ST.GONSALO GARCIA

COLLEGE

OF

ARTS AND COMMERCE

CERTIFICATE

This is to certify that Master _____ARNOLD AWLYN VAZ_____, Roll No.:___55___ of T.Y.B.SC. (IT) Sem-V has successfully completed practical's of the subject **'Advanced Web Programming'** the year 2020-2021 under the guidance of Prof. Brensa Cerejo.

Internal Examiner's Sign

Head of Department

Date:

Principal

INDEX

Sr.no	Practical	Date	Sign
1	**Working with basic C# and ASP.NET**		
	a. Create an application to demonstrate string operations.		
	b. Create an application to demonstrate following operations i. Generate Fibonacci Series ii. Test for prime numbers iii. Test for vowels iv. Use of foreach loop with arrays v. Reverse a number and find sum of digits of a number		
2	**Working with Object Oriented C# and ASP.Net**		
	a. Create a simple application to perform following operations i. Finding Factorial Value ii. Money Conversion iii. Quadratic Equation iv. Temperature Conversion		
	b. Create a simple application to perform following concepts i. Multiple Inheritance ii. Interfaces		
	c. Create a simple application to perform following concepts i. Using Delegates and events ii. Exception handling		
3	**Working with Web Forms and Controls**		
	a. Create a simple web page with various server controls to demonstrate setting and use of their properties		
	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) Selected day in a calendar control using style		

	d) Difference between two calendar dates		
	c. Demonstrate the use of Treeview control perform the following operations a) Treeview control and datalist b) Treeview operations		
4	**Working with Form Controls**		
	a. Create a Registration form to demonstrate use of various Validation controls		
	b. Create Web Form to demonstrate use of Adrotator Control		
	c. Create Web Form to demonstrate use User Controls		
5	**Working with Navigation, Beautification and Master page**		
	a. Create Web Form to demonstrate use of Website Navigation controls and Site Map		
	b. Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification		
	c. Create a web application to demonstrate various states of ASP.NET Pages		
6	**Working with Database**		
	a. Create a web application bind data in a multiline textbox by querying in another textbox		
	b. Create a web application to display records by using database.		
	c. Demonstrate the use of Datalist link control.		
7	**Working with Database**		
	a. Create a web application to display Databinding using dropdownlist control		
	b. Create a web application for to display the phone of an author using database		
	c. Create a web application for inserting and deleting record from a database		
8	**Working with data controls**		
	a. Create a web application to demonstrate various uses and properties of SqlDataSource		
	b. Create a web application to demonstrate data binding using DetailsView and FormView		
9	**Working with GridView control**		

	a. Create a web application to demonstrate use of GridView control template and GridView hyperlink		
	b. Create a web application to demonstrate use of GridView button column and GridView events		
10	**Working with AJAX and XML**		
	a. Create a web application to demonstrate reading and writing operation with XML		
	b. Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization properties		
	c. Create a web application to demonstrate use of various AJAX controls		
11	**Program to create and use DLL**		

Practical 1

Practical no :- 1

Aim :- Working with basic C# and ASP.Net

Theory :- * String functions :-

1) String.ToUpper() - To Upper function converts string to upper case.

2) String.ToLower() - ToLower converts string to lower case.

3) String.Trim() - Trim function removes extra spaces from the beginning and the ending of string

4) String.Substring() - substring method returns substring of a string

5) Length - length is a string property that return a number of characters in a string and here spaces count as characters.

Conclusion : Hence, we have performed above practical successfully.

Practical 1(a)

Aim: Create an application to demonstrate string operations

Default.aspx.cs

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

public partial class _Default : System.Web.UI.Page
{
protected void Button1_Click1(object sender, EventArgs e)
    {
        string s = TextBox1.Text;
        Label1.Text = "String Length:" + s.Length;
        Label2.Text = "Substring:" + s.Substring(4, 3);
        Label3.Text = "Upper String" + s.ToUpper();
        Label4.Text = "Lower String:" + s.ToLower();
        string rev = "";
        for (int i = s.Length - 1; i >= 0; i--)
        {
            rev = rev + s[i];
        }
        Label5.Text = "Reverse String:" + rev.ToString();
        Label6.Text = "Replace 's' by 't' in String:" + s.Replace('s', 't');
        Label7.Text = "Insert 'u' in String:" + s.Insert(3, "u");
        Label8.Text = "String Truncate:" + s.Trim();
        Label9.Text = "Remove String:" + s.Remove(4);
        Label10.Text = "Index of String:" + s.IndexOf('e');
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Label1.Text = "";
        Label2.Text = "";
        Label3.Text = "";
        Label4.Text = "";
        Label5.Text = "";
        Label6.Text = "";
        Label7.Text = "";
        Label8.Text = "";
        Label9.Text = "";
        Label10.Text = "";
        TextBox1.Text = "";
    }
}
```

OUTPUT:

Enter a String: Vedshree

Result Reset

String Length:8

Substring:hre

Upper StringVEDSHREE

Lower String:vedshree

Reverse String:eerhsdeV

Replace 's' by 't' in String:Vedthree

Insert 'u' in String:Vedushree

String Truncate:Vedshree

Remove String:Veds

Index of String:1

Practical 1(b)

Aim: Create an application to demonstrate following operations:

i) **Generate Fibonacci series**
ii) **Test for prime numbers**
iii) **Test for vowels**
iv) **Use of foreach loop with arrays**
v) **Reverse a number and find sum of digits of a number.**

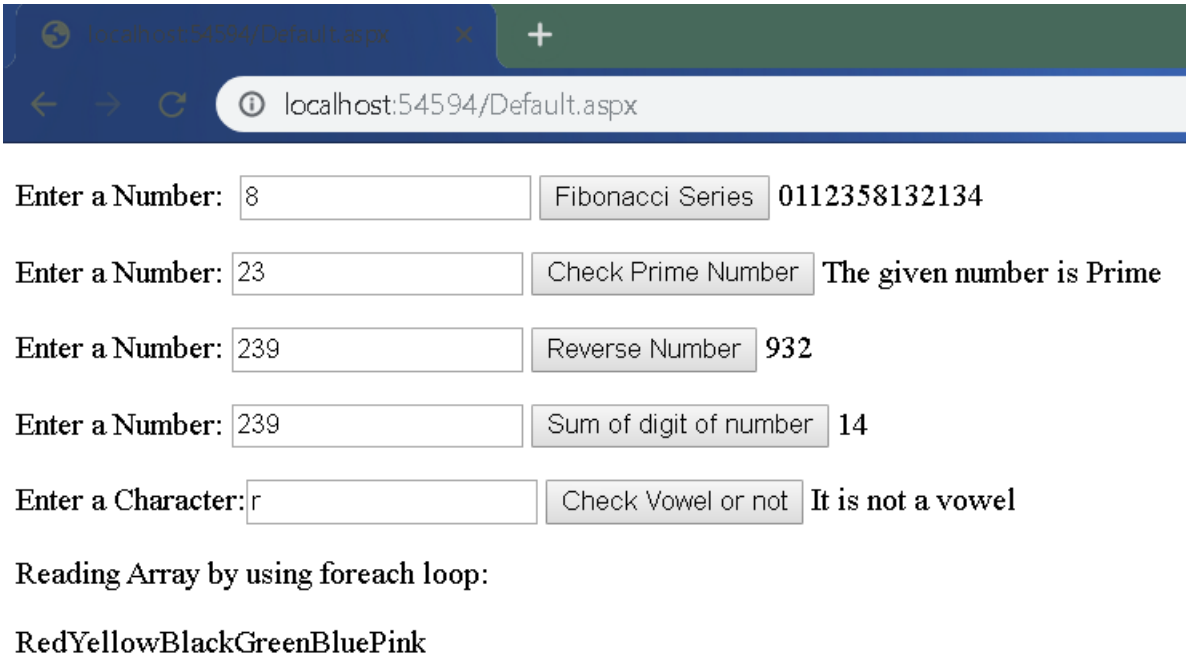
Default.aspx

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

public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      Label6.Text = "";
      string[] ColorNames = new string[] { "Red", "Yellow", "Black", "Green", "Blue", "Pink" };
      foreach (string ColorName in ColorNames)
      {
         Label6.Text = Label6.Text + "" + ColorName.ToString();
      }
   }


   protected void Button1_Click(object sender, EventArgs e)
   {
      int a, b, c, i, n;
      a = 0;
      b = 1;
      Label1.Text = a.ToString() + b.ToString();
      n = Convert.ToInt32(TextBox1.Text);
      for (i=1; i<=n; ++i)
      {
         c = a + b;
         Label1.Text = Label1.Text + c.ToString();
         a = b;
         b = c;
      }
   }

   protected void Button2_Click(object sender, EventArgs e)
   {
      int i, c = 0, j, num;
      num = Convert.ToInt32(TextBox2.Text);
```

```
    for (j = 1; j <= num; j++)
    {
        i = num % j;
        if (i==0)
        {
            c = c + 1;
        }
    }
    if (c == 2)
        Label2.Text = "The given number is Prime";
    else
        Label2.Text = "The given number is not Prime";
}

protected void Button3_Click(object sender, EventArgs e)
{
    long num, i, sum = 0;
    num = Convert.ToInt32(TextBox3.Text);
    while (num > 0)
    {
        i = num % 10;
        sum = i + sum * 10;
        num = num / 10;
    }
    Label3.Text = sum.ToString();
}

protected void Button4_Click(object sender, EventArgs e)
{
    long num, i, sum = 0;
    num = Convert.ToInt32(TextBox4.Text);
    while (num > 0)
    {
        i = num % 10;
        sum = i + sum;
        num = num / 10;
    }
    Label4.Text = sum.ToString();
}

protected void Button5_Click(object sender, EventArgs e)
{
    char c = Convert.ToChar(TextBox5.Text);
    switch (c)
    {

        case 'a':
            Label5.Text = "a is a vowel";
            break;
        case 'e':
            Label5.Text = "e is a vowel";
            break;
        case 'i':
```

```
            Label5.Text = "i is a vowel";
            break;
        case 'o':
            Label5.Text = "o is a vowel";
            break;
        case 'u':
            Label5.Text = "u is a vowel";
            break;
        default:
            Label5.Text = "It is not a vowel";
            break;
    }
  }
}
```

OUTPUT:



Practical 2

Practical no :- 2

Aim : Working with object oriented C# and asp.net

Theory : Exceptions provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords : try, catch, finally, and throw.

- try - A try block identifies a block of code for which particular expression of exceptions is activated. It is followed by one or more catch blocks.

- catch - A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.

- finally - The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example - if you open a file, it must be closed whether an exception is raised or not.

- throws :- A program throws an exception when a problem shows up. This is done using a Throw keyword.

Conclusion :- Hence, we successfully performed the above practical.

Practical 2(a)

Aim: Create a simple application to perform following operations:

i) **Finding factorial Value**
ii) **Money Conversion**
iii) **Quadratic Equation**
iv) **Temperature Conversion**

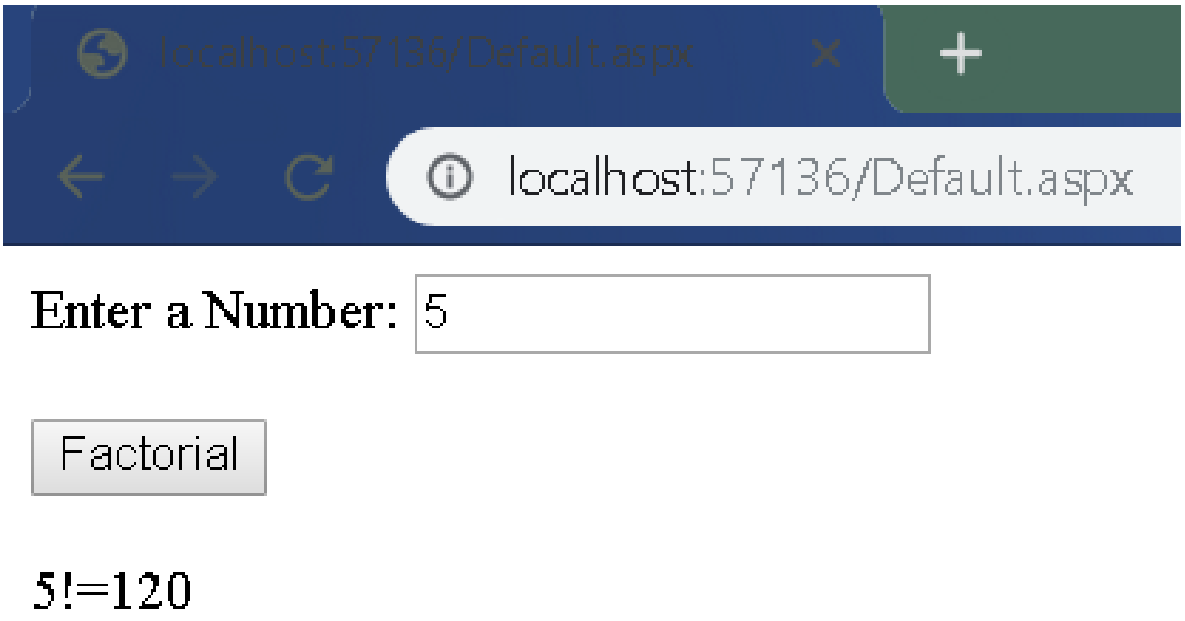
Finding factorial Value

Default.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
class fact
{
   public int n, f;
   public fact()
   {
      f = 1;
   }
   public void cal()
   {
      int i;
      for (i = 1; i <= n; i++)
      {
         f = f * i;
      }
   }
}
public partial class _Default : 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)
   {
      fact f1 = new fact();
      f1.n = 5;
      f1.cal();
      Label1.Text = (f1.n + "!=" + f1.f);
```

```
}
  }
```

OUTPUT:

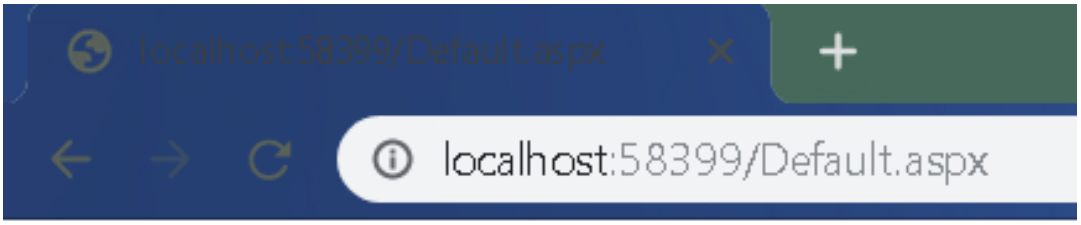

localhost:57136/Default.aspx
localhost:57136/Default.aspx
Enter a Number: 5
Factorial
5!=120

Money Conversion

Default.aspx

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

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

   }

   protected void Button1_Click(object sender, EventArgs e)
   {
      int amt;
      int dollar;
      amt = Convert.ToInt16(TextBox1.Text);
      dollar = amt / 70;
      Label1.Text = dollar.ToString();
   }
}
```

OUTPUT:



Enter amount in Rupees: 28000

US Dollars

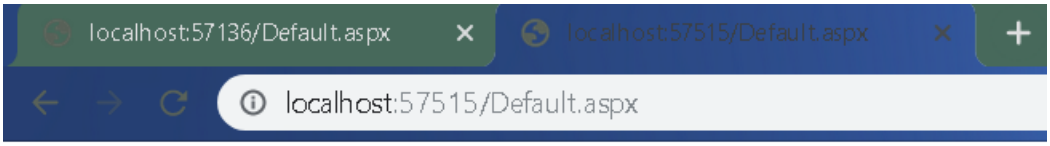
400

Quadratic Equation

Default.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   public void demo()
   {
      double a, b, c, r1, r2, x;
      double det;
      a = Convert.ToInt32(TextBox1.Text);
      b = Convert.ToInt32(TextBox2.Text);
      c = Convert.ToInt32(TextBox3.Text);
      det = (b * b) - (4 * a * c);
      if(det > 0)
      {
         x = Math.Sqrt(det);
         r1 = (-b + x) / (2 * a);
         r2 = (-b - x) / (2 * a);
         Label3.Text = "There are two roots::";
         Label1.Text = r1.ToString();
         Label2.Text = r2.ToString();

      }
      else  if (det == 0)
      {
         x = Math.Sqrt(det);
         r1 = (-b + x) / (2 * a);
         Label1.Text = "There is only one root:";
         Label2.Text = r1.ToString();
      }
      else
      {
         Label1.Text = "There is no root!!!!";
      }
   }
   protected void Page_Load(object sender, EventArgs e)
   {

   }


   protected void Button1_Click(object sender, EventArgs e)
   {
      demo();
   }
}
```

OUTPUT:



Enter a: 2

Enter b: 5

Enter c: -3

Result Reset

There are two roots::

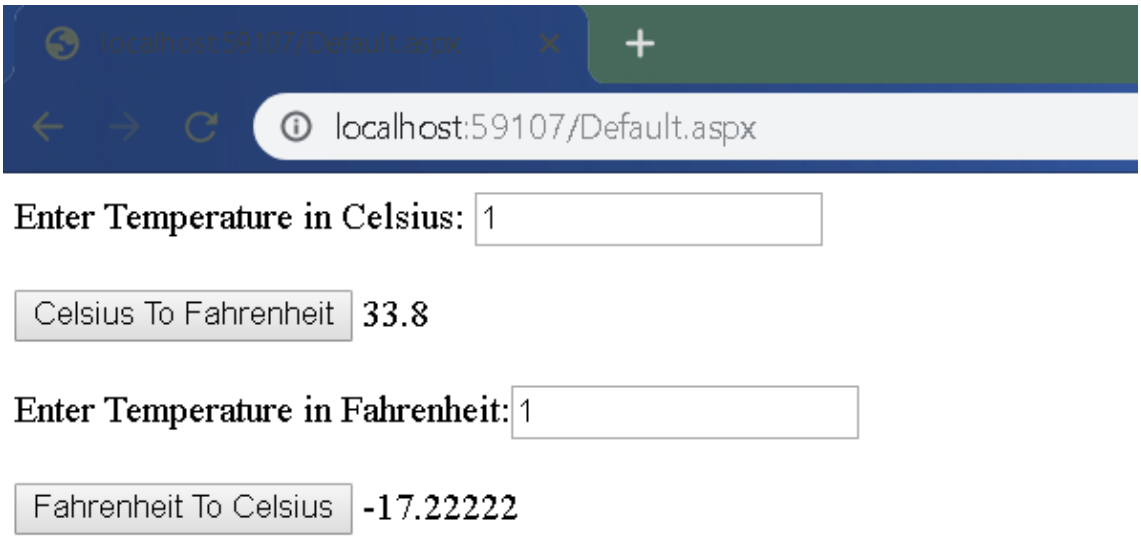
0.5 -3

Temperature Conversion

Default.aspx

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

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

   }

   protected void Button1_Click(object sender, EventArgs e)
   {
      float celsius;
      float fahrenheit;
      celsius = Convert.ToInt32(TextBox1.Text);
      fahrenheit = celsius * 9 / 5 + 32;
      Label1.Text = fahrenheit.ToString();

   }

   protected void Button2_Click(object sender, EventArgs e)
   {
      float celsius;
      float fahrenheit;
      fahrenheit = Convert.ToInt32(TextBox2.Text);
      celsius = (fahrenheit - 32) * 5 / 9;
      Label2.Text = celsius.ToString();
   }
}
```

OUTPUT:


localhost:59107/Default.aspx
localhost:59107/Default.aspx
Enter Temperature in Celsius: 1
Celsius To Fahrenheit
33.8
Enter Temperature in Fahrenheit: 1
Fahrenheit To Celsius
-17.22222

Practical 2(b)

Aim: Create simple application to demonstrate use of following concepts

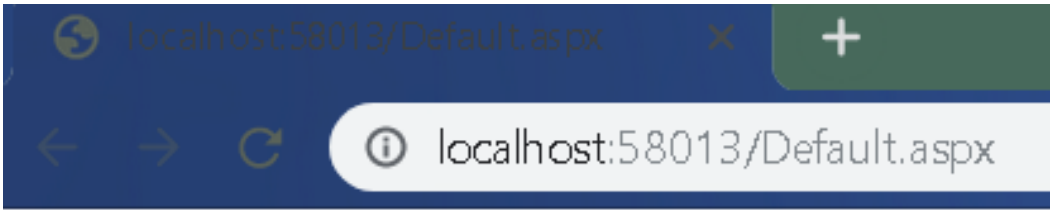
i) **Multiple Inheritance**
ii) **Interfaces**

Multiple Inheritance

Dafault.aspx

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

public partial class _Default : System.Web.UI.Page
{
   interface Area
   {
      double show(double s, double t);
   }
   class Rect : Area
   {
      public double show(double s, double t)
      {
         return s * t;
      }
   }
   class Circle : Area
   {
     public double show(double s, double t)
      {
         return (3.14 * s * s);
      }
   }

   protected void Page_Load(object sender, EventArgs e)
   {
      Rect r1 = new Rect();
      double x = r1.show(3, 4);
      Circle c1 = new Circle();
      double y = c1.show(3, 4);
      Label1.Text = x.ToString();
      Label2.Text = y.ToString();
   }
}
```

OUTPUT:



Multiple Inheritance

Area of a Circle:12

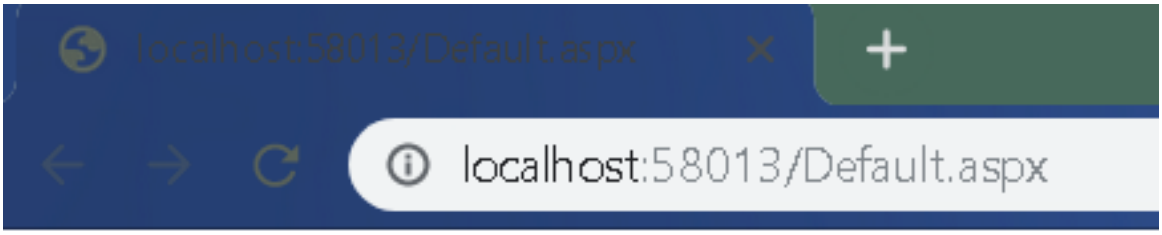
Area of a Rectangle: 28.26

Interfaces

Default.aspx

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

public partial class _Default : System.Web.UI.Page
{
   interface Area
   {
      double show(double s, double t);
   }
   class Rect : Area
   {
      public double show(double s, double t)
      {
         return s * t;
      }
   }
   class Circle : Area
   {
     public double show(double s, double t)
      {
         return (3.14 * s * s);
      }
   }

   protected void Page_Load(object sender, EventArgs e)
   {
      Rect r1 = new Rect();
      double x = r1.show(3, 4);
      Circle c1 = new Circle();
      double y = c1.show(3, 4);
      Label1.Text = x.ToString();
      Label2.Text = y.ToString();
   }
}
```

OUTPUT:



Area of a circle and rectangle using interface

Area of a Circle:12

Area of a Rectangle: 28.26

Practical 2(c)

Aim: Create simple application to demonstrate use of following concepts

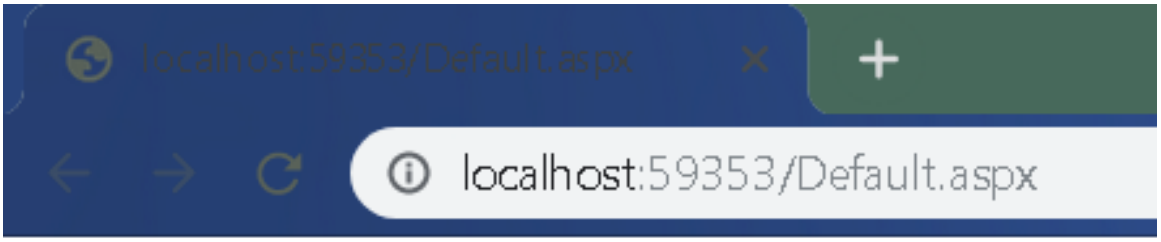
i) **Using Delegates and events**
ii) **Exception Handling**

Using Delegates and events

Default.aspx

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

public partial class _Default : System.Web.UI.Page
{
   public delegate string dele();
   public static string display1()
   {
      string s1 = "Yashashree Sambare";
      return s1;
   }
   public static string display2()
   {
      string s2 = "Vedshree Sambare";
      return s2;
   }
   protected void Page_Load(object sender, EventArgs e)
   {
      dele d1 = new dele(display1);
      d1();
      dele d2 = new dele(display2);
      d2();
      Label1.Text = d1();
      Label2.Text = d2();
   }
}
```

OUTPUT:



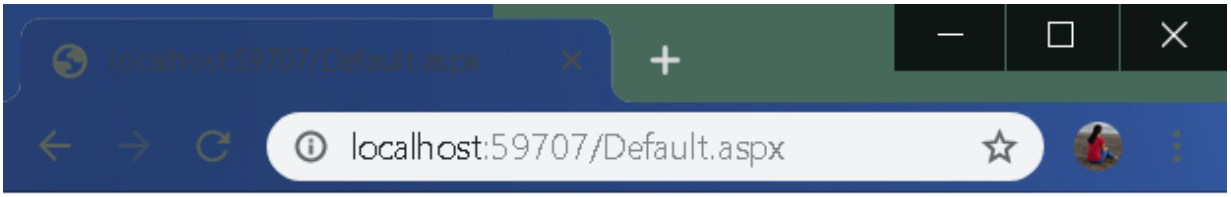
Yashashree Sambare
Vedshree Sambare

Exception Handling

Default.aspx

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

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

   }

   protected void Button1_Click(object sender, EventArgs e)
   {
      try
      {
         int a = Convert.ToInt32(TextBox1.Text);
         int[] b = { 12, 23, 33 };
         int resultVal;
         resultVal = (b[3] / a);
         Label1.Text = "The Result is:" + resultVal.ToString();
      }
      catch (System.DivideByZeroException ex)
      {
         Label1.Text = ex.ToString();
      }
      catch (System.IndexOutOfRangeException ex)
      {
         Label1.Text = ex.ToString();
      }
   }
}
```

OUTPUT:



Division of two numbers:

Num1: 23

Result

System.IndexOutOfRangeException: Index was outside the bounds of the array. at _Default.Button1_Click(Object sender, EventArgs e) in c:\Users\Geeta\Documents\Visual Studio 2015\WebSites\WebSite25\Default.aspx.cs:line 22

Practical 3

Practical no:- 3

Aim: Working with Web Forms and Controls

Theory: Server Controls are the tags that are understood by the server. There are basically three types of server controls.

- HTML Server Controls - Traditional HTML tags
- Web Server Controls - New ASP.NET tags

ASP.NET HTML Server Controls :-

- ASP.NET provides a way to work with HTML Server controls on the server side; programming with a set of controls collectively is called HTML controls

ASP.NET Web Server Controls :-

- Web server controls are special ASP.NET tags are understood by the server.
- Like HTML server controls, Web server controls are also created on the server and they require a runat = "server" attribute to work.
- However, Web server controls do not necessarily map to any existing HTML elements and they may represent more complex elements.
- Mostly all Web Server controls inherit from a common base class, namely the Web Control class defined in the System.Web.UI.WebControls namespace.

Conclusion: Hence, we have successfully created forms and performed the practical.

Practical 3(a)

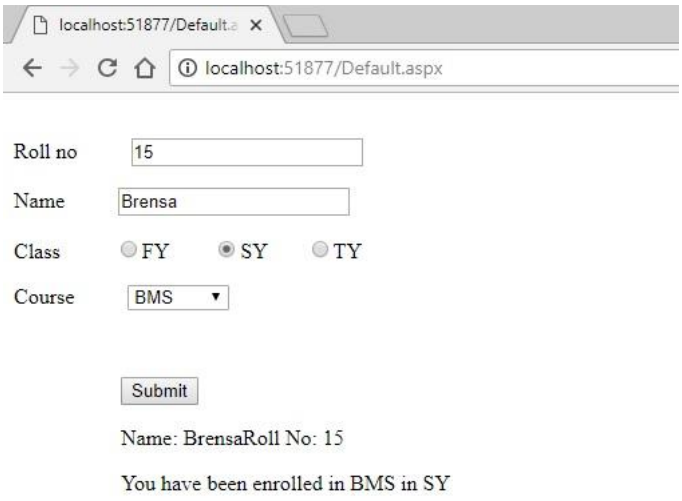
Aim: Create a simple web page with various server controls to demonstrate setting and use of their properties (Example: AutoPostBack)

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   { }
   protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
   {
      RadioButton2.Checked = false;
      RadioButton3.Checked = false;
   }
  protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
   {
      Label6.Text = "You have been enrolled in " + DropDownList1.SelectedItem;
   }
  protected void Button1_Click(object sender, EventArgs e)
   {
      string s,name;
      int rno;
      name = TextBox2.Text;
      rno = Convert.ToInt32(TextBox1.Text);
      if (RadioButton1.Checked == true)
      {
         s = RadioButton1.Text;
      }
      else if (RadioButton2.Checked == true)
      {
         s = RadioButton2.Text;
       }
      else
      {
         s = RadioButton3.Text;
      }
      Label5.Text += "Name: " + name + "Roll No: " + rno.ToString();
      Label6.Text += " in " + s;
   }
   protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
   {
      RadioButton1.Checked = false;
      RadioButton3.Checked = false;
   }
   protected void RadioButton3_CheckedChanged(object sender, EventArgs e)
   {
      RadioButton1.Checked = false;
      RadioButton2.Checked = false;
```

```
    }
}
```

OUTPUT:


localhost:51877/Default.a
localhost:51877/Default.aspx
Roll no
15
Name
Brensa
Class
FY
SY
TY
Course
BMS
Submit
Name: BrensaRoll No: 15
You have been enrolled in BMS in SY

Practical 3(b)

Aim: 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) **Selected day in calendar control using style**
d) **Difference between two dates**

Default.aspx

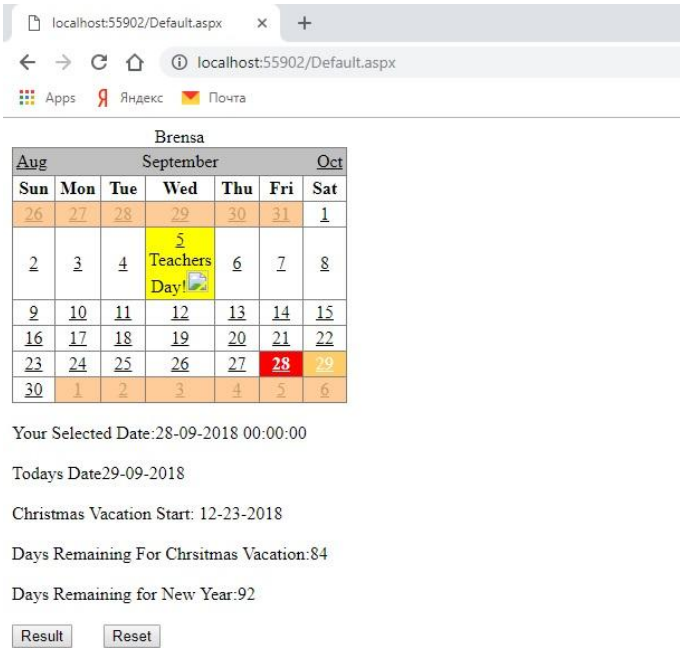
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>  </div>
 <asp:Calendar ID="Calendar1" runat="server"  NextPrevFormat="ShortMonth" OnDayRender="Calendar1_DayRender"
ShowGridLines="True" Width="300px"OnSelectionChanged="Calendar1_SelectionChanged">
  <OtherMonthDayStyle BackColor="#FFCC99" BorderStyle="Solid" ForeColor="#CC9966" />
<SelectedDayStyle BackColor="Red" Font-Bold="True" />
<TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
  </asp:Calendar><br />
      <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label5" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Result" />
     
      <asp:Button ID="Button2" runat="server" Text="Reset" />
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
```

```
{
   protected void Page_Load(object sender, EventArgs e)
   { }
  protected void Button1_Click(object sender, EventArgs e)
   {
      Calendar1.Caption = "Brensa";
      Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;
      Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;
      Calendar1.TitleFormat = TitleFormat.Month;
      Label2.Text = "Todays Date" + Calendar1.TodaysDate.ToShortDateString();
      Label3.Text = "Christmas Vacation Start: 12-23-2018";
      TimeSpan d = new DateTime(2018, 12, 23) - DateTime.Now;
      Label4.Text = "Days Remaining For Chrsitmas 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() == "12-23-2018")
         Label3.Text = "<b>Christmas Start</b>";
      if (Calendar1.SelectedDate.ToShortDateString() == "1-2-2019")
         Label3.Text = "<b>Christmas End</b>";
   }
 protected void Calendar1_SelectionChanged(object sender, EventArgs e)
   {
      Label1.Text = "Your Selected Date:" + Calendar1.SelectedDate.Date.ToString();
   }
   protected void btnReset_Click(object sender, EventArgs e)
   {
      Label1.Text = "";
      Label2.Text = "";
      Label3.Text = "";
      Label4.Text = "";
      Label5.Text = "";
      Calendar1.SelectedDates.Clear();
   }
  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 == 7 && e.Day.Date.Month == 11)
      {
         Calendar1.SelectedDate = new DateTime(2018, 11, 7);
         Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate,
         Calendar1.SelectedDate.AddDays(5));
         Label lbl1 = new Label();
         lbl1.Text = "<br>Diwali!";
         e.Cell.Controls.Add(lbl1);
      }
   }
}
```

OUTPUT:


localhost:55902/Default.aspx
localhost:55902/Default.aspx
Apps
Яндекс
Почта
Brensa
Aug
September
Oct
Sun Mon Tue Wed Thu Fri Sat
26 27 28 29 30 31 1
2 3 4 5 Teachers Day! 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 1 2 3 4 5 6
Your Selected Date:28-09-2018 00:00:00
Todays Date29-09-2018
Christmas Vacation Start: 12-23-2018
Days Remaining For Chrsitmas Vacation:84
Days Remaining for New Year:92
Result
Reset

Practical 3(c)

Aim: Demonstrate the use of Treeview Control and Datalist to perform following operations.

a) **Treeview control and datalist**
b) **Treeview operation**

Treeview Control and datalist

XMLFile

```
<?xml version="1.0" encoding="utf-8" ?>
<studentdetail>
 <student>
  <sid>1</sid>
  <sname>ABC</sname>
  <sclass>TYIT</sclass>
 </student>
<student>
  <sid>2</sid>
  <sname>XYZ</sname>
  <sclass>SYIT</sclass>
 </student>
<student>
  <sid>3</sid>
  <sname>PQR</sname>
  <sclass>FYIT</sclass>
 </student>
</studentdetail>
```

Default.aspx

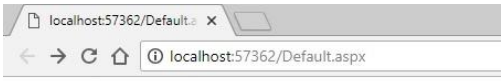
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>   Treeview Conrol navigation
    <asp:TreeView ID="TreeView1" runat="server" ImageSet="Arrows">
      <HoverNodeStyle Font-Underline="True" ForeColor="#336699" />
      <Nodes>
        <asp:TreeNode Text="ASP.NET Pracs" Value="ASP.NET Pracs">
          <asp:TreeNode NavigateUrl="~/CalendarControl.aspx" Text="Calendar Control" Value="Calendar
          Control"></asp:TreeNode>
          <asp:TreeNode NavigateUrl="~/ConstructorOverloading.aspx" Text="Constructor Overloading" Value="Constructor
          Overloading"></asp:TreeNode>
          <asp:TreeNode NavigateUrl="~/SingleInheritance.aspx" Text="Single Inheritance" Value="Single
          Inheritance"></asp:TreeNode>
          <asp:TreeNode NavigateUrl="~/ClassProperties.aspx" Text="Class properties" Value="Class
          properties"></asp:TreeNode>
        </asp:TreeNode>
      </Nodes>
    </asp:TreeView>
    <br /> <br /> Datalist Control<br />
  </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") %><br/>
              </td>
            </tr>
          </table>
        </ItemTemplate>
      </asp:DataList>
    </form>
</body>
</html>
```

Default.aspx.cs

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



Treeview Conrol navigation

- ASP.NET Pracs
  - Calendar Control
  - Constructor Overloading
  - Single Inheritance
  - Class properties

Datalist Control

Roll num:1 Name:ABC Class:TYIT
Roll num:2 Name:XYZ Class:SYIT
Roll num:3 Name:PQR Class:FYIT

Treeview Operations

Default.aspx

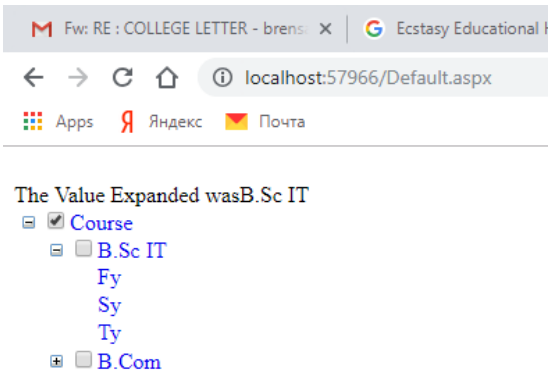
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
    <asp:TreeView ID="TreeView1" runat="server" OnSelectedNodeChanged="TreeView1_SelectedNodeChanged"
OnTreeNodeCollapsed="TreeView1_TreeNodeCollapsed" OnTreeNodeExpanded="TreeView1_TreeNodeExpanded">
        <Nodes>
          <asp:TreeNode Checked="True" ShowCheckBox="True" Text="Course" Value="Course">
            <asp:TreeNode Checked="True" ShowCheckBox="True" Text="B.Sc IT" Value="B.Sc IT">
              <asp:TreeNode Text="Fy" Value="Fy"></asp:TreeNode>
              <asp:TreeNode Text="Sy" Value="Sy"></asp:TreeNode>
              <asp:TreeNode Text="Ty" Value="Ty"></asp:TreeNode>
            </asp:TreeNode>
            <asp:TreeNode Checked="True" ShowCheckBox="True" Text="B.Com" Value="B.Com">
              <asp:TreeNode Text="Fy" Value="Fy"></asp:TreeNode>
              <asp:TreeNode Text="Sy" Value="Sy"></asp:TreeNode>
              <asp:TreeNode Text="Ty" Value="Ty"></asp:TreeNode>
            </asp:TreeNode>
          </asp:TreeNode>
        </Nodes>
      </asp:TreeView>
   </div>
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   { }
  protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
   {
      Response.Write("You have selected the option" + TreeView1.SelectedValue);
   }
   protected void TreeView1_TreeNodeCollapsed(object sender,TreeNodeEventArgs e)
   {
      Response.Write("The Value collapsed was" + e.Node.Value);
   }
   protected void TreeView1_TreeNodeExpanded(object sender, TreeNodeEventArgs e)
   {
```

```
        Response.Write("The Value Expanded was" + e.Node.Value);
    }
}
```

OUTPUT:



Practical 4

Practical no :- 4

Aim :- Working with Form Controls

Theory :- Validation Control

→ Validation is important part of any web application User's input must always be validated before sending across different layers of the application.

Validation Control	Description
Required Field Validation	Makes an input control a required field.
Compare Validator	Compares the value of one input control to the value of another input control or to a fixed value
Range Validator	Checks that the user enters a value that falls between two value.
Regular Expression Validator	Ensures that the value of an input control matches a specified pattern
Custom Validator	Allows you to write a method to handle the validation of the value entered

Conclusion :- Hence, we have successfully completed the above practical.

Practical 4(a)

Aim: Create a Registration form to demonstrate use of various Validation controls

Default.aspx

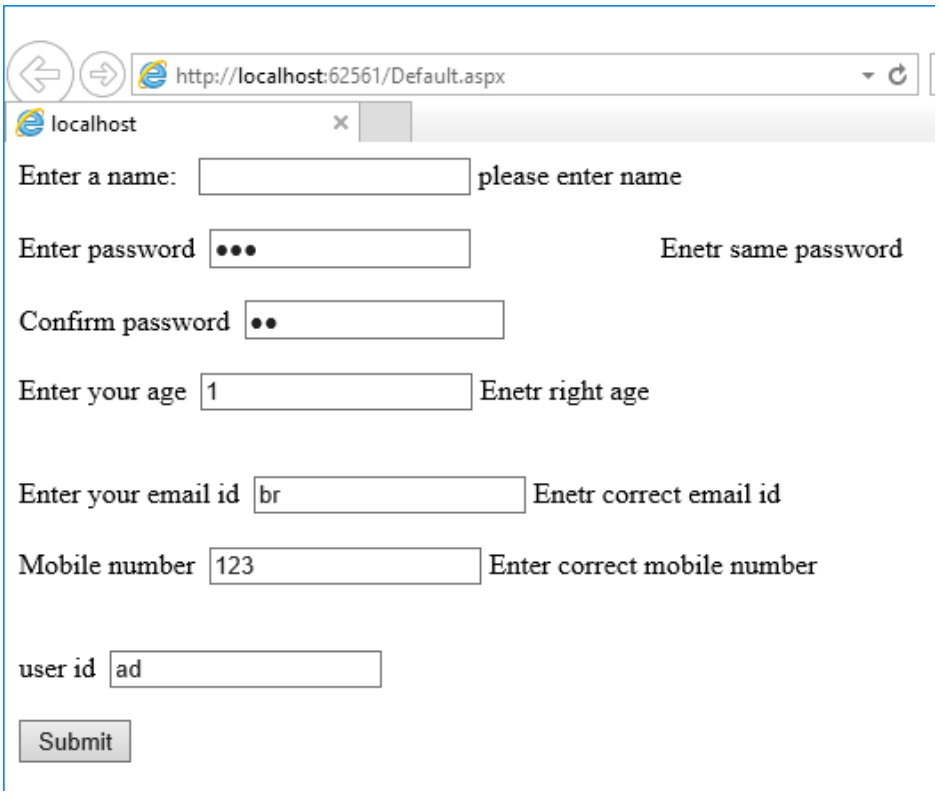
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
  Enter a name:  
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="TextBox1"
ErrorMessage="please enter name"></asp:RequiredFieldValidator>
    <br /> <br />
    Enter password 
    <asp:TextBox ID="TextBox2" runat="server" TextMode="Password"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="TextBox2"
ErrorMessage="Enter passwrd"></asp:RequiredFieldValidator>
 
    <asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="TextBox2"
ControlToValidate="TextBox3" ErrorMessage="Enetr same password"></asp:CompareValidator>
    <br />  <br />
    Confirm password 
    <asp:TextBox ID="TextBox3" runat="server" TextMode="Password"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="TextBox3"
ErrorMessage="Reenter password"></asp:RequiredFieldValidator>
    <br />  <br />
    Enter your age 
    <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
    <asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="TextBox4" ErrorMessage="Enetr right age"
MaximumValue="50" MinimumValue="18"></asp:RangeValidator>
    <br /> <br /> <br />
    Enter your email id 
    <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
    <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="TextBox5"
ErrorMessage="Enetr correct email id" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
    <br /> <br />
    Mobile number 
    <asp:TextBox ID="TextBox7" runat="server"></asp:TextBox>
    <asp:RegularExpressionValidator ID="RegularExpressionValidator2" runat="server" ControlToValidate="TextBox7"
ErrorMessage="Enter correct mobile number" ValidationExpression="^[0-9]{10}"></asp:RegularExpressionValidator>
    <br />  <br />  <br />
    user id 
    <asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>
    <asp:CustomValidator ID="CustomValidator1" runat="server" ControlToValidate="TextBox6" ErrorMessage="Enetr right
value" OnServerValidate="CustomValidator1_ServerValidate"></asp:CustomValidator>
    <br />  <br />
    <asp:Button ID="Button1" runat="server" Text="Submit" OnClick="Button1_Click" />
  </div>
  </form>
```

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

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {}
  protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
   {
      if(args.Value.Length<5)
      {
         args.IsValid = false;
      }
      else
      {
         args.IsValid = true;
      }
}
}
```

OUTPUT:



Practical 4(b)
Aim: Create a web form to demonstrate use of Adrotator Control

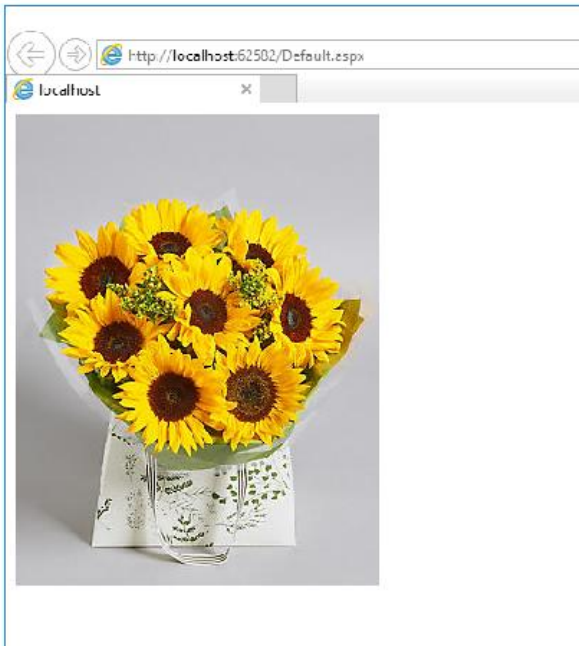
XMLFile.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
 <Ad>
  <ImageUrl>h1.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>h2.jpg</ImageUrl>
  <NavigateUrl>http://www.babybouquets.com.au</NavigateUrl>
  <AlternateText>Order roses and flowers</AlternateText>
  <Impressions>20</Impressions>
  <Keyword>gifts</Keyword>
 </Ad>
<Ad>
  <ImageUrl>h3.jpg</ImageUrl>
  <NavigateUrl>http://www.google.com.au</NavigateUrl>
  <AlternateText>Order flowers</AlternateText>
  <Impressions>20</Impressions>
  <Keyword>petals</Keyword>
 </Ad>
</Advertisements>
```

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
      <asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/XMLFile.xml" Target =  "_blank"/>
   </div>
  </form>
</body>
</html>
```

Output



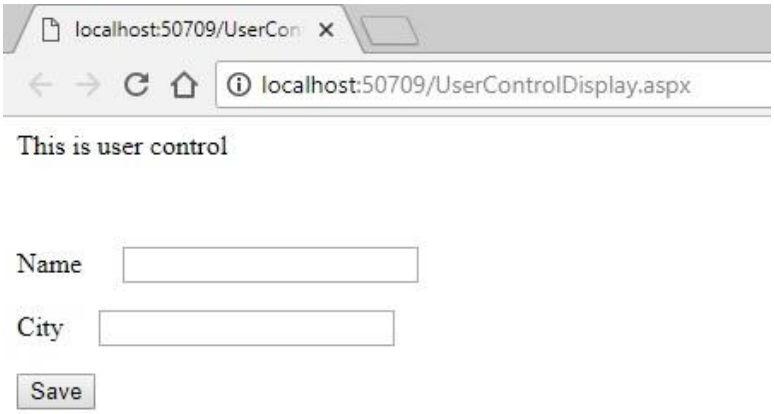
Practical 4(c)
Aim: Create Web form to demonstrate use of User Control

Add Web User Control
Website→Add→Web User Control→Name it MyUserControl

UserControlDisplay.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="UserControlDisplay.aspx.cs" Inherits="UserControlDisplay" %>
<%@ Register Src="~/MyUserControl.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>
```

Output:



Practical 5

Practical no:- 5

Aim:- Working with Navigation, Beautification and Master page.

Theory:- Basically ASP.NET 2.0 has three navigation controls
1. Dynamic menus
2. Tree Views
3. Site Map Path

1) Dynamic menus
→ It was the very difficult task to maintain the menu of a large website and time consuming. It is used to display the Menus. You can use it as easy as other Navigation controls. Menu can be stored in a file to make it easier to maintain. This file is normally called web. Sitemap, and is stored in the root directory of the web.

2) Tree Views
→ A Tree View control displays a hierarchial list of items using lines to connect related items in a hierarchy. Each item consists of a label and an optional bitmap. Windows Explorer use a Tree View.

3) Site Map Path
→ Use of this control is very simple. You can add this control to your page then view your page in browser.

Conclusion: Hence, we successfully performed practicals.

Practical 5(a)
Aim: Create Web form to demonstrate use of Website Navigation and Site Map

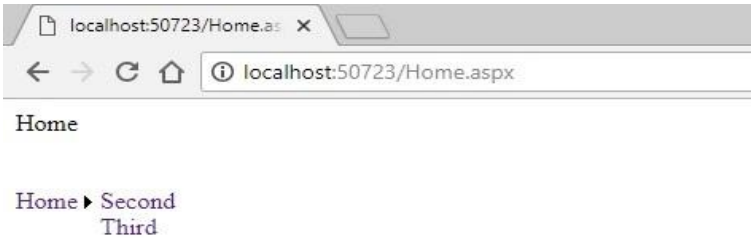
Web.Sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
   <siteMapNode url="Home.aspx" title="Home"  description="">
      <siteMapNode url="default.aspx" title="Second"  description="" />
      <siteMapNode url="default2.aspx" title="Third"  description="" />
   </siteMapNode>
</siteMap>
```

Home.aspx

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

OUTPUT:



Practical 5(b)

Aim: Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification.

StyleSheet.css

```
body {
   background-color:gray;
   font-style:italic;
   font-size:18px;
   font-family:Cambria;
}
```

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></title>
   <asp:ContentPlaceHolder id="head" runat="server">
   </asp:ContentPlaceHolder>
   <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
       </asp:ContentPlaceHolder>
      <br /> Welcome to master<br /> <br />
   </div>
   </form>
</body>
</html>
```

Default.aspx

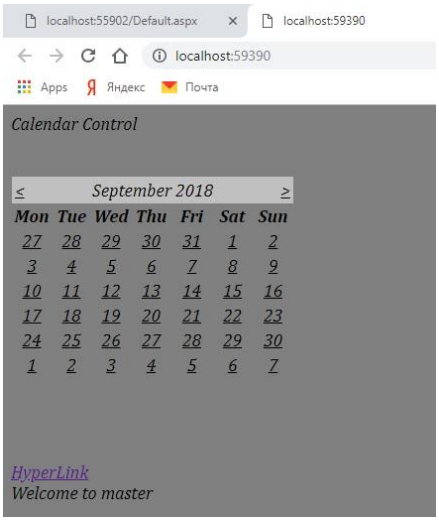
```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
   <asp:Label ID="Label1" runat="server" Text="Calendar Control"></asp:Label>
  <br /> <br />  <br />
   <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
   <br />  <br />   <br />  <br />
   <asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Default2.aspx" >HyperLink</asp:HyperLink>
</asp:Content>
```

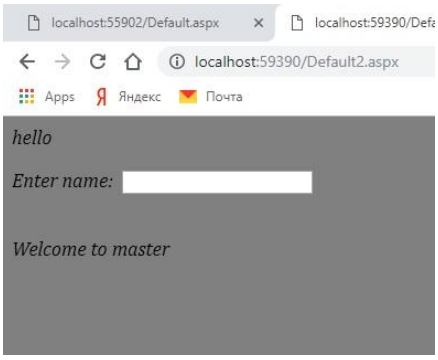
Default2.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
   hello
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
   <p>
```

```
    Enter name: 
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
</p></asp:Content>
```

OUTPUT:





Practical 5(c)

Aim: Create a web application to demonstrate various States of ASP.NET pages

1) View State

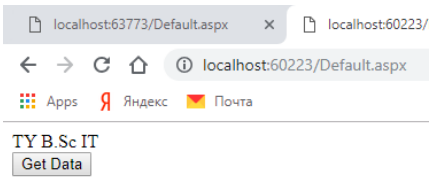
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
   <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
     <br />
     </div>
     <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Get Data" />
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if(!IsPostBack)
      {
         string str = "TY B.Sc IT";
         if(ViewState["nam"]==null)
         {
            ViewState["nam"] = str;
         }
      }
   }
  protected void Button1_Click(object sender, EventArgs e)
   {
      Label1.Text = ViewState["nam"].ToString();
   }
}
```

OUTPUT:



2. Query String

Default3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default3.aspx.cs" Inherits="Default3" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
   Querystring parameter<br />
     <br />
     User id<br />
     <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
     <br />
     <br />
     UserName<br />
     <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
  </div>
  </form>
</body>
</html>
```

Default3.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Default3 : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    if(!IsPostBack)
    {
      Label1.Text = Request.QueryString["UserId"];
      Label2.Text = Request.QueryString["Username"];
    }
  }
```

```
}
```

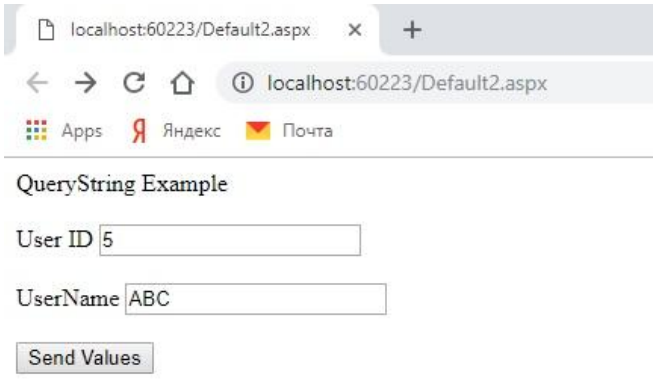
Default2.aspx

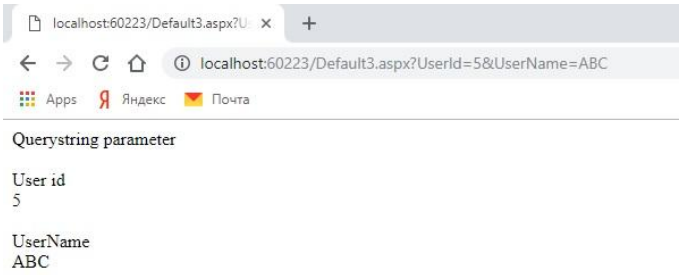
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default3.aspx.cs" Inherits="Default3" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
     Querystring parameter<br />
      <br />
      User id<br />
      <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      UserName<br />
      <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
      </div>
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Default2 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   { }
  protected void Button1_Click(object sender, EventArgs e)
   {
      Response.Redirect("Default3.aspx?UserId=" + TextBox1.Text + "&UserName=" + TextBox2.Text);
   }
}
```

OUTPUT:





3)Cookie

Default4.aspx

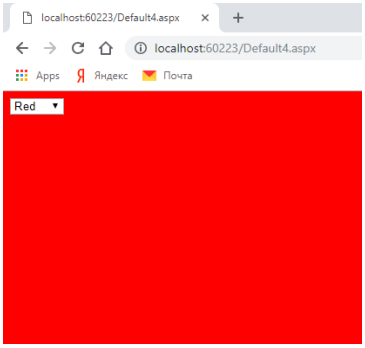
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default4.aspx.cs" Inherits="Default4" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body id="BodyTag" runat="server">
   <form id="form1" runat="server">
   <div>
      <asp:DropDownList ID="ColorSelector" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ColorSelector_IndexChanged">
         <asp:ListItem>White</asp:ListItem>
         <asp:ListItem>Red</asp:ListItem>
         <asp:ListItem>Green</asp:ListItem>
         <asp:ListItem>Blue</asp:ListItem>
      </asp:DropDownList>
    </div>
   </form>
</body>
</html>
```

Default4.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;
public partial class Default4 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if(Request.Cookies["BackgroundColor"]!=null)
      ColorSelector.SelectedValue = Request.Cookies["BackgroundColor"].Value;
      BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
  }
 protected void ColorSelector_IndexChanged(object sender, EventArgs e)
   {
```

```
        BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
        HttpCookie cookie = new HttpCookie("BackgroundColor");
        cookie.Value = ColorSelector.SelectedValue;
        cookie.Expires = DateTime.Now.AddMilliseconds(20);
        Response.SetCookie(cookie);
    }
}
```

OUTPUT:



4) Session and Application State

Global.asax

```
<%@ Application Language="C#" %>
<script runat="server">
  void Application_Start(object sender, EventArgs e)
    {
        Application["OnlineUsers"] = 0;
        // Code that runs on application startup
 }
    void Application_End(object sender, EventArgs e)
    {
        //  Code that runs on application shutdown
    }
  void Application_Error(object sender, EventArgs e)
    {
        // Code that runs when an unhandled error occurs
 }
  void Session_Start(object sender, EventArgs e)    {
        Application.Lock();
        Application["OnlineUsers"] = (int)Application["OnlineUsers"] + 1;
        Application.UnLock();
        // Code that runs when a new session is started
 }
    void Session_End(object sender, EventArgs e)
    {
        // Code that runs when a session ends.
        // Note: The Session_End event is raised only when the sessionstate mode
        // is set to InProc in the Web.config file. If session mode is set to StateServer
        // or SQLServer, the event is not raised.
        Application.Lock();
        Application["OnlineUsers"] = (int)Application["OnlineUsers"] - 1;
        Application.UnLock();
 }
</script>
```

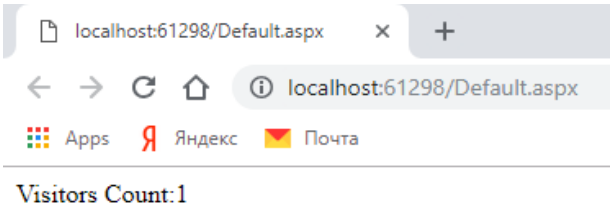
Web.config

```
<?xml version="1.0"?>
<!--
 For more information on how to configure your ASP.NET application, please visit
 http://go.microsoft.com/fwlink/?LinkId=169433
 -->
<configuration>
 <system.web>
    <sessionState mode="InProc" cookieless="false" timeout="1"/>
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
  </system.web>
</configuration>
```

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    Visitors Count:<%=Application["OnlineUsers"].ToString() %>
  </div>
  </form>
</body>
</html>
```

OUTPUT:

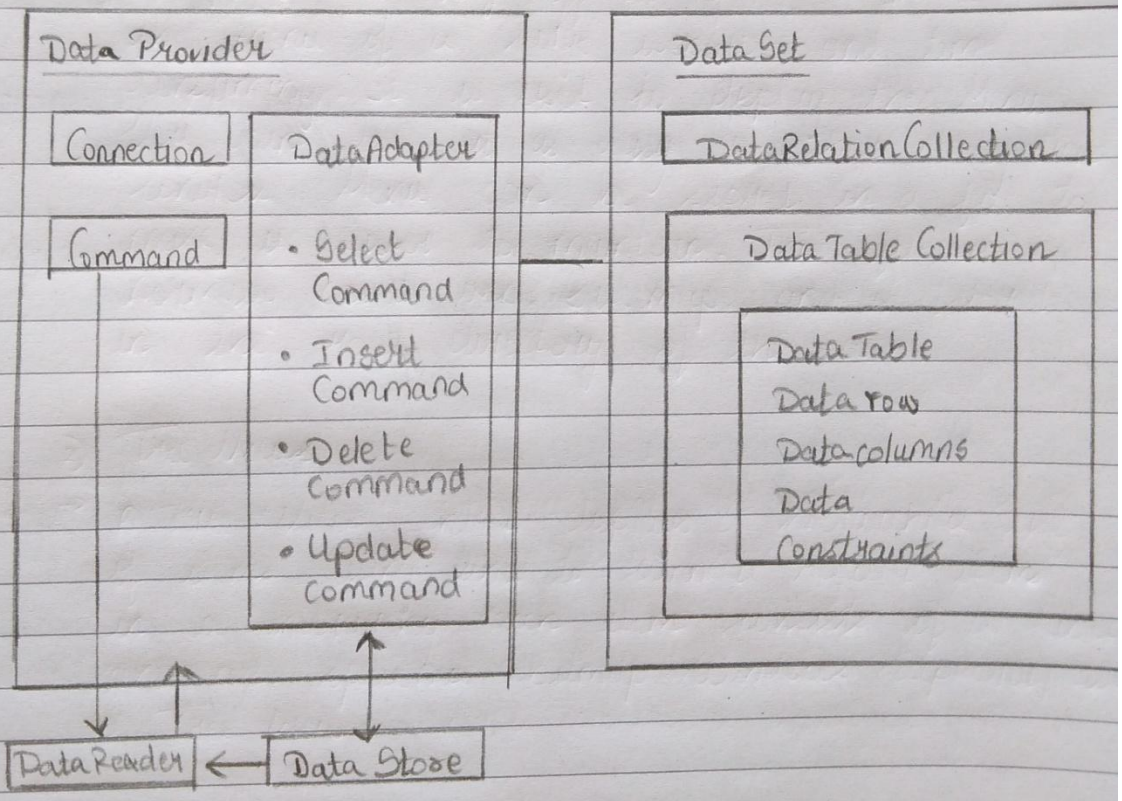


Practical 6

Practical no :- 6

Aim :- Working with database

Theory :- ADO.NET provides a bridge between the front end controls and the back end database. The ADO.NET objects encapsulate all the data access operations and the controls interact with these objects to display data, thus hiding the details of movement of data

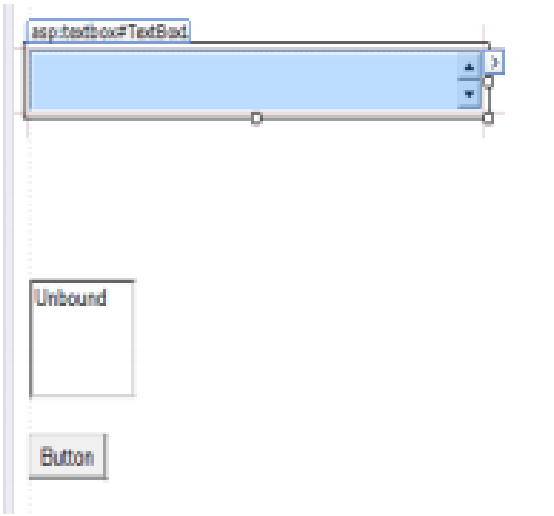


Conclusion : Hence, we have successfully performed the above practical.

Practical 6(a)

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

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.



Default.cs

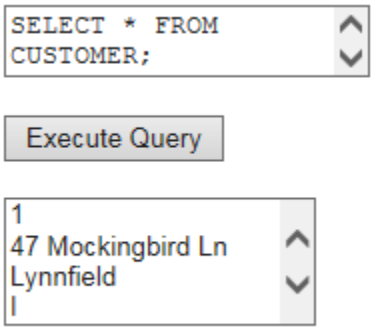
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   { }
  protected void Button1_Click(object sender, EventArgs e)
   {
      string connStr =ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
      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();    }
}
```

WEB.CONFIG

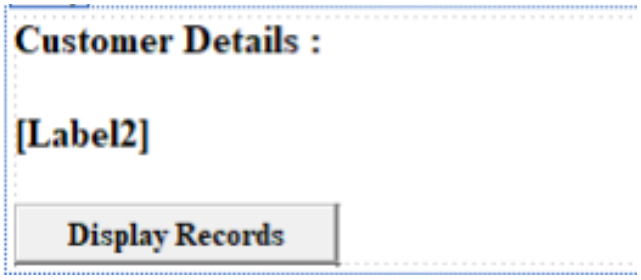
```
<?xml version="1.0"?>
<configuration>
  <system.web>
     <compilation debug="true" targetFramework="4.5.2" />
     <httpRuntime targetFramework="4.5.2" />
   </system.web>
  <connectionStrings>
   <add name="connStr"   connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\Admin\Documents\Visual Studio
2015\WebSites\WebSite1\App_Data\Database.mdf';Integrated Security=True" />
  </connectionStrings>
</configuration>
```

OUTPUT:



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

Create a web page with following design:



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

```
<?xml version="1.0"?>
<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\Admin\Documents\Visual Studio
2015\WebSites\WebSite1\App_Data\Database.mdf';Integrated Security=True" />
  </connectionStrings>
</configuration>
```

Add the following code on Button click event in 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 WebSites_WebSite1_Default2 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
```

```
  {

  }
  protected void Button1_Click(object sender, EventArgs e)
  {
     string connStr =ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
     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:

CUSTOMER DETAILS:

LabelLynnfield MA
Woburn MA
Quincy MA
Waltham MA
Salem NH
Waltham MA
Wilmington MA
Salem NH
Newton MA
Salem NH
Wilmington MA
Salem NH
Quincy MA

Button

Practical 6 (c)

Aim: 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.
8. After selection of Database file. We can also Test the connection.
9. Then Click on OK button.
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.
14. After successful steps form the Datalist controls option wizard our web page design and output will look like following.

OUTPUT:

TXN_ID: 1
AMOUNT: 100
FUNDS_AVAIL_DATE: 15-01-2000 00:00:00
TXN_DATE: 15-01-2000 00:00:00
TXN_TYPE_CD: CDT
ACCOUNT_ID: 1
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 2
AMOUNT: 100
FUNDS_AVAIL_DATE: 15-01-2000 00:00:00
TXN_DATE: 15-01-2000 00:00:00
TXN_TYPE_CD: CDT
ACCOUNT_ID: 2
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 3
AMOUNT: 100
FUNDS_AVAIL_DATE: 30-06-2004 00:00:00
TXN_DATE: 30-06-2004 00:00:00
TXN_TYPE_CD: CDT
ACCOUNT_ID: 3
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

TXN_ID: 4
AMOUNT: 100
FUNDS_AVAIL_DATE: 12-03-2001 00:00:00
TXN_DATE: 12-03-2001 00:00:00
TXN_TYPE_CD: CDT
ACCOUNT_ID: 4
EXECUTION_BRANCH_ID:
TELLER_EMP_ID:

Practical 7

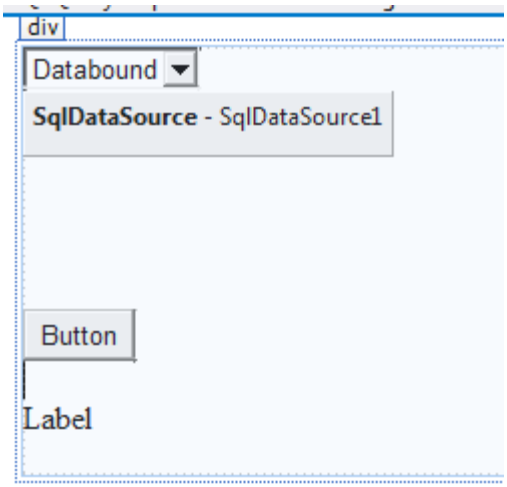
Practical 7 (a)

Aim: Create a web application to display Databinding using Dropdownlist control.

Create a web page with DropDownList control, one Button and one Label control.

Attach datasource to drop down list or you can add code to bind data in cs file

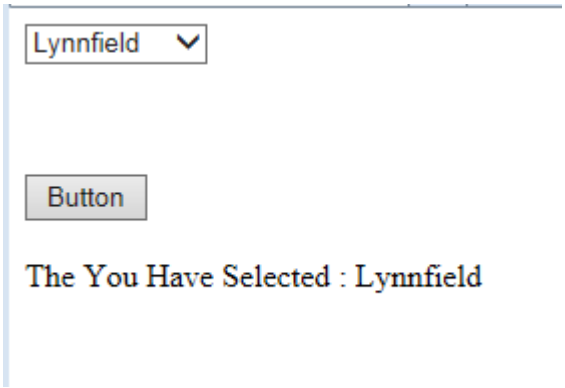
Write don't on button control to display data



Default.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (IsPostBack == false)
      {
         string DatabaseConnectionString =
ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].ConnectionString;
         SqlConnection con = new SqlConnection(DatabaseConnectionString);
         SqlCommand cmd = new SqlCommand("Select Distinct City from Customer", con);
```

```
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            DropDownList1.DataSource = reader;
            DropDownList1.DataSourceID = null;
            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;
    }
}
```



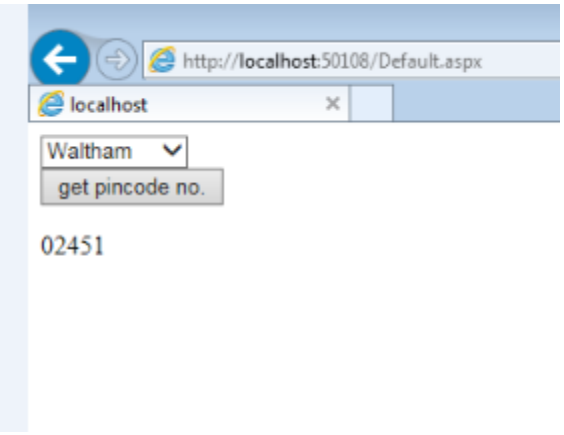
Practical 7 (b)

Aim: Create a web application for to display the phone no of an author using database.
Create a web page with DropDownList, Button and with Label control as shown below.

Default.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (IsPostBack == false)
      {
         string DatabaseConnectionString =
ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].ConnectionString;
         SqlConnection con = new SqlConnection(DatabaseConnectionString);
         SqlCommand cmd = new SqlCommand("Select Distinct POSTAL_CODE from Customer", con);
         con.Open();
         SqlDataReader reader = cmd.ExecuteReader();
         DropDownList1.DataSource = reader;
         DropDownList1.DataSourceID = null;
         DropDownList1.DataTextField = "City";
         DropDownList 1.DataValueField = "POSTAL_CODE";
         DropDownList1.DataBind();
         reader.Close();
         con.Close();
      }
   }
 protected void Button1_Click(object sender, EventArgs e)
   {
      Label1.Text = "The You Have Selected : " + DropDownList1.SelectedValue;
   }
}
```

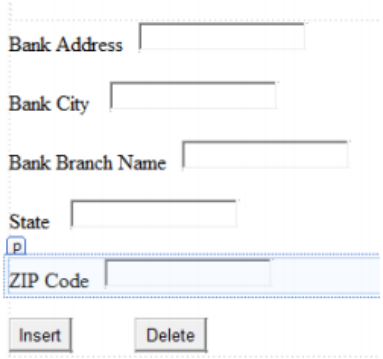
OUTPUT:


http://localhost:50108/Default.aspx
localhost
Waltham
get pincode no.
02451

Practical 7 (c)

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

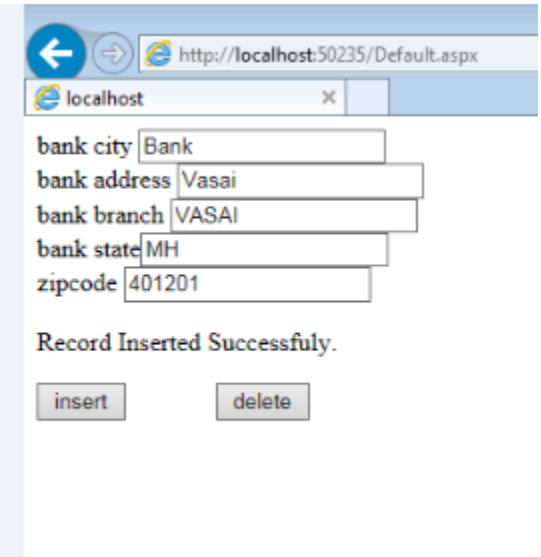
Create a web page with TextBox, and Two Button and one Label control as shown below. And follow the database related steps

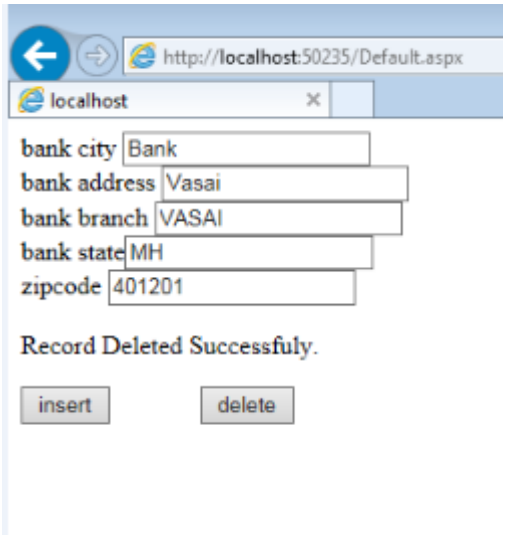



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
   protected void Button1_Click(object sender, EventArgs e)
   {
      string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
      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"].ConnectionString;
        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();
    }
}
```

OUTPUT:




http://localhost:50235/Default.aspx
localhost
bank city Bank
bank address Vasai
bank branch VASAI
bank state MH
zipcode 401201
Record Deleted Successfuly.
insert
delete

Practical 8

Practical 8 (a)

Aim: Create a web application to demonstrate various uses and properties of SqlDataSource.

Create a webpage with Dropdown list,details view and two sql datasource. And follow the database related steps

Code behind:

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
  {
        SqlDataSource2.SelectCommand="Select * from Product where
        NAME='"+DropDownList1.SelectedValue +"'";
  }
```

OUTPUT:

home mortgage

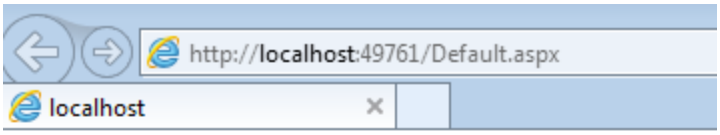
PRODUCT_CD	MRT
DATE_OFFERED	01-01-2000 00:00:00
DATE_RETIRED	
PRODUCT_TYPE_CD	LOAN
NAME	home mortgage

Practical 8 (b)

Aim: Create a web application to demonstrate data binding using DetailsView and FormView Control.

1) Add a new webpage with Detailsview and form view.
2) After adding, configure data dource property to these controls
3) After finishing data source configuration, we can use auto format option and edit template option to configure display settings

OUTPUT:



EMP_ID	1
END_DATE	
FIRST_NAME	Michael
LAST_NAME	Smith
START_DATE	22-06-2001 00:00:00
TITLE	President
ASSIGNED_BRANCH_ID	1
DEPT_ID	3
SUPERIOR_EMP_ID	

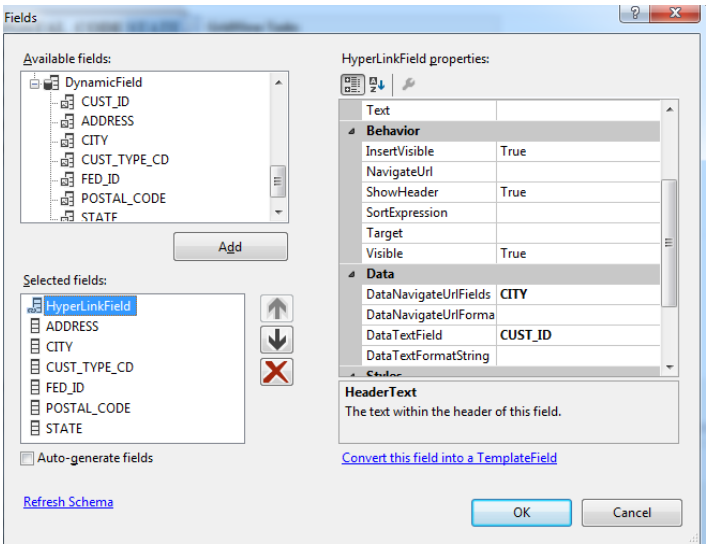
OFFICER_ID: 1
END_DATE:
FIRST_NAME: John
LAST_NAME: Chilton
START_DATE: 01-05-1995 00:00:00
TITLE: President
CUST_ID: 10

Practical 9

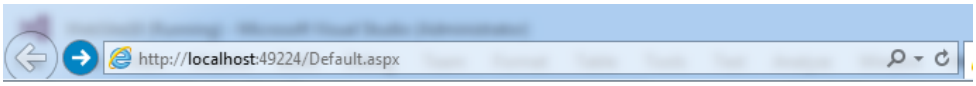
Practical 9 (a)

Aim: Create a web application to demonstrate use of GridView control template and GridView hyperlink.

1) Create a web page with grid view control
2) Connect grid view to datasource control
3) Select Edit Columns option from grid view tasks
4) In the next window from Available Fields options select the HyperLinkField and Add it to Selected Field
5) Navigate the HyperLinkField to the top of the selected fileds using arrow option.
6) Remove repeated or unwanted columns from from selected fields.
7) Select the HyperLink filed from selected fields and change DataTextFiled property to Id. Press ok and run the application.



OUTPUT:

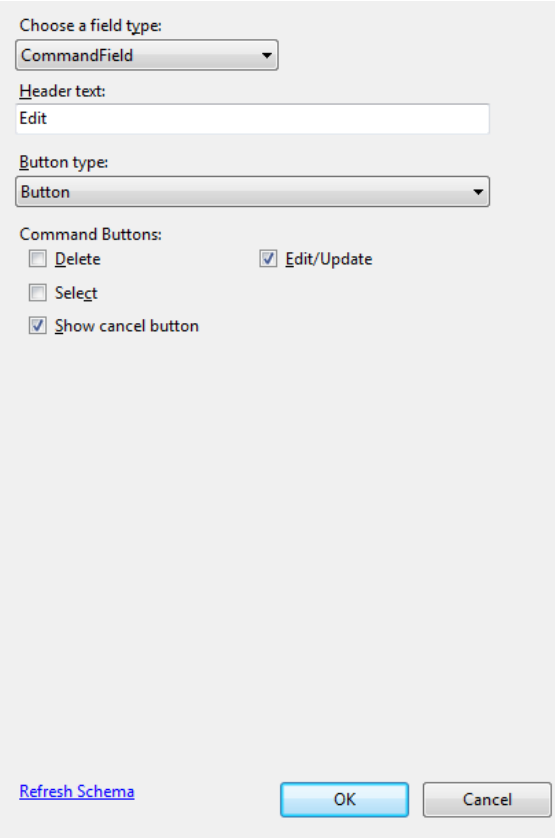


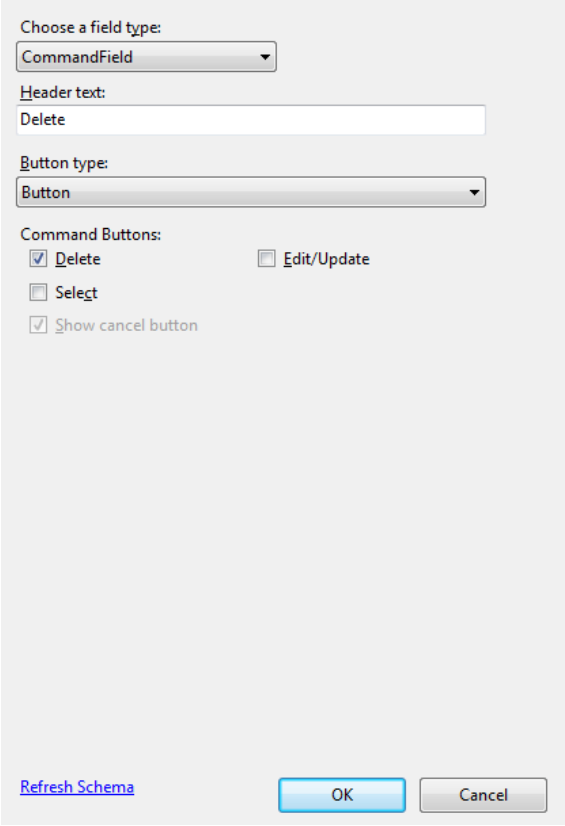
	ADDRESS	CITY	CUST_TYPE_CD	FED_ID	POSTAL_CODE	STATE
1	47 Mockingbird Ln	Lynnfield	I	111-11-1111	01940	MA
2	372 Clearwater Blvd	Woburn	I	222-22-2222	01801	MA
3	18 Jessup Rd	Quincy	I	333-33-3333	02169	MA
4	12 Buchanan Ln	Waltham	I	444-44-4444	02451	MA
5	2341 Main St	Salem	I	555-55-5555	03079	NH
6	12 Blaylock Ln	Waltham	I	666-66-6666	02451	MA
7	29 Admiral Ln	Wilmington	I	777-77-7777	01887	MA
8	472 Freedom Rd	Salem	I	888-88-8888	03079	NH
9	29 Maple St	Newton	I	999-99-9999	02458	MA
10	7 Industrial Way	Salem	B	04-1111111	03079	NH
11	287A Corporate Ave	Wilmington	B	04-2222222	01887	MA
12	789 Main St	Salem	B	04-3333333	03079	NH
13	4772 Presidential Way	Quincy	B	04-4444444	02169	MA

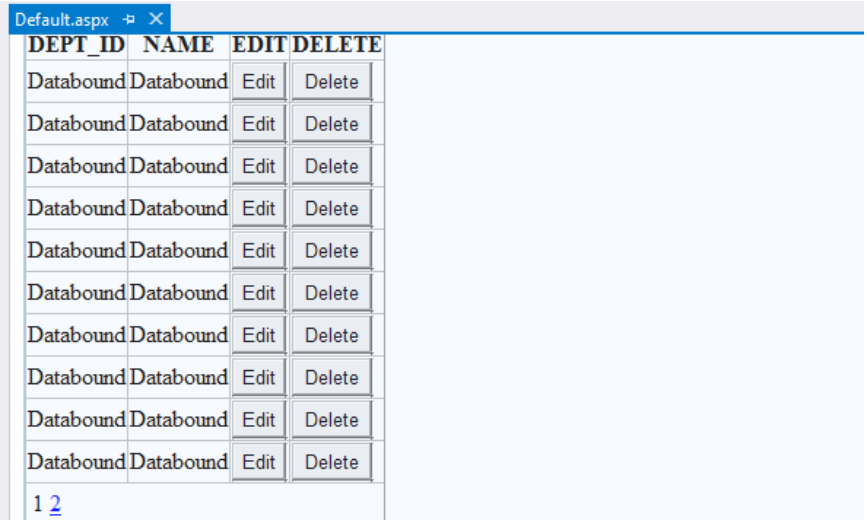
Practical 9 (b)

Aim: Create a web application to demonstrate use of GridView button column and GridView events.

1) Create a web page with grid view control
2) Connect grid view to datasource control
3) Select Add new column option from grid view tasks
4) In add filed choose filed type command field, header text as edit, Button type as button, alos select Edit/Update checkbox with show cancel button checkbox
5) Again Select Add new column option from grid view tasks
6) In add filed choose filed type command field, header text as delete, Button type as button, alos select delete checkbox







OUTPUT:

DEPT_ID	NAME	EDIT	DELETE
1	ACC	Update Cancel	
2	Loans	Edit	Delete
3	Administration	Edit	Delete
4	IT	Edit	Delete

Practical 10

Practical 10(a)

Aim: Create a web application to demonstrate reading and writing operation with XML
Create demo.xml file

Default.aspx

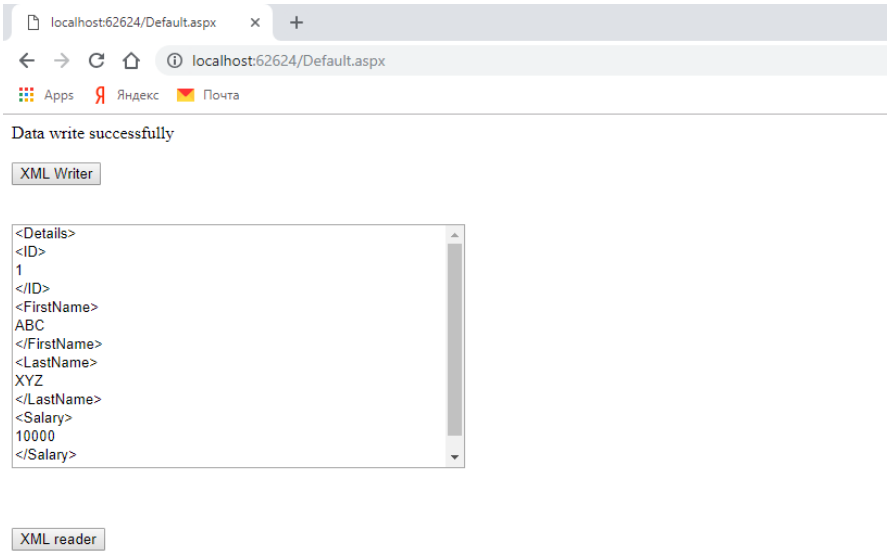
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
   <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
     <br />  <br />
     <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="XML Writer" />
     <br />   <br />   <br />
     <asp:ListBox ID="ListBox1" runat="server" Height="225px" Width="417px"></asp:ListBox>
     <br /> <br /> <br /> <br />
     <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="XML reader" />
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml;
public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  { }
  protected void Button1_Click(object sender, EventArgs e)
  {
   XmlTextWriter writer = new XmlTextWriter("C:\\Users\\dell\\Documents\\Visual Studio
2015\\WebSites\\xmlwrite\\demo.xml", null);
     writer.WriteStartDocument();
     writer.WriteStartElement("Details", "");
     writer.WriteElementString("ID", "1");
     writer.WriteElementString("FirstName", "ABC");
     writer.WriteElementString("LastName", "XYZ");
     writer.WriteElementString("Salary", "10000");
```

```
        writer.WriteEndElement();
        writer.WriteEndDocument();
        writer.Close();
        Label1.Text = "Data write successfully";

    }
protected void Button2_Click(object sender, EventArgs e)
    {
        String xmlNode = "C:\\Users\\dell\\Documents\\Visual Studio 2015\\WebSites\\xmlwrite\\demo.xml";
        XmlReader xReader = XmlReader.Create(xmlNode);
        while(xReader.Read())
        {
            switch(xReader.NodeType)
            {
                case XmlNodeType.Element:
                    ListBox1.Items.Add("<" + xReader.Name + ">");
                    break;
                case XmlNodeType.Text:
                    ListBox1.Items.Add(xReader.Value);
                    break;
                case XmlNodeType.EndElement:
                    ListBox1.Items.Add("</" + xReader.Name + ">");
                    break;
            }
        }
    }
}
```

OUTPUT



Practical 10(b)

Aim: Create a web application to demonstrate form security and windows security with proper authentication properties

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <p>
    <br />
  </p>
  <form id="form1" runat="server">
    <p>
      Useer name
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </p>
    <p>
      User Password 
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    </p>
    <p>
       </p>
    <p>
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Login" />
    </p>
  <div>
    </div>
    <asp:CheckBox ID="CheckBox1" runat="server" Text="Check if it is not a public computer" />
    <br />
    <br />
    <br />
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;
public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {  }
  protected bool authenticate (String uname, String pass)
  {
    if(uname =="ABC")
    {
```

```
            if (pass == "abc123")
            {
                return true;
            }
        }
        if (uname == "XYZ")
        {
            if (pass == "xyz123")
            {
                return true;
            }
        }
    if (uname == "PQR")
        {
            if (pass == "pqr123")
            {
                return true;
            }
        }
        return false;
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        if(authenticate(TextBox1.Text,TextBox2.Text))
        {
            FormsAuthentication.RedirectFromLoginPage(TextBox1.Text, CheckBox1.Checked);
            Session["Username"] = TextBox1.Text;
            Response.Redirect("Default2.aspx");
        }
        else
        {
            Response.Write("Inavalid username and password");
        }
    }
}
```

Default2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<!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:Label ID="Label1" runat="server" Text="Label"></asp:Label>
 Welcome to .NET world</form>
</body>
</html>
```

Default.aspx.cs

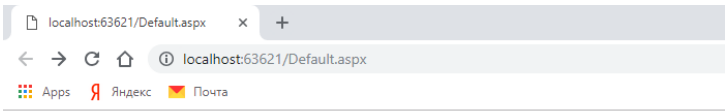
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if(Session["Username"]!=null)
      {
         Label1.Text = Session["Username"].ToString();
      }
   }
}
```

Web.config

```
<?xml version="1.0"?>
<!--
 For more information on how to configure your ASP.NET application, please visit
 http://go.microsoft.com/fwlink/?LinkId=169433
 -->
<configuration>
  <system.web>
    <authentication mode="Forms">
     <forms loginUrl="Default.aspx"/>
       </authentication>
    <authorization>
     <deny users="?"/>
    </authorization>
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
  </system.web>
</configuration>
```

OUTPUT



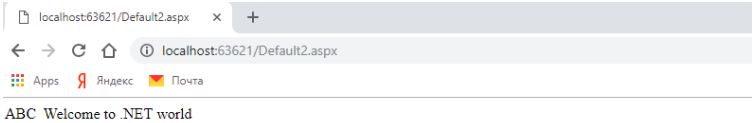
Useer name ABC

User Password abc123

Login

Check if it is not a public computer

Label



ABC Welcome to .NET world

Practical 10(c)
Aim: Create a web application to demonstrate various Ajax Controls

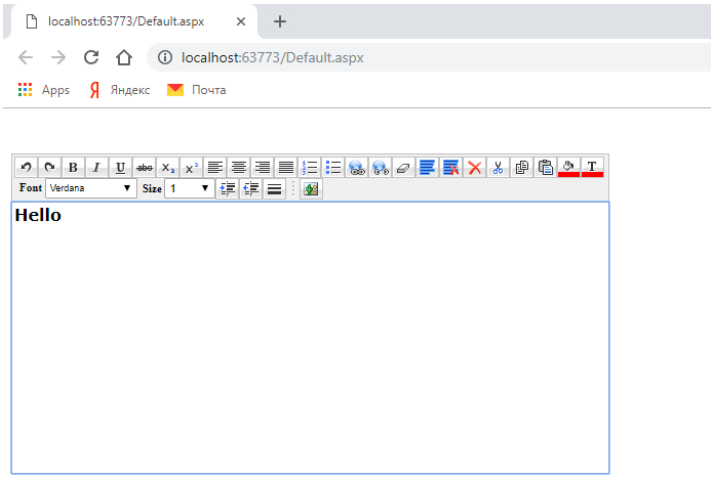
Program to demonstrate HTML Extender

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="ajaxToolkit" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
      </asp:ScriptManager>
      <br />
      <br />
      <asp:TextBox ID="TextBox1" runat="server" Columns="80" Rows="20" TextMode="MultiLine"></asp:TextBox>
      <ajaxToolkit:HtmlEditorExtender ID="TextBox1_HtmlEditorExtender" runat="server" TargetControlID="TextBox1"
EnableSanitization="false">
  <Toolbar>
            <ajaxToolkit:Undo />
           <ajaxToolkit:Redo />
           <ajaxToolkit:Bold />
           <ajaxToolkit:Italic />
           <ajaxToolkit:Underline />
           <ajaxToolkit:StrikeThrough />
           <ajaxToolkit:Subscript />
           <ajaxToolkit:Superscript />
           <ajaxToolkit:JustifyLeft />
           <ajaxToolkit:JustifyCenter />
           <ajaxToolkit:JustifyRight />
           <ajaxToolkit:JustifyFull />
           <ajaxToolkit:InsertOrderedList />
           <ajaxToolkit:InsertUnorderedList />
           <ajaxToolkit:CreateLink />
           <ajaxToolkit:UnLink />
           <ajaxToolkit:RemoveFormat />
           <ajaxToolkit:SelectAll />
           <ajaxToolkit:UnSelect />
           <ajaxToolkit:Delete />
           <ajaxToolkit:Cut />
           <ajaxToolkit:Copy />
           <ajaxToolkit:Paste />
           <ajaxToolkit:BackgroundColorSelector />
           <ajaxToolkit:ForeColorSelector />
           <ajaxToolkit:FontNameSelector />
           <ajaxToolkit:FontSizeSelector />
           <ajaxToolkit:Indent />
           <ajaxToolkit:Outdent />
           <ajaxToolkit:InsertHorizontalRule />
```

```
            <ajaxToolkit:HorizontalSeparator />
            <ajaxToolkit:InsertImage />
        </Toolbar>
    </ajaxToolkit:HtmlEditorExtender>
     </div>
    </form>
</body>
</html>
```

OUTPUT:



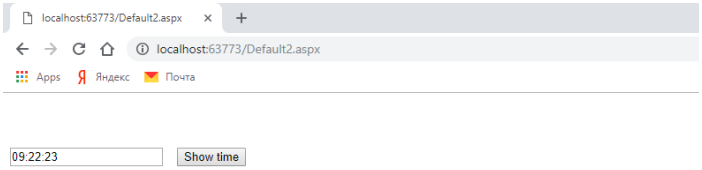
Program to demonstrate Ajax control

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="ajaxToolkit" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
      </asp:ScriptManager>
      <br />  <br /> <br />
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
  
      <asp:Button ID="Button1" runat="server" Text="Show time" OnClick="Button1_Click" />
    <br />   <br />

   </div>
      <asp:UpdateProgress ID="UpdateProgress1" runat="server">
         <ProgressTemplate>
            Please wait for sometime
         </ProgressTemplate>
      </asp:UpdateProgress>
 <asp:UpdatePanel ID="UpdatePanel1" runat="server">   </asp:UpdatePanel>
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Default2 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      System.Threading.Thread.Sleep(2000);
   }
 protected void Button1_Click(object sender, EventArgs e)
   {
      TextBox1.Text = DateTime.Now.ToLongTimeString();
   }
}
```


localhost:63773/Default2.aspx
localhost:63773/Default2.aspx
Apps
Яндекс
Почта
09:22:23
Show time

Practical 11

Practical 11
Aim: Programs to create and use DLL
In TSClassLib

Class1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TSClassLib
{
   public class Class1
   {
      public string UpperConvert(string text)
      {
         return text.ToUpper();
      }
      public string LowerConvert(string text)
      {
         return text.ToLower();
      }
   }
}
```

In Website

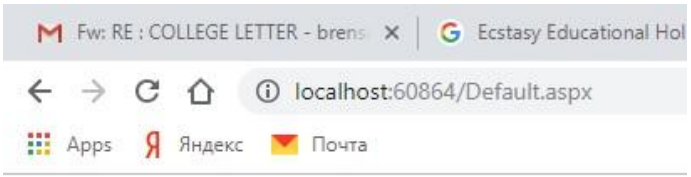
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <p>
      <br />
   </p>
   <form id="form1" runat="server">
      <p>
         <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Upper" />
      </p>
   <div>
    </div>
      <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Lower" />
      <br />
      <br />
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
```

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

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

   }
 protected void Button1_Click(object sender, EventArgs e)
   {
      Class1 t =new Class1();
      TextBox1.Text = t.UpperConvert(TextBox1.Text);
}
 protected void Button2_Click(object sender, EventArgs e)
   {
      Class1 t = new Class1();
      TextBox1.Text = t.LowerConvert(TextBox1.Text);
   }
}
```



Upper

Lower

arnold