**Practical - 1**

**1. Working with basic C# and ASP.NET**

**a. Create an application that obtains four int values from the user and displays the product.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication3

{

class Program

{

static void Main(string[] args)

{

int a, b, c, d, prod;

Console.Write("Enter first value: ");

a = int.Parse(Console.ReadLine());

Console.Write("Enter second value: ");

b = int.Parse(Console.ReadLine());

Console.Write("Enter third value: ");

c = int.Parse(Console.ReadLine());

Console.Write("Enter fourth value: ");

d = int.Parse(Console.ReadLine());

prod = a + b + c + d;

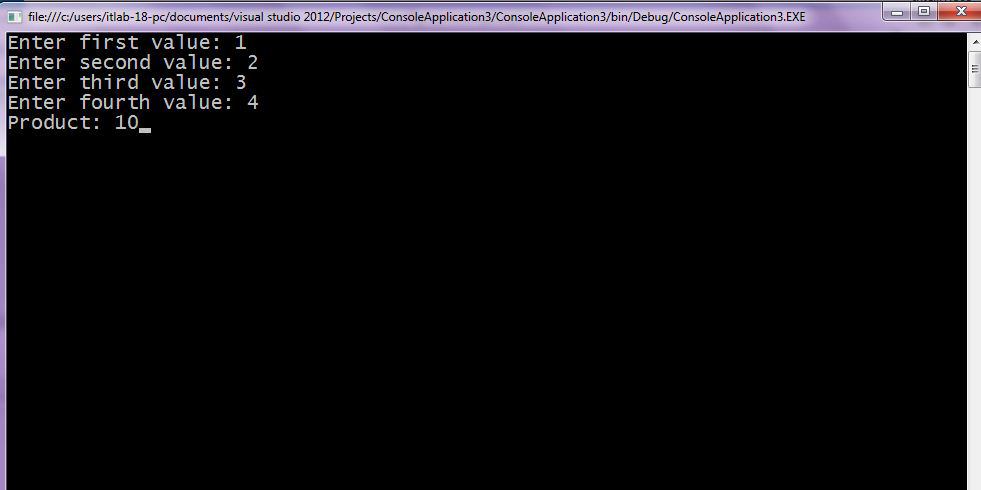
Console.Write("Product: "+prod);

Console.ReadKey();

}

}

}

**Output:**

**b. Create an application to demonstrate string operations.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication5

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter a string:");

string s=Console.ReadLine();

Console.WriteLine(s.Length);

Console.WriteLine(s.ToUpper());

Console.WriteLine(s.ToLower());

Console.WriteLine(s.Contains("Programming"));

Console.WriteLine(s.Trim().Length);

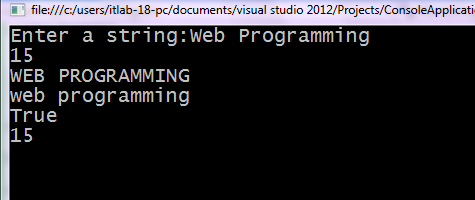
Console.ReadKey();

}

}

}

**Output:**



**c. Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("\_Enter Student Details\_");

int stud\_id;

string stud\_name, course\_name,dob;

Console.Write("Enter Name: ");

stud\_name = Console.ReadLine();

Console.Write("Enter ID: ");

stud\_id = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter Course Name: ");

course\_name = Console.ReadLine();

Console.Write("Enter Date of Birth: ");

dob = Console.ReadLine();

Console.WriteLine("\_Student Details\_");

Console.WriteLine("Name of Student " + stud\_name);

Console.WriteLine("Student ID: " + stud\_id);

Console.WriteLine("Course Name: "+course\_name);

Console.WriteLine("Date of Birth: " +dob);

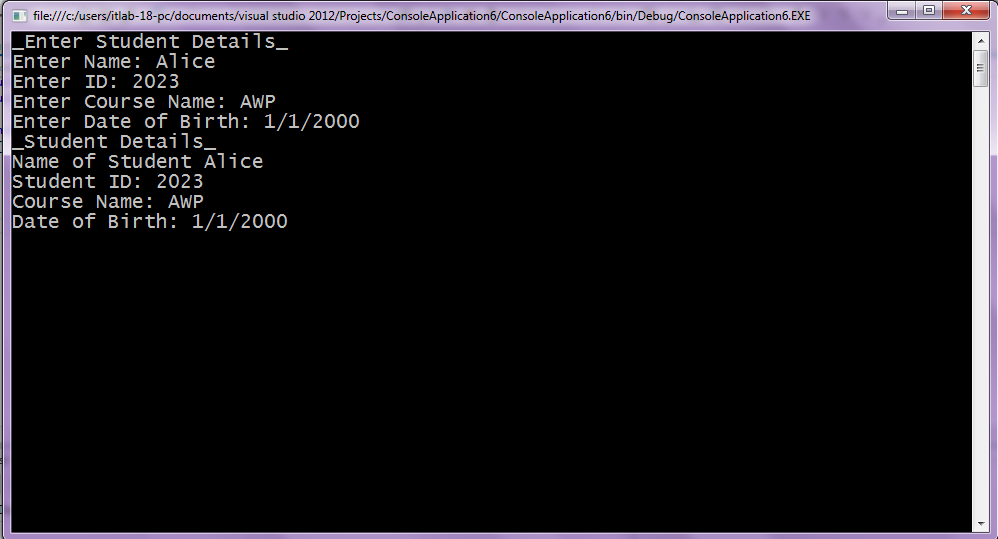
Console.ReadKey();

}

}

}

**Output:**



**Create an application to demonstrate following operations.**

**i. Generate Fibonacci Series**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

int f1, f2, f, i, num;

Console.Write("Enter Value: ");

num = Convert.ToInt32(Console.ReadLine());

f1 = 0;

f2 = 1;

Console.WriteLine(f1);

Console.WriteLine(f2);

for (i = 2; i <= num; i++)

{

f = f1 + f2;

Console.WriteLine(f);

f1 = f2;

f2 = f;

}

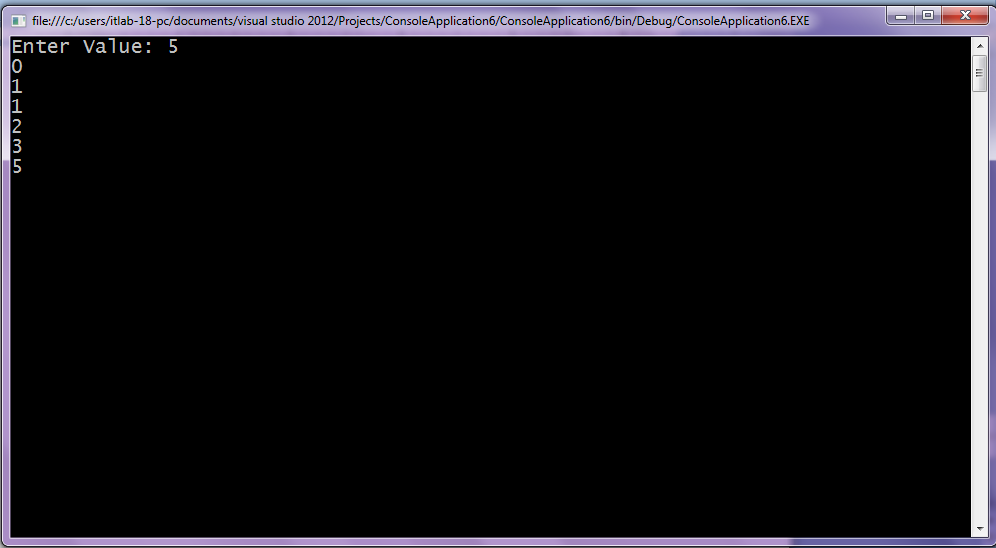
Console.ReadKey();

}

}

}

**Output:**



**ii. Test for prime numbers.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

int num, i;

Console.Write("Enter the Number: ");

num = int.Parse(Console.ReadLine());

for (i = 2; i <= num/2; i++)

{

if (num % i == 0)

break;

}

if (num == 1)

{

Console.WriteLine(num+" is neither prime nor composite");

}

else if(i<(num/2))

{

Console.WriteLine(num+" is not a prime number");

}

else

{

Console.WriteLine(num + " is a prime number");

}

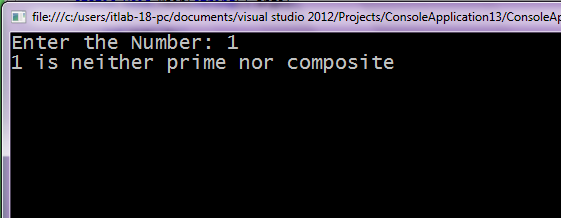
Console.ReadKey();

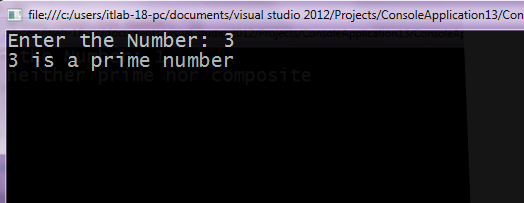
}

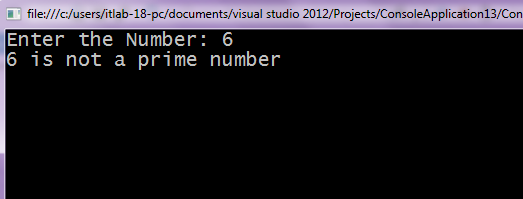
}

}

**Output:**







**iii.** **Test for vowels.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication10

{

class Program

{

static void Main(string[] args)

{

char ch;

Console.Write("Enter a Character: ");

ch = (char)Console.Read();

switch (ch)

{

case 'A':

case 'E':

case 'I':

case 'O':

case 'U':

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

Console.WriteLine(ch+ " is vowel");

break;

default:

Console.Write(ch+" is not vowel");

break;

}

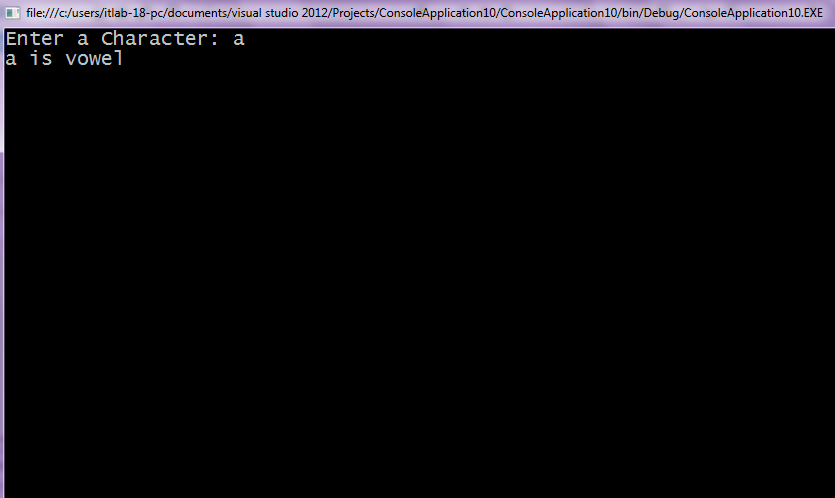
Console.ReadKey();

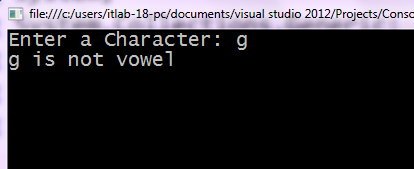
}

}

}

**Output:**

****

****

**iv. Use of foreach loop with arrays.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication11

{

class Program

{

static void Main(string[] args)

{

string[] str = { "BSCTY", "IT", "AWP" };

foreach (String s in str)

{

Console.WriteLine(s);

}

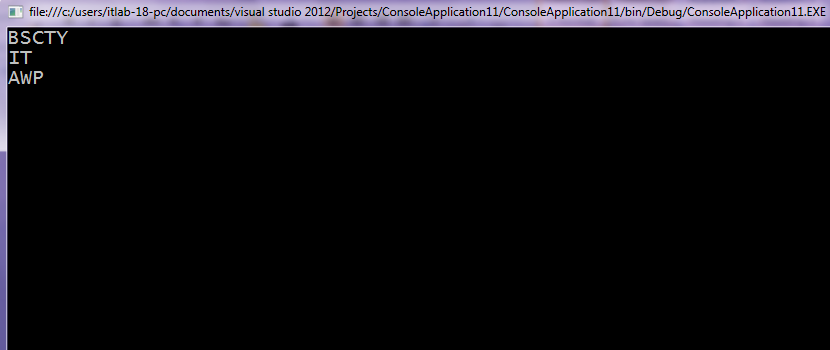
Console.ReadKey();

}

}

}

**Output:**



**v. Reverse a number and find sum of digits of a number.**

**Code:-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

static void Main(string[] args)

{

int num, n, rev = 0, sum = 0;

Console.Write("Enter the Number: ");

num = Convert.ToInt32(Console.ReadLine());

while (num != 0)

{

n = num % 10;

sum = n + sum;

rev = rev \* 10 + n;

num = num / 10;

}

Console.WriteLine("Reverse of a given number: "+rev);

Console.WriteLine("Sum of a given number: " +sum);

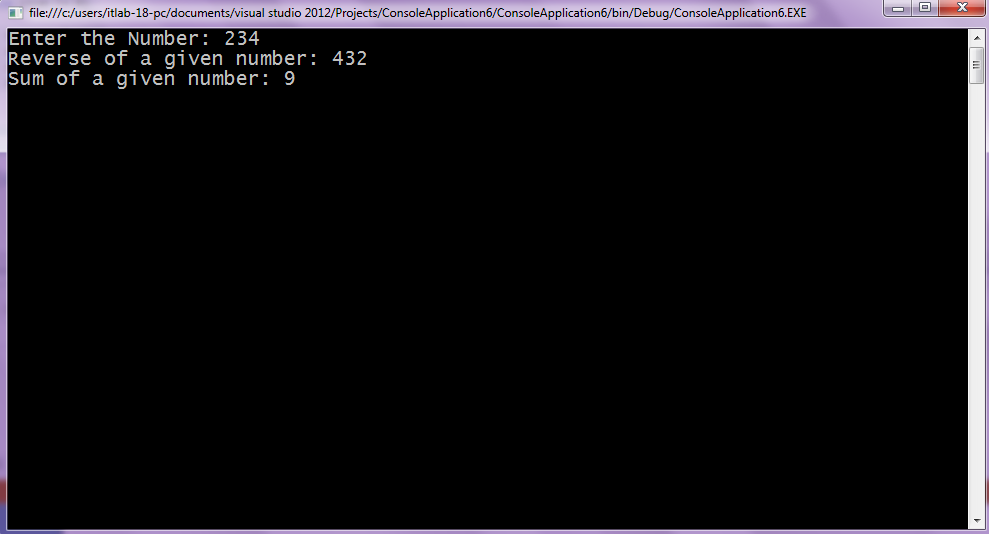
Console.ReadKey();

}

}

}

**Output:**



**Practical - 2**

**2. Working with Object Oriented C# and ASP.NET**

**a. Create simple application to perform following operations.**

**i) Finding Factorial Value.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace factorial

{

class Program

{

static void Main(string[] args)

{

int i, fact = 1, num;

Console.Write("Enter any number=");

num = int.Parse(Console.ReadLine());

for (i = 1; i <= num; i++)

{

fact = fact \* i;

}

Console.Write("Factorial of " + num + " is " + fact);

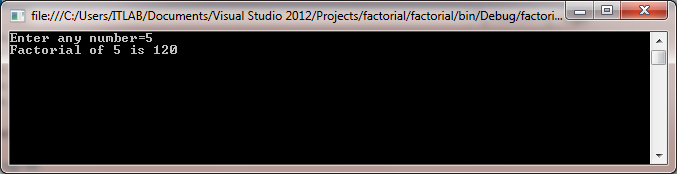
Console.ReadLine();

}

}

}

**Output:**



**ii) Money Conversion.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Money\_Conversion

{

class Program

{

static void Main(string[] args)

{

Double usd, inr, exchnge;

Console.WriteLine("Enter amount to be converted");

usd = Double.Parse(Console.ReadLine());

Console.WriteLine("Enter the current exchange rate.");

exchnge = Double.Parse(Console.ReadLine());

inr = usd \* exchnge;

Console.WriteLine("{0} Dollar = {1} INR", usd, inr);

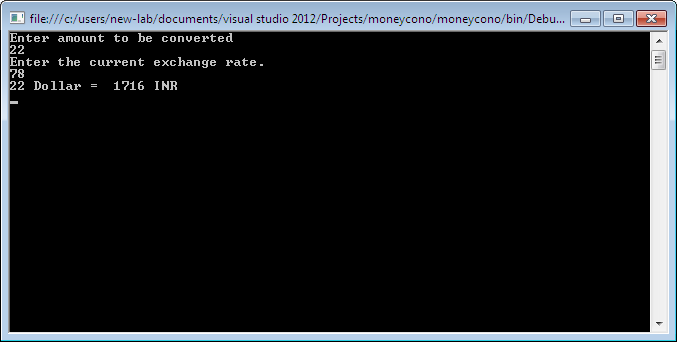
Console.ReadKey();

}

}

}

**Output:**



**iii) Quadratic Equation.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace quadratic

{

class Program

{

static void Main(string[] args)

{

double a, b, c, d1, d2, d3, r1, r2;

Console.WriteLine("Enter the value of a=");

a = double.Parse(Console.ReadLine());

Console.WriteLine("enter the value of b=");

b = double.Parse(Console.ReadLine());

Console.WriteLine("enter the value of c=");

c = double.Parse(Console.ReadLine());

d1 = b \* b;

d2 = 4 \* a \* c;

d3 = Math.Sqrt(d1 - d2);

r1 = (-b + d1) / (2 \* a);

r2 = (-b - d1) / (2 \* a);

Console.WriteLine("first root is " + r1);

Console.WriteLine("second root is " + r2);

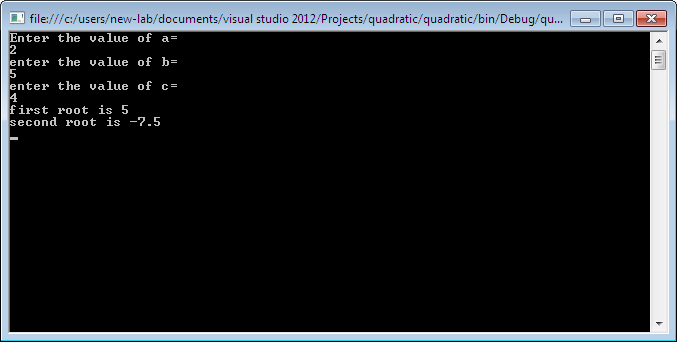
Console.ReadLine();

}

}

}

**Output:**

****

**iv) Temperature Conversion.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Temperature\_Conversion

{

class Program

{

static void Main(string[] args)

{

double fahrenheit,celsius;

Console.WriteLine("Enter the temperature:");

celsius = Double.Parse(Console.ReadLine());

Console.WriteLine("Celsius: " + celsius);

fahrenheit = (celsius \* 9) / 5 + 32;

Console.WriteLine("Fahrenheit: " + fahrenheit);

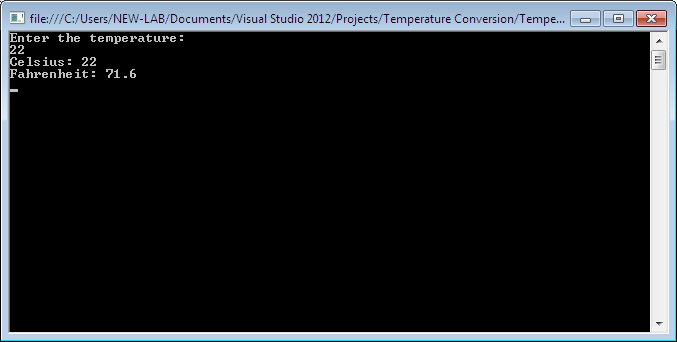
Console.ReadLine();

}

}

}

**Output:**

****

**b. Create simple application to demonstrate use of following concepts**

**i) Function Overloading.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace func\_overloading

{

class operaton

{

double result;

public void area1(int x)

{

result = x \* x;

Console.WriteLine("Area of Square=" + result);

}

public void area2(int x, int y)

{

result = x \* y;

Console.WriteLine("Area of Rectangle=" + result);

}

public void area3(double p, int x, int y)

{

result =p \* x \* y;

Console.WriteLine("Area of Triangle=" + result);

}

public void area4(int r)

{

result = 3.14 \* r \* r;

Console.WriteLine("Area of Circle=" + result);

}

}

class Program

{

static void Main(string[] args)

{

operaton o1= new operaton();

o1.area1(5);

o1.area2(5, 2);

o1.area3(0.5, 5, 2);

o1.area4(5);

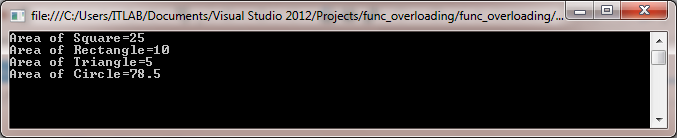
Console.ReadLine();

}

}

}

**Output:**

****

**ii) Inheritance (all types).**

**I) Single Level Inheritance**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SingleInheritance

{

public class Employee

{

public float salary = 40000;

}

public class Programmer: Employee

{

public float bonus = 10000;

}

class Program

{

static void Main(string[] args)

{

Programmer p1 = new Programmer();

Console.WriteLine("Salary:" + p1.salary);

Console.WriteLine("Bonus:" + p1.bonus);

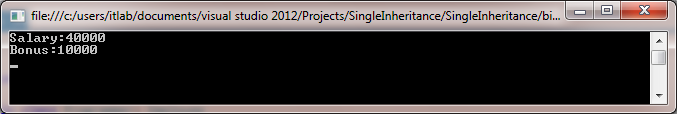
Console.ReadLine();

}

}

}

**Output:**

****

**II) Multi Level Inheritance**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Multi\_Level\_Inheritance

{

public class Animal

{

public void eat() { Console.WriteLine("Eating...."); }

}

public class Dog:Animal

{

public void bark() { Console.WriteLine("Barking...."); }

}

public class BabyDog:Dog

{

public void weep() { Console.WriteLine("weeping...."); }

}

class Program

{

static void Main(string[] args)

{

BabyDog d1 = new BabyDog();

d1.eat();

d1.bark();

d1.weep();

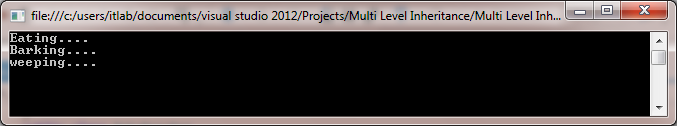
Console.ReadLine();

}

}

}

**Output:**

****

**III) Hierarchical Inheritance.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HierarchicalInheritance

{

public class parent

{

public void parentF()

{

Console.WriteLine("A and b are my parent");

}

}

public class ChildC : parent

{

public void ChildCF()

{

Console.WriteLine("I am ChildC");

}

}

public class ChildD : parent

{

public void ChildDF()

{

Console.WriteLine("I am ChildD");

}

}

class Program

{

static void Main(string[] args)

{

ChildC cc = new ChildC();

ChildD cd = new ChildD();

cc.ChildCF();

cc.parentF();

cd.ChildDF();

cd.parentF();

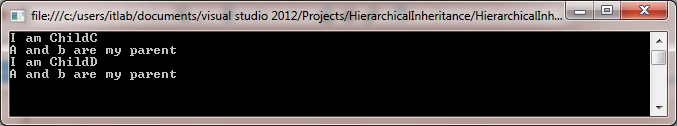
Console.ReadLine();

}

}

}

**Output:**

****

**IV) Multiple Inheritance using Interface.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace MultipleInheritApplication

{

interface calc1

{

int add(int a, int b);

}

interface calc2

{

int sub(int x, int y);

}

interface calc3

{

int mul(int r, int s);

}

interface calc4

{

int div(int c, int d);

}

class Calculation : calc1, calc2, calc3, calc4

{

public int result1;

public int add(int a, int b)

{

return result1 = a + b;

}

public int result2;

public int sub(int x, int y)

{

return result2 = x - y;

}

public int result3;

public int mul(int r, int s)

{

return result3 = r \* s;

}

public int result4;

public int div(int c, int d)

{

return result4 = c / d;

}

class Program

{

static void Main(string[] args)

{

Calculation c = new Calculation();

c.add(8, 2);

c.sub(20, 10);

c.mul(5, 2);

c.div(20, 10);

Console.WriteLine("Multiple Inheritance concept Using Interfaces :\n ");

Console.WriteLine("Addition: " + c.result1);

Console.WriteLine("Substraction: " + c.result2);

Console.WriteLine("Multiplication :" + c.result3);

Console.WriteLine("Division: " + c.result4);

Console.ReadKey();

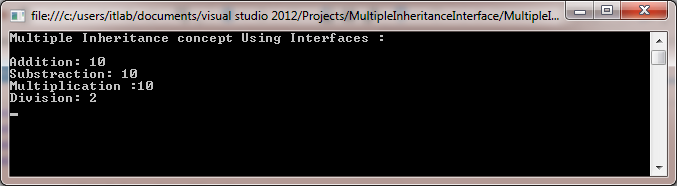
}

}

}

}

**Output:**

****

**iii) Constructor Overloading.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Constr\_Overloading\_EX

{

class ABC

{

public ABC()

{

Console.WriteLine("I am constructor of ABC With 0 parameter");

}

public ABC(string s)

{

Console.WriteLine("I am constructor of ABC With 1 parameter = " + s);

}

public ABC(string s1, string s2)

{

Console.WriteLine("I am constructor of ABC With 2 parameter" + s1 + " " + s2);

}

}

class Program

{

static void Main(string[] args)

{

ABC a1 = new ABC("Constructor", "Overlaoding");

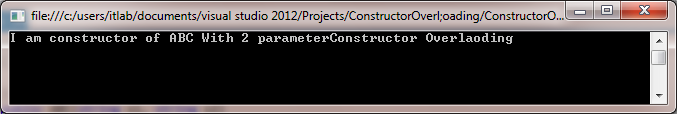
Console.ReadLine();

}

}

}

**Output:**

****

**iv) Interfaces.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace interfaceeg

{

interface college

{

void department();

}

interface year

{

void year();

}

class student: college, year

{

public void department()

{

Console.Write("IT DEPARTMENT");

}

public void year()

{

Console.Write("\nT.Y");

}

}

class Program

{

static void Main(string[] args)

{

student s1= new student();

s1.department();

s1.year();

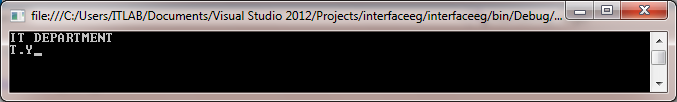
Console.ReadLine();

}

}

}

**Output:**

****

**c. Create simple application to demonstrate use of following concept:**

**i) Exception Handling.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exception\_Handling\_Example

{

class Program

{

static void Main(string[] args)

{

int[] numbers={10,11,12,13,14};

//Console.Write(numbers[0]);

//Console.Write(numbers[1]);

//Console.Write(numbers[2]);

//Console.Write(numbers[3]);

//Console.Write(numbers[4]);

try

{

Console.WriteLine(numbers[5]);

}

catch (Exception e)

{

Console.WriteLine("array out of limits, cannot accsss");

}

finally

{

Console.WriteLine("i alws execute as finally block");

}

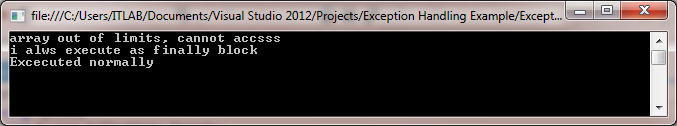
Console.Write("Excecuted normally");

Console.ReadKey();

}

}

}

**Output:**

**Practical – 3**

**3. Working with Web Forms and Controls**

**a. Demonstrate the use of Calendar control to perform following operations:**

**i. Display messages in calendar control**

**ii. Display vacation in a calendar control**

**iii. Selected day in a calendar control using style dates**

**iv. Difference between two calendar dates**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace calendarexample

{

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

{

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

{

}

protected void Calendar1\_DayRender(object sender, DayRenderEventArgs e)

{

if (e.Day.Date.Day == 5 && e.Day.Date.Month == 9)

{

e.Cell.BackColor = System.Drawing.Color.Green;

Label lbl = new Label();

lbl.Text = "<br> Teachers Day";

e.Cell.Controls.Add(lbl);

}

if (e.Day.Date.Day == 28 && e.Day.Date.Month == 9)

{

e.Cell.BackColor = System.Drawing.Color.Green;

Label lbl1 = new Label();

lbl1.Text = "<br> Zainab Birthday";

e.Cell.Controls.Add(lbl1);

}

if (e.Day.Date.Day == 22 && e.Day.Date.Month == 10)

{

Calendar1.SelectedDate = new DateTime(2022, 10, 22);

Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate,

Calendar1.SelectedDate.AddDays(16));

Label lbl2 = new Label();

lbl2.Text = "<br> Diwali Vacation";

e.Cell.Controls.Add(lbl2);

}

}

protected void Calendar1\_SelectionChanged(object sender, EventArgs e)

{

Label1.Text = "Your Selected Date:" + Calendar1.SelectedDate.Date.ToString();

}

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

{

Calendar1.Caption = "College";

Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;

Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;

Calendar1.TitleFormat = TitleFormat.Month;

Label2.Text = "Todays Date" + Calendar1.TodaysDate.ToShortDateString();

Label3.Text = "Diwali Vacation Start : 22-10-2022";

TimeSpan d = new DateTime(2022, 10, 22) - DateTime.Now;

Label4.Text = "Days remaining for Diwali Vacation" + d.Days.ToString();

TimeSpan d1 = new DateTime(2022, 12, 31) - DateTime.Now;

Label5.Text = "Days remaining for New Year" + d1.Days.ToString();

if (Calendar1.SelectedDate.ToShortDateString() == "22-10-2022")

Label3.Text = "<b> Diwali Vacation Start </b>";

if (Calendar1.SelectedDate.ToShortDateString() == "6-10-2022")

Label3.Text = "<b> Diwali Vacation End </b>";

}

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

{

Label1.Text = "";

Label2.Text = "";

Label3.Text = "";

Label4.Text = "";

Label5.Text = "";

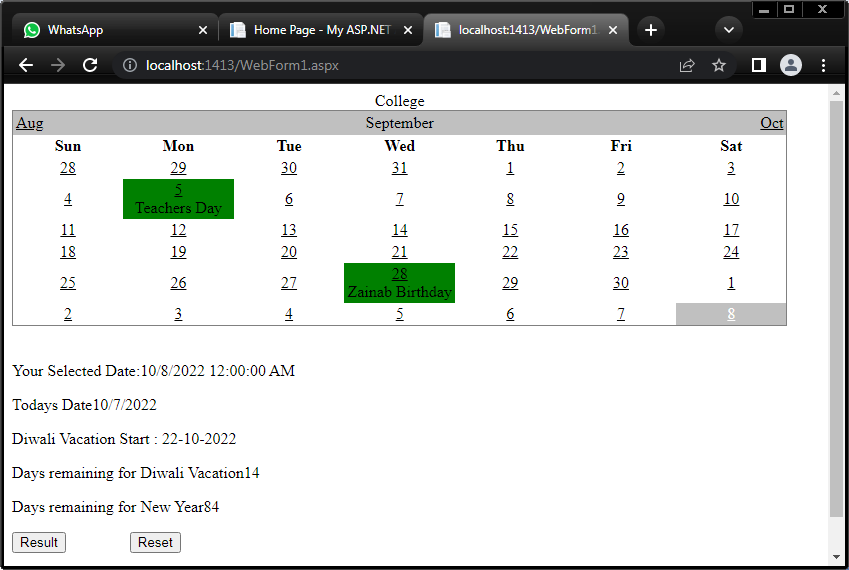
Calendar1.SelectedDates.Clear();

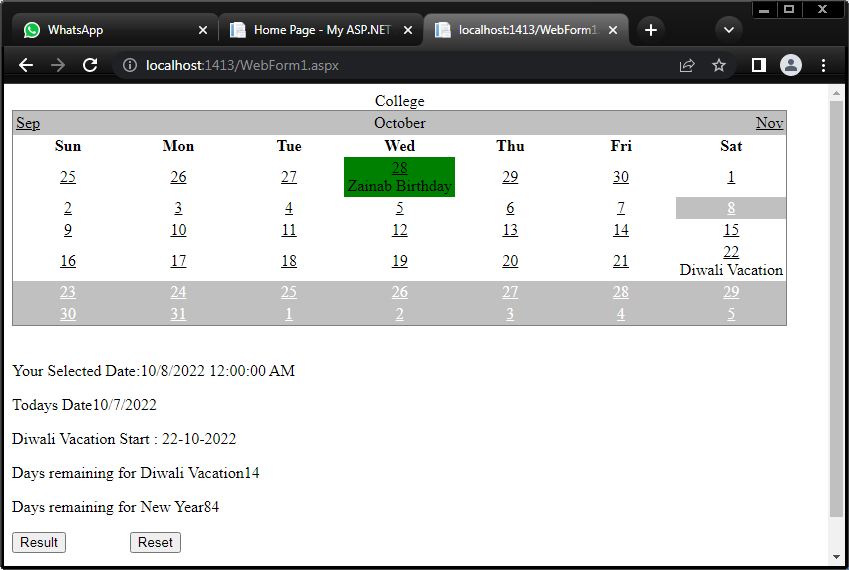
}

}

}

**Output:**





**b. Demonstrate the use of Treeview control perform following operations.**

**i. Treeview control and datalist**

**Code:**

**WebForm1.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="tree\_view.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

</div>

<asp:DataList ID="DataList1" runat="server">

<ItemTemplate>

<table class="table" border="2">

<tr>

<td> Roll no: <%# Eval("sid") %>

<br> </br>

Name: <%# Eval("sname") %>

<br> </br>

Class: <%# Eval("sclass") %>

</td>

</tr>

</table>

</ItemTemplate>

</asp:DataList>

<asp:TreeView ID="TreeView1" runat="server">

<Nodes>

<asp:TreeNode NavigateUrl="~/WebForm2.aspx" Text="Constructor" Value="Constructor"></asp:TreeNode>

<asp:TreeNode NavigateUrl="~/WebForm3.aspx" Text="Destructor" Value="Destructor"></asp:TreeNode>

<asp:TreeNode NavigateUrl="~/WebForm4.aspx" Text="OOP" Value="OOP"></asp:TreeNode>

<asp:TreeNode NavigateUrl="~/WebForm5.aspx" Text="Assembly" Value="Assembly"></asp:TreeNode>

</Nodes>

</asp:TreeView>

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

</form>

</body>

</html>

**WebForm2.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="tree\_view.WebForm2" %>

<!DOCTYPE html>

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

<head runat="server">

<title>Constructor</title>

</head>

<body>

<h1> Constructor</h1>

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

<div>

</div>

</form>

</body>

</html>

**WebForm3.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs" Inherits="tree\_view.WebForm3" %>

<!DOCTYPE html>

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

<head runat="server">

<title>Destructor</title>

</head>

<body>

<h1> Destructor</h1>

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

<div>

</div>

</form>

</body>

</html>

**WebForm4.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm4.aspx.cs" Inherits="tree\_view.WebForm4" %>

<!DOCTYPE html>

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

<head runat="server">

<title>OOP</title>

</head>

<body>

<h1> OOP</h1>

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

<div>

</div>

</form>

</body>

</html>

**WebForm5.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm5.aspx.cs" Inherits="tree\_view.WebForm5" %>

<!DOCTYPE html>

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

<head runat="server">

<title>Assembly</title>

</head>

<body>

<h1>E</h1>

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

<div>

<h1>Assembly</h1>

</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.Data;

namespace tree\_view

{

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

{

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

{

}

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

{

DataSet ds = new DataSet();

ds.ReadXml(Server.MapPath("XMLFile1.xml"));

DataList1.DataSource = ds;

DataList1.DataBind();

}

}

}

**XMLFile1.xml**

<?xml version="1.0" encoding="utf-8" ?>

<studentdetail>

<student>

<sid>1</sid>

<sname>John</sname>

<sclass>TYIT</sclass>

</student>

<student>

<sid>2</sid>

<sname>Alice</sname>

<sclass>TYIT</sclass>

</student>

<student>

<sid>3</sid>

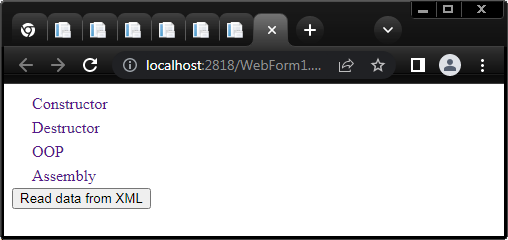
<sname>Tom</sname>

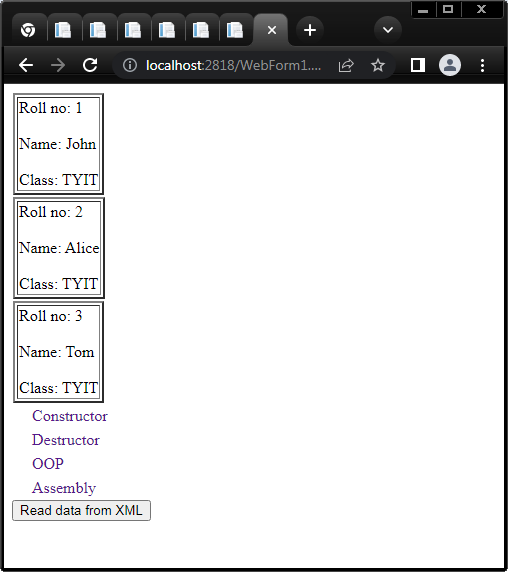
<sclass>TYIT</sclass>

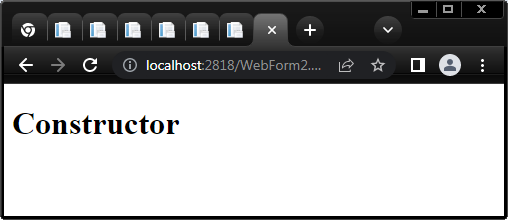
</student>

</studentdetail>

**Output:**







**Practical - 4**

**4. Working with Form Controls**

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

**Code:**

**Valid.aspx**

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div style="margin-left: 0px">

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

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

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="TextBox1" ErrorMessage="Name Should not be empty"></asp:RequiredFieldValidator>

<br />

<br />

Age&nbsp;&nbsp;&nbsp; <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

<asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="TextBox2" ErrorMessage="Age should be between 18 to 40" MaximumValue="40" MinimumValue="18"></asp:RangeValidator>

<br />

<br />

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

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

<asp:CustomValidator ID="CustomValidator1" runat="server" ControlToValidate="TextBox3" ErrorMessage="Enter Password" OnServerValidate="CustomValidator1\_ServerValidate"></asp:CustomValidator>

<br />

<br />

Confirm Password&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <asp:TextBox ID="TextBox4" runat="server" TextMode="Password"></asp:TextBox>

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

<asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="TextBox4" ControlToValidate="TextBox3" ErrorMessage="Not Equal"></asp:CompareValidator>

<br />

<br />

Email ID <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="TextBox5" ErrorMessage="In Valid Email Address" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*"></asp:RegularExpressionValidator>

&nbsp;<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="TextBox5" ErrorMessage="Email Should not be empty"></asp:RequiredFieldValidator>

<br />

<br />

<br />

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

</div>

</form>

</body>

</html>

**Valid.aspx.cs**

protected void CustomValidator1\_ServerValidate(object source, ServerValidateEventArgs args)

{

string str = args.Value;

args.IsValid = false;

if (str.Length < 7 || str.Length > 20)

{

return;

}

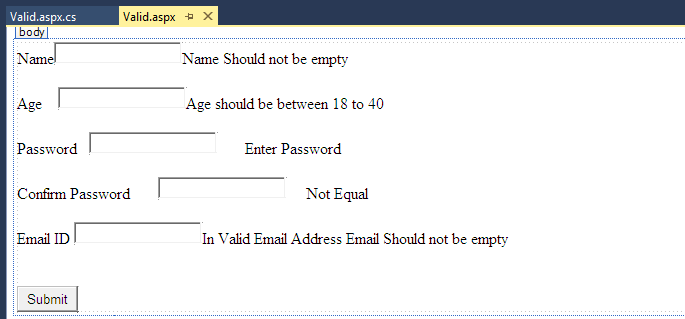
}

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

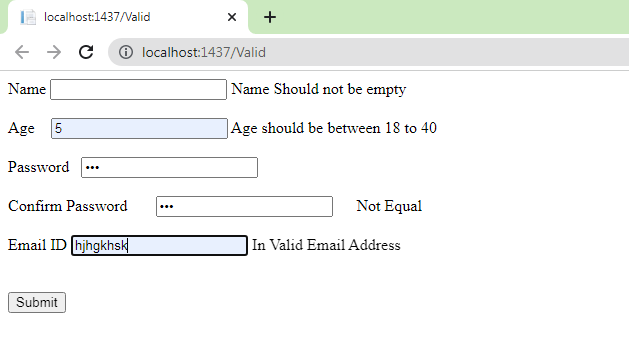
{

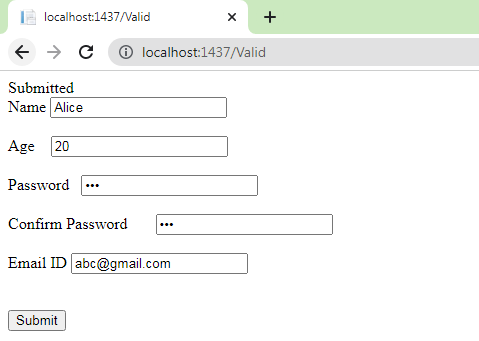
Response.Write("Submitted");

}

**Design**:

**Output:**





**b. Create web form to demonstrate use User Controls.**

**Code:**

**WebUserControl1.ascx**

<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="WebUserControl1.ascx.cs" Inherits="userControl.WebUserControl1" %>

<style type="text/css">

.auto-style1 {

width: 28%;

height: 65px;

}

.auto-style2 {

width: 67px;

}

.auto-style3 {

width: 166px;

}

</style>

<h3>This is User Contro1 </h3>

<p>

&nbsp;</p>

<table border="0" class="auto-style1">

<tr>

<td class="auto-style2">

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

</td>

<td class="auto-style3">

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

</td>

</tr>

<tr>

<td class="auto-style2">

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

</td>

<td class="auto-style3">

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

</td>

</tr>

<tr>

<td class="auto-style2">&nbsp;</td>

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

</tr>

<tr>

<td class="auto-style2">&nbsp;</td>

<td class="auto-style3">

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

</td>

</tr>

</table>

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

**WebUserControl1.ascx.cs**

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

{

Label3.Text = "Your Name is " + TextBox1.Text + " and you are from " + TextBox2.Text;

}

**WebForm1.aspx**

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

<%@ Register Src="~/WebUserControl1.ascx" TagPrefix="uc" TagName="Student"%>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

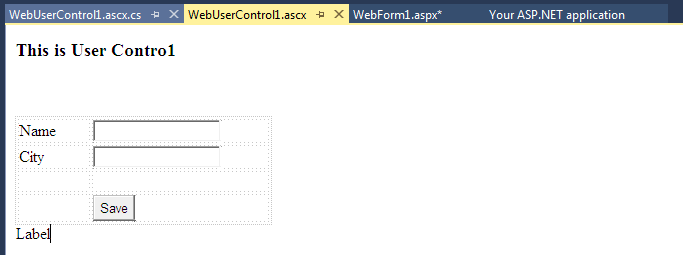
<uc:Student ID="studentcontrol" runat="server" />

</div>

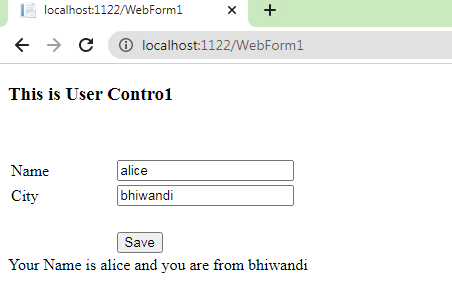
</form>

</body>

</html>

**Design:**

**Output:**



**Practical - 5**

**a. Create web form to demonstrate use of Website Navigation controls and Site Map.**

**Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="home.aspx.cs" Inherits="WebApplication20.home" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:TreeView ID="TreeView1" runat="server">

<Nodes>

<asp:TreeNode NavigateUrl="~/about us.aspx" Text="about" Value="about"></asp:TreeNode>

<asp:TreeNode NavigateUrl="~/Contact.aspx" Text="contact" Value="contact"></asp:TreeNode>

<asp:TreeNode NavigateUrl="~/review.aspx" Text="review" Value="review"></asp:TreeNode>

</Nodes>

</asp:TreeView>

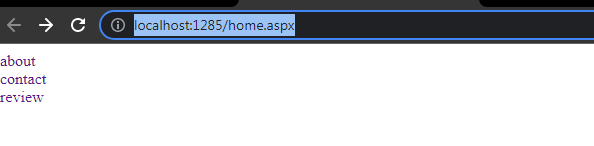
</div>

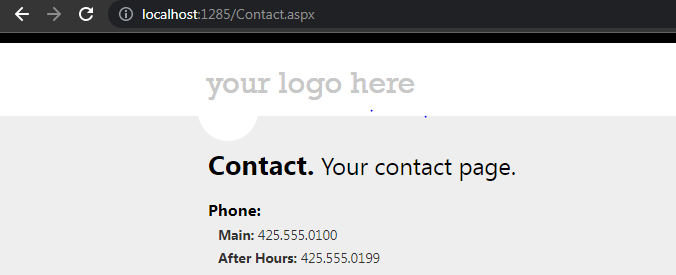
</form>

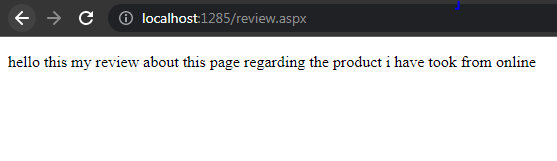
</body>

</html>

**Output:**

****

****

****

**b. Create a web application to demonstrate use of master page with applying styles and themes for page beautification.**

**Code:**

**Master code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication8

{

public partial class MyMaster : System.Web.UI.MasterPage

{

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

{

}

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

{

Response.Redirect("~/mYhOME.aspx");

}

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

{

Response.Redirect("~/MyContact.aspx");

}

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

{

Response.Redirect("~/Myabout.aspx");

}

}

}

**After clicking on the home button:**



**After clicking on the about button:**

**After clicking on the contact button:**



**c) Create a web application to demonstrate various states os ASP.NET pages.**

**Code:**

**WebForm1.aspx**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication37

{

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

{

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

{

if (IsPostBack)

{

if (ViewState["count"] != null)

{

int ViewStateVal = Convert.ToInt32(ViewState["count"]) + 1;

Label2.Text = "View State :" + ViewStateVal.ToString();

ViewState["count"] = ViewStateVal.ToString();

}

else

{

ViewState["count"] = "1";

}

}

}

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

{

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

}

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

{

if (HiddenField1.Value != null)

{

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

HiddenField1.Value = val.ToString();

}

}

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

{

HttpCookie h = new HttpCookie("name");

h.Value = TextBox1.Text;

Response.Cookies.Add(h);

Response.Redirect("WebForm2.aspx");

}

}

}

**WebForm2.aspx**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication37

{

public partial class WebForm2 : System.Web.UI.Page

{

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

{

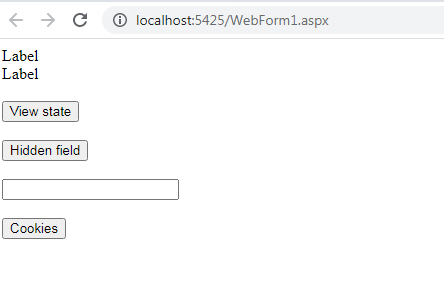
if (Request.Cookies["name"] != null)

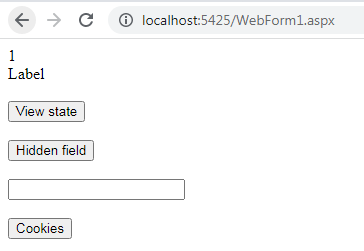
Response.Write("Welcome: " + Request.Cookies["name"].Value);

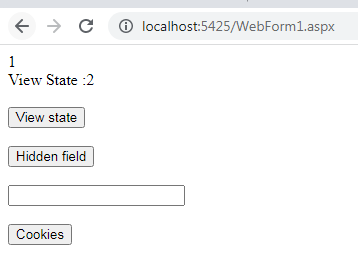
}

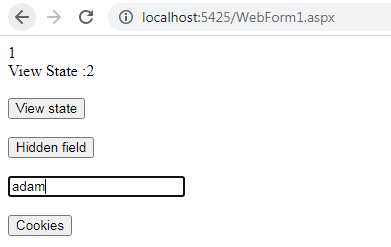
}

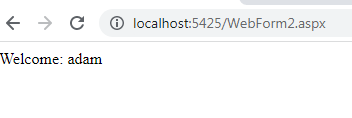
}

**Output:**

****

****

****

****

**Practical – 6**

**a. Create a web application to bind data in a multiline textbox by querying in another textbox.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace WebApplication14

{

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

{

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

{

}

protected void TextBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

string constr = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string a = "SELECT \* FROM [company].[dbo].[employee details] where Address='"+TextBox1.Text+"'";

SqlConnection con = new SqlConnection(constr);

con.Open();

SqlCommand cmd = new SqlCommand(a, con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

TextBox2.Text =TextBox2.Text + dr[0].ToString() +" "+ dr[1].ToString()+"\n";

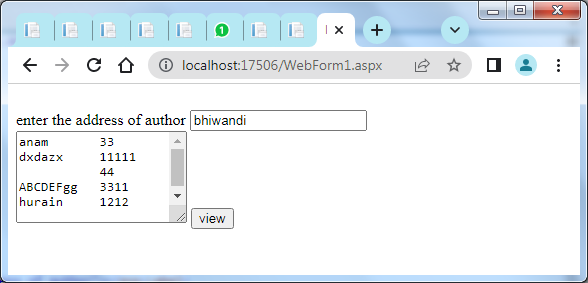
}

}

}

}

**Output**:



**b. Create a web application to display records by using database**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace WebApplication39

{

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

{

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

{

}

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

{

string conStr1 = "Data Source=ITLAB-18;Initial Catalog=Records;Integrated Security=True";

string Select1 = "SELECT \* FROM [Records].[dbo].[record1] where ID='"+TextBox1.Text+"'";

SqlConnection con1 = new SqlConnection(conStr1);

SqlCommand cmd1 = new SqlCommand(Select1, con1);

con1.Open();

SqlDataReader dr1 = cmd1.ExecuteReader();

while (dr1.Read())

{

Label1.Text = dr1[0].ToString();

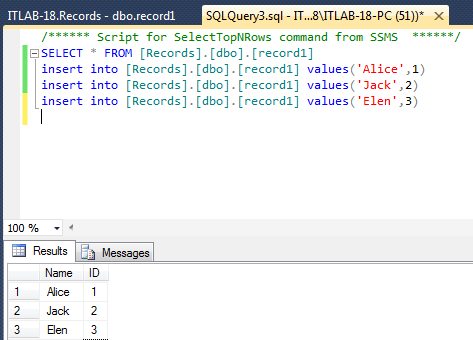
}

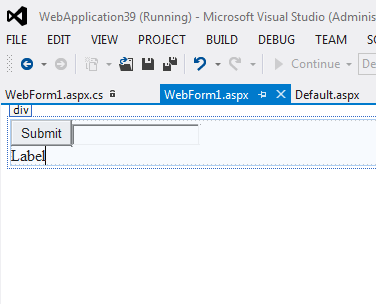
con1.Close();

}

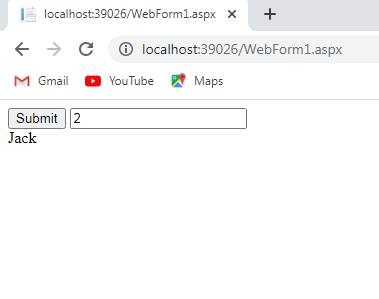
}

}





**Output:**



**c. Demonstrate the use of Data List Control.**

**Steps:**

1. Drag and drop data list control, sqldatasource control

2. Right click on the sql data source control

3. Click on configure data source.

4. Choose your data connection.

5. Select the table to connect.

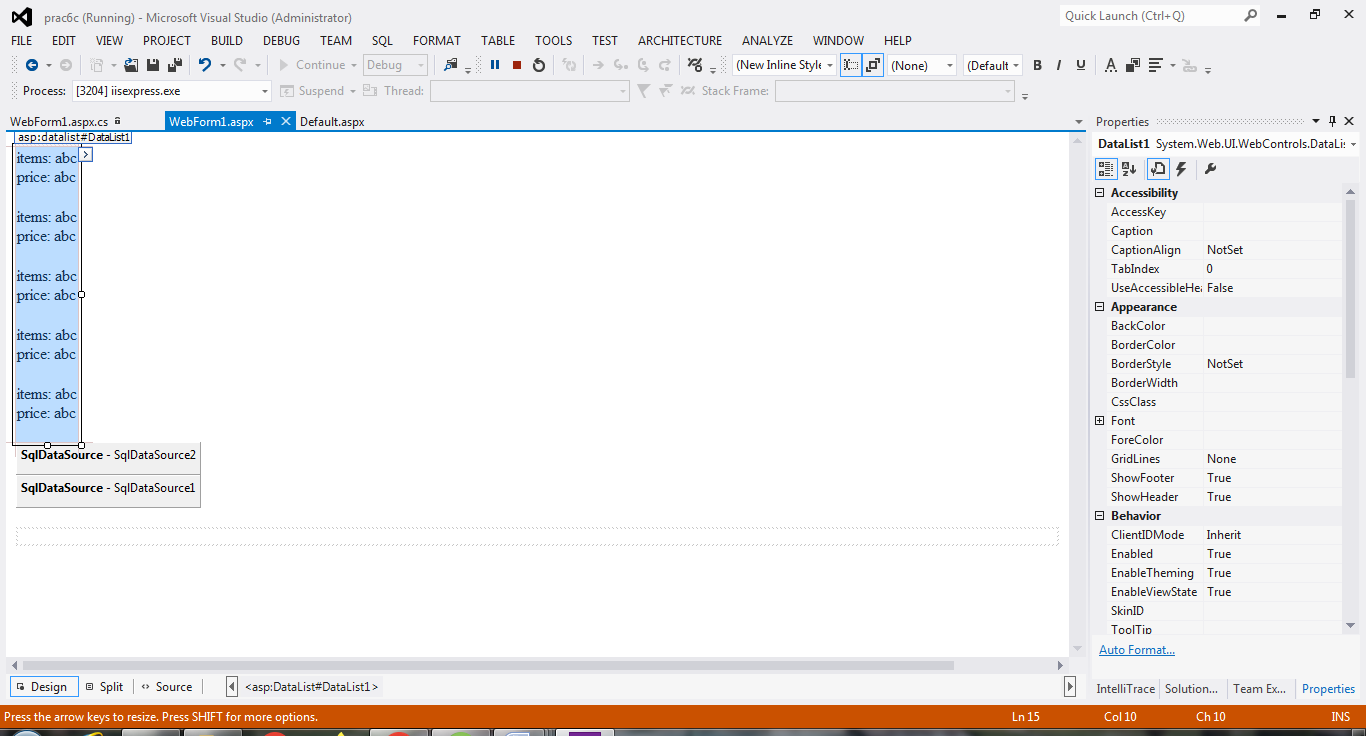
6. Test the query.

7. Click on finish button.

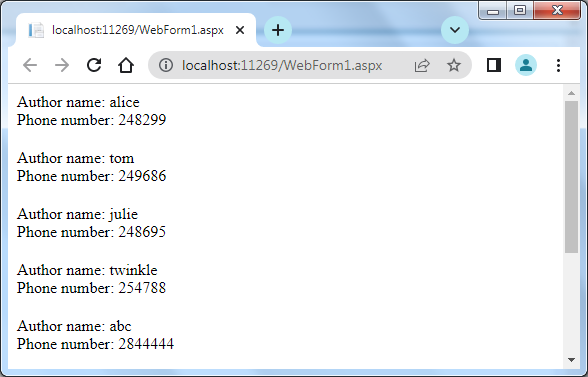
8. Connect the table on the data list control too.

8. Run the program.

**Design:**



**Output:**



**Practical – 7**

**Working with Database**

**a. Create a Web Application to Display Data binding Using Dropdown list Control.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace practical\_7a

{

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

{

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

{

}

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

{

Label1.Text = DropDownList1.SelectedItem.Text;

}

protected void Button2\_Click1(object sender, EventArgs e)

{

string constr = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string a = "select \* from [dbo].[city\_name]";

SqlConnection con = new SqlConnection(constr);

con.Open();

SqlCommand cmd = new SqlCommand(a, con);

DataTable dt = new DataTable();

SqlDataAdapter da = new SqlDataAdapter(cmd);

da.Fill(dt);

DropDownList1.DataSource = dt;

DropDownList1.DataTextField = "cities";

DropDownList1.DataBind();

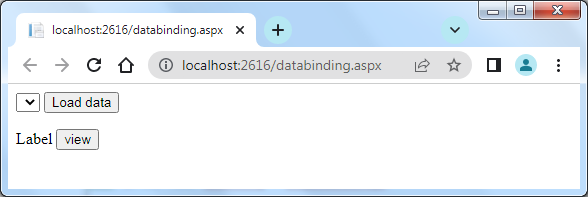
}

}

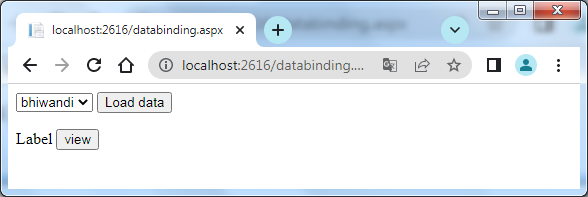
}

**Output:**

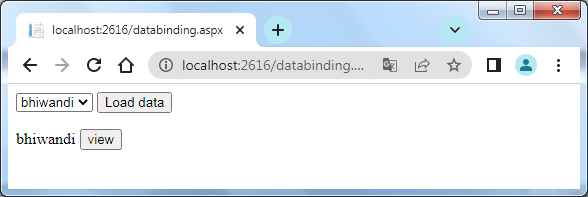
Before clicking anywhere:



After clicking on” load data” button:



After clicking on “view” button:



**b. Create a web application to display the phone number of an author using database.**

**Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace WebApplication13

{

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

{

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

{

}

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

{

String Aut\_name = DropDownList1.SelectedItem.Text;

string constr = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string selectquery = "select [Phone number] from [dbo].[author] where [Author name]='" + Aut\_name.Trim() + "'";

SqlConnection con = new SqlConnection(constr);

con.Open();

SqlCommand cmd = new SqlCommand(selectquery, con);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

Label1.Text = reader["Phone number"].ToString();

}

}

protected void DropDownList1\_SelectedIndexChanged(object sender, EventArgs e)

{

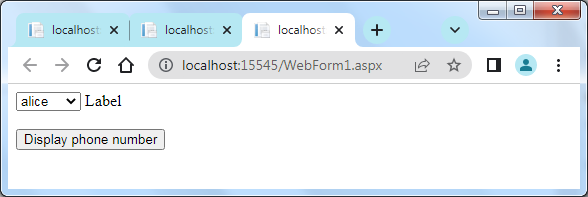
}

}

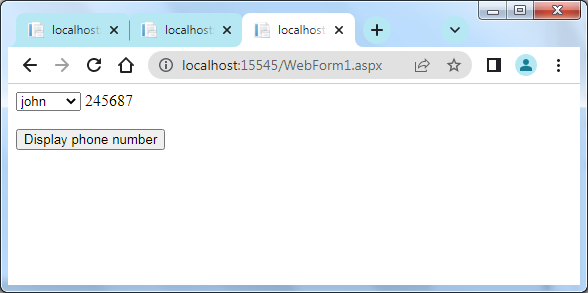
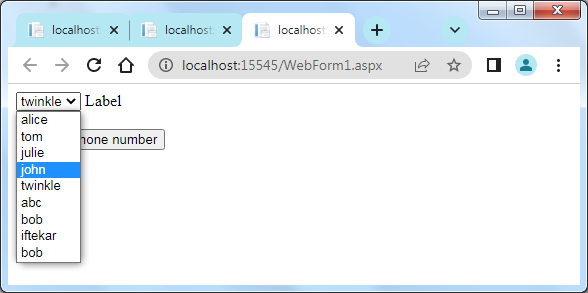
}

**Output:**

Before clicking anywhere, this is initially displayed



Select the author from the dropdownlist, whose number is required to be displayed.



After clicking on the “display phone number” button, the number is displayed on the label.

**c. Create a web application for inserting and deleting record from a database.**

**Code**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace practical7c

{

public partial class insert\_and\_delete : System.Web.UI.Page

{

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

{

}

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

{

string constr = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string insert\_query = "INSERT INTO [dbo].[author]([Author name],[Phone number])VALUES('" + TextBox1.Text + "','" + TextBox2.Text + "')";

SqlConnection con = new SqlConnection(constr);

con.Open();

SqlCommand cmd = new SqlCommand(insert\_query, con);

cmd.ExecuteNonQuery();

con.Close();

ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", "alert('Record Inserted Successfully')", true);

TextBox1.Text = "";

TextBox2.Text = "";

}

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

{

string constr2 = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string delete\_query = "DELETE FROM [dbo].[author] WHERE [Author name]='" + TextBox1.Text + "'";

SqlConnection con2 = new SqlConnection(constr2);

con2.Open();

SqlCommand cmd2 = new SqlCommand(delete\_query, con2);

cmd2.ExecuteNonQuery();

con2.Close();

}

protected void Button5\_Click(object sender, EventArgs e)

{

string constr4 = "Integrated Security=SSPI;Initial Catalog=company;Data Source=ITLAB-09";

string selectquery = "select \* from [dbo].[author]";

SqlConnection con4 = new SqlConnection(constr4);

con4.Open();

SqlCommand cmd4 = new SqlCommand(selectquery, con4);

SqlDataAdapter da = new SqlDataAdapter(cmd4);

DataSet ds = new DataSet();

da.Fill(ds);

GridView1.DataSource = ds;

GridView1.DataBind();

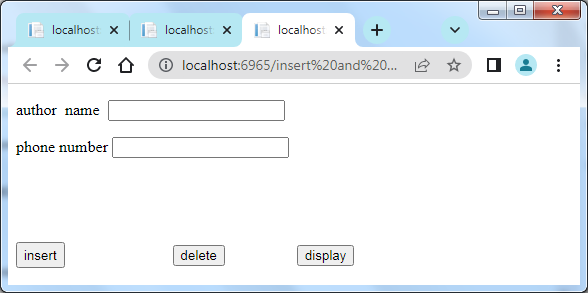
}

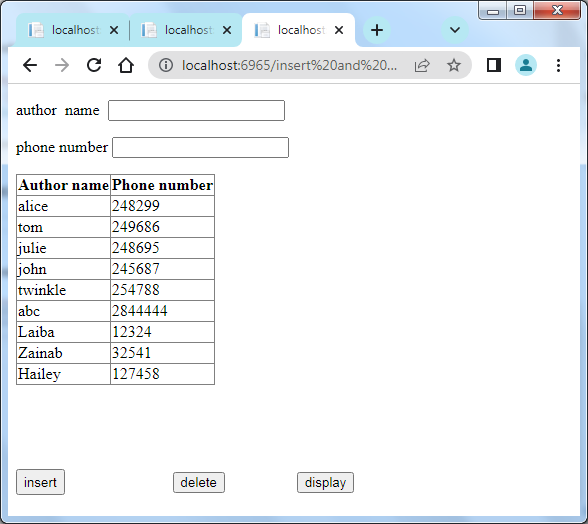
}

}

**Output:**

Initial screen,





Enter the Author name to be deleted, after clicking on the delete button and again displaying the table the entered author name information will be deleted.

