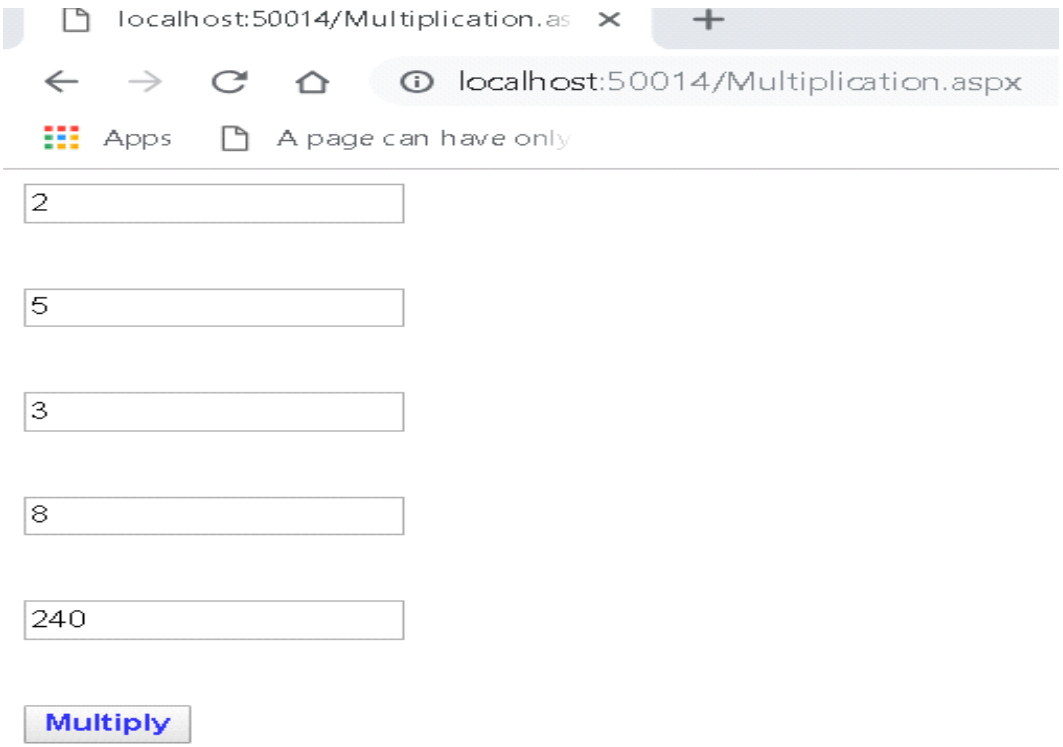
1. Working with Basic C# and ASP.NET

1(a). create an application that obtain four int values from the users and displays the product.

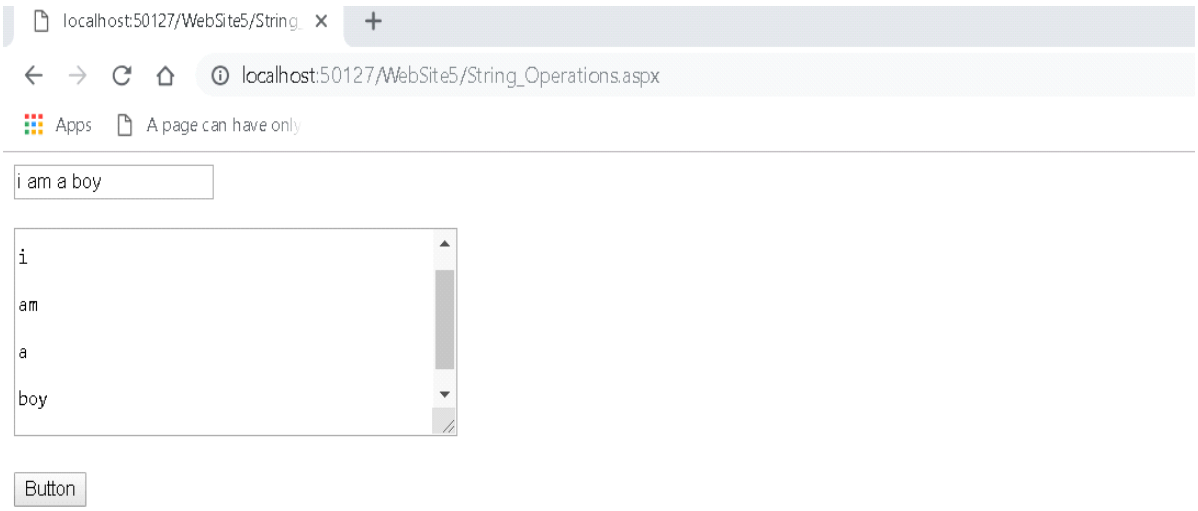
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Practical_1a
{
    public partial class Multiplication : System.Web.UI.Page
    {
        protected void Button1_Click(object sender, EventArgs e)
        {
            int num1, num2, num3, num4, prod;
            num1 = Convert.ToInt32(TextBox1.Text);
            num2 = Convert.ToInt32(TextBox2.Text);
            num3 = Convert.ToInt32(TextBox3.Text);
            num4 = Convert.ToInt32(TextBox4.Text);
            prod = num1 * num2 * num3 * num4;
            TextBox5.Text = Convert.ToString(prod);
        }
    }
}
```


localhost:50014/Multiplication.as
localhost:50014/Multiplication.aspx
Apps
A page can have only
2
5
3
8
240
Multiply

1(b). Create an application to demonstrate string operations.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class String_Operations : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
      string str1 = TextBox1.Text;
       string[] words = str1.Split();
       for(int i=0;i<words.Length;i++)
       {
           TextBox2.Text =TextBox2.Text+words[i]+"\n\r";
       }
    }
}
```

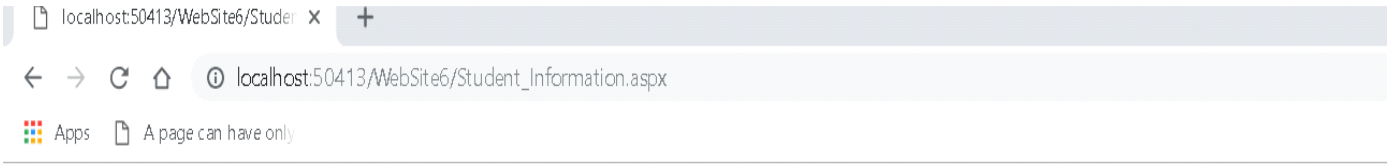
Practical No: Date:



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

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
struct Student
{
    public string studid, name, cname;
    public string dob;
}
public partial class Student_Information : System.Web.UI.Page
{
    Student[] s = new Student[1];
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
       int i;
```

```
        for ( i = 0; i < 1; i++ )
        {
            s[i].studid = TextBox2.Text;
            s[i].name = TextBox3.Text;
            s[i].cname = TextBox4.Text;
            s[i].dob = TextBox5.Text;


        }
        for (i = 0; i < 1; i++)
        {
            Response.Write("\nStudent ID:" + s[i].studid+"<br>");
            Response.Write("\nStudent name:" + s[i].name + "<br>");
            Response.Write("\nCourse name:" + s[i].cname + "<br>");
            Response.Write("\nDate of Birth(dd-mm-yy):" + s[i].dob + "<br>");
            Response.Write("<br>");
            Response.Write("<br>");
        }
    }
}
```



Student ID:1
Student name:abc
Course name:c#
Date of Birth(dd-mm-yy):1998-07-01

Student Id 1

Student Name abc

Course Name c#

Date Of Birth 01-07-1998

Button

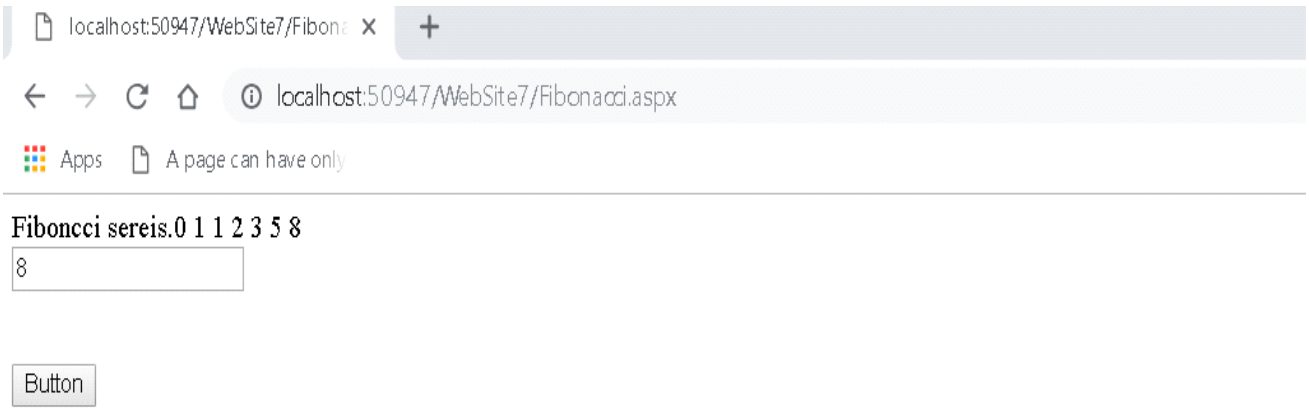
1.(d) Create an application to demonstrate following operations.

(i) Generate Fibonacci series.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Fibonacci : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        int num1 = 0, num2 = 1, num3, num4, num;
        num4= int.Parse(TextBox1.Text);
        num = 3;
        Response.Write("Fiboncci sereis.");
        Response.Write(num1 + "\t" + num2);
        while (num<= num4)
        {
            num3 = num1 + num2;
            if (num>= num4)
                break;
            Response.Write("\t" + num3);
            num1 = num2;
```

```
            num2 = num3;
            num++;
        }
}
```

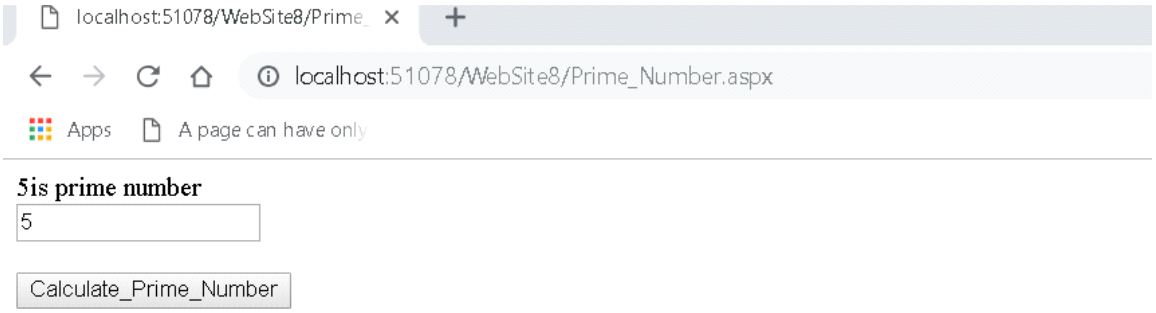
Practical No: Date:

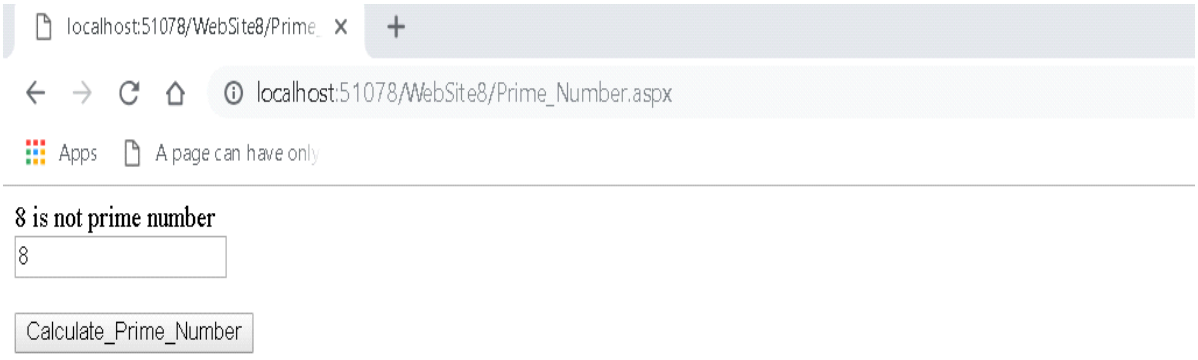


(ii) Test for prime numbers.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Prime_Number : System.Web.UI.Page
{
  protected void Button1_Click(object sender, EventArgs e)
    {
        int n, i, c;
        n = int.Parse(TextBox1.Text);
        for (c = 2; c <= n; c++)
        {
            if ((n % c) == 0)
                break;
        }
        if (n == 1)
            Response.Write(n + " is neither prime nor composite");
        else if (c < n - 1)
            Response.Write(n + " is not prime number");
        else
            Response.Write(n + "is prime number");
    } }
```

Practical No: Date:

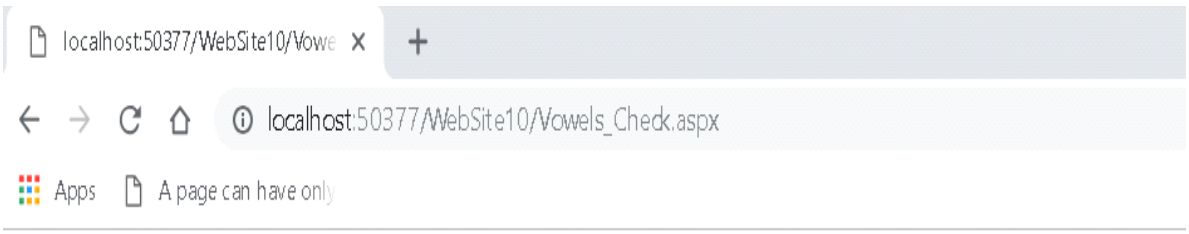




(iii) Test for vowels.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

    protected void Button1_Click(object sender, EventArgs e)

    {

        String ch;

        int count = 0;

        ch = TextBox1.Text;

        for (int i = 0; i < ch.Length; i++)

        {

       if ((ch.Substring(i, 1) == "a") || (ch.Substring(i, 1) == "e") ||
(ch.Substring(i, 1) == "i") || (ch.Substring(i, 1) == "o") || (ch.Substring(i, 1) ==
"u"))

            {

                count++;

            }

    }

        Response.Write("Given String:\n" + ch+"<br>");

        Response.Write("Total Number of vowels:\n" + count+"<br>");

        Response.Write("<br>");
```

```
        Response.Write("<br>");
    }
}
```



Given String: This is a vowel example
Total Number of vowels: 8

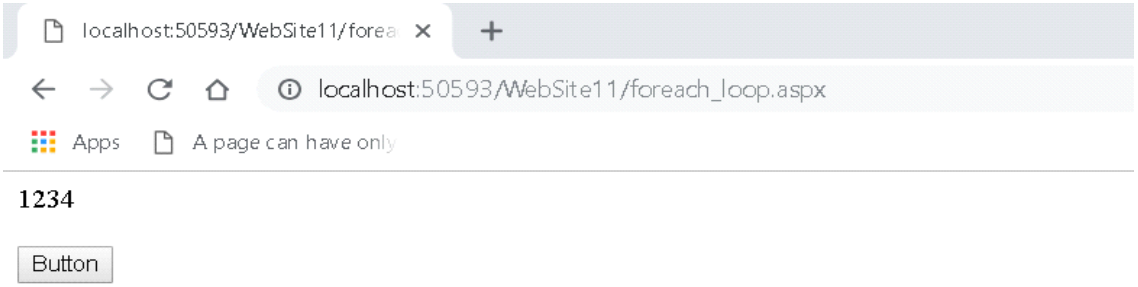
Enter a String: This is a vowel example

Button

(iv) Use foreach loop with arrays.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class foreach_loop : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        int[] a = { 1, 2, 3, 4 };
        foreach (int x in a)
            Response.Write(x);
    }
}
```

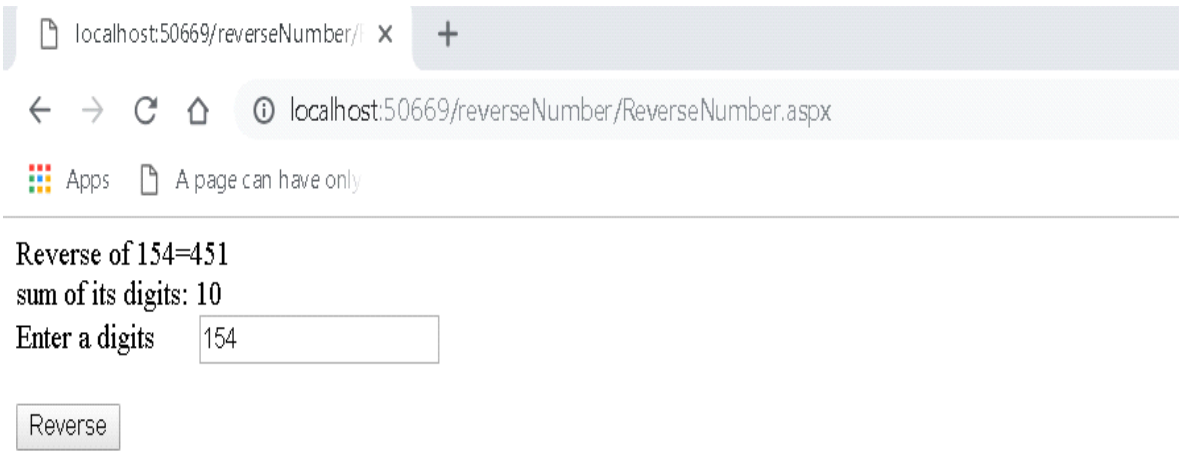
Practical No: Date:



(v). Reverse a number and find sum of digits of a number.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class ReverseNumber : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        int num, actualnumber, revnum = 0, digit, sumDigits = 0;
        num = int.Parse(TextBox1.Text);
        actualnumber = num;
        while (num > 0)
        {
            digit = num % 10;
            revnum = revnum * 10 + digit;
            sumDigits = sumDigits + digit;
            num = num / 10;
        }
        Response.Write("Reverse of\n" + actualnumber + "=" + revnum+"<br>");
```

```
        Response.Write("sum of its digits:\n" + sumDigits + "<br>");
    }
}
```

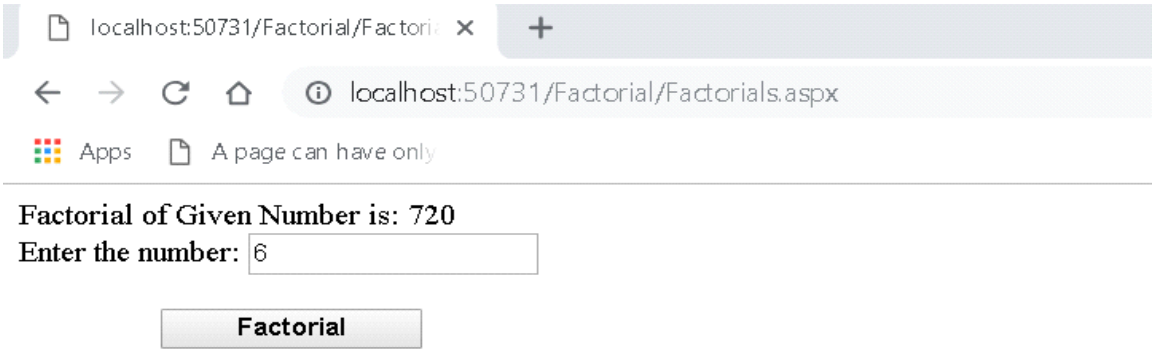

localhost:50669/reverseNumber/
localhost:50669/reverseNumber/ReverseNumber.aspx
Apps
A page can have only
Reverse of 154=451
sum of its digits: 10
Enter a digits
154
Reverse

2. Working With Object Oriented C# And ASP.NET.

2(a) Create simple application to perform follwoing operations.

(i) Finding factorial Value.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;


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

{

    protected void Button1_Click(object sender, EventArgs e)

    {

        int i, f=1, num;

        num =Convert.ToInt32(TextBox1.Text);

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


            f= f * i;

        Response.Write("\nFactorial of  Given Number is:\n"+ f);

        Response.Write("<br>");

    }}
```


localhost:50731/Factorial/Factori
localhost:50731/Factorial/Factorials.aspx
Apps
A page can have only
Factorial of Given Number is: 720
Enter the number: 6
Factorial

(ii) Money Conversion.

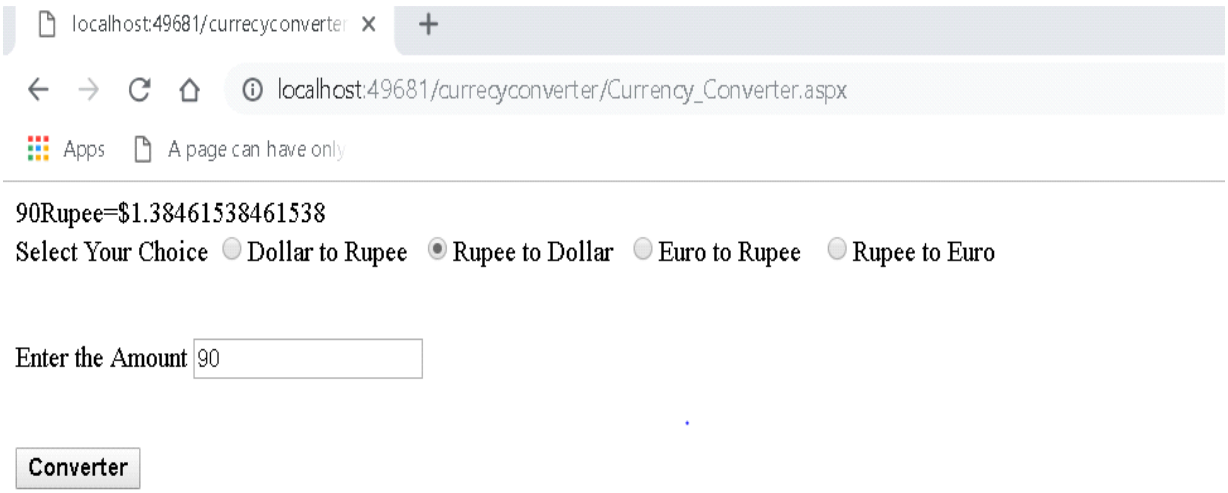
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public class Class1
{
    public double r, e, d;
    public Class1()
    {
        r = 0; e = 0; d = 0;
    }
    public void convertdtor()
    {
        double ev = 60;
        r = d * ev;
    }
    public void convertetor()
    {
        double ev = 80;
        r = e * ev;
    }
    public void convertrtod()
```

```
    {
        double ev = 65;
        d = r / ev;
    }
    public void convertrtoe()
    {
        double ev = 80;
        e = r / ev;
    }
}
public partial class Currency_Converter : System.Web.UI.Page
{
    Class1 f1;
    protected void Page_Load(object sender, EventArgs e)
    {
        f1 = new Class1();
    }

    protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton2.Checked == true)
        {
            f1.r = Convert.ToInt16(TextBox1.Text);
            f1.convertrtod();
            Response.Write(f1.r + "Rupee" + "=$" + f1.d);
```

```
        }
    }
 protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton1.Checked == true)
        {
            f1.d = Convert.ToInt16(TextBox1.Text);
            f1.convertdtor();
            Response.Write(f1.d + "Dollar" + "=Rs." + f1.r);
        }
    }
    protected void RadioButton3_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton3.Checked == true)
        {
            f1.e = Convert.ToInt16(TextBox1.Text);
            f1.convertetor();
            Response.Write(f1.e + "Euro" + "=Rs." + f1.r);
        }
    }
    protected void RadioButton4_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton4.Checked == true)
        {
            f1.r = Convert.ToInt16(TextBox1.Text);
```

```
            f1.convertrtoe();
            Response.Write(f1.r + "=Rs. to Euro" + f1.e);
        }
    }
}
```

Practical No: Date:

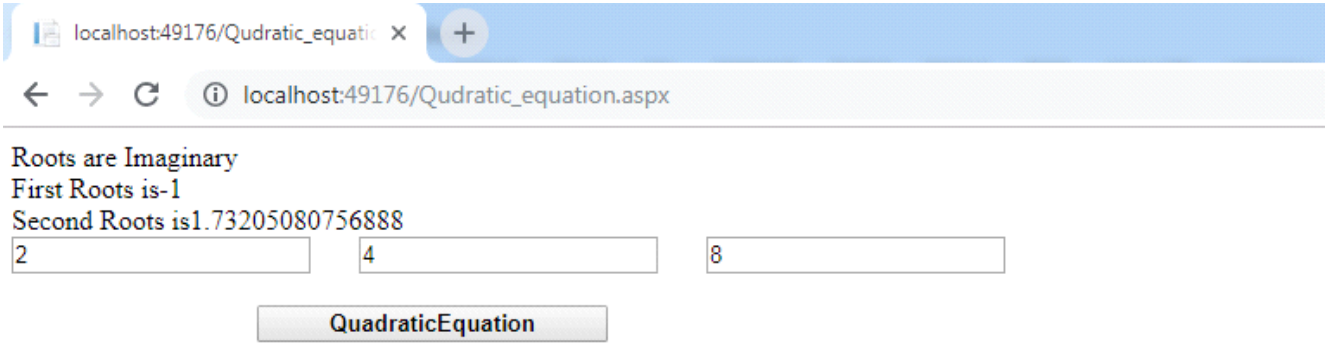


(iii) Quadratic Equation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class Quadraticroots
{
    public double a, b, c, r1, r2;
    public double compute()
    {
        int m;
        double d1;
        d1 = b * b - 4 * a * c;
        if (d1 == 0)
        {
            r1 = r2 = (-b) / (2 * a);
            return d1;
        }
        else if (d1 > 0)
        {
            r1 = (-b + Math.Sqrt(d1)) / (2 * a);
            r2 = (-b - Math.Sqrt(d1)) / (2 * a);
            return d1;

        }
        else
        {
            r1 = (-b) / (2 * a);
            r2 = Math.Sqrt(-d1) / (2 * a);
            return d1;

        }
    }
}

namespace practical
{
    public partial class Qudratic_equation : System.Web.UI.Page
    {
        Quadraticroots q;


        protected void Page_Load(object sender, EventArgs e)
        {
            q = new Quadraticroots();
```

```
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            q.a = Convert.ToInt16(TextBox1.Text);
            q.b = Convert.ToInt16(TextBox2.Text);
            q.c = Convert.ToInt16(TextBox3.Text);
            double d = q.compute();
            if (d == 0)
            {
                Response.Write("\n Roots are  Real and Equal<br>");
                Response.Write("First root and second root is" + q.r1);

            }
            else if (d > 0)
            {
                Response.Write("\nRoots are Real and Distinct<br>");
                Response.Write("\nFirst Root is" + q.r1 + "<br>");
                Response.Write("\nSecond Root is" + q.r2 + "<br>");
            }
            else
            {
                Response.Write("\nRoots are Imaginary <br>");
                Response.Write("\nFirst Roots is"+q.r1+"<br>");
                 Response.Write("\nSecond Roots is"+q.r2+"<br>");

            }
        }
    }
}
```

Practical No: Date:

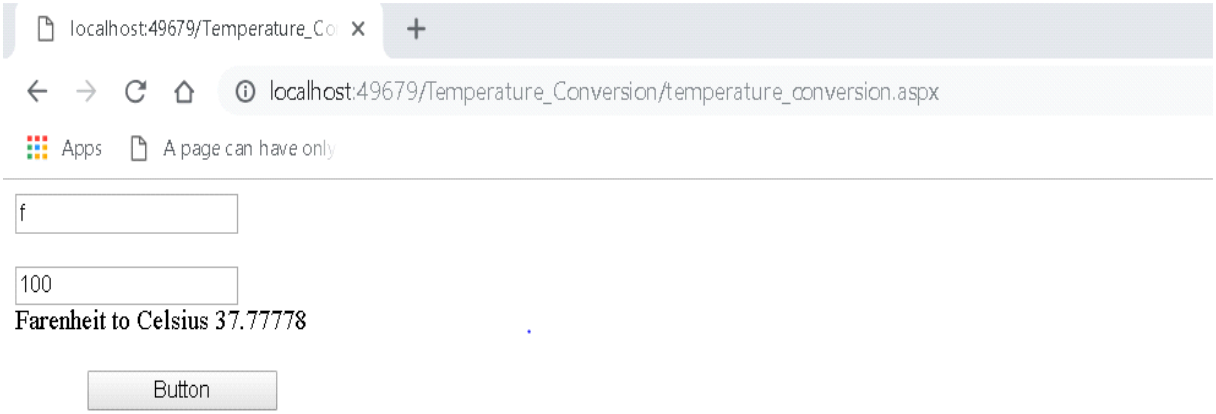


(iv) Temperature Conversion.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class converttemp
{
    public float celsius, faren;
    public converttemp()
    {
        celsius = 0;
        faren = 0;
    }
    public void converttofaren()
    {
        faren = ((celsius * 9.0f / 5.0f) + 32.0f);
    }
    public void converttocel()
    {
        celsius = (faren - 32) * (5.0f / 9.0f);
    }
}
public partial class temperature_conversion : System.Web.UI.Page
```

```
{
    converttemp c;
    protected void Page_Load(object sender, EventArgs e)
    {
        c = new converttemp();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        char ch;
        ch = Convert.ToChar(TextBox1.Text);
        if (ch == 'c')
        {
            c.celsius = float.Parse(TextBox2.Text);
            c.converttofaren();
            Label1.Text = "Celsius to Farenheit\n" + c.faren;
        }
        else
        {
            c.faren = float.Parse(TextBox2.Text);
            c.converttocel();
            Label1.Text="Farenheit to Celsius\n"+c.celsius;
        }
    }
}
```

Practical No: Date:



2 Woriking With Object Oriented C# and ASP.NET.

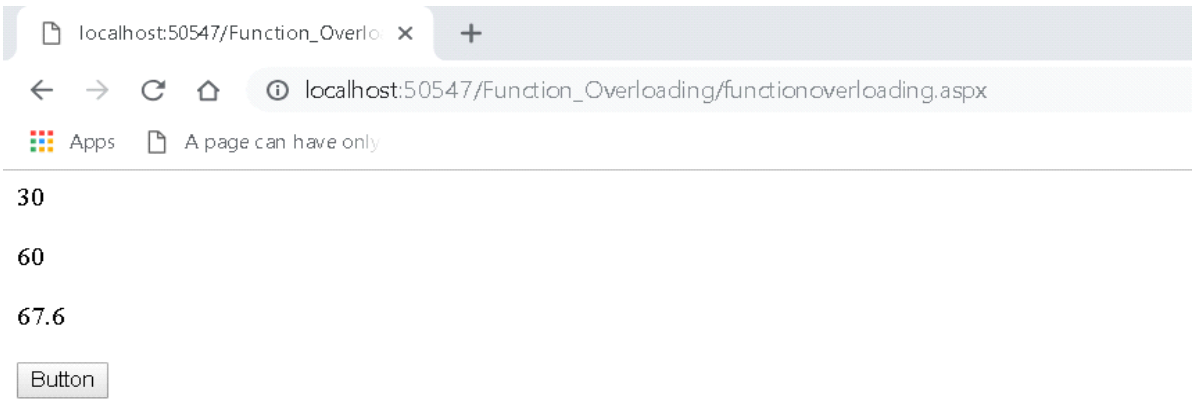
(b) Create simple application to demonstrate use of following concepts.

(i) Function Overloading.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class overloading
{
    public int sum(int a, int b)
    {
        int x;
        return x = a + b;
    }
    public int sum(int a, int b, int c)
    {
        int y;
        return y = a + b + c;
    }
    public float sum(float a, float b)
    {
        float u;
```

```
        return u = a + b;
    }
    public float sum(float a, float b, float c)
    {
        float v;
        return v = a + b + c;
    }
}
public partial class functionoverloading : System.Web.UI.Page
{
    overloading o;

    protected void Page_Load(object sender, EventArgs e)
    {
        o = new overloading();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = Convert.ToString(o.sum(10, 20));
        Label2.Text = Convert.ToString(o.sum(10, 20, 30));
        Label3.Text = Convert.ToString(o.sum(12.0f, 23.1f, 32.5f));
    }
}
```

Practical No: Date:

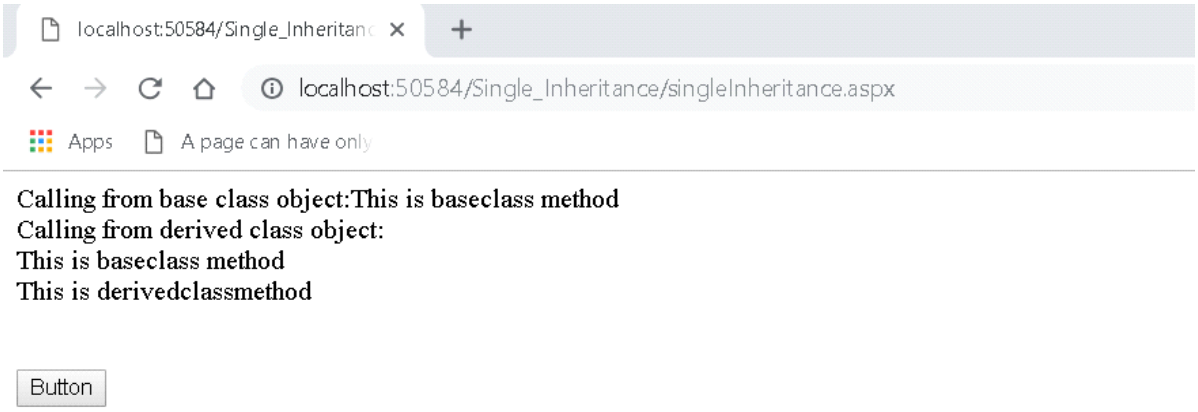


(ii) Inheritance (All Types)

Single Inheritance

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public class basec
{
    public int d;
    public string basemethod()
    {
        string p = "This is baseclass method";
        return p;
    }
}
public class Derived : basec
{
    public string derivedmethod()
    {
        string s = "This is derivedclassmethod";
        return s;
    }
```

```
}
public partial class singleInheritance : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        basec b=new basec();
        Response.Write("Calling from base class object:" +b.basemethod());
        Derived d = new Derived();
        Response.Write("<br>Calling from derived class object:<br>" +
d.basemethod());
        Response.Write("<br>" + d.derivedmethod());
    }
}
```

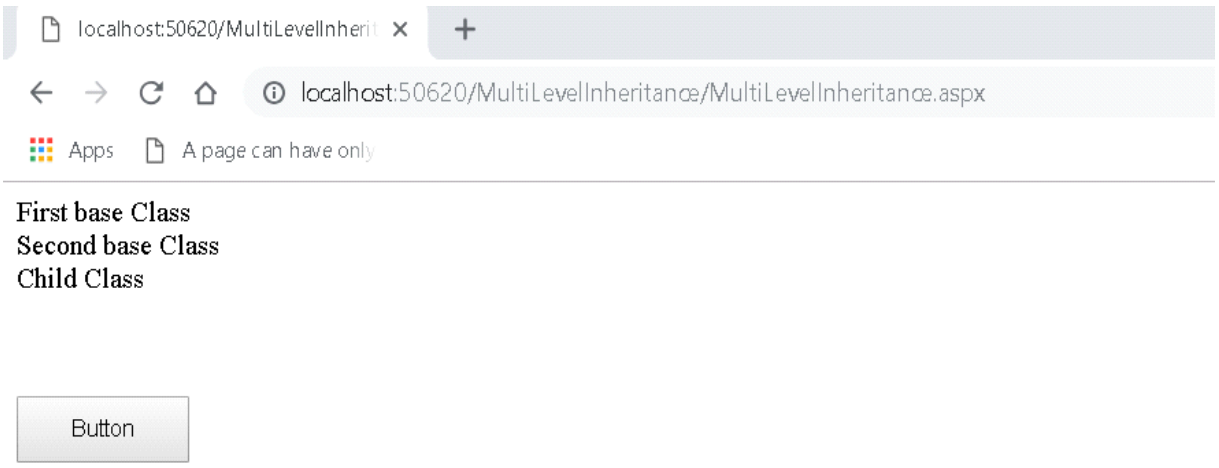

localhost:50584/Single_Inheritanc
localhost:50584/Single_Inheritance/singleInheritance.aspx
Apps
A page can have only
Calling from base class object:This is baseclass method
Calling from derived class object:
This is baseclass method
This is derivedclassmethod
Button

Multi Level Inheritance

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class A
{
    public string show()
    {
        return ("First base Class");
    }
}
class B : A
{
    public string display()
    {
        return ("Second base Class");
    }
}
class C : B
{
    public string show1()
    {
```

```
        return "Child Class";
    }
}
public partial class MultiLevelInheritance : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        C obj = new C();
        Response.Write(obj.show() + "<br>");
        Response.Write(obj.display() + "<br>");
        Response.Write(obj.show1() + "<br>");
    }
}
```

Practical No: Date:



Multiple Inheritance

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class Shape
{
    public int side;
    public void setSide(int s)
    {
        side = s;
    }
}
public interface Cost
{
    int getCost(int area);
}
class square : Shape, Cost
{
    public int getArea()
    {
        return (side * side);
    }
```

```
    public int getCost(int area)
    {
        return area * 10;
    }
}
public partial class MULTIPLE_INHERITANCE : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        square sq = new square();
        int area;
        sq.setSide(15);
        area = sq.getArea();
        Label1.Text = "Area:" + area;
        int c = sq.getCost(area);
        Label2.Text = "Cost is Rs:" + c;
    }
}
```

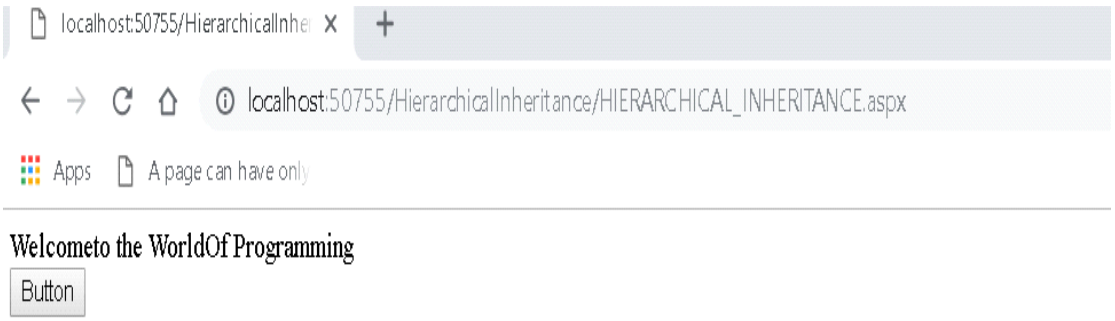
Practical No: Date:

Hierarchical Inheritance

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class A
{
    public string show()
    {
        return "Welcome";
    }
}
class B : A
{
    public string display()
    {
        return "to the World";
    }
}
class C : A
{
    public string show1()
    {
```

```
        return "Of Programming";
    }
}
public partial class HIERARCHICAL_INHERITANCE : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        C c1 = new C();
        B b1 = new B();
        string s = "";
        s += c1.show();
        s += b1.display();
        s += c1.show1();
        Label1.Text=s;
    }
}
```

Practical No: Date:



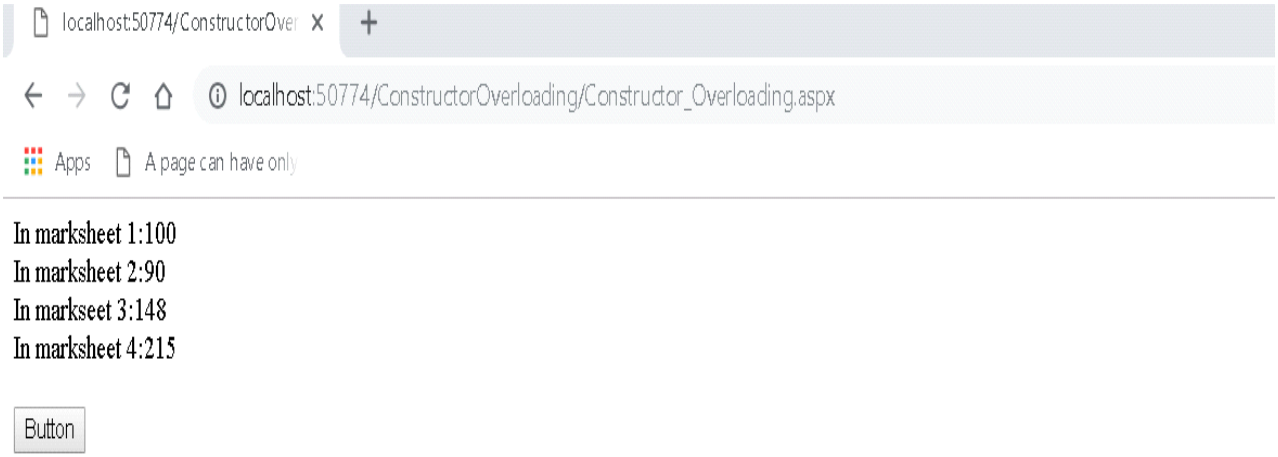
(iii) Constructor overloading

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class MarkSheet
{
    private float m1, m2, m3;
    string name;
    public MarkSheet()
    {
        m1 = 20;
        m2 = 40;
        m3 = 40;
    }
    public MarkSheet(float ms)
    {
        m1 = ms;
    }
    public MarkSheet(float ms1, float ms2)
    {
        m1 = ms1;
        m2 = ms2;
```

```
    }
    public MarkSheet(float ms1, float ms2, float ms3)
    {
        m1 = ms1;
        m2 = ms2;
        m3 = ms3;
    }
    public float tot()
    {
        float t = m1 + m2 + m3;
        return t;
    }
}
public partial class Constructor_Overloading : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        MarkSheet a = new MarkSheet();
        MarkSheet b = new MarkSheet(90);
        MarkSheet c = new MarkSheet(88, 60);
        MarkSheet d = new MarkSheet(70, 90, 55);
        Response.Write("In marksheet 1:");
```

```
        Response.Write(a.tot() + "<br>");
        Response.Write("In marksheet 2:");
        Response.Write(b.tot() + "<br>");
        Response.Write("In markseet 3:");
        Response.Write(c.tot() + "<br>");
        Response.Write("In marksheet 4:");
        Response.Write(d.tot() + "<br>");
    }
}
```

Practical No: Date:

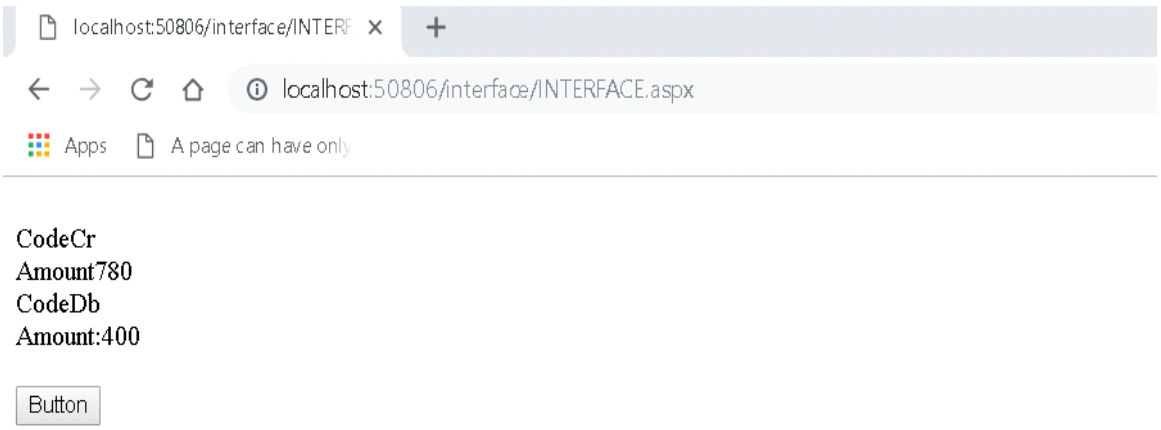


(iv) Interface

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public interface ITransactions
{
    string retcode();
    double amtfunc();
}
public class Transaction : ITransactions
{
    private string tCode;
    private double amount;
    public Transaction()
    {
        tCode = "";
        amount = 0.0;
    }
    public Transaction(string c, double a)
    {
        tCode = c;
        amount = a;
```

```
    }
    public double amtfunc()
    {
        return amount;
    }
    public string retcode()
    {
        return tCode;
    }
}
public partial class INTERFACE : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Transaction t1 = new Transaction("Cr", 780.00);
        Transaction t2 = new Transaction("Db", 400.00);
        Response.Write("<br>Code" + t1.retcode());
        Response.Write("<br>Amount" + t1.amtfunc());
        Response.Write("<br>Code" + t2.retcode());
        Response.Write("<br>Amount:" + t2.amtfunc());
    }
}
```

Practical No: Date:



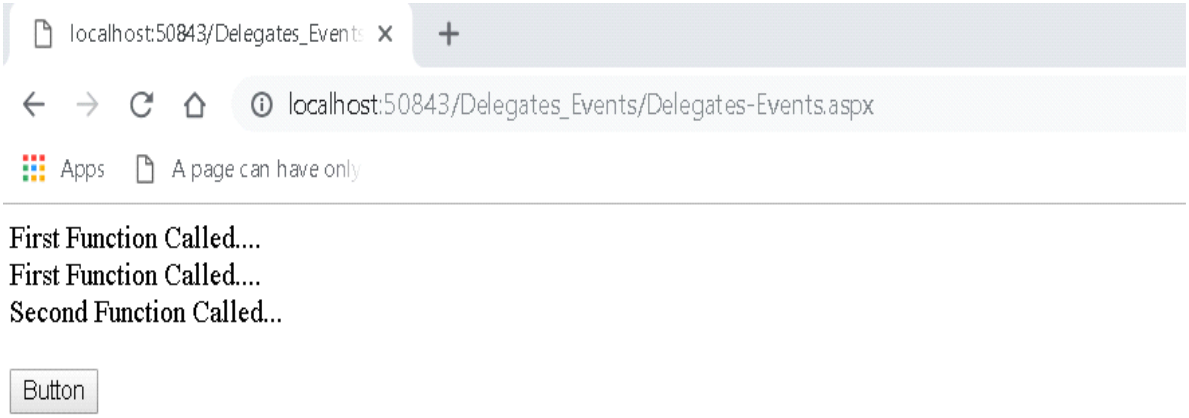
2(c). Create simple applicaion to demonstrate use of following concepts

(i) Using Delegates and events

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Delegates_Events : System.Web.UI.Page
{
    public delegate void SimpleDelegate();
    public void callingFunction()
    {
        Response.Write("First Function Called....<br>");
    }
    public void secfunction()
{
    Response.Write("Second Function Called...<br>");
}
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
```

```
    {
        SimpleDelegate sd = new SimpleDelegate(callingFunction);
        sd();
        sd += new SimpleDelegate(secfunction);
        sd();
    }
}
```

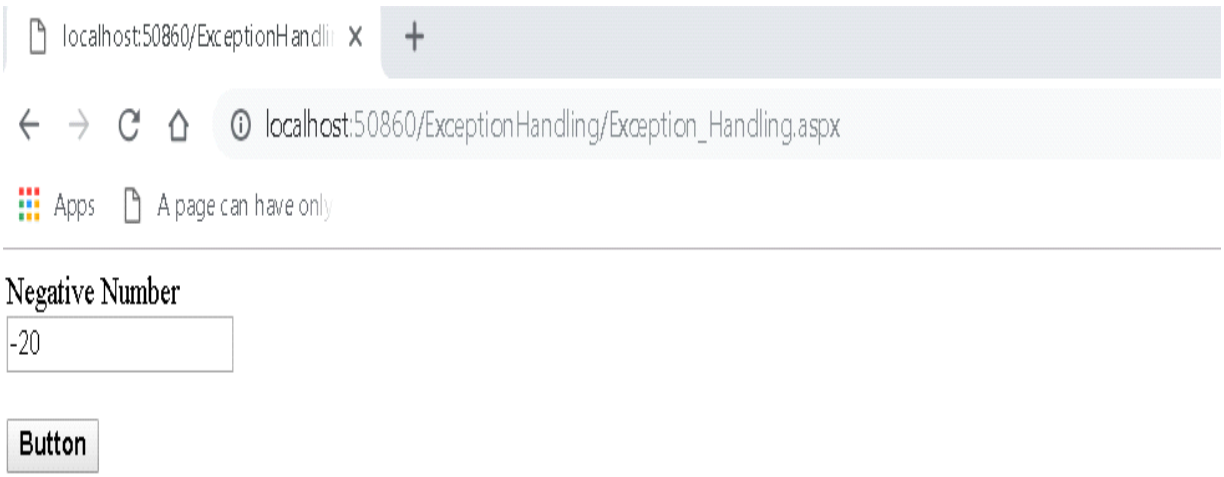
Practical No: Date:



(ii) Exception Handling

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class NegativeException : Exception
{
    public NegativeException(string msg)
        : base(msg)
    {
    }
}
public partial class Exception_Handling : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        int num;
        try
        {
            num = int.Parse(TextBox1.Text);
```

```
            if (num < 0)
                throw new NegativeException("Negative Number");
            else
                Console.WriteLine("Positive Number");
        }
        catch(NegativeException en)
        {
            Response.Write(en.Message);
        }
    }
    }
```


localhost:50860/ExceptionHandli
localhost:50860/ExceptionHandling/Exception_Handling.aspx
Apps
A page can have only
Negative Number
-20
Button

Practical No: Date:

Practical 3(b).Demonstrate the use of Calendar control to perform following operations.

a) Display messages in a calendar control.
b) Display vacation in a calendar control
c) Select Day in calendar control using style.
d) Difference between two calendar dates.

Calndrctrl.aspx

Calender properties set for this example:

<asp:Calendar ID="Calendar1" runat="server"
BackColor="#FFFFCC"

BorderColor="#FFCC66" BorderWidth="1px"
DayNameFormat="Shortest"

Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399"
Height="200px"

NextPrevFormat="ShortMonth"
OnDayRender="Calendar1_DayRender"

ShowGridLines="True" Width="300px"

OnSelectionChanged="Calendar1_SelectionChanged" >

<DayHeaderStyle BackColor="#FFCC66" Font-Bold="True"
Height="1px" /> <NextPrevStyle BorderStyle="Solid"
BorderWidth="2px" Font-Size="9pt"
ForeColor="#FFFFCC"/>

<OtherMonthDayStyle BackColor="#FFCC99" BorderStyle="Solid"

ForeColor="#CC9966" />

<SelectedDayStyle BackColor="Red" Font-
Bold="True" /> <SelectorStyle
BackColor="#FFCC66" />

```
<TitleStyle BackColor="#990000" Font-Bold="True" Font
Size="9pt"
ForeColor="#FFFFCC" />
<TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
<WeekendDayStyle Height="50px" />
</asp:Calendar>
```

calndrCtrl.aspx.cs

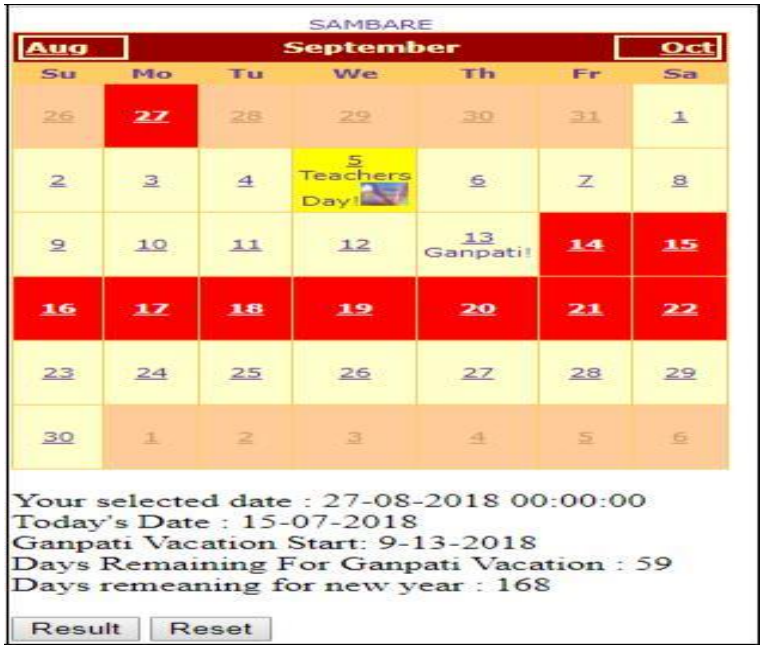
```
protected void btnResult_Click(object sender, EventArgs e)
  {
    Calendar1.Caption = "SAMBARE";
    Calendar1.FirstDayOfWeek =
    FirstDayOfWeek.Sunday;
    Calendar1.NextPrevFormat =
    NextPrevFormat.ShortMonth;
    Calendar1.TitleFormat = TitleFormat.Month;



    Label2.Text = "Todays
    Date"+Calendar1.TodaysDate.ToShortDateString();
    Label3.Text = "Ganpati Vacation Start: 9-13-2018";
    TimeSpan d = new DateTime(2018, 9, 13) -
    DateTime.Now;

    Label4.Text = "Days Remaining For Ganpati
    Vacation:"+d.Days.ToString(); TimeSpan d1 = new
    DateTime(2018, 12, 31) - DateTime.Now; Label5.Text
    = "Days Remaining for New Year:"+d1.Days.ToString();

     if (Calendar1.SelectedDate.ToShortDateString() == "9-13-
     2018")
      Label3.Text = "<b>Ganpati Festival Start</b>";
     if (Calendar1.SelectedDate.ToShortDateString() == "9-23-
     2018")
```

```
    Label3.Text = "<b>Ganpati Festival End</b>";
}

protected void Calendar1_DayRender(object sender,
System.Web.UI.WebControls.DayRenderEventArgs e)

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

  {
    e.Cell.BackColor = System.Drawing.Color.Yellow;
    Label lbl = new Label();
    lbl.Text = "<br>Teachers Day!";
    e.Cell.Controls.Add(lbl);

    Image g1 = new Image();
    g1.ImageUrl = "td.jpg";
    g1.Height = 20;
    g1.Width = 20;

    e.Cell.Controls.Add(g1);
```

```
    }
    if (e.Day.Date.Day == 13 && e.Day.Date.Month == 9)
    {
      Calendar1.SelectedDate = new
      DateTime(2018, 9, 12);
      Calendar1.SelectedDates.SelectRange(Calenda
      r1.SelectedDate,
Calendar1.SelectedDate.AddDays(10));
      Label lbl1 = new Label();
      lbl1.Text = "<br>Ganpati!";
      e.Cell.Controls.Add(lbl1);
    }
  }
  protected void btnReset_Click(object sender, EventArgs e)
  {
    Label1.Text = "";
    Label2.Text = "";
    Label3.Text = "";
    Label4.Text = "";
    Label5.Text = "";
    Calendar1.SelectedDates.Clear();
  }
```

```
protected void Calendar1_SelectionChanged(object sender,
EventArgs e)
{
  Label1.Text = "Your Selected Date:" +
  Calendar1.SelectedDate.Date.ToString();
}
```

OUTPUT


SAMBARE
Aug
September
Oct
Su Mo Tu We Th Fr Sa
26 27 28 29 30 31 1
2 3 4 5 Teachers Day! 6 7 8
9 10 11 12 13 Ganpati! 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 1 2 3 4 5 6
Your selected date : 27-08-2018 00:00:00
Today's Date : 15-07-2018
Ganpati Vacation Start: 9-13-2018
Days Remaining For Ganpati Vacation : 59
Days remeaning for new year : 168
Result
Reset

Practical 3(c).Demonstrate the use of Treeview control perform following operations.

a) Treeview control and datalist b) Treeview operations

Add XML File

Website -> Add -> XML File and Name it ‘stdetail’.

stdetail.xml

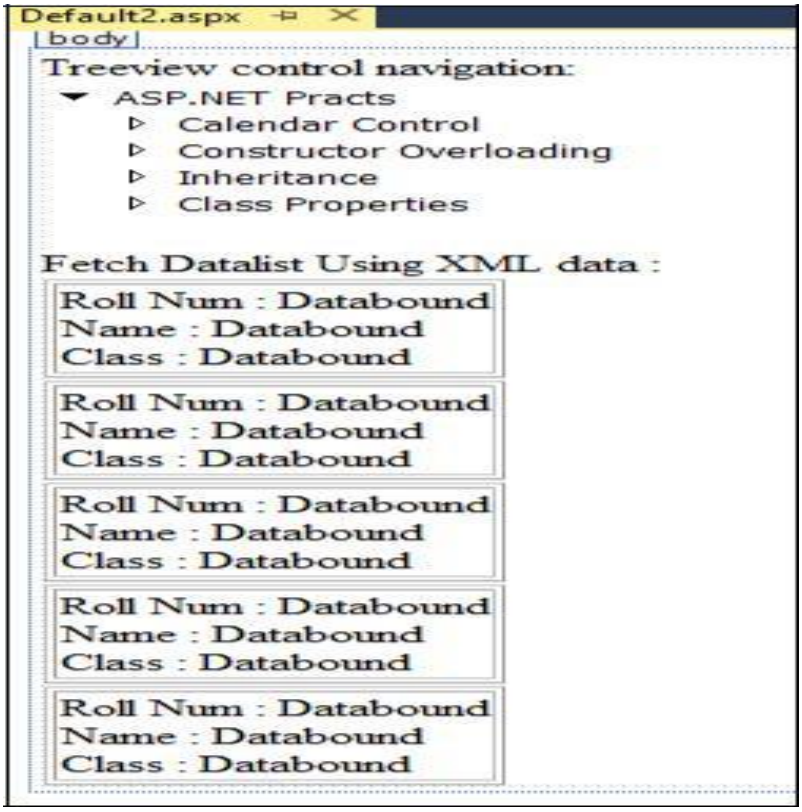
```
<?xml version="1.0" encoding="utf-8" ?>
<studentdetail>
 <student>
  <sid>1</sid>
  <sname>Tushar</sname>
  <sclass>TYIT</sclass>
 </student>
 <student>
  <sid>2</sid>
  <sname>Sonali</sname>
  <sclass>TYCS</sclass>
 </student>
<student>
  <sid>3</sid>
  <sname>Yashashree</sname>
  <sclass>TYIT</sclass>
 </student>
<student>
```

```
  <sid>4</sid>
  <sname>Vedshree</sname>
  <sclass>TYCS</sclass>
 </student>
</studentdetail>
```

Default2.aspx

```
<form id="form1" runat="server">
<div>
Treeview control navigation:<asp:TreeView ID = "TreeView1" runat
= "server" Width = "150px" ImageSet="Arrows">
<HoverNodeStyle Font-Underline="True"
ForeColor="#5555DD" /> <Nodes>
<asp:TreeNode Text = "ASP.NET Practs" Value = "New Node">
<asp:TreeNode Text = "Calendar Control" Value = "RED"
NavigateUrl="~/calndrCtrl.aspx"> </asp:TreeNode>
<asp:TreeNode Text = "Constructor Overloading" Value = "GREEN"
NavigateUrl="~/clsconstrc.aspx"> </asp:TreeNode>
<asp:TreeNode NavigateUrl="~/singleInh.aspx" Text="Inheritance"
Value="BLUE"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/clsProp.aspx" Text="Class
Properties" Value="Class Properties"></asp:TreeNode>
</asp:TreeNode>
</Nodes>
```

```
    <NodeStyle Font-Names="Tahoma" Font-Size="10pt"
ForeColor="Black" HorizontalPadding="5px"
NodeSpacing="0px" VerticalPadding="0px" />
    <ParentNodeStyle Font-Bold="False" />
    <SelectedNodeStyle Font-Underline="True"
ForeColor="#5555DD" HorizontalPadding="0px"
VerticalPadding="0px" /> </asp:TreeView>
<br />
Fetch Datalist Using XML data :
</div> <asp:DataList
ID="DataList1" runat="server">
      <ItemTemplate>
<table class = "table" border="1">
        <tr>
        <td>Roll Num : <%# Eval("sid") %><br />
          Name : <%# Eval("sname") %><br />
          Class : <%# Eval("sclass")%>
        </td>
      </tr>
    </table>
</ItemTemplate>
</asp:DataList>
```

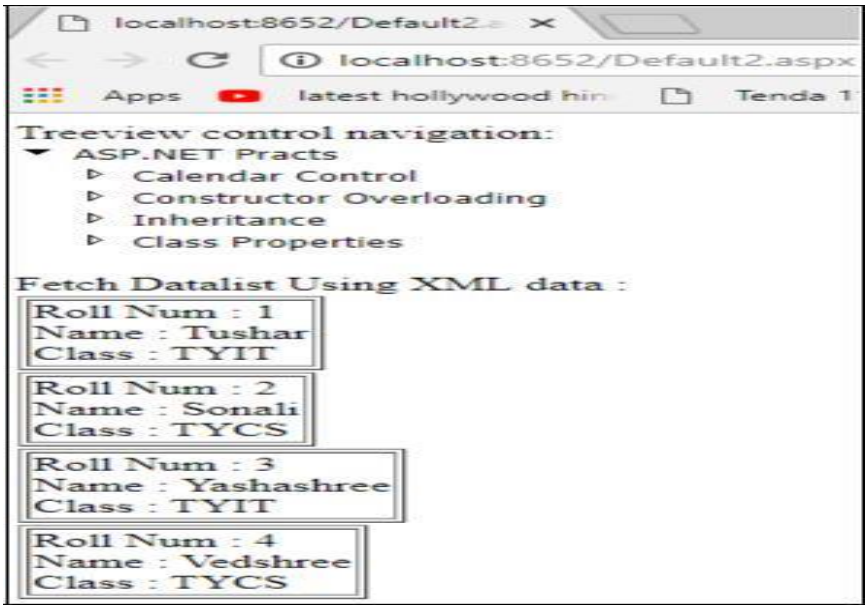

Default2.aspx
body
Treeview control navigation:
ASP.NET Practs
Calendar Control
Constructor Overloading
Inheritance
Class Properties
Fetch Datalist Using XML data :
Roll Num : Databound
Name : Databound
Class : Databound
Roll Num : Databound
Name : Databound
Class : Databound
Roll Num : Databound
Name : Databound
Class : Databound
Roll Num : Databound
Name : Databound
Class : Databound
Roll Num : Databound
Name : Databound
Class : Databound

Default2.aspx.cs

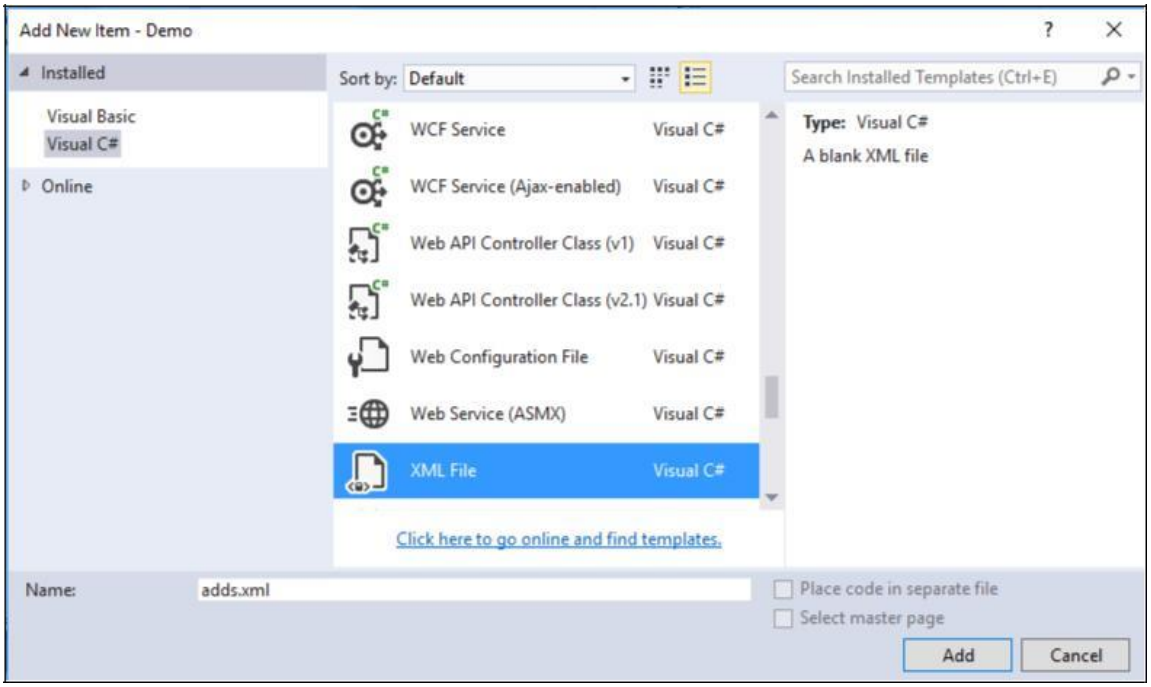
```
using System.Data;
public partial class _Default :
System.Web.UI.Page {
  protected void Page_Load(object sender, EventArgs e)
  {
    if (!IsPostBack)
    {
      BindData();
    }
  }
  protected void BindData()
  {
    DataSet ds = new DataSet();
    ds.ReadXml(Server.MapPath("stdetail.xml"));
    if (ds != null && ds.HasChanges())
    {
        DataList1.DataSource = ds;
        DataList1.DataBind();
    }
    else
    {
        DataList1.DataBind();
```

```
    }
  }
}
```

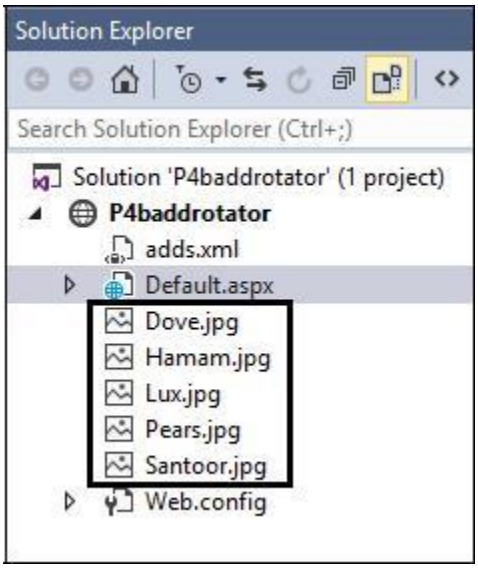
OUTPUT


localhost:8652/Default2
localhost:8652/Default2.aspx
Apps
latest hollywood hin
Tenda 1
Treeview control navigation:
ASP.NET Practs
Calendar Control
Constructor Overloading
Inheritance
Class Properties
Fetch Datalist Using XML data :
Roll Num : 1
Name : Tushar
Class : TYIT
Roll Num : 2
Name : Sonali
Class : TYCS
Roll Num : 3
Name : Yashashree
Class : TYIT
Roll Num : 4
Name : Vedshree
Class : TYCS

Practical 4(b).Create Web Form to demonstrate use of Adrotator Control. Add a XML file, name it "adds.xml"



Add images to test out the adrotator functionality.



XML File

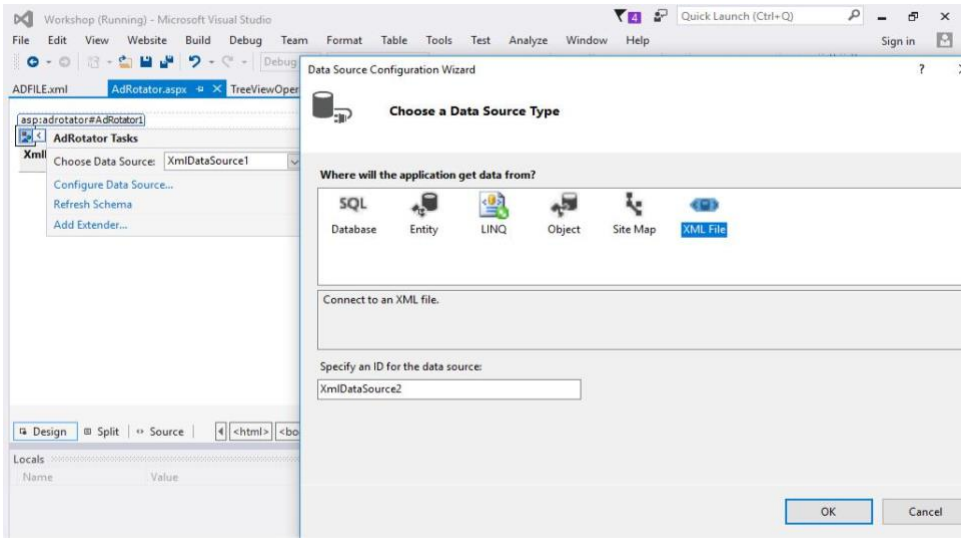
```
<Advertisements>
<Ad>
 <ImageUrl>rose1.jpg</ImageUrl>
 <NavigateUrl>http://www.1800flowers.com</NavigateUrl>
 <AlternateText>
 Order flowers, roses, gifts and more
 </AlternateText>
 <Impressions>20</Impressions>
 <Keyword>flowers</Keyword>
</Ad>

<Ad>
 <ImageUrl>rose2.jpg</ImageUrl>
 <NavigateUrl>http://www.babybouquets.com.a
 u</NavigateUrl> <AlternateText>Order roses
 and flowers</AlternateText>
 <Impressions>20</Impressions>
 <Keyword>gifts</Keyword>
</Ad>
```

```
<Ad>
 <ImageUrl>rose3.jpeg</ImageUrl>
 <NavigateUrl>http://www.flowers2moscow.co
 m</NavigateUrl> <AlternateText>Send flowers
 to Russia</AlternateText>
 <Impressions>20</Impressions>
 <Keyword>russia</Keyword>
</Ad>

</Advertisements>
```

Default.aspx

<asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1" />

<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile="~/ADFILE.xml"></asp:XmlDataSource>

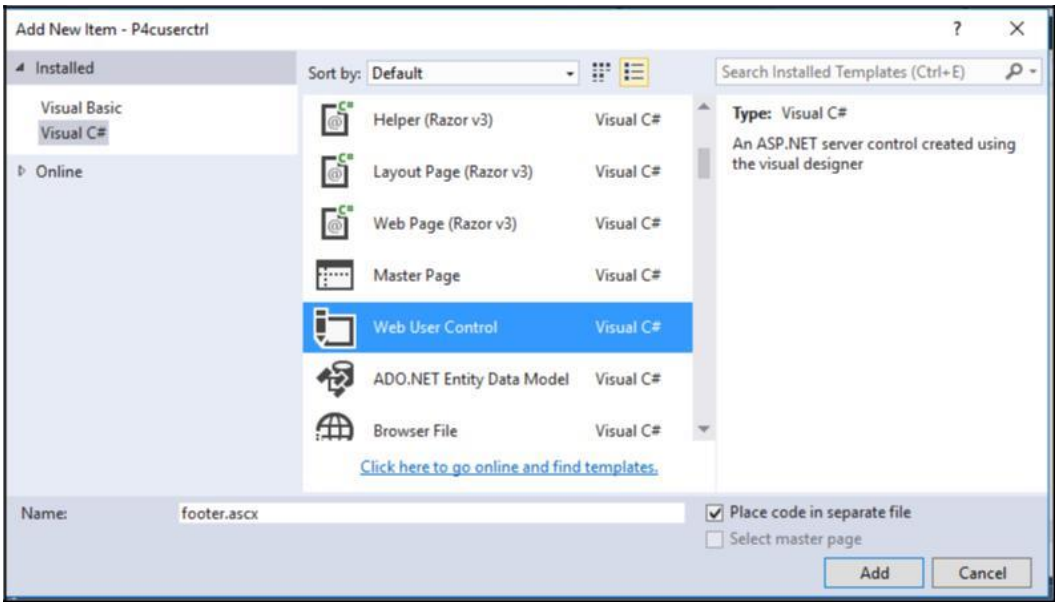


OUTPUT:

Practical 4(c).Create Web Form to demonstrate use User Controls.

Add Web User Control

Website -> Add -> Web User Control and Name it 'MyUserControl.



.

MyUserControl.ascx

```
<%@ Control Language="C#"
AutoEventWireup="true"
CodeFile="MyUserControl.ascx.cs"
Inherits="MyUserControl" %> <h3>This is User
Contro1 </h3> <table>
<tr>
<td>Name</td>
<td>

  <asp:TextBox ID="txtName"
runat="server"></asp:TextBox> </td>
</tr>
<tr>
<td>City</td>
<td><asp:TextBox ID="txtcity"
runat="server"></asp:TextBox></td>
</tr>

<tr>
<td></td>
<td>
  </td>
</tr>
<tr>
```

```
<td></td>

<td>

  <asp:Button ID="txtSave" runat="server" Text="Save"
onclick="txtSave_Click" /> </td>

</tr>

</table><br />

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

MyUserControl.ascx.cs

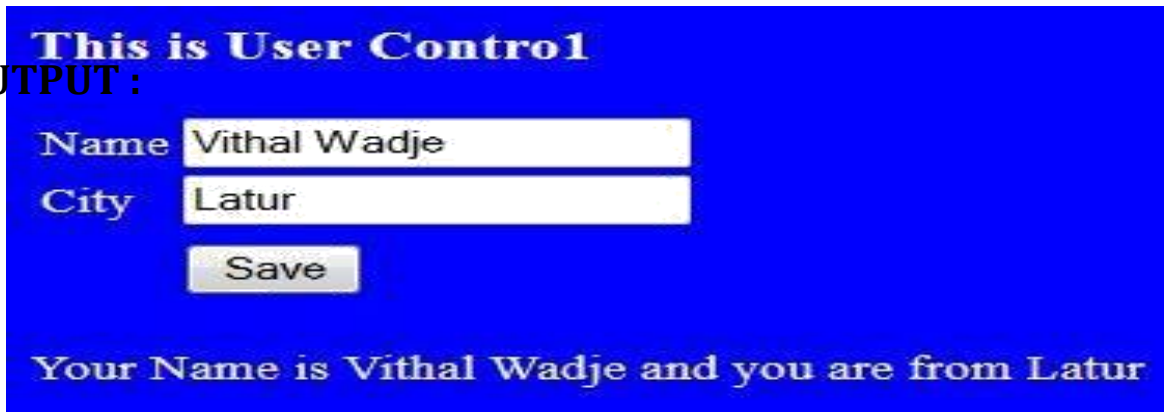
```
protected void txtSave_Click(object sender, EventArgs e)
  {
    Label1.Text = "Your Name is " + txtName.Text + " and
you are from " + txtcity.Text;
  }
```

UserControlDisplay**.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UserControlDisplay.aspx.cs"
Inherits="UserControlDisplay" %>
<%@ Register Src="~/MyUserControl.asc TagPrefix="uc"
TagName="Student"%>
<!DOCTYPE html>

<html
xmlns="http://www.w3.org/1999/x
html"> <head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  <uc:Student ID="studentcontrol" runat="server" />
```

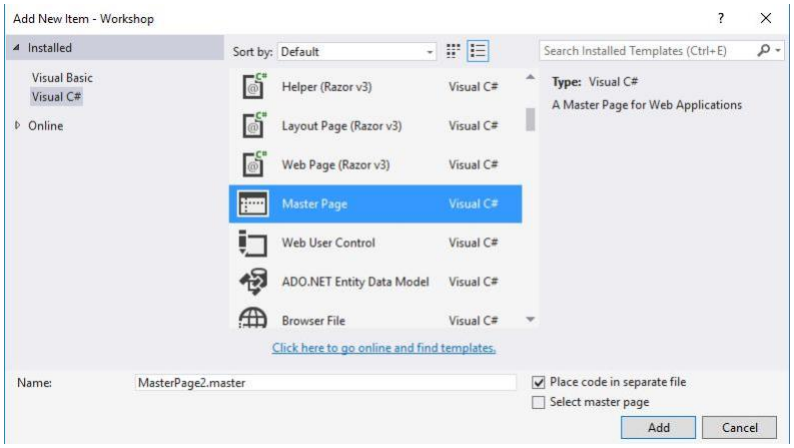
```
    </div>
    </form>
</body>
</html>
```

OUTPUT :

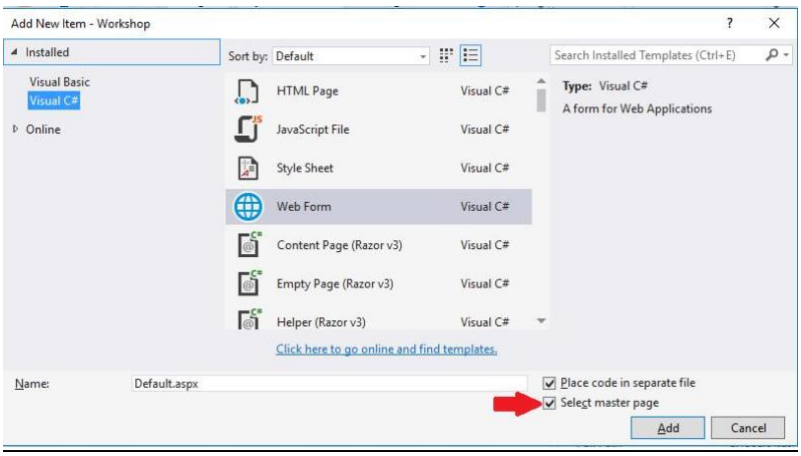


Practical 5(b).Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification.

Adding Master Page



Adding Web page For Master page



MasterPage.master

```
<%@ Master Language="C#"
AutoEventWireup="true"
CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Master Page Demo</title>
  <link href="css/my.css" rel="stylesheet" />
  <asp:ContentPlaceHolder ID="head" runat="server">
  </asp:ContentPlaceHolder>
  <style type="text/css">
    .auto-style1 {
      position: absolute;
      top: 373px;
      left: 1028px;
      bottom: 303px;
    }
    .auto-style2 {
      position: absolute;
```

```
        top: 537px;
        left: 1016px;
        z-index: 1;
    }
  </style>
</head>
```

```
<body>
   <!DOCTYPE html>
   <form id="form1" runat="server">

<html>
<head>
   <title>Master</title>
   <link rel="stylesheet" type="text/css"
   href="StyleSheet.css">
</head>
<body>
<header id="header">
<h1>Demo Of Master Page</h1>
</header>
<nav id="nav">
   <ul>
      <li><a href="home.aspx">Insight</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">Downloads</a></li>
      <li><a href="#">Contact Us</a></li>
   </ul>
</nav>
```

```
    <aside
id="side">
<h1>Info</
       h1>

   <a href="#"><p>Product Type 1</p></a>

   <a href="#"><p>Product Type 2</p></a>

   <a href="#"><p>Product Type 3<a
href="#"><asp:ScriptManager ID="ScriptManager1"
runat="server">

     </asp:ScriptManager>

     </a>

   </p>

   <asp:Button ID="Button2" runat="server"
CssClass="auto-style1" style="z-index: 1"
Text="Button" />
   <asp:Button ID="Button1" runat="server"
   CssClass="auto-style2" Text="Button" />


</aside>
   <div id="con">
```

```
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
        runat="server">

        </asp:ContentPlaceHolder>
    </div>

<footer id="footer">
    copyright @Sambare
</footer>
</body>
</html>

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

MasterDisplay.aspx

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

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

</asp:Content>

<asp:Content ID="Content2"
   ContentPlaceHolderID="ContentPlaceHolder1"
   runat="server"> <h1>Home page</h1>

</asp:Content>
```

StyleSheet.css

```
#header{

   color: blueviolet;

   text-align: center;
   font-size: 20px;

}

#nav{
   background-color:darkseagreen;
```

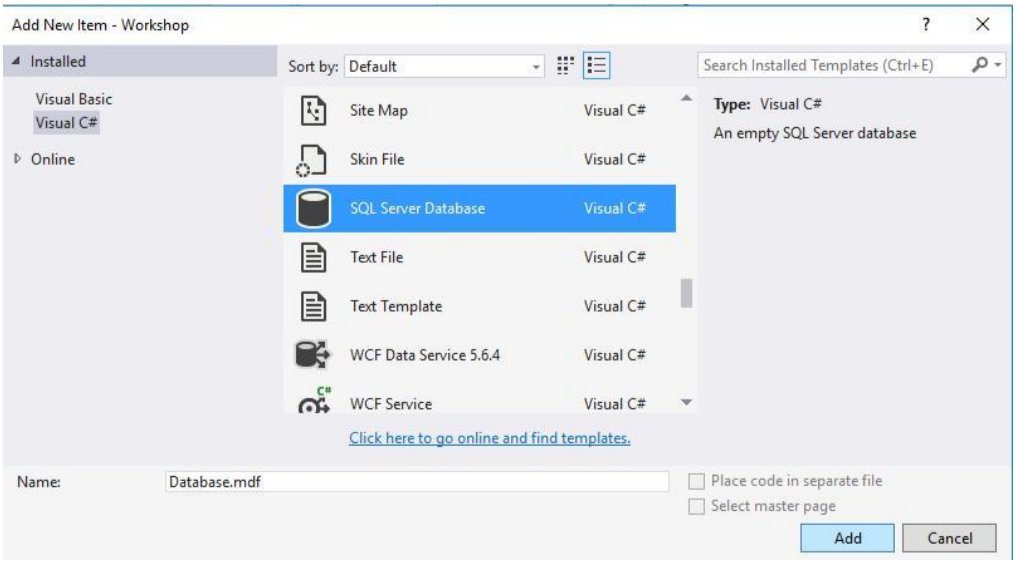
```
   padding: 5px;
}
ul{
   list-style-type: none;
}
li a {
   color:crimson ;
font-size: 30px;
column-width: 5%;
   }
   li
  {
   display: inline;
   padding-left: 2px;
   column-width: 20px;
  }
  a{
 text-decoration: none;
 margin-left:20px
   }
```

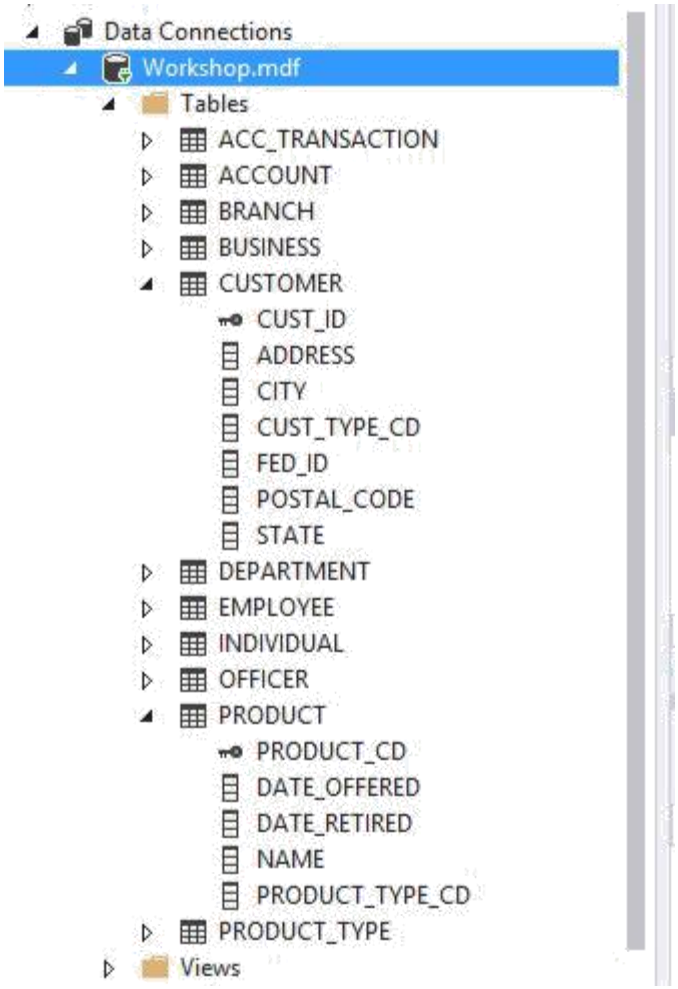
```
li a:hover{
 background-color: aqua;
 color:coral ;
 padding:1%;
}
#side{
 text-align: center;
 float: right;
 width: 15%;
 padding-bottom: 79%;
 background-color: #F1FAEE;
}
#article{
 background-color: burlywood;
 padding: 10px;
 padding-bottom: 75%;
}
#footer{
 background-color: #C7EFCF;
 text-align:center;
 padding-bottom: 5%;
```

```
   font-size: 20px;
}
  #con{
     border:double;
     border-color:burlywood;
  }
```

Database Practicals*******

Here we to add new database in our website, as shown below. Add this database inside App_Data folder.

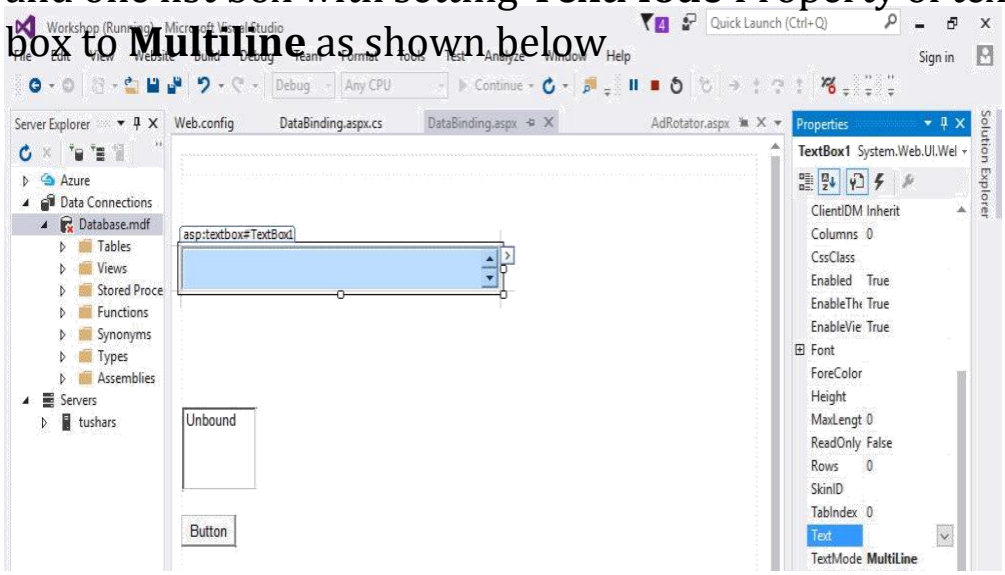



Data Connections
Workshop.mdf
Tables
ACC_TRANSACTION
ACCOUNT
BRANCH
BUSINESS
CUSTOMER
CUST_ID
ADDRESS
CITY
CUST_TYPE_CD
FED_ID
POSTAL_CODE
STATE
DEPARTMENT
EMPLOYEE
INDIVIDUAL
OFFICER
PRODUCT
PRODUCT_CD
DATE_OFFERED
DATE_RETIRED
NAME
PRODUCT_TYPE_CD
PRODUCT_TYPE
Views

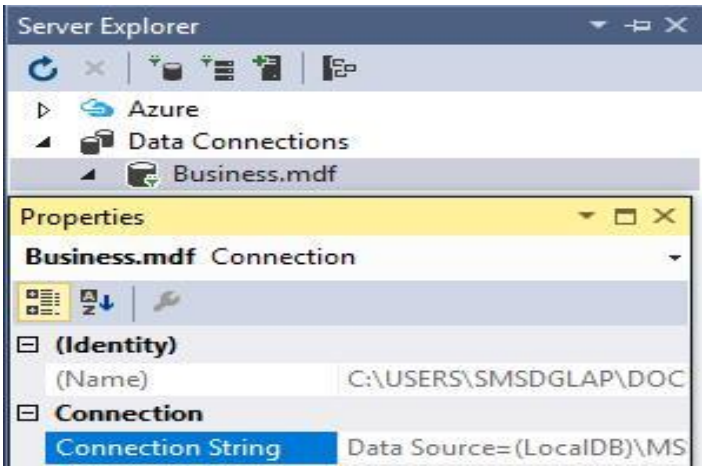
Practical No: Date:

Practical 6 (a): Create a web application to bind data in a multiline textbox by querying in another textbox.

1. Create a webpage with one **Button,** one Multiline TextBox and one list box with setting **TextMode** Property of text box to **Multiline** as shown below.



2. Write the Database related code in code behind C# file as given below.



3. Add this string to configuration file (web.config) as given below.

Web.confing

```
<configuration>

  <system.web>

   <compilation debug="true"
   targetFramework="4.5.2" />
   <httpRuntime
   targetFramework="4.5.2" />

  </system.web>

  <connectionStrings>

   <add name="connStr" connectionString="Data
```

```
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\tushars\Documents\Visual Studio

2015\WebSites\Workshop\App_Data\Database.mdf';Integrated Security=True" />


</connectionStrings>

</configuration>
```

4. Now use the following code C# in Default.aspx.cs (**Note** : First write following using statements at the top of file

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

public partial class DataBinding : System.Web.UI.Page
{
   protected void Button1_Click(object sender,
   EventArgs e)
   {
      string connStr =
ConfigurationManager.ConnectionStrings["connStr"].Con
nectionString;
```

```
        SqlConnection con = new
        SqlConnection(connStr);
        con.Open();

        SqlCommand cmd = new
        SqlCommand(TextBox1.Text,
        con); SqlDataReader reader =
        cmd.ExecuteReader();
        ListBox1.Items.Clear();
        while (reader.Read())
        {
           //To add new blank line in the text area

           for (int i = 0; i < reader.FieldCount - 1;
           i++)
           {
              ListBox1.Items.Add(reader[i].ToString());
           }
        }
        reader.Close();
        con.Close();

    }
}
```

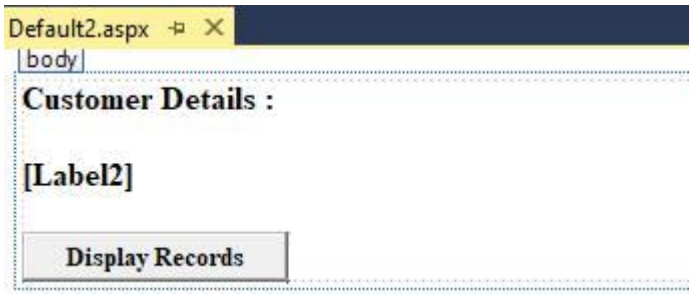

localhost:3772/DataBinding.aspx

Output:


select * from customer
1
47 Mockingbird Ln
Lynnfield
l
Button

Practical 6 (b): Create a web application to display records by using database.

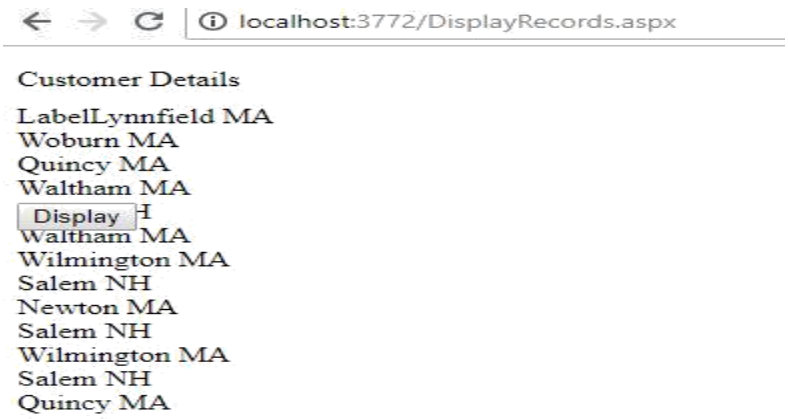
Create a web page with following design:




```
protected void Button1_Click(object sender, EventArgs
e)
   {
      string connStr =
ConfigurationManager.ConnectionStrings["connStr"].Con
nectionString;
      SqlConnection con = new SqlConnection(connStr);
      SqlCommand cmd = new SqlCommand("Select
      City, State from Customer", con);
      con.Open();
      SqlDataReader reader = cmd.ExecuteReader();
```

```
        while (reader.Read())
        {

        Label1.Text += reader["City"].ToString() + " " +
                                reader["State"].ToString() +
"<br>";

        }
        reader.Close();
        con.Close();

    }
```

Output:

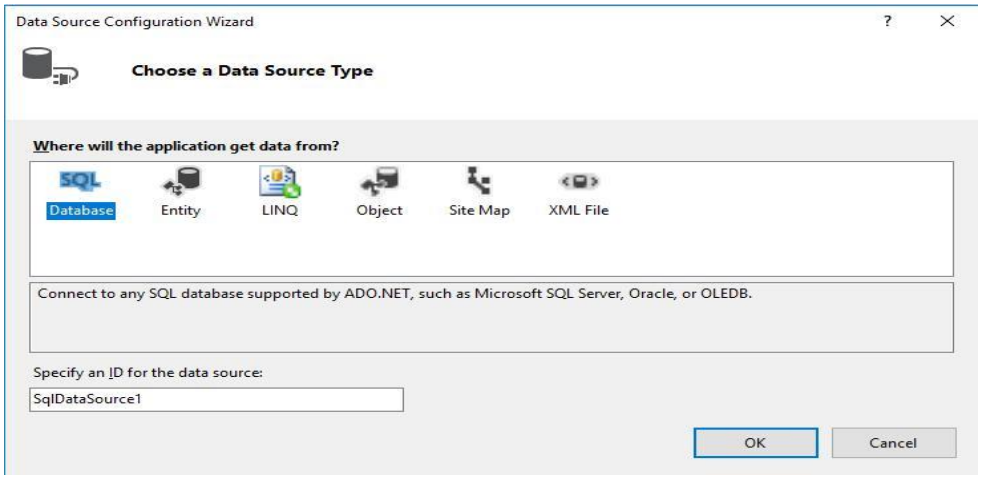


Practical 6 (c): Demonstrate the use of Datalist link control.

1. Drag the Datalist control to our web page form toolbox->Data-> Datalist.

2. Then select **Choose Data Source** Option and select **<New Data Source>.**

3. Now Select SQL Database from options and Click Ok button.



4. In next window click on New Connection button.

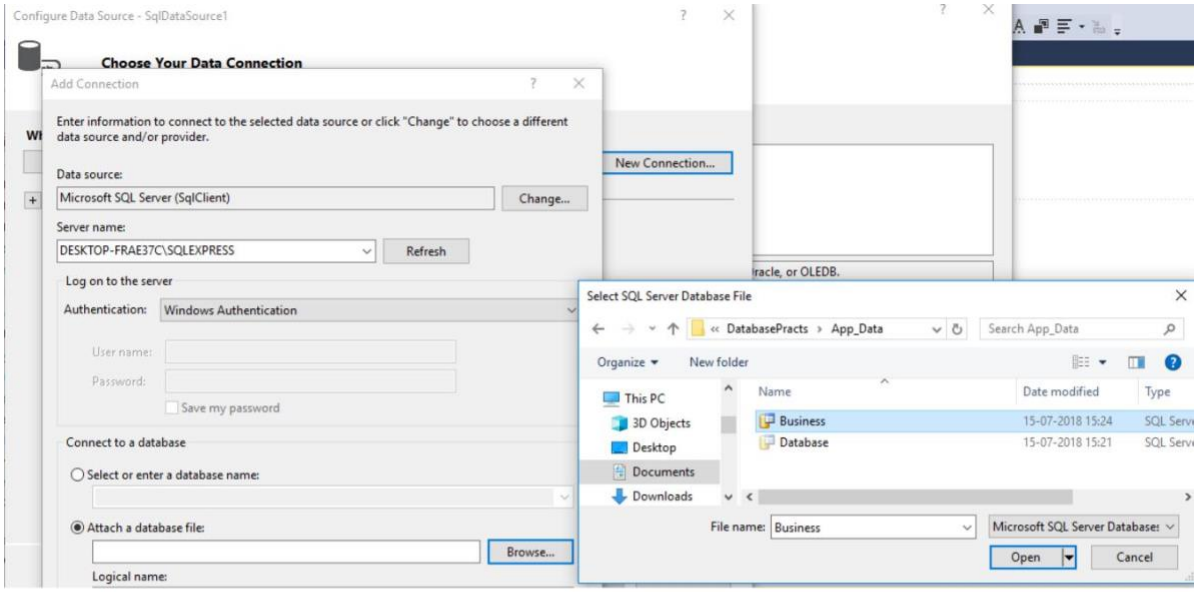
5. In add connection window Select the available SQL Server Name

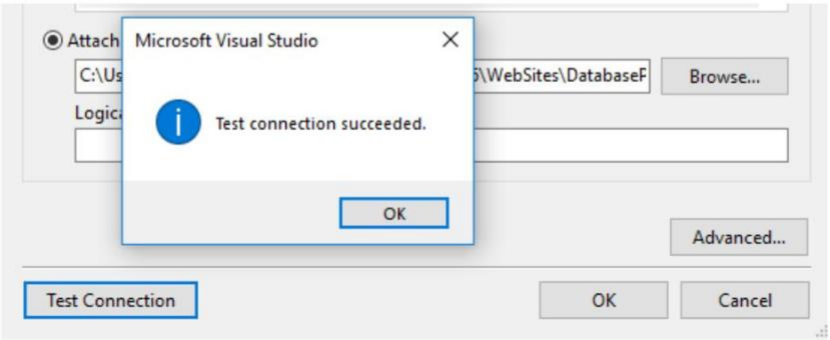
6. Keep the Authentication as Windows Authentication.

7. After that select Attach a Database file radio button. Here we have to select the database that we have created in our application. (Usually it will be in Documents folder under Visual Studio 2015/ Websites).

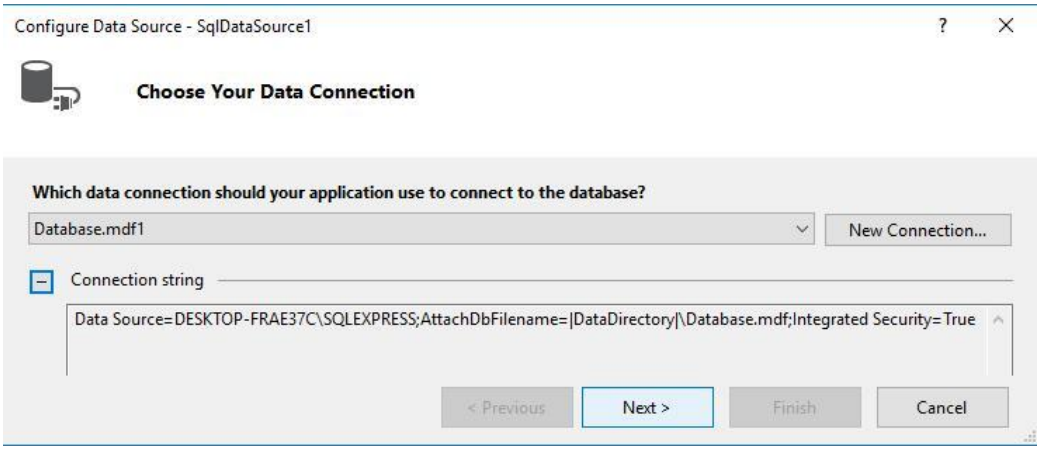
8. After selection of Database file. We can also Test the connection.

9. Then Click on OK button.



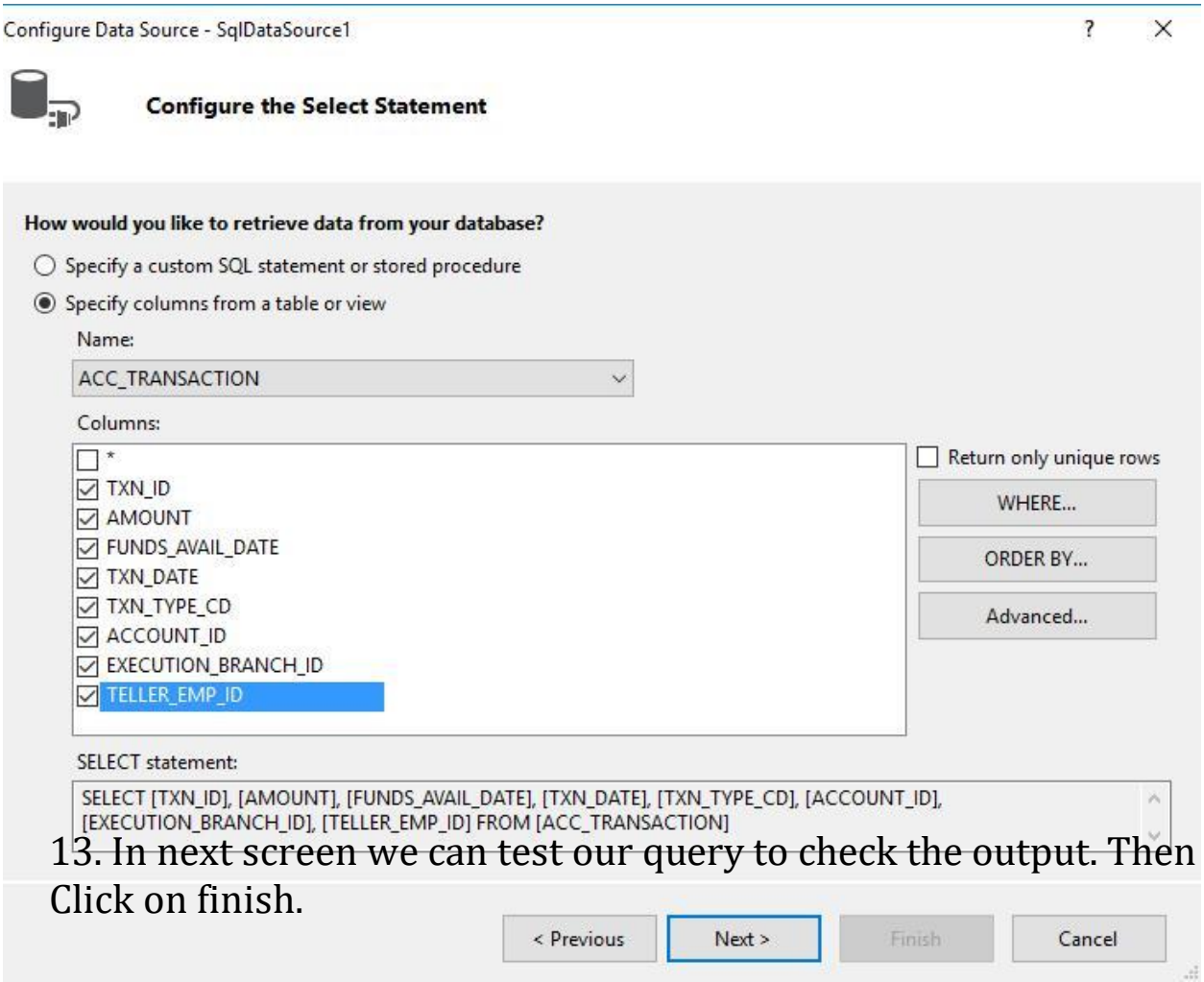


10. Once the Connection is made then click on Next button from Data Source Wizard.



11. Then wizard ask for saving the connection string in configuration file. If you already stored it web.config file then uncheck check box, if you haven't, then select the checkbook. Then click on next button.

12. The next screen gives option to configure the select statement. Here we can choose the table as well as configure the select statement as we need to display the data on web page.



13. In next screen we can test our query to check the output. Then Click on finish.

After successful steps form the Datalist controls option wizard our web page design and output will look like following.

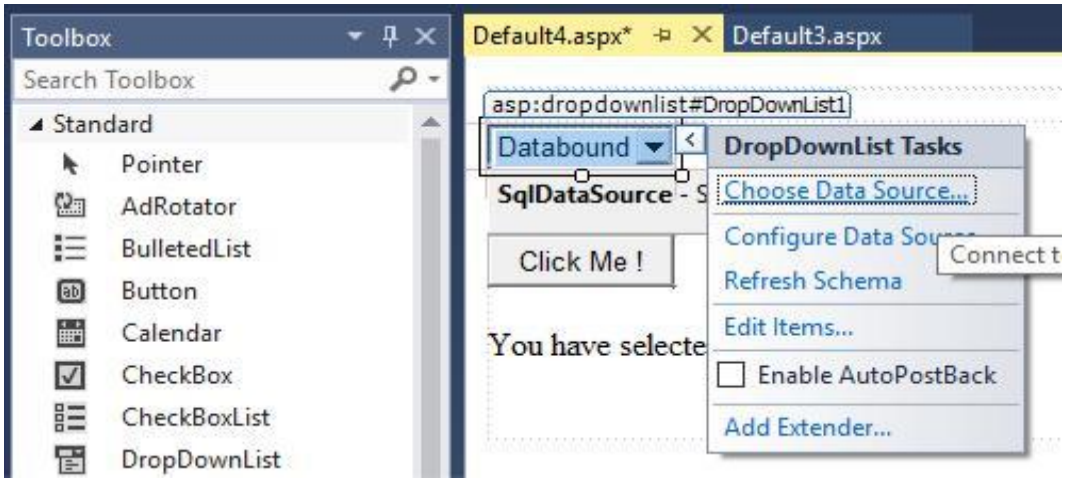
TXN_ID: 1
AMOUNT: 100
FUNDS_AVAIL_DATE: 1/15/2000 12:00:00 AM
TXN_DATE: 1/15/2000 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 1
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 2
AMOUNT: 100
FUNDS_AVAIL_DATE: 1/15/2000 12:00:00 AM
TXN_DATE: 1/15/2000 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 2
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 3
AMOUNT: 100
FUNDS_AVAIL_DATE: 6/30/2004 12:00:00 AM
TXN_DATE: 6/30/2004 12:00:00 AM
TXN_TYPE_CD: CDT
ACCOUNT_ID: 3
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

Practical 7 (a): Create a web application to display Databinding using Dropdownlist control.

1. Create a web page with DropDownList control, one Button and one Label control.
2. Use code to bind the data to DropDownList.



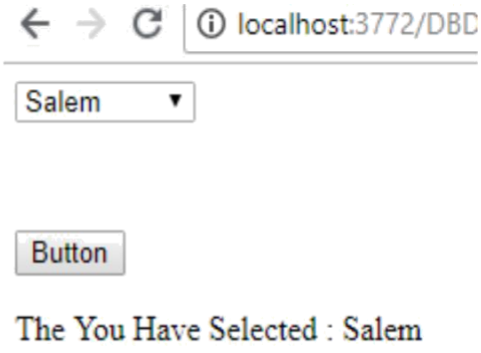
Code of C# Code behind file

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


public partial class DBDropDown : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs
   e)
   {
      if (IsPostBack == false)
      {
         string connStr =
```

```
ConfigurationManager.ConnectionStrings["connStr"].Con
nectionString;

        SqlConnection con = new
        SqlConnection(connStr);

        SqlCommand cmd = new SqlCommand("Select
        Distinct City from Customer", con);
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();

        DropDownList1.DataSource = reader;

        DropDownList1.DataTextField = "City";
        DropDownList1.DataBind();

        reader.Close();

        con.Close();
    }

  }

  protected void Button1_Click(object sender,
  EventArgs e)

  {

    Label1.Text = "The You Have Selected : " +
    DropDownList1.SelectedValue;

  }
}
```

Output:

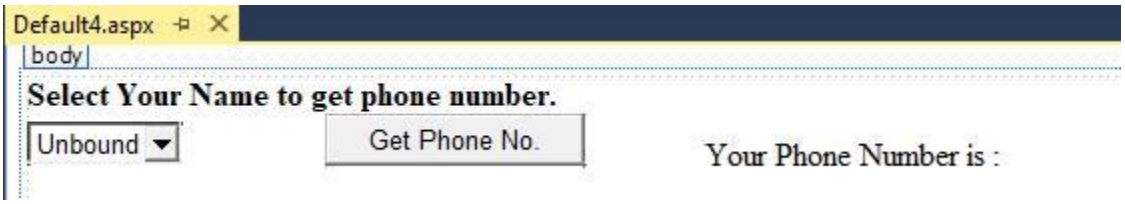

Lynnfield
Lynnfield
Newton
Quincy
Salem
Waltham
Wilmington
Woburn


localhost:3772/DBD
Salem
Button
The You Have Selected : Salem

Practical No: Date:

Practical 7 (b): Create a web application for to display the Postal Code no of Customer using database.

Create a web page with DropDownList, Button and with Label control as shown below.



Code of C# Code behind file

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

```
public partial class PostalCodeByCity :
System.Web.UI.Page
{


   protected void Button1_Click(object sender,
   EventArgs e)
   {
      Label1.Text = ListBox1.SelectedValue;

   }

   protected void Page_Load(object sender, EventArgs
   e)
   {
      if (IsPostBack == false)
      {
         string connStr =
ConfigurationManager.ConnectionStrings["connStr"].Con
nectionString;
         SqlConnection con = new
         SqlConnection(connStr);
         SqlCommand cmd = new SqlCommand("Select
         Distinct POSTAL_CODE from Customer",
con);
         con.Open();
```

```
        SqlDataReader reader = cmd.ExecuteReader();
        ListBox1.DataSource = reader;
        ListBox1.DataTextField = "City";
        ListBox1.DataValueField = "POSTAL_CODE";
        ListBox1.DataBind();

        reader.Close();
        con.Close();

    }
  }
}
```

Output:

Practical 7 (c): Create a web application for inserting and deleting record from a database. (Using Execute-Non Query).

Create a web page with TextBox, and Two Button and one Label control as shown below.

And follow the database related steps same as it is in previous examples.

Code of C# Code behind file

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


public partial class ExecuteNonQuery :
System.Web.UI.Page {
  protected void Button1_Click(object sender, EventArgs e)
  {
    string connStr =
    ConfigurationManager.ConnectionStrings["connStr"].Conne
```

```
    ctionString; SqlConnection con = new
    SqlConnection(connStr);

    string InsertQuery = "insert into BRANCH
values(@ADDRESS, @CITY, @NAME, @STATE, @ZIP_CODE)";

    SqlCommand cmd = new SqlCommand(InsertQuery, con);

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

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

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

    cmd.Parameters.AddWithValue("@STATE", TextBox4.Text);

    cmd.Parameters.AddWithValue("@ZIP_CODE", TextBox5.Text);

    con.Open();

    cmd.ExecuteNonQuery();

    Label1.Text = "Record Inserted Successfuly.";

    con.Close();


  }
  protected void Button2_Click(object sender, EventArgs e)

  {

    string connStr =
    ConfigurationManager.ConnectionStrings["connStr"].Conne
    ctionString; SqlConnection con = new
    SqlConnection(connStr);
```

```
    string InsertQuery = "delete from branch where
    NAME=@NAME";

    SqlCommand cmd = new SqlCommand(InsertQuery, con);

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

    con.Open( );

    cmd.ExecuteNonQuery( );

    Label1.Text = "Record Deleted Successfuly.";

    con.Close( );

  }
}
```