{

public int length;

public int width;

public void set(int a = 0, int b = 0)

{

length = a;

width = b;

}

public abstract int area();

}

class rectangle : Shape

{

public override int area()

{

return (width \* length);

}

}

}

**Rectangle.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace Abstract\_Class

{

class Rectangle : Shape

{

public override int area()

{

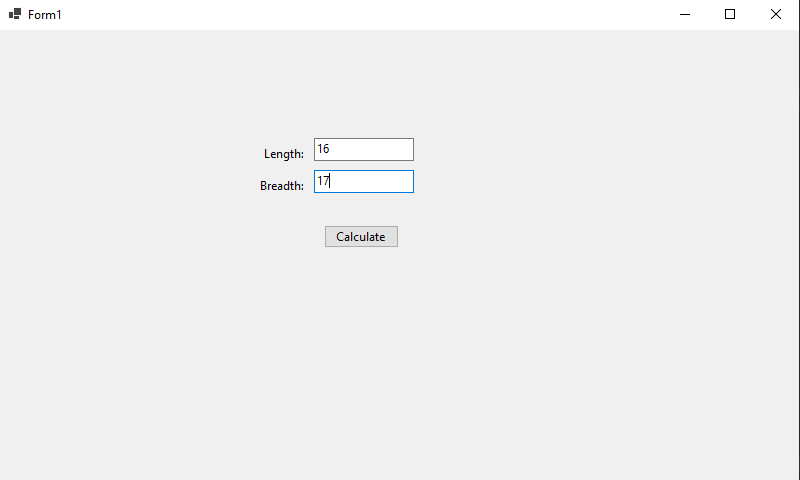
return (width \* length);

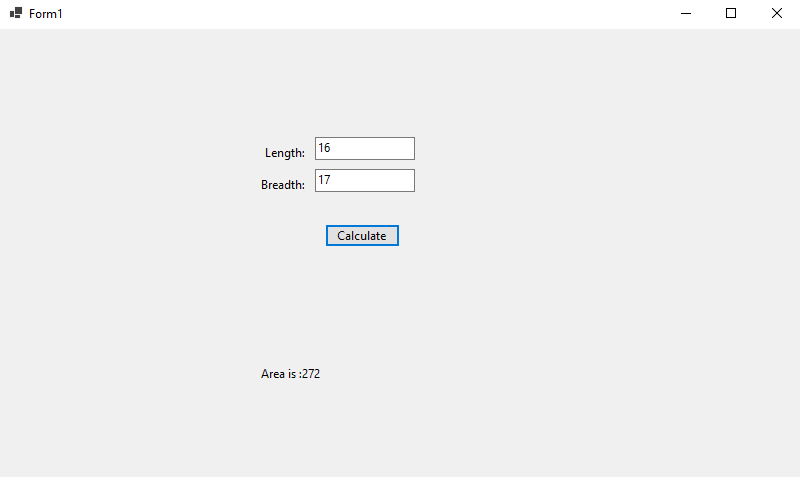
}

}

}

**Output:-**

****

****

Practical 6

Aim. Create a simple Web application to Display your Details entered in registration form using advanced controls.

CODE:

registration.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

<style type="text/css">

.auto-style2 {

width: 41%;

height: 267px;

}

</style>

</head>

<body>

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

<div>

<br />

<table class="auto-style2">

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem>Mumbai</asp:ListItem>

<asp:ListItem>Pune</asp:ListItem>

<asp:ListItem>Ratnagiri</asp:ListItem>

<asp:ListItem>Raigad</asp:ListItem>

</asp:DropDownList>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label5" runat="server" Text="Gender"></asp:Label>

</td>

<td>:</td>

<td>

<asp:RadioButton ID="RadioButton1" runat="server" GroupName="gen" Text="Male" />

<asp:RadioButton ID="RadioButton2" runat="server" GroupName="gen" Text="Female"/>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label6" runat="server" Text="Gmail"></asp:Label>

</td>

<td>:</td>

<td>

<asp:TextBox ID="TextBox4" runat="server" TextMode="Email"></asp:TextBox>

</td>

</tr>

<tr>

<td colspan="3">

<asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1\_Click" />

</td>

</tr>

<tr>

<td colspan="3">

<asp:Label ID="Label7" runat="server" Text=""></asp:Label>

</td>

</tr>

</table>

</div>

</form>

</body>

</html>

**registration.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace RegistrationForm {

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

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click(object sender,EventArgs e) {

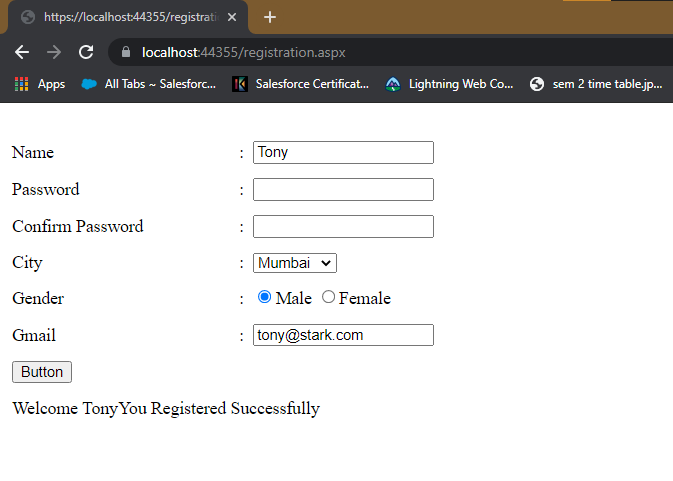
Label7.Text = "Welcome " + TextBox1.Text + "You Registered Successfully";

}

}

}

OUTPUT:



Practical 7

Aim: Develop a windows application to create StudentInfo (Name, RollNo, Sem) table & add, delete, modify, search records(Connected Architecture)

CODE:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace StudentInfo {

public partial class Form1:Form {

SqlConnection conn = new SqlConnection();

SqlDataReader dr;

SqlCommand cmd;

public Form1() {

InitializeComponent();

conn.ConnectionString = "Data Source=(localdb)\\MyInstance;Initial Catalog=master;Integrated Security=True";

}

private void button1\_Click(object sender,EventArgs e) {

label4.Text = "";

label5.Text = "";

label6.Text = "";

//int roll = Convert.ToInt16(textBox2.Text);

conn.Open();

cmd = new SqlCommand("Select \* from Studentinfo where roll =" + textBox2.Text);

cmd.Connection = conn;

dr = cmd.ExecuteReader();

while (dr.Read()) {

label4.Text += dr[0].ToString() + "\n";

label5.Text += dr[1].ToString() + "\n";

label6.Text += dr[2].ToString() + "\n";

}

conn.Close();

}

private void button2\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("insert into StudentInfo values('" + textBox1.Text + "'," + textBox2.Text + "," + textBox3.Text + ")");

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Inserted Successfully!!!");

}

private void button3\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("update StudentInfo set Name='" + textBox1.Text + "',Sem=" + textBox3.Text + " where roll= " + textBox2.Text);

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Updated Successfully!!!");

}

private void button4\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("delete from StudentInfo where roll = " + textBox2.Text);

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Deleted Successfully!!!");

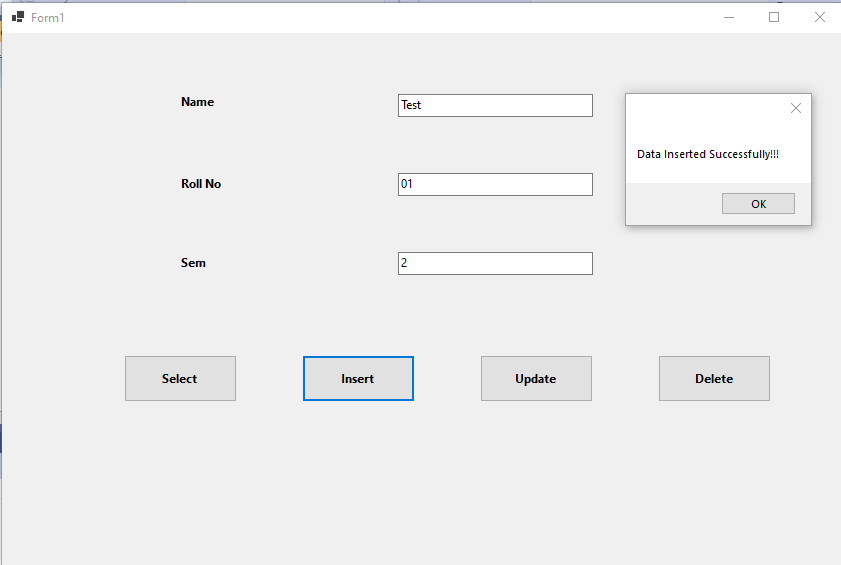
}

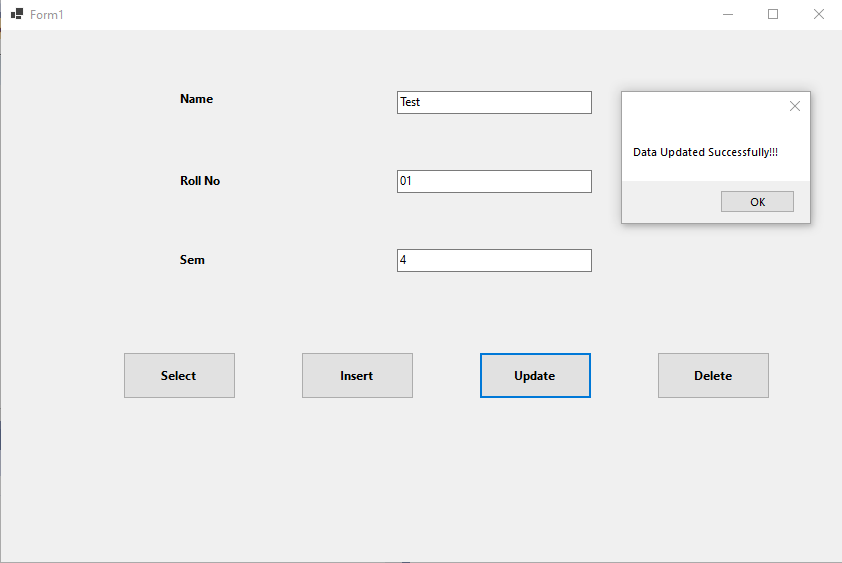
private void Form1\_Load(object sender,EventArgs e) {

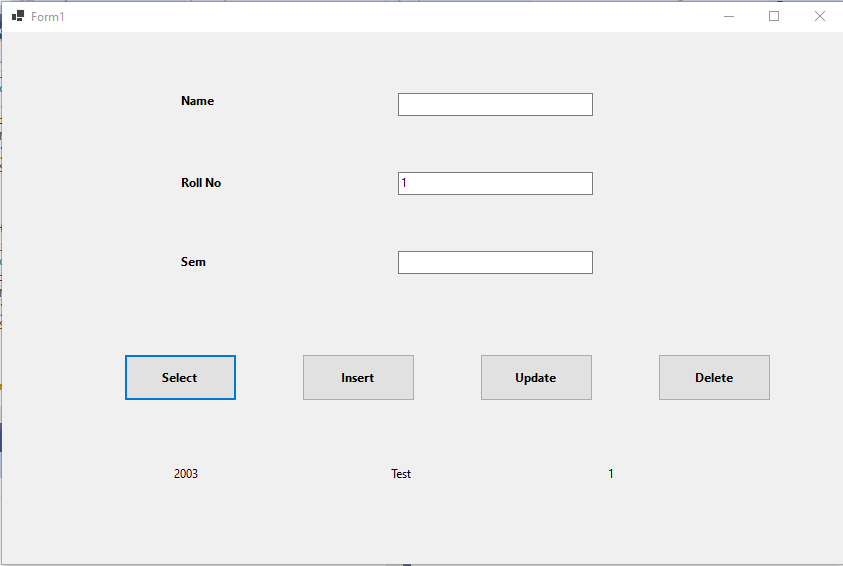
}

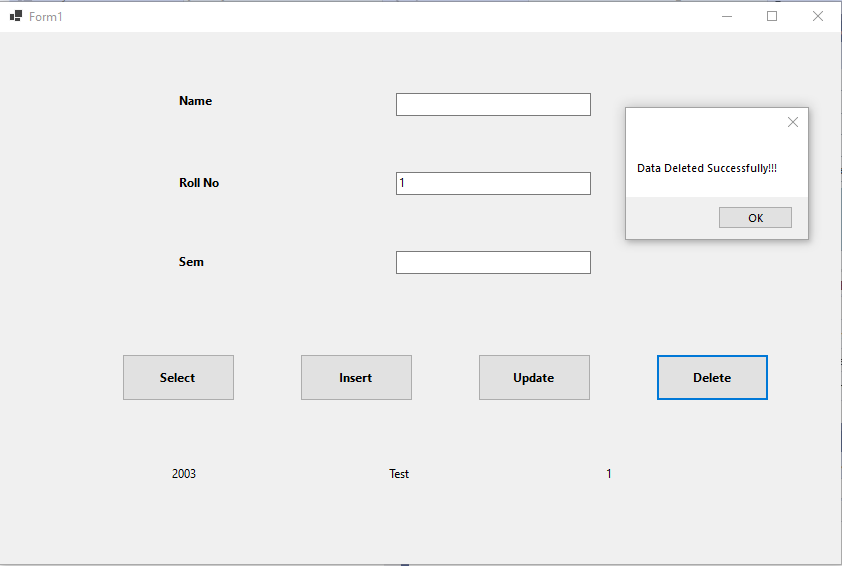
}

}









Practical 5

Aim: Design a program for multiple choice quiz application. Store at least 3 questions with their 4 optional answers, the correct answer & marks assigned in sql database table. Calculate total score & display result in message box on click of submit button. (Disconnected Architecture)

CODE:

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Quizz\_App

{

public partial class Form1 : Form

{

SqlConnection con = new SqlConnection();

public Form1()

{

InitializeComponent();

con.ConnectionString = "Data Source=(localdb)\\MyInstance;Initial Catalog=master;Integrated Security=True";

con.Open();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string q = "select ans from quizz";

SqlCommand cm = new SqlCommand(q, con);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataTable dt = new DataTable();

da.Fill(dt);

int count = 0;

foreach (DataRow row in dt.Rows)

{

if (radioButton2.Checked)

{

if (radioButton2.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton6.Checked)

{

if (radioButton6.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton9.Checked)

{

if (radioButton9.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton16.Checked)

{

if (radioButton16.Text == row["ans"].ToString())

{ count++; }

}

}

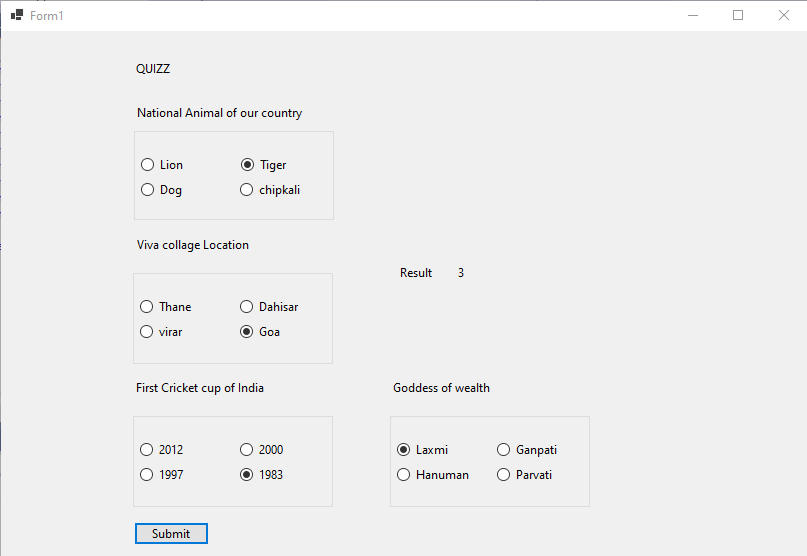
label7.Text = count.ToString();

}

}

}

Output:



Practical 9

9. Write a program on LinqToSQLClass. Show select, selectWith, insert, update, delete command on Course\Product Table.

Code :

ProductData.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title>Product</title>

<style type="text/css">

.auto-style2 {

width: 229px;

}

.auto-style3 {

width: 118px;

}

.auto-style4 {

width: 420px;

}

</style>

</head>

<body>

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

<div>

<br />

<br />

<table>

<tr>

<td class="auto-style2">Enter Product\_Name</td>

<td class="auto-style3">:</td>

<td class="auto-style4"><asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Enter Product\_Price</td>

<td class="auto-style3">:</td>

<td class="auto-style4"><asp:TextBox ID="TextBox2" runat="server"></asp:TextBox></td>

</tr>

<tr>

<th class="auto-style2"><asp:Button ID="Button1" runat="server" Text="Add" OnClick="Button1\_Click1" /></th>

<th class="auto-style3"><asp:Button ID="Button2" runat="server" Text="Select" OnClick="Button2\_Click" /></th>

<th class="auto-style4">

<asp:Button ID="Button4" runat="server" OnClick="Button4\_Click" Text="Update" />

&emsp;

<asp:Button ID="Button3" runat="server" Text="Delete" OnClick="Button3\_Click" /></th>

</tr>

<tr>

<th class="auto-style2">

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

</th>

<th class="auto-style3">&nbsp;</th>

<th class="auto-style4">&nbsp;</th>

</tr>

</table>

<br />

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataKeyNames="ID" DataSourceID="SqlDataSource1">

<Columns>

<asp:BoundField DataField="ID" HeaderText="ID" ReadOnly="True" SortExpression="ID" InsertVisible="False" />

<asp:BoundField DataField="P\_Name" HeaderText="P\_Name" SortExpression="P\_Name" />

<asp:BoundField DataField="P\_Price" HeaderText="P\_Price" SortExpression="P\_Price" />

</Columns>

</asp:GridView>

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:masterConnectionString %>" SelectCommand="SELECT \* FROM [Product]"></asp:SqlDataSource>

<br />

<br />

</div>

</form>

</body>

</html>

ProductData.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace LinqToSql {

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

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click1(object sender,EventArgs e) {

var d = new DataClassesDataContext();

var c = new Product();

c.P\_Name = TextBox1.Text;

c.P\_Price = Convert.ToInt16(TextBox2.Text);

d.Products.InsertOnSubmit(c);

d.SubmitChanges();

Label1.Text = "Successfully Added Data";

}

protected void Button2\_Click(object sender,EventArgs e) {

GridView1.DataBind();

}

protected void Button3\_Click(object sender,EventArgs e) {

var d = new DataClassesDataContext();

Product c = d.Products.FirstOrDefault(e1 => e1.P\_Name.Equals(TextBox1.Text));

c.P\_Name = TextBox1.Text;

d.Products.DeleteOnSubmit(c);

d.SubmitChanges();

Label1.Text = "Successfully Deleted Data";

GridView1.DataBind();

}

protected void Button4\_Click(object sender,EventArgs e) {

var d = new DataClassesDataContext();

Product c = d.Products.FirstOrDefault(e1 => e1.P\_Name.Equals(TextBox1.Text));

c.P\_Name = TextBox1.Text;

c.P\_Price = Convert.ToInt16(TextBox2.Text);

d.SubmitChanges();

Label1.Text = "Successfully Updated Data";

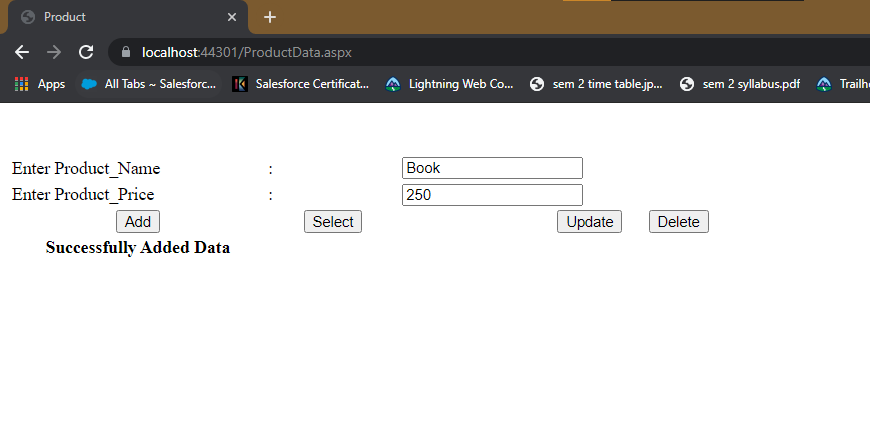
GridView1.DataBind();

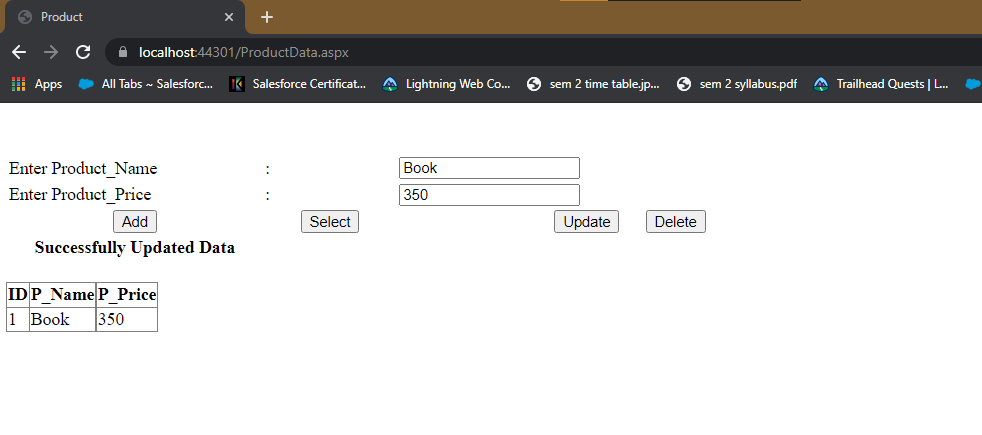
}

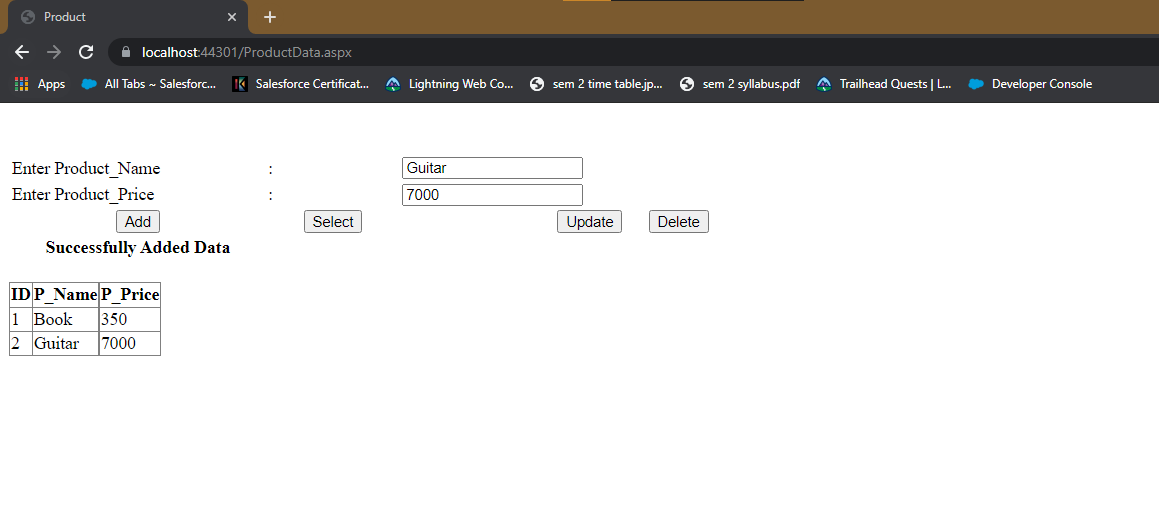
}

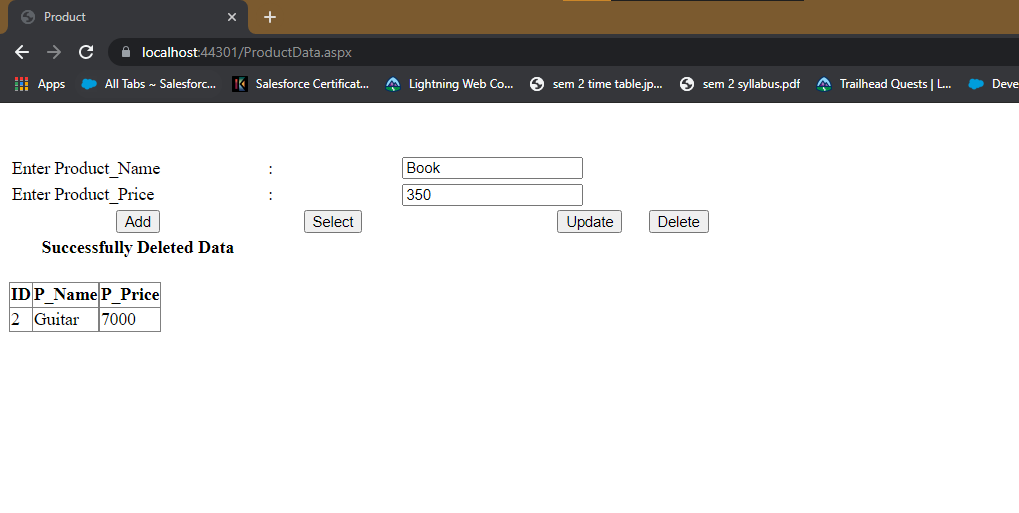
}

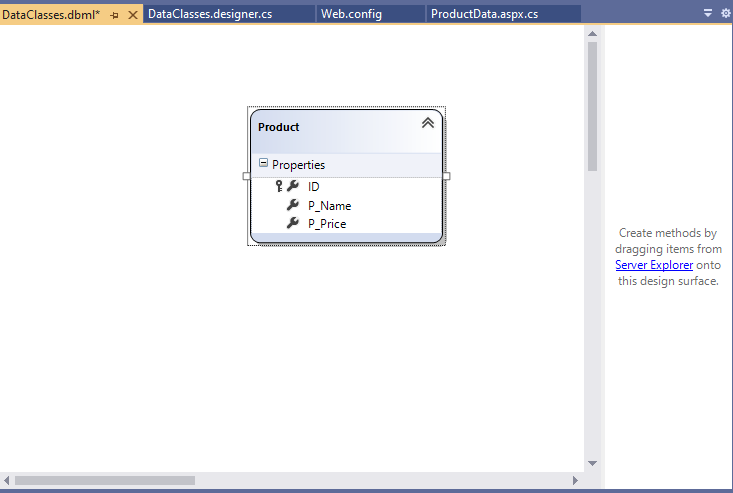
OUTPUT











Practical 10

10. Create ASP.NET program to demonstrate binding of different controls from database and display those records on grid view.

CODE :

TableList.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:masterConnectionString %>" SelectCommand="Select name from sys.tables"></asp:SqlDataSource>

<br />

<br />

List of Table :

<asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="SqlDataSource1" DataTextField="name" DataValueField="name">

</asp:DropDownList>

<br />

<br />

<br />

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Show Data" />

<br />

<br />

&nbsp;<br />

<br />

<asp:SqlDataSource ID="SqlDataSource2" runat="server"></asp:SqlDataSource>

<br />

<asp:GridView ID="GridView1" runat="server" >

</asp:GridView>

</div>

</form>

</body>

</html>

TableList.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.Data.SqlClient;

using System.Data;

namespace DataBinding {

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

static string conn = System.Configuration.ConfigurationManager.ConnectionStrings["masterConnectionString"].ConnectionString;

SqlConnection cn = new SqlConnection(conn);

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter adapt;

DataTable dt;

DataSet ds;

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click(object sender,EventArgs e) {

String q = "select \* from " + DropDownList1.Text;

cn.Open();

adapt = new SqlDataAdapter(q,cn);

dt = new DataTable();

adapt.Fill(dt);

GridView1.DataSource = dt;

cn.Close();

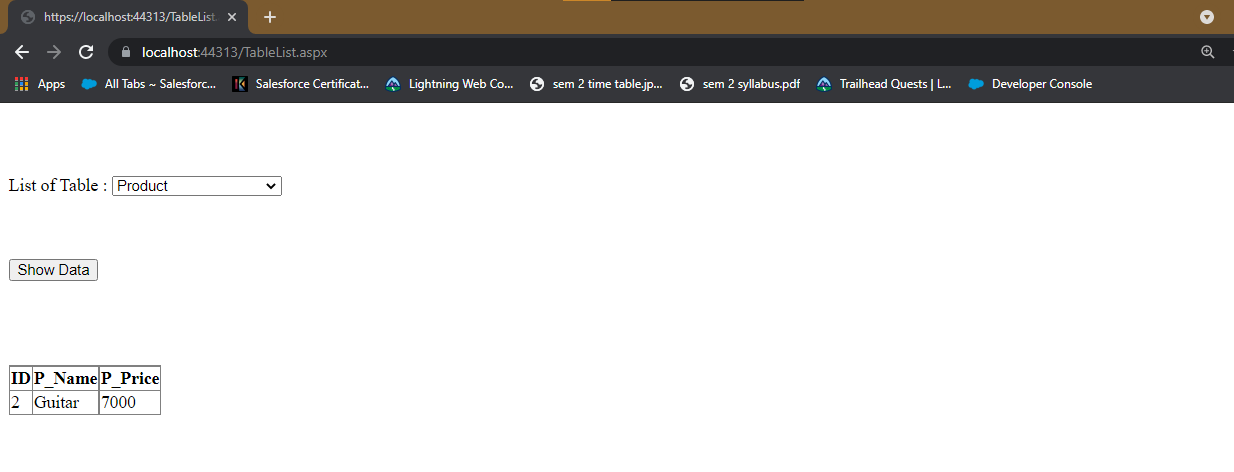
GridView1.DataBind();

}

}

}

Output:



Practical No:-11

**Aim : Create a windows application to implement Simple & Parameterized Stored Procedure.**

**Code :**

**File name**:-Default.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

<style type="text/css">

.auto-style1 {

width: 57%;

}

.auto-style2 {

height: 33px;

}

</style>

</head>

<body>

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

<div>

&nbsp;<br />

<br />

<table class="auto-style1">

<tr>

<td>Name</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td class="auto-style2">Age</td>

<td class="auto-style2">:</td>

<td class="auto-style2">

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

</td>

</tr>

<tr>

<td>City</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td><asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Add Record" /></td>

<td><asp:Button ID="Button2" runat="server" OnClick="Button2\_Click" Text="Update Record" /></td>

<td><asp:Button ID="Button3" runat="server" OnClick="Button3\_Click" Text="Delete Record" /></td>

</tr>

</table>

<br />

<br />

<br />

&nbsp;

&nbsp;

<br />

<br />

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataSourceID="SqlDataSource1">

<Columns>

<asp:BoundField DataField="C\_Name" HeaderText="C\_Name" SortExpression="C\_Name" />

<asp:BoundField DataField="C\_Age" HeaderText="C\_Age" SortExpression="C\_Age" />

<asp:BoundField DataField="C\_Country" HeaderText="C\_Country" SortExpression="C\_Country" />

</Columns>

</asp:GridView>

<br />

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

<br />

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:mcaConnectionString %>" SelectCommand="SELECT [C\_Name], [C\_Age], [C\_Country] FROM [UserRegistration]"></asp:SqlDataSource>

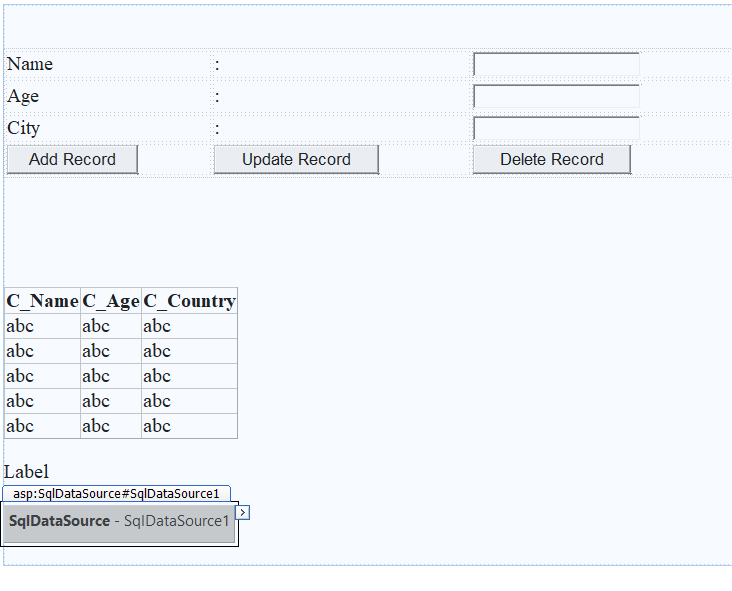
<br />

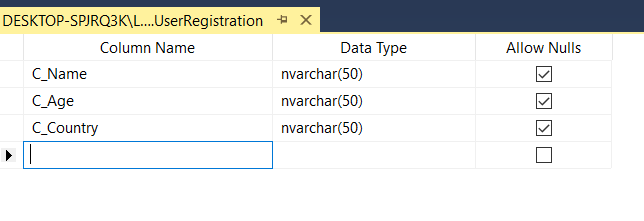
</div>

</form>

</body>

</html>





**Create your Stored Procedure**

Create PROCEDURE All\_Operation\_StoredProcedure(

@Name varchar(100)= null,

@Age varchar(100)= null,

@Country varchar(100)= null,

@Action varchar(100)= null

)

As begin

if @Action = 'Insert' Insert into UserRegistration(C\_Name, C\_Age,

C\_Country) values(@Name, @Age, @Country)

if @Action = 'Update' Update UserRegistration set C\_Name = @Name,

C\_Age = @Age, C\_Country = @Country where C\_Name = @Name

if @Action= 'Delete' Delete from UserRegistration where C\_Name =

@Name

End

**without parameter:**

Create PROCEDURE Select\_Simple\_StoredProcedure1

AS

begin

Select C\_Name from UserRegistration

end

**Default .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.Data;

using System.Data.SqlClient;

public partial class \_Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;Initial Catalog=mca;Integrated Security=True");

SqlCommand cmd = new SqlCommand("Select\_Simple\_StoredProcedure1 ", cn);

cmd.CommandType = CommandType.StoredProcedure;

cn.Open();

SqlDataReader dr = cmd.ExecuteReader();

Label1.Text = " All users are: " ;

while (dr.Read())

Label1.Text += dr[0].ToString() + "\n " ;

dr.Close();

cn.Close();

}

protected void Button1\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;Initial Catalog=mca;Integrated Security=True");

SqlCommand cmd = new SqlCommand("All\_Operation\_StoredProcedure", cn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Action", "Insert");

cmd.Parameters.AddWithValue("@Name", TextBox1.Text);

cmd.Parameters.AddWithValue("@Age", TextBox2.Text);

cmd.Parameters.AddWithValue("@Country", TextBox3.Text);

cn.Open();

cmd.ExecuteNonQuery();

GridView1.Visible = true;

GridView1.DataBind();

cn.Close();

}

protected void Button2\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;Initial Catalog=mca;Integrated Security=True");

SqlCommand cmd = new SqlCommand("All\_Operation\_StoredProcedure", cn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Action", "Update");

cmd.Parameters.AddWithValue("@Name", TextBox1.Text);

cmd.Parameters.AddWithValue("@Age", TextBox2.Text);

cmd.Parameters.AddWithValue("@Country", TextBox3.Text);

cn.Open();

cmd.ExecuteNonQuery();

GridView1.Visible = true;

GridView1.DataBind();

cn.Close();

}

protected void Button3\_Click(object sender, EventArgs e)

{

SqlConnection cn = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;Initial Catalog=mca;Integrated Security=True");

SqlCommand cmd = new SqlCommand("All\_Operation\_StoredProcedure", cn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Action", "Delete");

cmd.Parameters.AddWithValue("@Name", TextBox1.Text);

cmd.Parameters.AddWithValue("@Age", TextBox2.Text);

cmd.Parameters.AddWithValue("@Country", TextBox3.Text);

cn.Open();

cmd.ExecuteNonQuery();

GridView1.Visible = true;

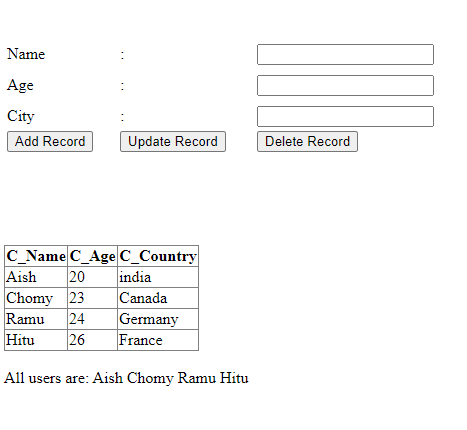
GridView1.DataBind();

cn.Close();

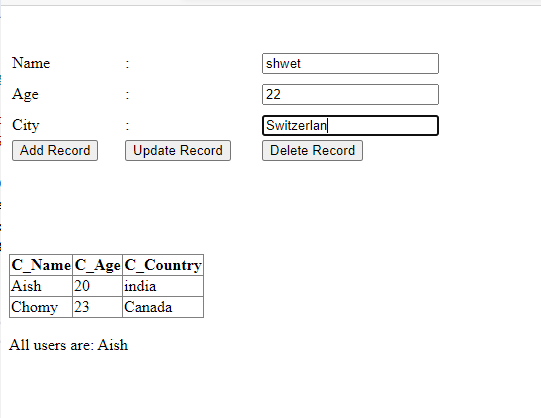
}

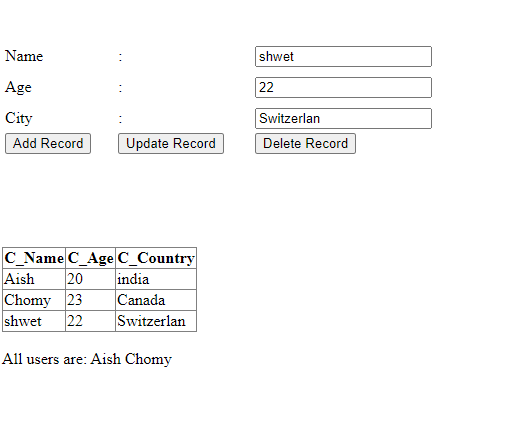
}

**Output:-**

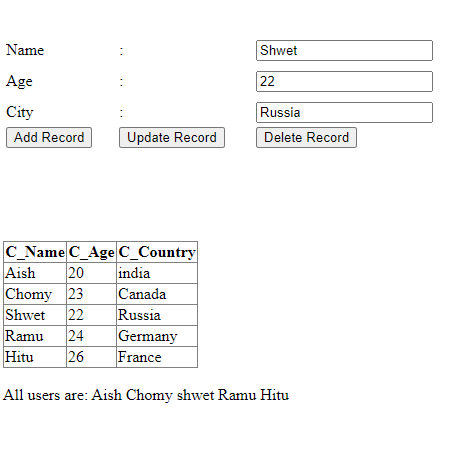


**Add Record:**

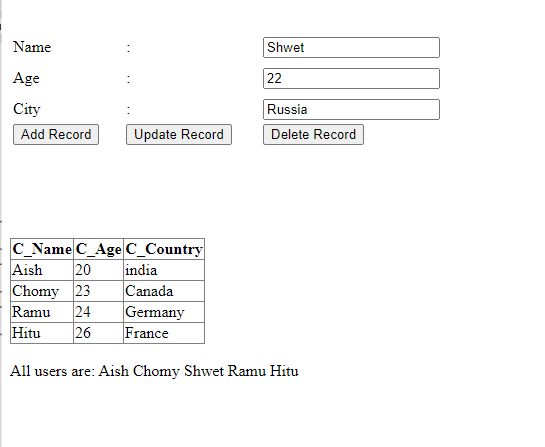




**Update Record:**



**Delete Record:**

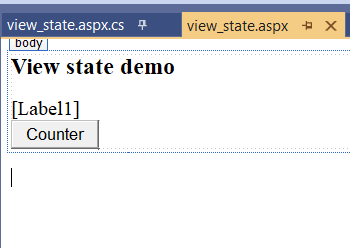


Practical No:-12

**Aim : .Create a windows applications on managing State: Client side (view state)**

**Code :**

File name:-view\_state.aspx



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="view\_state.aspx.cs" Inherits="hidden\_field\_sm\_.view\_state" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<h3>View state demo</h3>

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

</br>

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Counter" />

</div>

</form>

</body>

</html>

File name:-view\_state.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace hidden\_field\_sm\_

{

public partial class view\_state : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

int counter;

if (ViewState["Counter"] == null)

{

counter = 1;

}

else

{

counter = (int)ViewState["Counter"] + 1;

}

ViewState["Counter"] = counter;

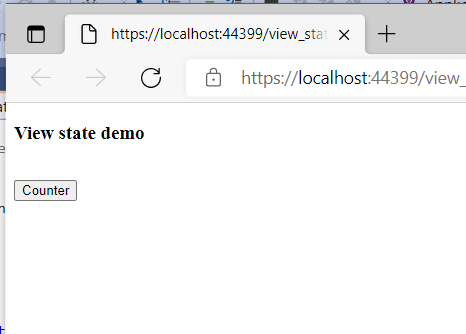
Label1.Text = "Page Counter" + counter.ToString();

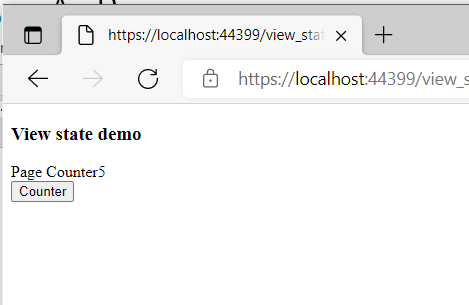
}

}

}

**Output:-**



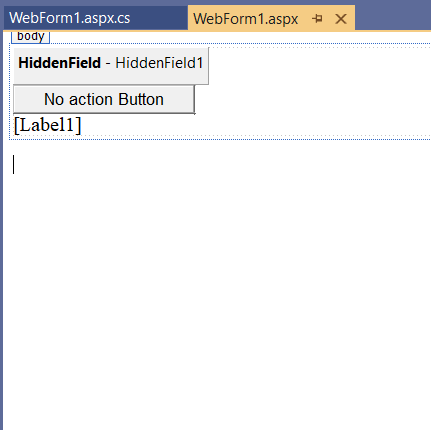


Practical No:-13

**Aim : Create a windows applications on managing State: Client side (Hidden field)**

**Code :**

File name:-WebForm1.aspx



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="hidden\_field\_sm\_.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:HiddenField ID="HiddenField1" runat="server" Value="0" />

<asp:Button ID="Button1" runat="server" Text="No action Button" OnClick="Button1\_Click" />

<br />

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

</div>

</form>

</body>

</html>

File name:-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 hidden\_field\_sm\_

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

if (HiddenField1.Value != null)

{

int val = Convert.ToInt32(HiddenField1.Value) + 1;

HiddenField1.Value = val.ToString();

Label1.Text = val.ToString();

}

}

protected void Button1\_Click(object sender, EventArgs e)

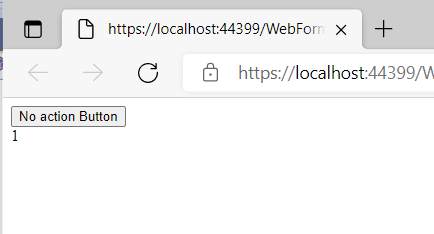
{

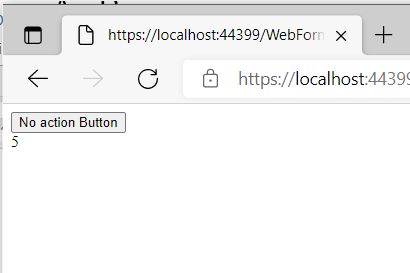
}

}

}

**Output:-**



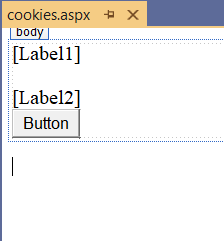


Practical No:-14

**Aim : .Create a windows applications on managing State: Client side (Persistent & Non Persistent Cookies)**

**Code :**

File name:-cookies.aspx



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="cookies.aspx.cs" Inherits="hidden\_field\_sm\_.cookies" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

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

<br />

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

</asp:Label><br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Button" />

<br />

</div>

</form>

</body>

</html>

File name:-cookies.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace hidden\_field\_sm\_

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.Cookies["Persistance"] != null)

Label1.Text = Request.Cookies["Persistance"].Value;

else

Label1.Text = "";

if (Request.Cookies["NonPersistance"] != null)

Label2.Text = Request.Cookies["NonPersistance"].Value;

}

protected void Button1\_Click(object sender, EventArgs e)

{

/\*Persistent Cookies\*/

HttpCookie aCookieValPer = new HttpCookie("Persistance");

aCookieValPer.Value = "This is A Persistance Cookie";

aCookieValPer.Expires = DateTime.Now.AddSeconds(10);

Response.Cookies.Add(aCookieValPer);

/\*Non Persistent Cookies\*/

HttpCookie aCookieValNonPer = new HttpCookie("NonPersistance");

aCookieValNonPer.Value = "This is A Non Persistance Cookie";

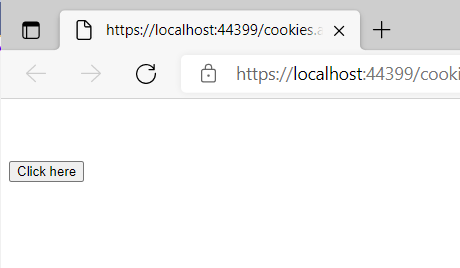
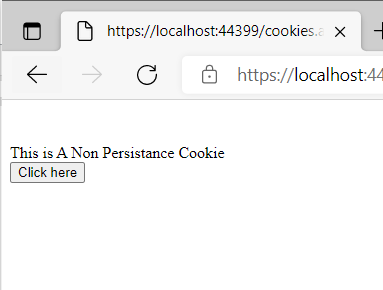
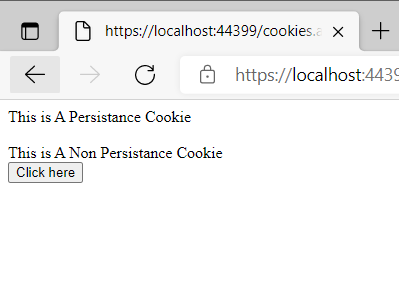
Response.Cookies.Add(aCookieValNonPer);

}

}

}

**Output:-**

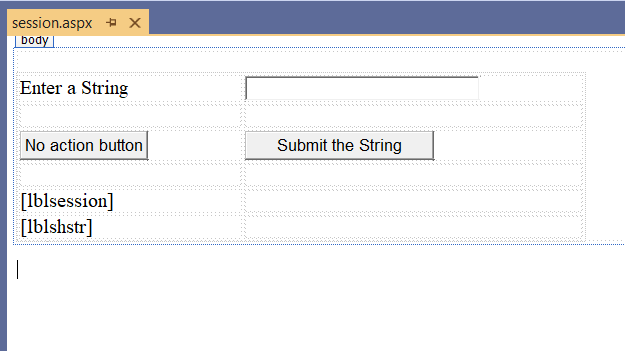


Practical No:-15

**Aim : Create a windows applications on managing State: Server Side(Session Management)**

**Code :**

File name:-session.aspx



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="session.aspx.cs" Inherits="hidden\_field\_sm\_.session" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

&nbsp; &nbsp; &nbsp;

<table style="width: 568px; height: 103px">

<tr>

<td style="width: 209px">

<asp:Label ID="lblstr" runat="server" Text="Enter a String"

style="width:94px">

</asp:Label>

</td>

<td style="width: 317px">

<asp:TextBox ID="txtstr" runat="server" style="width:227px">

</asp:TextBox>

</td>

</tr>

<tr>

<td style="width: 209px"> </td>

<td style="width: 317px"> </td>

</tr>

<tr>

<td style="width: 209px">

<asp:Button ID="btnnrm" runat="server"

Text="No action button" style="width:128px" />

</td>

<td style="width: 317px">

<asp:Button ID="btnstr" runat="server"

OnClick="btnstr\_Click" Text="Submit the String" />

</td>

</tr>

<tr>

<td style="width: 209px"> </td>

<td style="width: 317px"> </td>

</tr>

<tr>

<td style="width: 209px">

<asp:Label ID="lblsession" runat="server" style="width:231px" >

</asp:Label>

</td>

<td style="width: 317px"> </td>

</tr>

<tr>

<td style="width: 209px">

<asp:Label ID="lblshstr" runat="server">

</asp:Label>

</td>

<td style="width: 317px"> </td>

</tr>

</table>

</div>

</form>

</body>

</html>

File name:-session.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace hidden\_field\_sm\_

{

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

{

String mystr;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnstr\_Click(object sender, EventArgs e)

{

this.mystr = this.txtstr.Text;

this.Session["str"] = this.txtstr.Text;

this.lblshstr.Text = this.mystr;

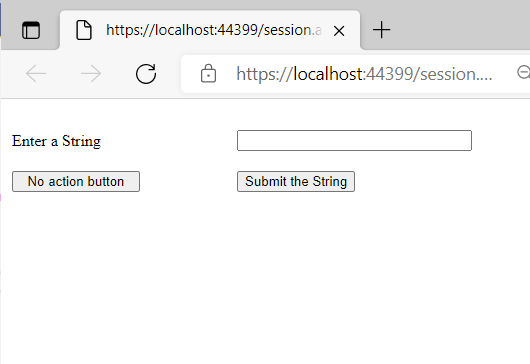
this.lblsession.Text = (String)this.Session["str"];

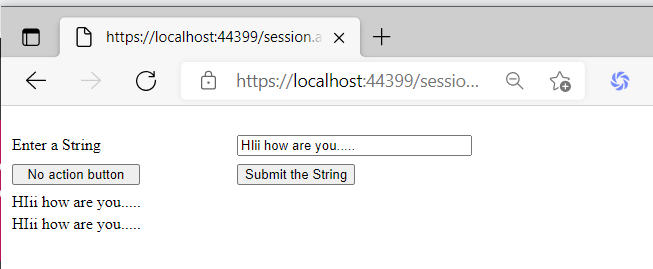
}

}

}

**Output:-**



[](https://drive.google.com/file/d/18iabSqyxlgtnWKVfatOTSkxKMPXtnk15/view?usp=drive_web&authuser=1)

Practical 16

**Aim: Display ASP.NET web page to demonstrate postback and crosspage posting with all web controls.**

**Web1.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Web1.aspx.cs" Inherits="Web1" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

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

&nbsp;&nbsp;

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

<br />

<br />

<asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>

<br />

<asp:Button ID="Button1" runat="server" Text="Same page post back" OnClick="Button1\_Click" />

&nbsp;&nbsp;

<asp:Button ID="Button2" runat="server" Text="cross page post back" PostBackUrl="~/web2.aspx" OnClick="Button2\_Click" />

<br />

<br />

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

<br />

<br />

<br />

</div>

</form>

</body>

</html>

**Web1.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Web1:System.Web.UI.Page {

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click(object sender,EventArgs e) {

Label2.Text = "Hi" + TextBox1.Text + ",here is the output of the same page post back button:" + Calendar1.SelectedDate.ToString();

}

protected void Button2\_Click(object sender,EventArgs e) {

}

}

**Web2**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Web2.aspx.cs" Inherits="Web2" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

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

</div>

</form>

</body>

</html>

Web2.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Web2:System.Web.UI.Page {

protected void Page\_Load(object sender,EventArgs e) {

Calendar Calendar1 = new Calendar();

TextBox TextBox1 = new TextBox();

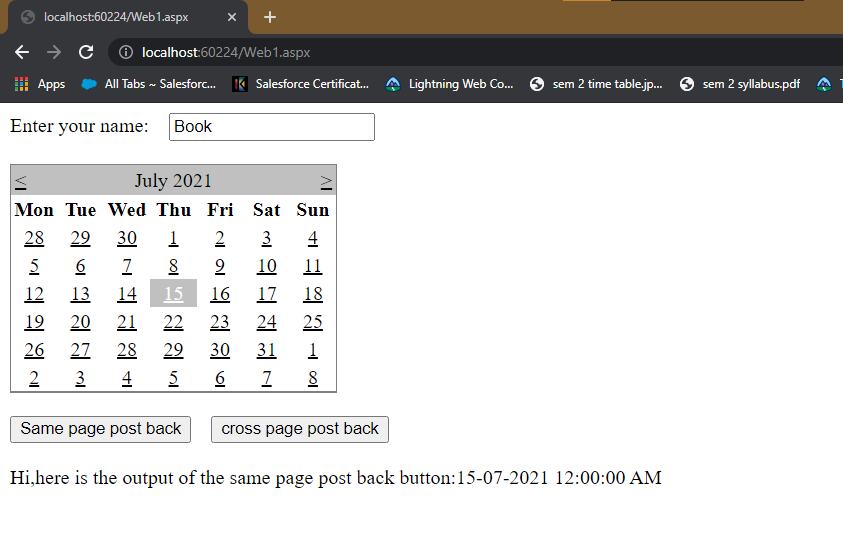
Calendar1 = (Calendar)PreviousPage.FindControl("Calendar1");

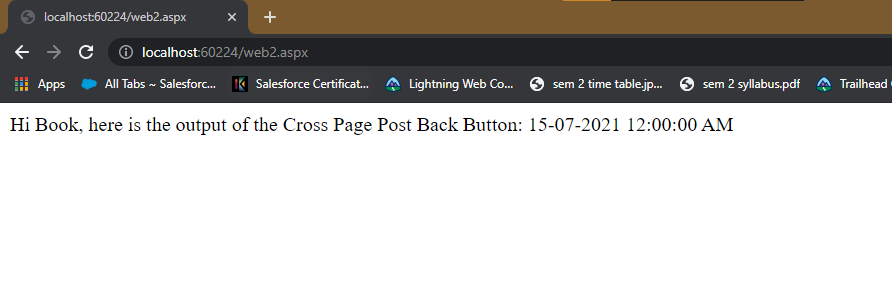
TextBox1 = (TextBox)PreviousPage.FindControl("TextBox1");

Label1.Text = "Hi " + TextBox1.Text + ", here is the output of the Cross Page Post Back Button: " + Calendar1.SelectedDate.ToString();

}

}





Practical 17

**Aim: Create ASP.NET program using master page & themes and skins.**

**Default2**

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.master" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h1>Welcome to Home Page</h1>

</asp:Content>

Default3

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.master" AutoEventWireup="true" CodeFile="Default3.aspx.cs" Inherits="Default3" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h1>Welcome to About Us</h1>

</asp:Content>

Default4

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.master" AutoEventWireup="true" CodeFile="Default4.aspx.cs" Inherits="Default4" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<h1>Welcome to Contact Us</h1>

</asp:Content>

Site1.master.cs

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

<link href="StyleSheet.css" rel="stylesheet" type="text/css" />

<asp:ContentPlaceHolder id="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

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

<div class="wrapper">

<div class="menu">

<ul>

<li><a href="Default2.aspx">Home Page</a></li>

<li><a href="Default3.aspx">Category</a></li>

<li><a href="Default4.aspx">About Us</a></li>

</ul>

</div>

<div class="content">

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

<div class="footer">

<h3>@ abc.com</h3>

</div>

</div>

</form>

</body>

</html>

StyleSheet.css

body

{ background-color:rgb(237,237,237);

font-family:Times New Roman;

font-size:13px;

}

.footer

{

background-color :rgb(10,110,178);

color:rgb(0,0,0);

position:fixed;

bottom:0px;

left:0px;

width:100%;

text-align:right;

}

ul{

list-style-type:none;

margin:0px;

padding:0px;

overflow:hidden;

}

li{

float:left;

}

li a{

display:block;

padding:14px 16px;

font-size:20px;

text-decoration:none;

}

.menu{

background-color:burlywood;

margin:auto;

}

li a:hover{

background-color:coral;

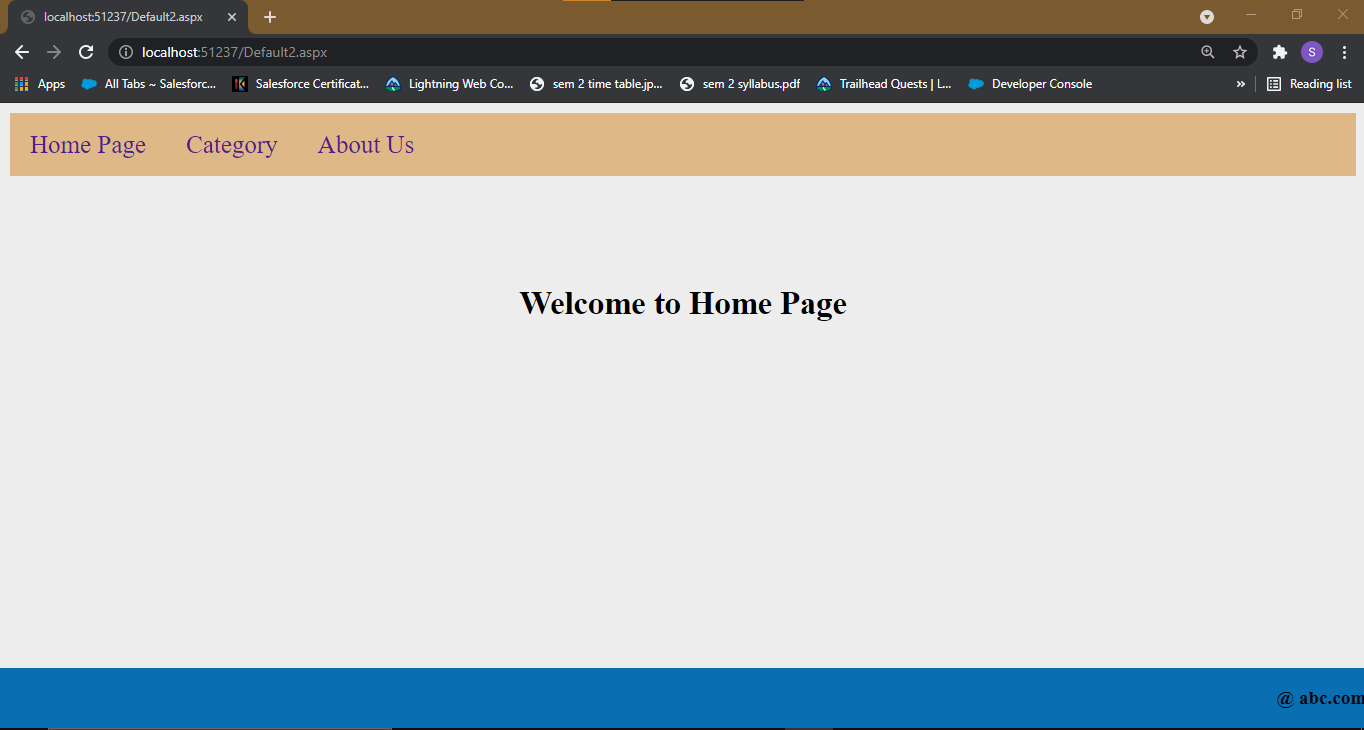
}

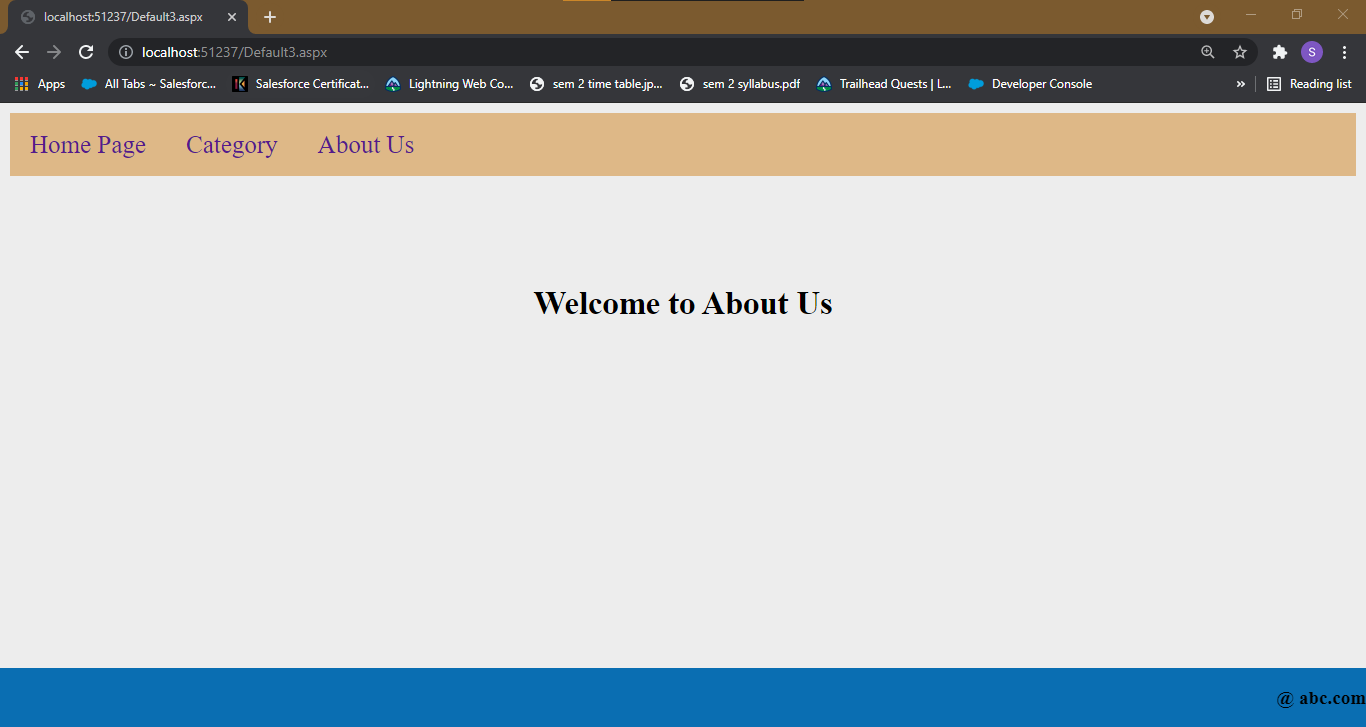
.content{

padding: 70px 0;

text-align:center;

}





Practical 18

**Aim: Create ASP.NET program based on validation controls.**

Valid.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<h1>VALIDATION CONTROLS: </h1>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<asp:Label ID="Label3" runat="server" Style="top: 241px; left: 32px; position: absolute; height: 22px; width: 128px; bottom: 54px;" Text="Enter your email id:"> </asp:Label>

<asp:Label ID="Label1" runat="server" Style="top: 65px; left: 31px; position: absolute; height: 22px; width: 128px" Text="Enter your name:"> </asp:Label>

<asp:TextBox ID="TextBox1" runat="server" Style="top: 66px; left: 212px; position: absolute; height: 22px; width: 128px; right: 765px;"> </asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" Style="top: 67px; left: 378px; position: absolute; height: 22px; width: 128px" ErrorMessage="RequiredFieldValidator" ControlToValidate="TextBox1"> name is mandatory </asp:RequiredFieldValidator>

</div>

<p>

<asp:Button ID="Button1" runat="server" Style="top: 311px; left: 267px; position: absolute; height: 26px; width: 61px"

Text="Submit" />

</p>

<asp:TextBox ID="TextBox3" runat="server" Style="top: 145px; left: 209px; position: absolute; height: 22px; width: 131px" TextMode="Password"> </asp:TextBox>

<p>

<asp:TextBox ID="TextBox2" runat="server" Style="top: 105px; left: 210px; position: absolute; height: 22px; width: 131px"

TextMode="Password"> </asp:TextBox>

<asp:Label ID="Label4" runat="server" Style="top: 105px; left: 31px; position: absolute; height: 22px; width: 128px" Text="Password"> </asp:Label>

<asp:TextBox ID="TextBox5" runat="server" Style="top: 239px; left: 210px; position: absolute; height: 22px; width: 132px"> </asp:TextBox>

</p>

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" Style="top: 104px; left: 367px; position: absolute; height: 26px; width: 162px" ErrorMessage="password required" ControlToValidate="TextBox2"> </asp:RequiredFieldValidator>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" Style="top: 145px; left: 367px; position: absolute; height: 26px; width: 162px" ErrorMessage="password required" ControlToValidate="TextBox3"> </asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator1" runat="server" Style="top: 149px; left: 512px; position: absolute; height: 26px; width: 162px" ErrorMessage="CompareValidator" ControlToValidate="TextBox3" ValueToCompare="hello"> </asp:CompareValidator>

<p>

<asp:Label ID="Label5" runat="server" Style="top: 148px; left: 32px; position: absolute; height: 22px; width: 128px; bottom: 147px;" Text="Confirm Password"> </asp:Label>

<asp:TextBox ID="TextBox4" runat="server" Style="top: 194px; left: 209px; position: absolute; height: 22px; width: 132px"> </asp:TextBox>

<asp:Label ID="Label6" runat="server" Style="top: 194px; left: 32px; position: absolute; height: 22px; width: 128px; bottom: 101px;" Text="Enter your age:"> </asp:Label>

</p>

<asp:RangeValidator ID="RangeValidator1" runat="server" Style="top: 194px; left: 365px; position: absolute; height: 22px; width: 105px" ErrorMessage="RangeValidator" ControlToValidate="TextBox4" MaximumValue="100" MinimumValue="18" Type="Integer"> </asp:RangeValidator>

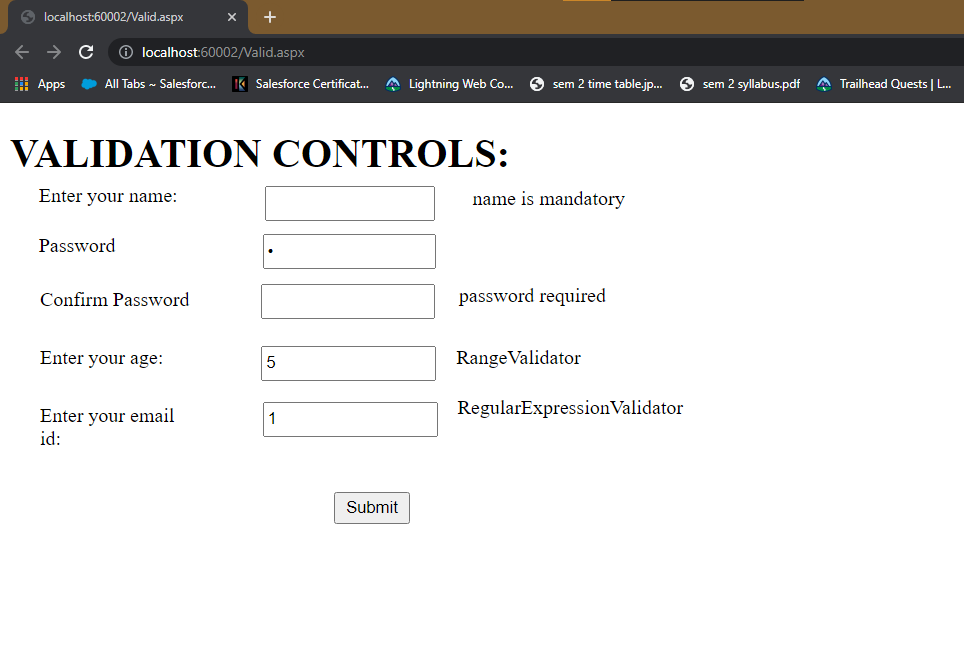
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" Style="top: 234px; left: 366px; position: absolute; height: 22px; width: 177px" ErrorMessage="RegularExpressionValidator" ControlToValidate="TextBox5" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*"> </asp:RegularExpressionValidator>

</form>

</body>

</html>

Output:



Practical 19

**Aim: Display digital clock using Ajax.**

Ajax.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Image ID="Image1" runat="server" Width="100" Height="100" ImageUrl="Image/photo.jpeg" />

<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

Current Time is:<br />

<asp:Timer ID="Timer1" runat="server" Interval="1000"

OnTick="Timer1\_Tick1">

</asp:Timer>

<asp:Label ID="Label1" runat="server" BackColor="#FF9999" BorderStyle="Ridge"

Font-Size="Larger"></asp:Label>

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

</html>

Ajax.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace AjaxPractical

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Timer1\_Tick1(object sender, EventArgs e)

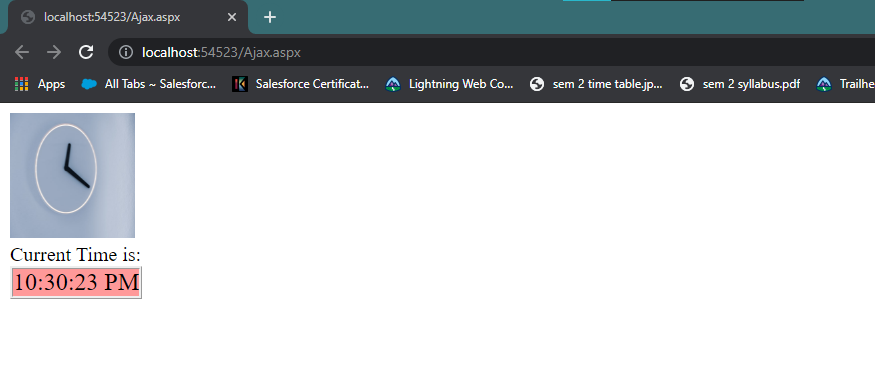
{

Label1.Text = DateTime.Now.ToLongTimeString();

}

}

}



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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

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

<br />

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

<br />

Enter Mobile No : <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>

<br />

<br />

<asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1\_Click" />

<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

<asp:Timer ID="Timer1" runat="server" Interval="1000" OnTick="Timer1\_Tick"></asp:Timer>

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

</ContentTemplate>

</asp:UpdatePanel>

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

</div>

</form>

</body>

</html>

Practical 20

**Aim: Design a registration form with current time as one field, update the time using Ajax while you are entering details in registration form.**

Registration.aspx

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

Label2.Text = TextBox1.Text +" " + TextBox2.Text +" "+ TextBox3.Text;

}

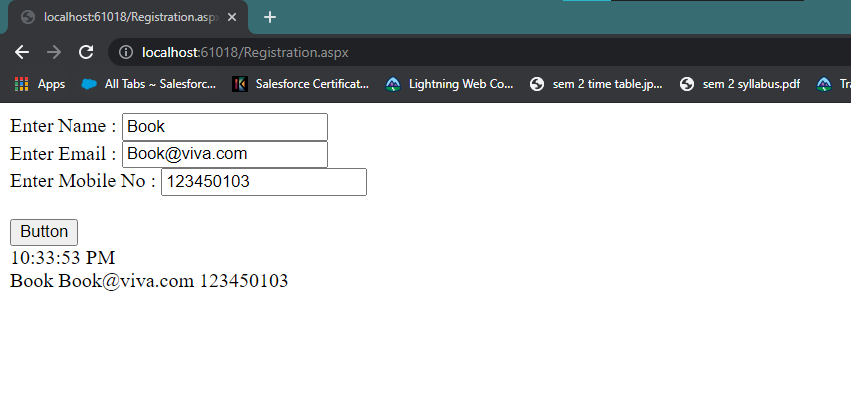
protected void Timer1\_Tick(object sender, EventArgs e)

{

Label1.Text = DateTime.Now.ToLongTimeString();

}

}



Practical 21

**Aim: Design a web service to access the method of BankAccount class, consume this web service using web client.**

BankWebService

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

/// <summary>

/// Summary description for BankWebService

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class BankWebService : System.Web.Services.WebService

{

public BankWebService()

{

//Uncomment the following line if using designed components

//InitializeComponent();

}

[WebMethod]

public string HelloWorld()

{

return "Hello World";

}

[WebMethod]

public string getName(string nm)

{

return nm;

}

[WebMethod]

public int getAccountNumber(int AccNum)

{

return AccNum;

}

}

Default.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<table align="center">

<caption>Account Details</caption>

<tr>

<td>Enter Name</td>

<td>

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

</td>

</tr>

<tr>

<td>Enter Account No </td>

<td>

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

</td>

</tr>

<tr>

<td colspan="2" align="center">

<asp:Button ID="Button1" runat="server" Text="Show" OnClick="Button1\_Click" />

</td>

</tr>

<tr>

<td colspan="2" align="center">

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

</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;

public partial class \_Default : System.Web.UI.Page

{

BankWebService obj = new BankWebService();

protected void Page\_Load(object sender, EventArgs e)

{

}

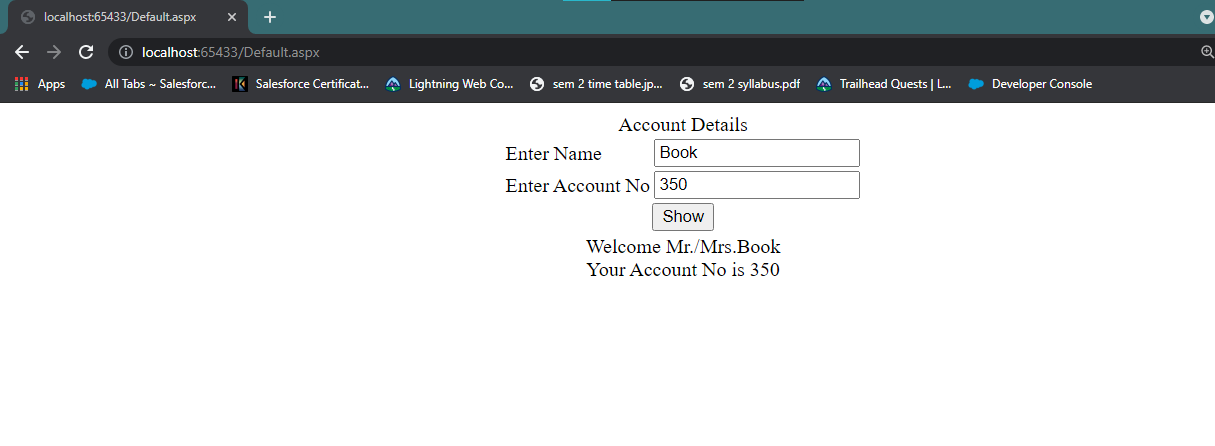
protected void Button1\_Click(object sender, EventArgs e)

{

Label1.Text = "Welcome Mr./Mrs." + obj.getName(TextBox1.Text) + "<br /> Your Account No is " + obj.getAccountNumber(Convert.ToInt32(TextBox2.Text)).ToString();

}

}



Practical 22

**Aim: Create a web service that returns all student details from student table. Write windows application that user this service to display student details in a DataGridView control.**

WebForm1.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="webservice\_database.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:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:CollegeConnectionString %>" SelectCommand="SELECT \* FROM [Course]"></asp:SqlDataSource>

<asp:GridView ID="GridView1" runat="server"></asp:GridView>

<asp:Button ID="Button1" runat="server" Text="Show Courses" OnClick="Button1\_Click" />

</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 webservice\_database {

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

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click(object sender,EventArgs e) {

WebService1 obj = new WebService1();

GridView1.DataSource = obj.getCourse();

GridView1.DataBind();

}

}

}

**WebService1.asmx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Data.SqlClient;

using System.Configuration;

using System.Data;

namespace webservice\_database {

/// <summary>

/// Summary description for WebService1

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class WebService1:System.Web.Services.WebService {

[WebMethod]

public DataSet getCourse() {

string con = ConfigurationManager.ConnectionStrings["CollegeConnectionString"].ConnectionString;

using (SqlConnection conn = new SqlConnection(con)) {

using (SqlCommand cmd = new SqlCommand("select \* from Course ",conn)) {

using (SqlDataAdapter da = new SqlDataAdapter(cmd)) {

DataSet ds = new DataSet();

da.Fill(ds);

return ds;

}

}

}

}

}

}







Practical 23

**Aim: Design WCF service for a simple arithmetic calculator; consume the service using a web client.**

**Customer.aspx**

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="Label1" runat="server" Text="Enter number 1: "></asp:Label>&nbsp;&nbsp;&nbsp;&nbsp;

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

<br>

<asp:Label ID="Label2" runat="server" Text="Enter number 2: "></asp:Label>&nbsp;&nbsp;&nbsp;&nbsp;

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

<br>

<asp:Label ID="Label3" runat="server" Text="Result: "></asp:Label>&nbsp;&nbsp;&nbsp;

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

<br>

<asp:Button ID="Button1" runat="server" Text="Add" OnClick="Button1\_Click" />

&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="Button2" runat="server" Text="Subtract"

OnClick="Button2\_Click" />&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="Button3" runat="server" Text="Multiply"

OnClick="Button3\_Click" />&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="Button4" runat="server" Text="Divide" OnClick="Button1\_Click" />

</div>

</form>

</body>

</html>

WebForm1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WCFAapplication {

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

ServiceReference1.Service1Client obj = new ServiceReference1.Service1Client();

protected void Page\_Load(object sender,EventArgs e) {

}

protected void Button1\_Click(object sender,EventArgs e) {

double x, y;

x = Convert.ToInt32(TextBox1.Text);

y = Convert.ToInt32(TextBox2.Text);

TextBox3.Text = obj.add(x,y).ToString();

}

protected void Button2\_Click(object sender,EventArgs e) {

double x, y;

x = Convert.ToInt32(TextBox1.Text);

y = Convert.ToInt32(TextBox2.Text);

TextBox3.Text = obj.sub(x,y).ToString();

}

protected void Button3\_Click(object sender,EventArgs e) {

double x, y;

x = Convert.ToInt32(TextBox1.Text);

y = Convert.ToInt32(TextBox2.Text);

TextBox3.Text = obj.mul(x,y).ToString();

}

protected void Button4\_Click(object sender,EventArgs e) {

double x, y;

x = Convert.ToInt32(TextBox1.Text);

y = Convert.ToInt32(TextBox2.Text);

TextBox3.Text = obj.div(x,y).ToString();

}

}

}

**Service1.svc.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace WCFAapplication {

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.

// NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc or Service1.svc.cs at the Solution Explorer and start debugging.

public class Service1:IService1 {

public double add(double a,double b) {

return a + b;

}

public double sub(double a,double b) {

return a - b;

}

public double mul(double a,double b) {

return a \* b;

}

public double div(double a,double b) {

return a / b;

}

}

}

WCFAapplication

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace WCFAapplication {

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.

[ServiceContract]

public interface IService1 {

[OperationContract]

double add(double a,double b);

[OperationContract]

double sub(double a,double b);

[OperationContract]

double mul(double a,double b);

[OperationContract]

double div(double a,double b);

}

}









Practical 24

**Aim: Design a simple MVC application to demonstrate use of ActionResult & ViewResult Method, ViewBag Object.**

HomeController

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MVC.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ViewResult Index()

{

int hour = DateTime.Now.Hour;

ViewBag.Greeting = hour < 12 ? "Good Moring" : "Good Day";

return View();

}

}

}

Index.cshtml

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Index</title>

</head>

<body>

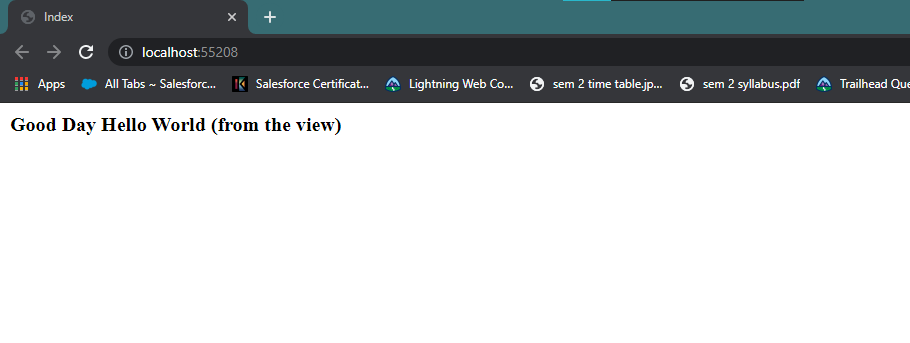
<div>

<b>@ViewBag.Greeting Hello World (from the view)</b>

</div>

</body>

</html>



Practical 25

**Aim: Design a simple Data-Entry Application( for Customer) with MVC using following:  
·  Creating & accessing strongly typed View & model  
·  Automatically implemented properties,  
·  Html helper Methods,  
·  Validations,  
·  Style sheet for highlighting Invalid fields  
·  Bootstrap functionality**

using CustomerMVC\_APP.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CustomerMVC\_APP.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

return View();

}

[HttpGet]

public ViewResult CustomerInput()

{

return View();

}

[HttpPost]

public ViewResult CustomerInput(Customer C1)

{

if (ModelState.IsValid)

{

return View("CustomerDisplayed", C1);

}

else

return View();

}

}

}

Customer

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace CustomerMVC\_APP.Models

{

public class Customer

{

[Required(ErrorMessage ="Please Enter ID")]

public int CustID { get; set;}

[Required(ErrorMessage = "Please Enter Name")]

public string CustName { get; set; }

[Required(ErrorMessage = "Please Enter Address")]

public string CustAdd { get; set; }

}

}

Index.cshtml

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<link href="~/Content/bootstrap.css" rel="stylesheet" />

<link href="~/Content/bootstrap-theme.css" rel="stylesheet" />

<title>Index</title>

<style>

</style>

</head>

<body >

<div class="text-center">

<b>Customer Information System to display all the customers details.</b>

<div class="btn-success">

@Html.ActionLink("CustomerInputLink", "CustomerInput")

</div>

</div>

</body>

</html>

CustomerInput.cshtml

@model CustomerMVC\_APP.Models.Customer

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>CustomerInput</title>

<link href="~/Content/StyleSheet1.css" rel="stylesheet" />

</head>

<body>

<div class="panel-success">

<div class="panel-heading">

<div class="panel-body">

@using (Html.BeginForm())

{

@Html.ValidationSummary()

<div class="form-group">

<p>Customer ID: @Html.TextAreaFor(x => x.CustID)</p>

<p>Customer Name: @Html.TextAreaFor(x => x.CustName)</p>

<p>Customer Address: @Html.TextAreaFor(x => x.CustAdd)</p>

<input id="Submit1" type="submit" value="submit" />

</div>

}

</div>

</div>

</div>

</body>

</html>

CustomerDisplayed.cshtml

@model CustomerMVC\_APP.Models.Customer

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>CustomerDisplayed</title>

</head>

<body>

<div>

<h1>Customer Information System</h1>

<p>

Customer ID : @Model.CustID

</p>

<p>

Customer Name : @Model.CustName

</p>

<p>

Customer Address : @Model.CustAdd

</p>

</div>

</body>

</html>

StyleSheet1.css

.field-validation-error{

color:#F00;

}

.field-validation-valid{

display:none;

}

.input-validation-error{

border:1px solid #f00;

background-color:#fee;

}

.validation-summary-errors{

font-weight:bold;

color:#f00;

}

.validation-summary-valid{

display:none;

}

