MAHATMA GANDHI MISSION'S
COLLEGE OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

(Affiliated to University Of Mumbai)
Kamothe, Navi-Mumbai – 410209
Batch: -2016-2019

Certificate

This is to certify that, Mr./Ms.______________________________________ of ________________ Class, Examination Seat No. ____________ has performed the experiments in subject - ____________________ of S.Y.B.Sc (CS) sem-IV , as per guidelines of University of Mumbai.

_________________ ________________
Subject in Charge Principal

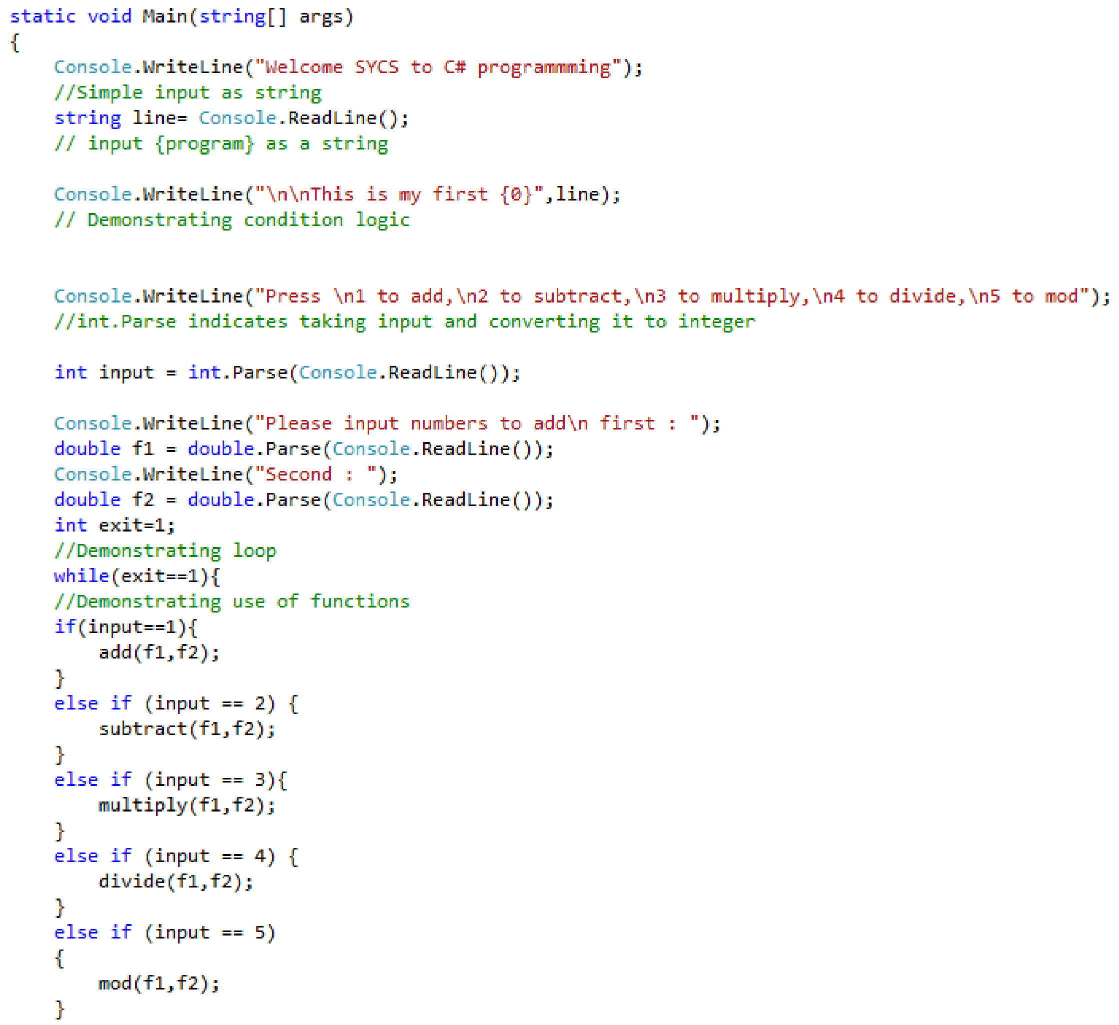
____

External Examiner

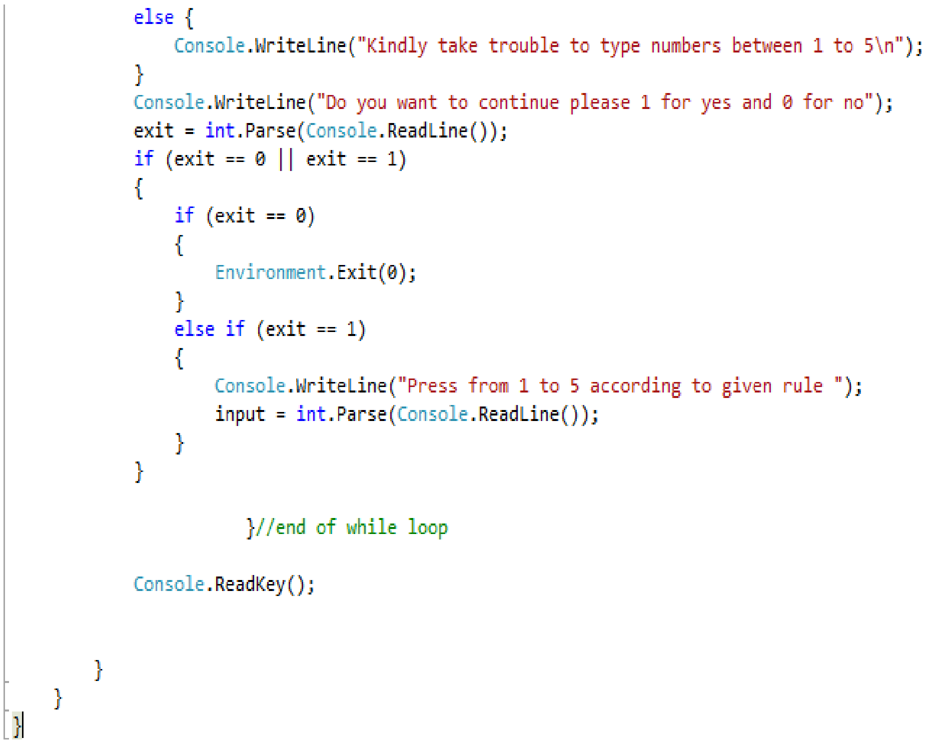
SR NO	EXPERIMENTS	DATE	SIGN
1	Demonstration of C# basics	15/1/2018	
2	Working on C# Object oriented features	22/1/2018	
3	Working with Master page in asp.net	29/1/2018	
4	Working with rich and navigational controls	6/2/2018	
5	Use of Validation control	20/2/2018	
6	Working with client side State management	27/2/2018	
7	Working with ADO.net data binding	28/3/2018	
8	Working with LINQ	28/3/2018	
9	Use of cache and working of xml classes	28/3/2018	
10	Working with AJAX in asp.net	28/3/2018	

Practical 1

```
namespace ConsoleApplication2
{

    class Program
    {
        static void add(double a,double b) {
            double c=a+b;
            Console.WriteLine("Addition of two numbers {0} and {1} is {2}",a,b,c);

        }
        static void subtract(double a, double b)
        {
            double c = a - b;
            Console.WriteLine("Substraction of two numbers {0} and {1} is {2}", a, b, c);

        }
        static void multiply(double a, double b)
        {
            double c = a * b;
            Console.WriteLine("Multiplication of two numbers {0} and {1} is {2}", a, b, c);

        }
        static void divide(double a, double b)
        {
            double c = a / b;
            Console.WriteLine("Division of two numbers {0} and {1} is {2}", a, b, c);

        }
        static void mod(double a, double b)
        {
            double c = a % b;
            Console.WriteLine("Mod of two numbers {0} and {1} is {2}", a, b, c);

        }
```



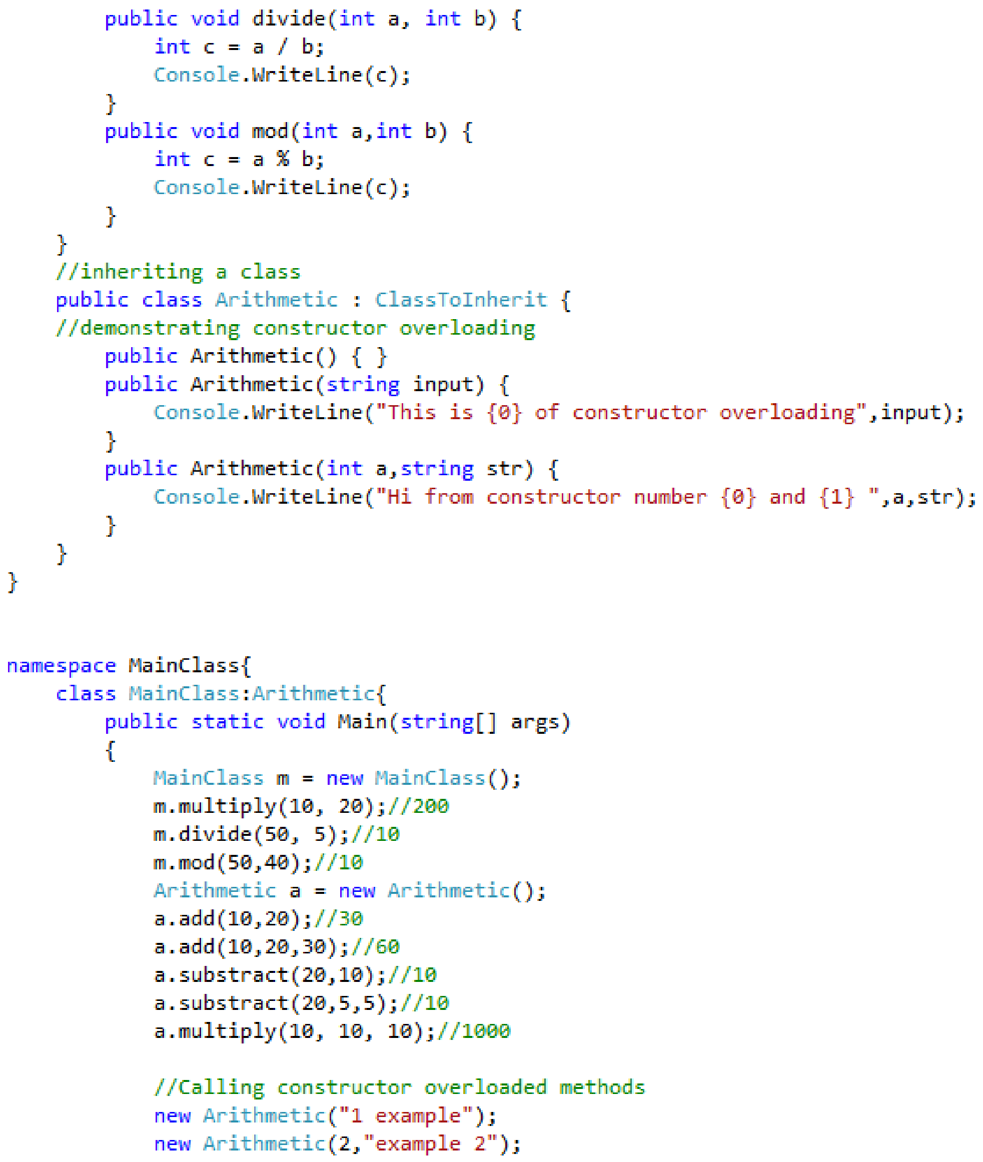
```
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome SYCS to C# programmming");
            //Simple input as string
            string line= Console.ReadLine();
            // input {program} as a string

            Console.WriteLine("\n\nThis is my first {0}",line);
            // Demonstrating condition logic


            Console.WriteLine("Press \n1 to add,\n2 to subtract,\n3 to multiply,\n4 to divide,\n5 to mod");
            //int.Parse indicates taking input and converting it to integer

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

            Console.WriteLine("Please input numbers to add\n first : ");
            double f1 = double.Parse(Console.ReadLine());
            Console.WriteLine("Second : ");
            double f2 = double.Parse(Console.ReadLine());
            int exit=1;
            //Demonstrating loop
            while(exit==1){
            //Demonstrating use of functions
            if(input==1){
                add(f1,f2);
            }
            else if (input == 2) {
                subtract(f1,f2);
            }
            else if (input == 3){
                multiply(f1,f2);
            }
            else if (input == 4) {
                divide(f1,f2);
            }
            else if (input == 5)
            {
                mod(f1,f2);
            }
```



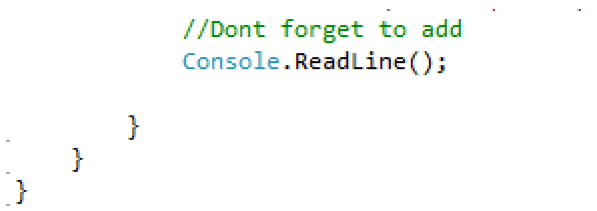
```
            else {
                Console.WriteLine("Kindly take trouble to type numbers between 1 to 5\n");
            }
            Console.WriteLine("Do you want to continue please 1 for yes and 0 for no");
            exit = int.Parse(Console.ReadLine());
            if (exit == 0 || exit == 1)
            {
                if (exit == 0)
                {
                    Environment.Exit(0);
                }
                else if (exit == 1)
                {
                    Console.WriteLine("Press from 1 to 5 according to given rule ");
                    input = int.Parse(Console.ReadLine());
                }
            }

                       }//end of while loop

            Console.ReadKey();


        }
    }
}
```

Practical 2

```
namespace ObjectBasedManipulation
{
    public interface InterfaceToInherit {
        void add(int a,int b);
        void substract(int a,int b);
    }
    public class ClassToInherit : InterfaceToInherit {
        //implementing interface methods
        public void add(int a, int b) {
            int c = a + b;
            Console.WriteLine(c);
        }
        public void substract(int a,int b) {
            int c = a - b;
            Console.WriteLine(c);
        }
        //now we overload these methods
        public void add(int a,int b,int c) {
            int d = a + b + c;
            Console.WriteLine(d);
        }
        public void substract(int a,int b,int c) {
            int d = a - b - c;
            Console.WriteLine(d);
        }

        //now specifying this class methods
        public void multiply(int a,int b) {
            int c = a * b;
            Console.WriteLine(c);
        }
        //lets overload the multiply method
        public void multiply(int a,int b,int c) {
            int d = a * b * c;
            Console.WriteLine(d);
        }
        public void divide(int a, int b) {
            int c = a / b;
            Console.WriteLine(c);
        }
```



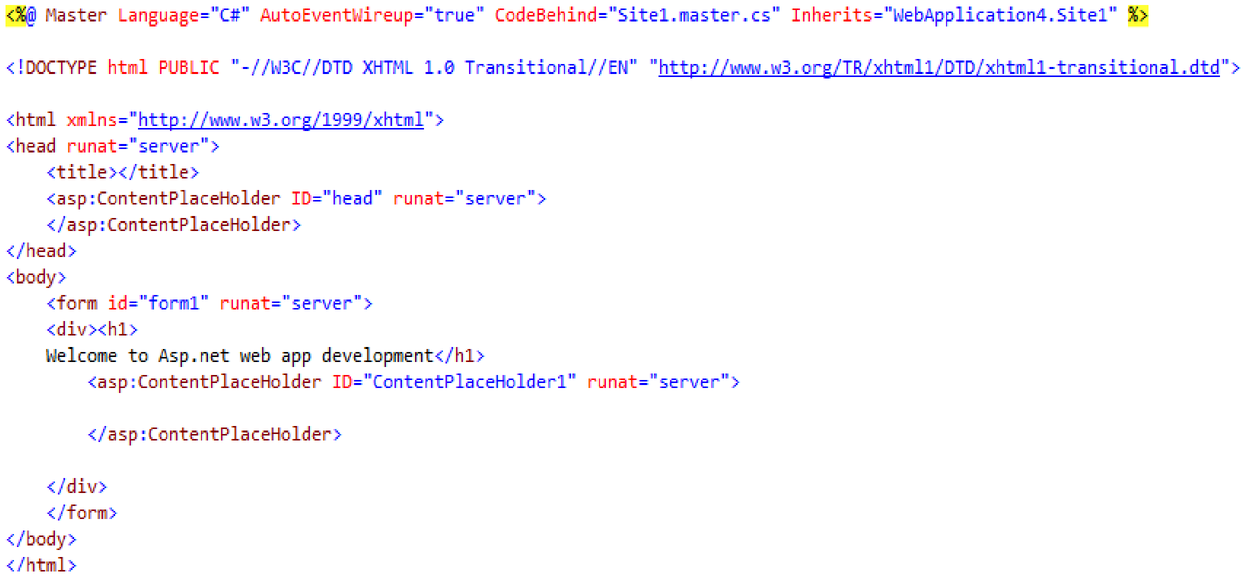
```
        public void divide(int a, int b) {
            int c = a / b;
            Console.WriteLine(c);
        }
        public void mod(int a,int b) {
            int c = a % b;
            Console.WriteLine(c);
        }
    }
    //inheriting a class
    public class Arithmetic : ClassToInherit {
    //demonstrating constructor overloading
        public Arithmetic() { }
        public Arithmetic(string input) {
            Console.WriteLine("This is {0} of constructor overloading",input);
        }
        public Arithmetic(int a,string str) {
            Console.WriteLine("Hi from constructor number {0} and {1} ",a,str);
        }
    }
}


namespace MainClass{
    class MainClass:Arithmetic{
        public static void Main(string[] args)
        {
            MainClass m = new MainClass();
            m.multiply(10, 20);//200
            m.divide(50, 5);//10
            m.mod(50,40);//10
            Arithmetic a = new Arithmetic();
            a.add(10,20);//30
            a.add(10,20,30);//60
            a.substract(20,10);//10
            a.substract(20,5,5);//10
            a.multiply(10, 10, 10);//1000

            //Calling constructor overloaded methods
            new Arithmetic("1 example");
            new Arithmetic(2,"example 2");
```



```
            //Dont forget to add
            Console.ReadLine();

        }
    }
}
```

Practical 3

Site1.Master

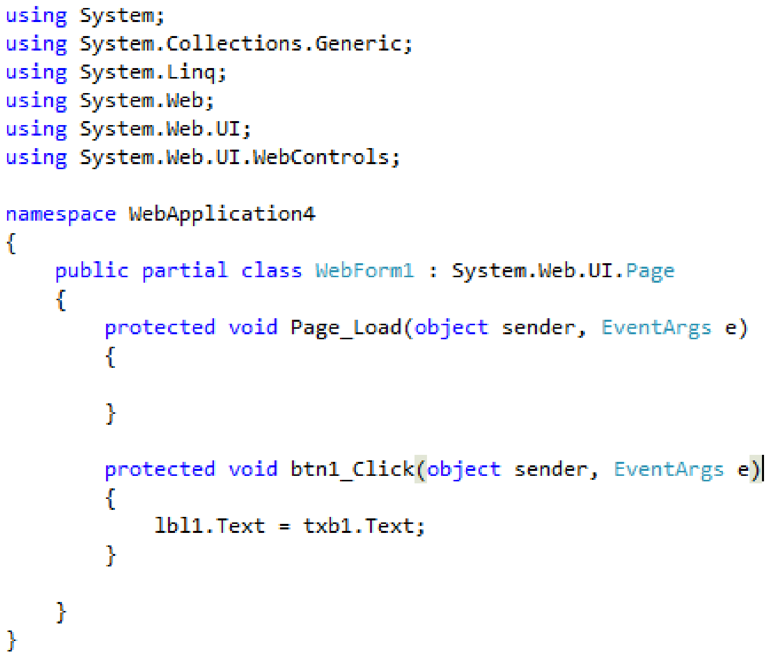


```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs" Inherits="WebApplication4.Site1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div><h1>
    Welcome to Asp.net web app development</h1>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

        </asp:ContentPlaceHolder>

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

WebForm1.aspx

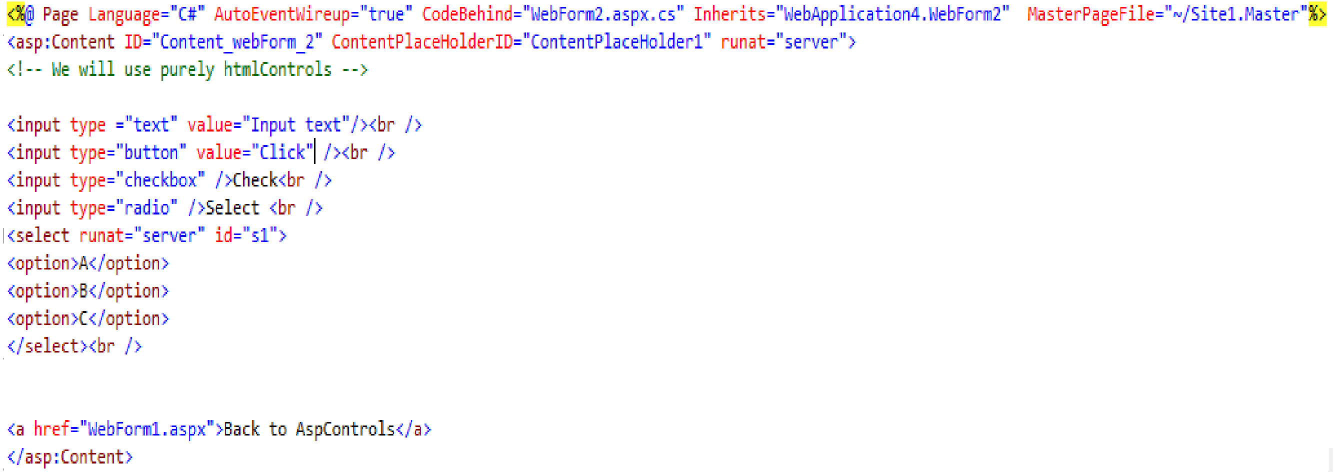
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication4.WebForm1" MasterPageFile="~/Site1.Master"%>
<asp:Content ID="Main_page" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <h3>Lets first demonstrate Asp Controls</h3>
<!-- Simple table demo -->
<asp:Table runat="server" BorderWidth="2" >
<asp:TableRow>
<asp:TableCell BorderWidth="2">AspControls</asp:TableCell>
<asp:TableCell BorderWidth="2">HtmlControls</asp:TableCell>
</asp:TableRow>
<asp:TableRow>
<asp:TableCell BorderWidth="2">It involves asp prefix<br/> and then control name</asp:TableCell>
<asp:TableCell BorderWidth="2">Use the regular way<br /> to control</asp:TableCell>
</asp:TableRow>
</asp:Table>
<!-- Try radio buttons
First group them using a list
-->
<asp:RadioButtonList runat="server">
<asp:ListItem>AspControls are dynamic you can handle them at server</asp:ListItem>
<asp:ListItem>AspControls need only runat which indicates it will be handled on server</asp:ListItem>
</asp:RadioButtonList>

<!-- Lets now use checkbox controls
   Again group them using list
-->
<asp:CheckBoxList runat="server">
<asp:ListItem>You can use htmlControls along side aspControls</asp:ListItem>
<asp:ListItem>But don't use them between two same child aspControls</asp:ListItem>
</asp:CheckBoxList>
<!--  A ListBox contains multiple items which can
    selected by the user
-->
<asp:ListBox runat="server">
<asp:ListItem>Remember why aspx is used for</asp:ListItem>
<asp:ListItem>Can you see aspx.cs</asp:ListItem>
<asp:ListItem>Don forget Master</asp:ListItem>
</asp:ListBox>
<br/>
<!--
Now we use textbox and a button control
-->
<asp:TextBox runat="server" Text="Input text" ID="txb1"/>
<asp:Button runat="server" Text="Its very easy" ID="btn1" onclick="btn1_Click"/>
<br />
<asp:Label runat="server" Text="Everything in parent control has a runat attribute" ID="lbl1"/>
<br />
<asp:HyperLink runat ="server"  NavigateUrl="~/WebForm2.aspx" Text="Next HTMLControls"/>
</asp:Content>
```

WebForm1.aspx.cs



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

namespace WebApplication4
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btn1_Click(object sender, EventArgs e)
        {
            lbl1.Text = txb1.Text;
        }

    }
}
```

WebFrom2.aspx



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="WebApplication4.WebForm2"  MasterPageFile="~/Site1.Master"%>
<asp:Content ID="Content_webForm_2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<!-- We will use purely htmlControls -->

<input type ="text" value="Input text"/><br />
<input type="button" value="Click" /><br />
<input type="checkbox" />Check<br />
<input type="radio" />Select <br />
<select runat="server" id="s1">
<option>A</option>
<option>B</option>
<option>C</option>
</select><br />


<a href="WebForm1.aspx">Back to AspControls</a>
</asp:Content>
```

Practical 4

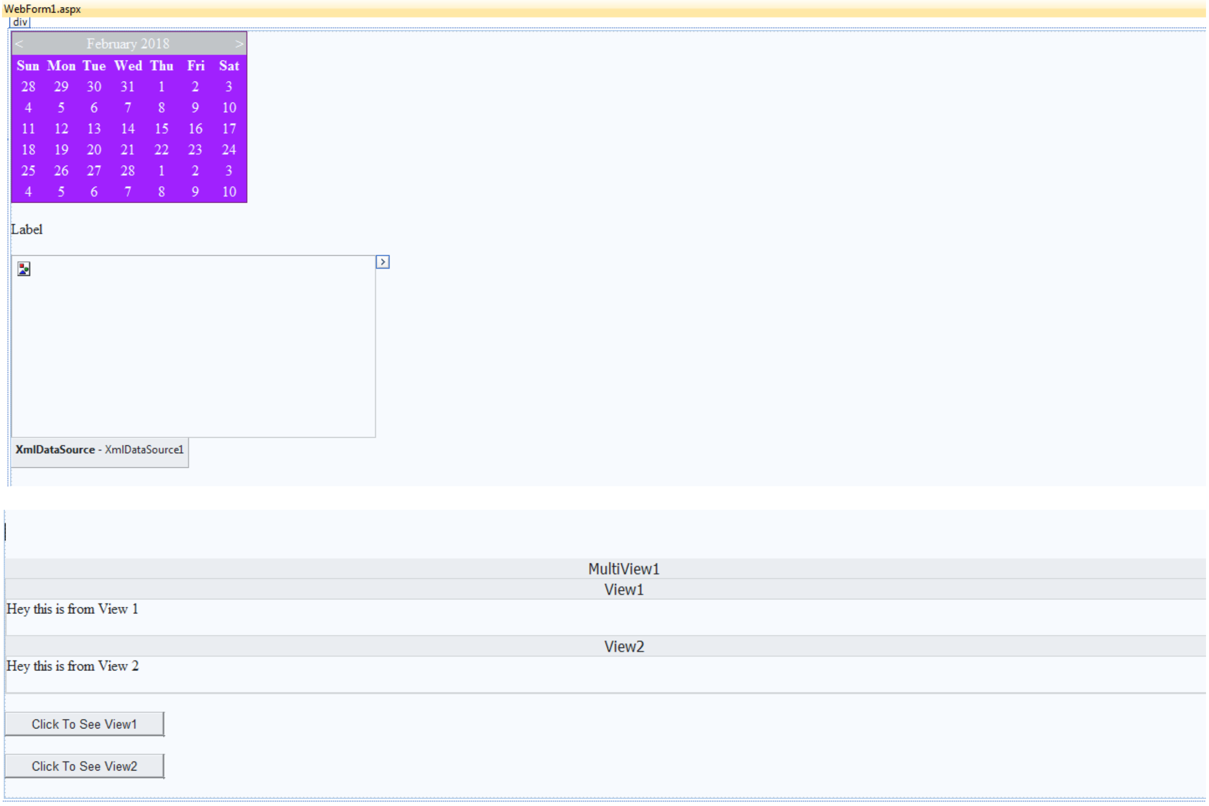
Demonstrating use of rich controls

Design (WebForm.aspx)

Take an CalenderControl, asp:Label, AdRotator, Multiview in that { two view controls },

two asp:button.

****In the adRotator you won't be able to see xmdDataSource as you have not added it yeta



<			February 2018			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

Hey this is from View 1

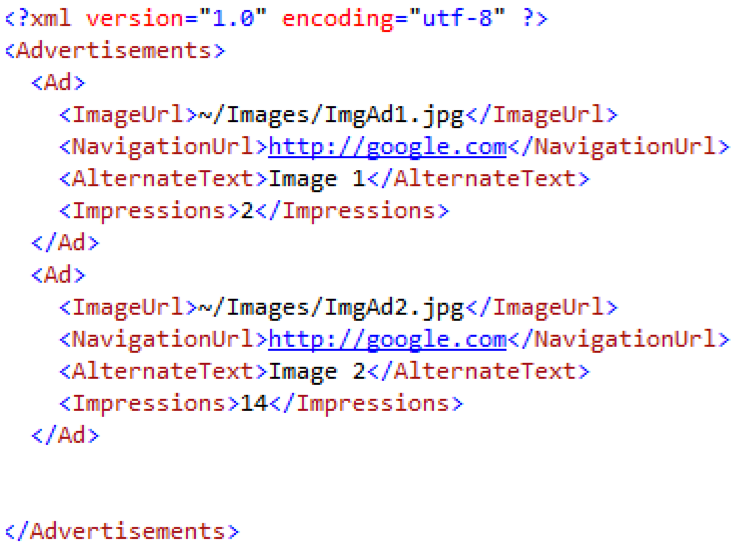
Hey this is from View 2

Use calendar control properties the way you to define its characteristics, also associate an event with it by double clicking on the control. This event should set the selected date to an label in our user interface.

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    Label1.Text ="Selected date is : "+Calendar1.SelectedDate.ToShortDateString();

}
```

Next when you add an ad rotator you need to provide some data to it,which we will do from an xml file.

Create an xml file and write the following code, make sure you have copied the images folder to your project.



```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
  <Ad>
    <ImageUrl>~/Images/ImgAd1.jpg</ImageUrl>
    <NavigationUrl>http://google.com</NavigationUrl>
    <AlternateText>Image 1</AlternateText>
    <Impressions>2</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>~/Images/ImgAd2.jpg</ImageUrl>
    <NavigationUrl>http://google.com</NavigationUrl>
    <AlternateText>Image 2</AlternateText>
    <Impressions>14</Impressions>
  </Ad>


</Advertisements>
```

Now attach this datasource file to the adrotator control.

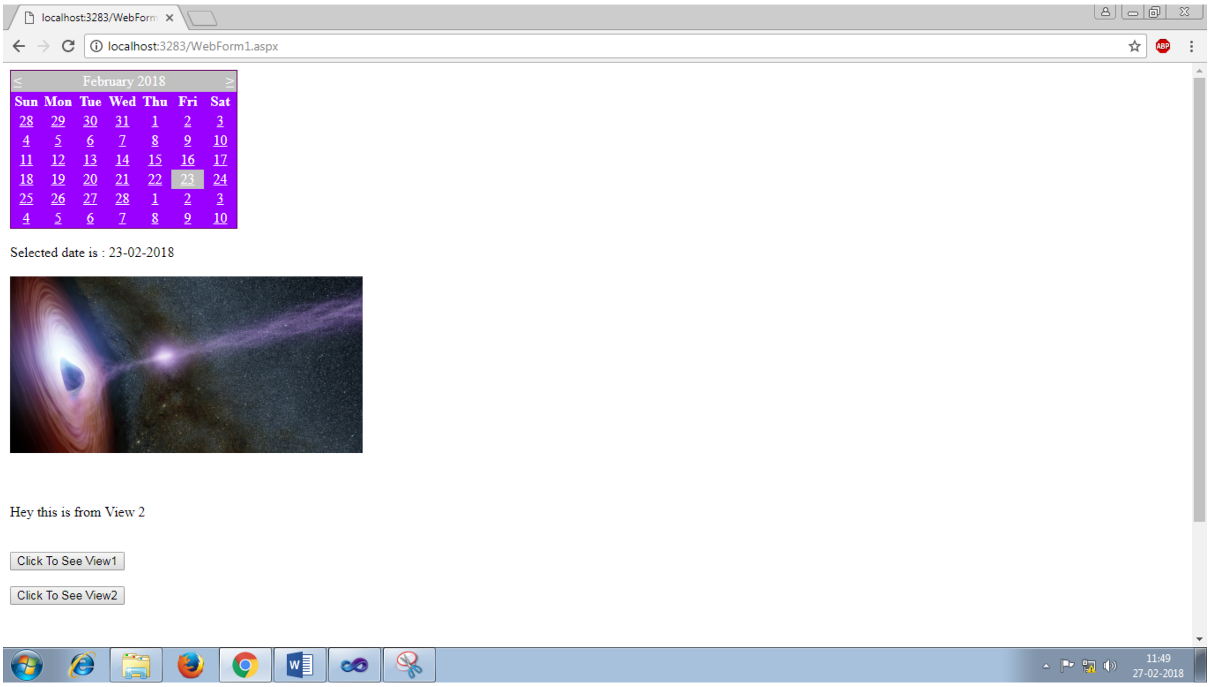
Now we will demonstrate multiview and view controls, insert view in mulitview control as it will contain the actual content which you will be able to see.

Now also associate events with the two buttons on double clicking on each one of them.

```
protected void Button1_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 0;
}

protected void Button2_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 1;
}
```

This code activates the view on clicking as our index starts from 0 ,the view 1 will be activate when ActiveViewIndex=0 and view 2 when ActiveViewIndex=1;

Output:



<			February 2018			>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

Selected date is : 23-02-2018

Hey this is from View 2

Demonstrating use of navigation controls

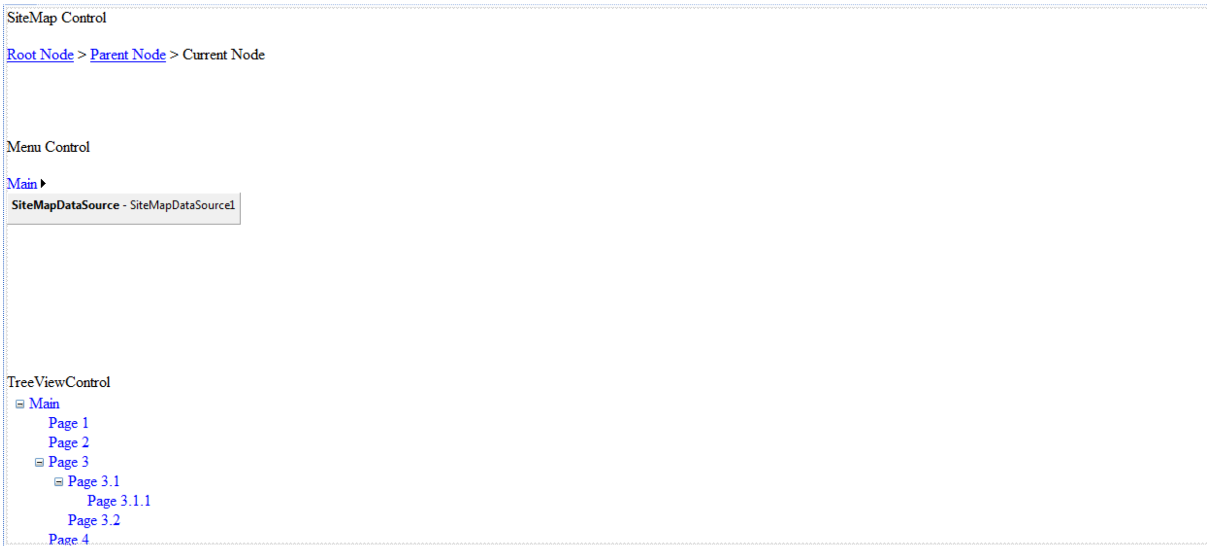
Firstly create an master page ,then add 8 web pages using the master page , names can be random but to follow along use the given convention:

WebForm1, WebForm2, WebForm3, WebForm3_1, WebForm3_1_1, WebForm3_2, WebForm4, MainPage;

Add the Navigation controls from the toolbox look on to navigational control part, and add a sitemapControl , menuControl, treeViewControl to the master page.

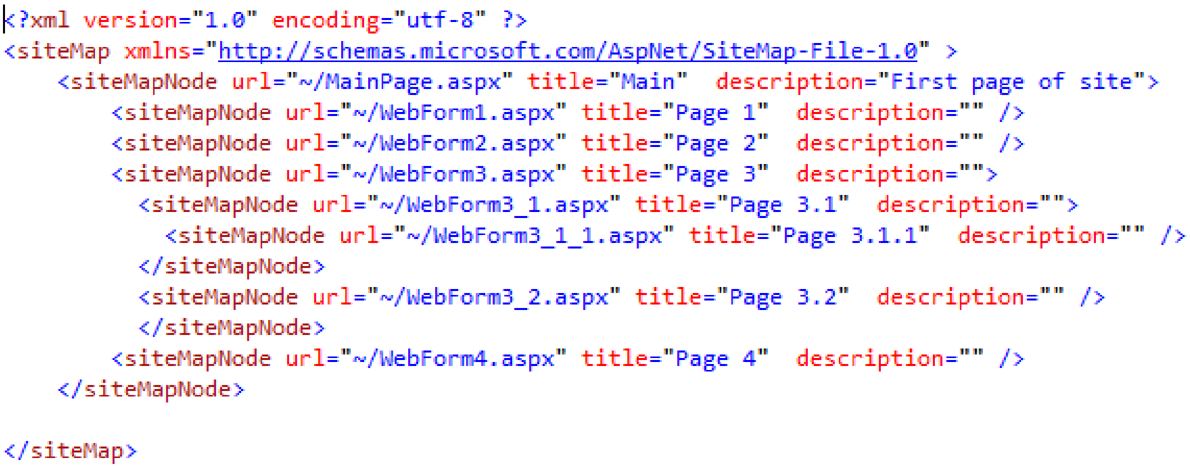
It should look similar to this,

*****Note you won't see content similar to this, as we still have to provide the datasource.



Now create as sitemap file (add ->NewItems ->Data ->sitemap),

By default it will provide some predefined content just edit it as shown below:

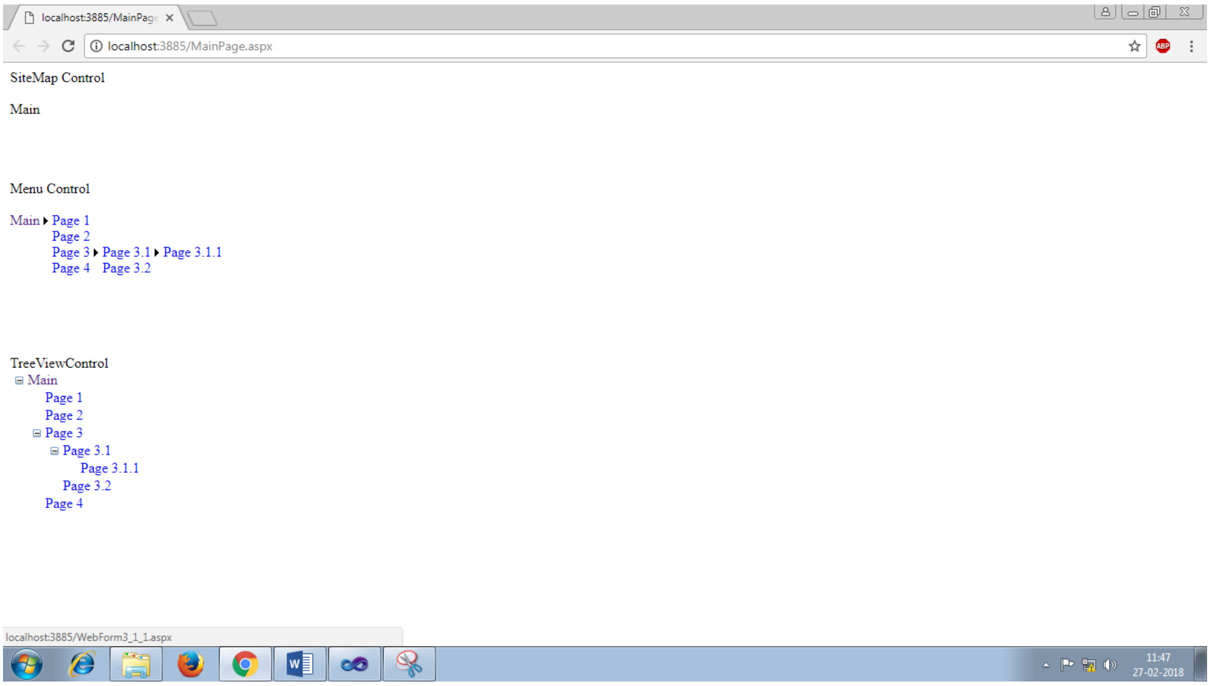


```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
    <siteMapNode url="~/MainPage.aspx" title="Main"  description="First page of site">
        <siteMapNode url="~/WebForm1.aspx" title="Page 1"  description="" />
        <siteMapNode url="~/WebForm2.aspx" title="Page 2"  description="" />
        <siteMapNode url="~/WebForm3.aspx" title="Page 3"  description="">
          <siteMapNode url="~/WebForm3_1.aspx" title="Page 3.1"  description="">
            <siteMapNode url="~/WebForm3_1_1.aspx" title="Page 3.1.1"  description="" />
          </siteMapNode>
          <siteMapNode url="~/WebForm3_2.aspx" title="Page 3.2"  description="" />
          </siteMapNode>
        <siteMapNode url="~/WebForm4.aspx" title="Page 4"  description="" />
    </siteMapNode>

</siteMap>
```

Now you do not need to attach a sitemap file to sitemap control but you do need to attach it to menucontrol and treeviewcontrol;

Just click on the arrow on upperright corner attach a new data source and select the sitemap file.

Output:


localhost:3885/MainPage
localhost:3885/MainPage.aspx
localhost:3885/WebForm3_1_1.aspx
11:47
27-02-2018

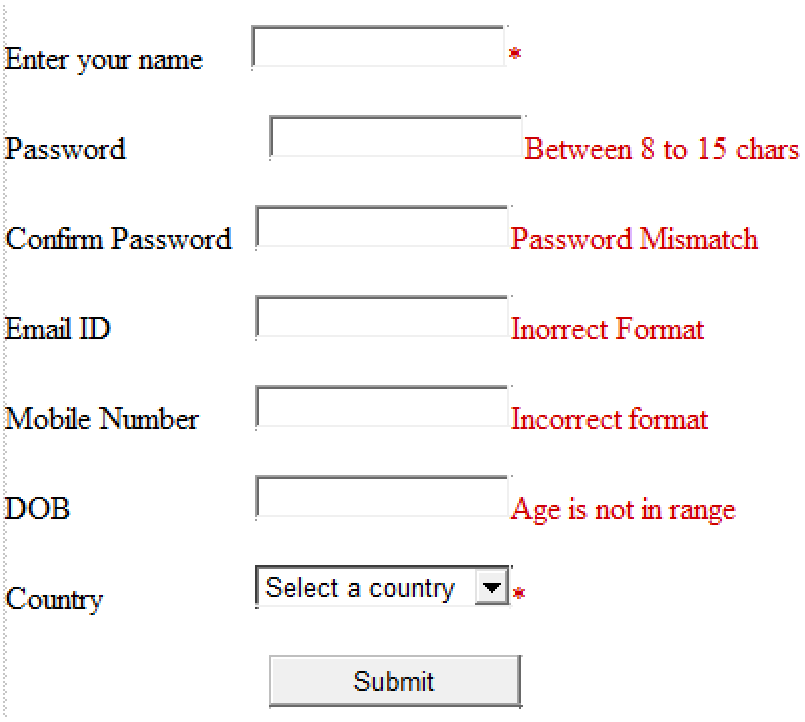
```
SiteMap Control

Main



Menu Control

Main ▸ Page 1
       Page 2
       Page 3 ▸ Page 3.1 ▸ Page 3.1.1
       Page 4   Page 3.2



TreeViewControl
⊟ Main
     Page 1
     Page 2
   ⊟ Page 3
      ⊟ Page 3.1
           Page 3.1.1
         Page 3.2
     Page 4
```

Practical 5

Server Side Validation Control

Design

Controls : 6 TextBoxes, a drop down list, a button.



Enter your name *

Password Between 8 to 15 chars

Confirm Password Password Mismatch

Email ID Inorrect Format

Mobile Number Incorrect format

DOB Age is not in range

Country Select a country *

Next we will attach a validation control to each of textbox controls

You only need to drag and drop the controls from validation section and further set its properties from property window as defined in the given code.

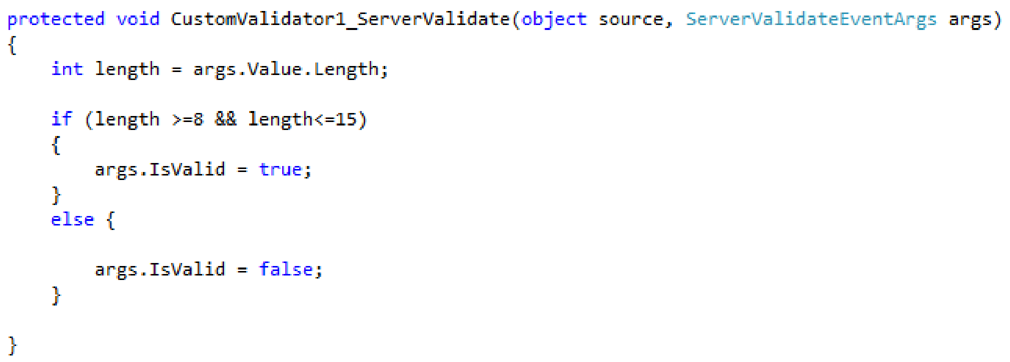
For Textbox1, attach require field validator control

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
    ControlToValidate="TextBox1" Display="Dynamic" ErrorMessage="Name is mandatory"
    ForeColor="#CC0000" SetFocusOnError="True">*</asp:RequiredFieldValidator>
```

For Textbox2, attach a custom validator control,

```
<asp:CustomValidator ID="CustomValidator1" runat="server"
    ControlToValidate="TextBox2" Display="Dynamic" ForeColor="#CC0000"
    onservervalidate="CustomValidator1_ServerValidate" SetFocusOnError="True"
    ValidateEmptyText="True">Between 8 to 15 chars</asp:CustomValidator>
```

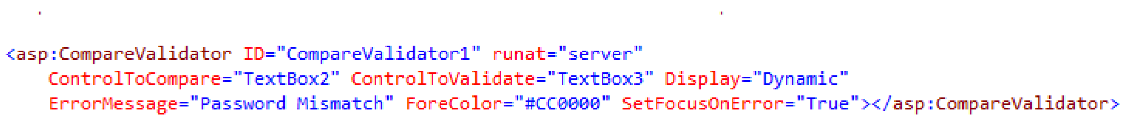
For this control to work you need to associate an event by double clicking on the control by default it will create CustomValidator1_ServerValidate method on aspx.cs file ,just make the necessary changes



```
protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
{
    int length = args.Value.Length;

    if (length >=8 && length<=15)
    {
        args.IsValid = true;
    }
    else {

        args.IsValid = false;
    }

}
```

For Textbox3,

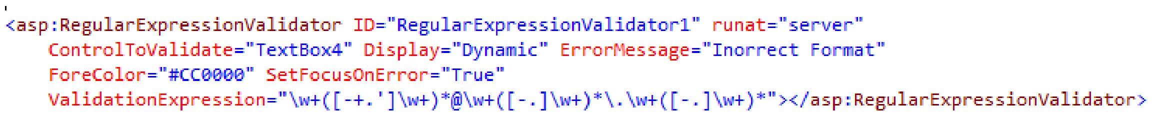
Now we will use a compare field validator, since we it will be attach to confirm the password user entered



```
<asp:CompareValidator ID="CompareValidator1" runat="server"
    ControlToCompare="TextBox2" ControlToValidate="TextBox3" Display="Dynamic"
    ErrorMessage="Password Mismatch" ForeColor="#CC0000" SetFocusOnError="True"></asp:CompareValidator>
```

By default textboxes will show you initially the data as you entered , so to change that to password mode, from property window change the TextMode to Password.

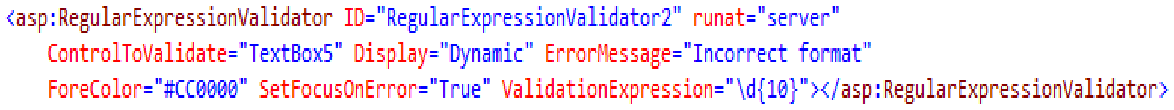
Next for TextBox4, attach a Regular Expression Field validator



```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
    ControlToValidate="TextBox4" Display="Dynamic" ErrorMessage="Inorrect Format"
    ForeColor="#CC0000" SetFocusOnError="True"
    ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"></asp:RegularExpressionValidator>
```

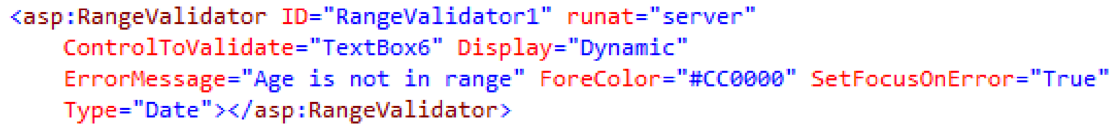
For validation expression shown here select Internet email type from its properties.

For TextBox5, attach another Regular Expression Field validator,



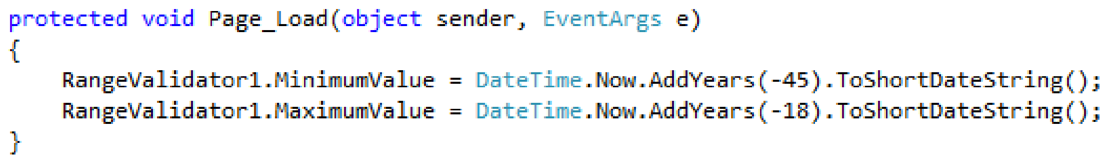
```
<asp:RegularExpressionValidator ID="RegularExpressionValidator2" runat="server"
    ControlToValidate="TextBox5" Display="Dynamic" ErrorMessage="Incorrect format"
    ForeColor="#CC0000" SetFocusOnError="True" ValidationExpression="\d{10}"></asp:RegularExpressionValidator>
```

For TextBox6, attach Range Field Validator



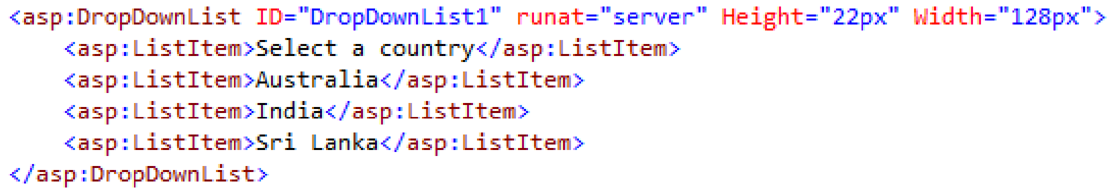
```
<asp:RangeValidator ID="RangeValidator1" runat="server"
    ControlToValidate="TextBox6" Display="Dynamic"
    ErrorMessage="Age is not in range" ForeColor="#CC0000" SetFocusOnError="True"
    Type="Date"></asp:RangeValidator>
```

Make sure you associate a range specified in page load event as shown,

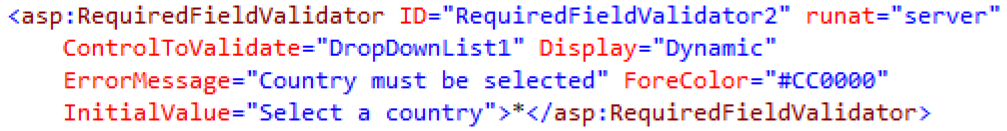


```
protected void Page_Load(object sender, EventArgs e)
{
    RangeValidator1.MinimumValue = DateTime.Now.AddYears(-45).ToShortDateString();
    RangeValidator1.MaximumValue = DateTime.Now.AddYears(-18).ToShortDateString();
}
```

For DropDownlist Add elements as shown and attach RequireFieldValidator,



```
<asp:DropDownList ID="DropDownList1" runat="server" Height="22px" Width="128px">
    <asp:ListItem>Select a country</asp:ListItem>
    <asp:ListItem>Australia</asp:ListItem>
    <asp:ListItem>India</asp:ListItem>
    <asp:ListItem>Sri Lanka</asp:ListItem>
</asp:DropDownList>
```



```
<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
    ControlToValidate="DropDownList1" Display="Dynamic"
    ErrorMessage="Country must be selected" ForeColor="#CC0000"
    InitialValue="Select a country">*</asp:RequiredFieldValidator>
```

Practical 6

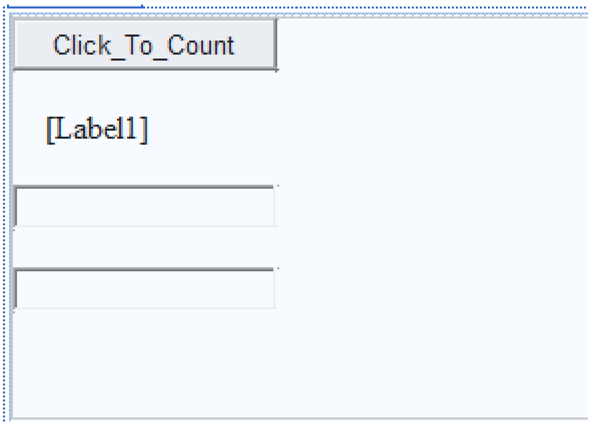
STATE MANAGEMENT

1.ViewState

Interface should like this, it contains:

AspControls: Button, label, AspTextBox

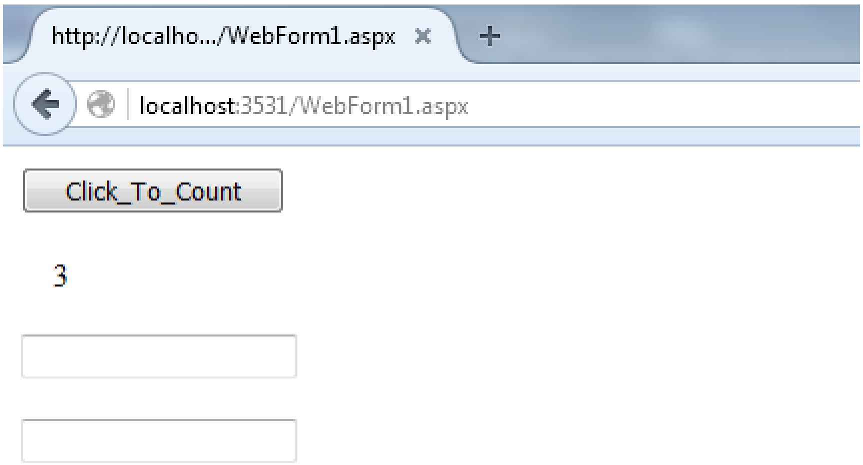
HtmlControl :HtmlTextField (Input type text)



.aspx.cs file

```
public partial class WebForm1 : System.Web.UI.Page
{
    int i = 0;
    protected void Page_Load(object sender, EventArgs e)
    {
        if(!IsPostBack)
        ViewState["i"]=i;
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        i = (int)ViewState["i"];
        Label1.Text = (++i).ToString();
        ViewState["i"] = i;
    }
}
```

O/p:

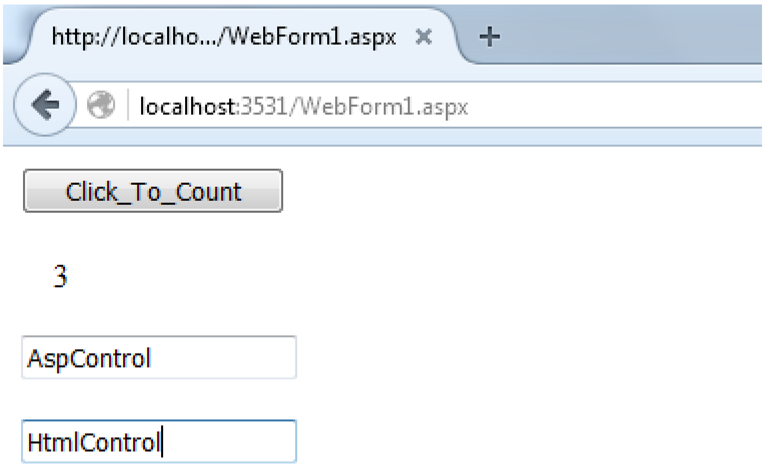


```
3
```

As the output shows the variable maintains it state ,since we saved its value in viewstate property;

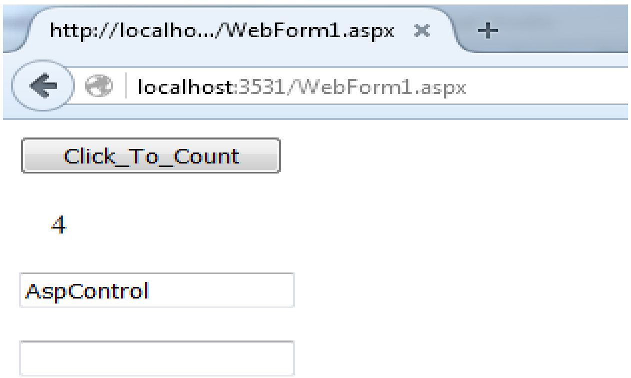
Now there are given two text boxes , where box1 is an asp control and box2 is htmlcontrol,

If you type text in both text boxes as given:



```
3
AspControl
HtmlControl
```

If we now trigger an event handler , as we click the button



The page is posted back when you click , and as a result refreshed now the AspControl text is remembered by textbox1 and HtmlControl text isn’t.

So what’s happening here is that the AspControl (asp:TextBox) has the property of view state enabled which helps it to remember the state, whereas the htmlContol(input type=”text”) is not able to do so.

But you can do it by running the control on server by setting runat property of htmlControl.

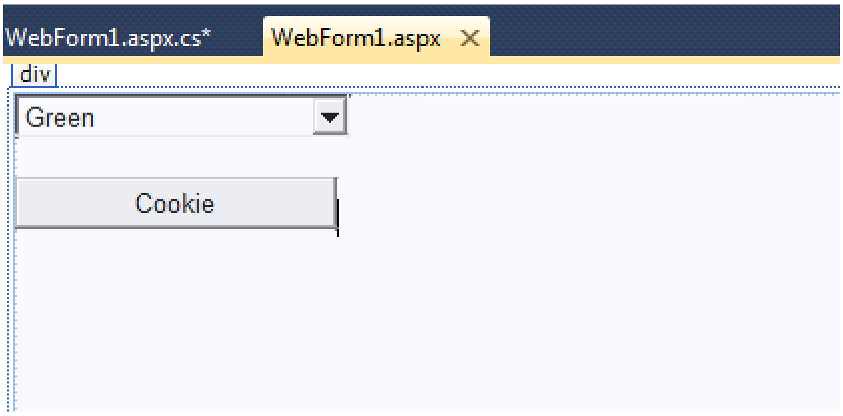
2.Cookies

There are two types of these one is temporary another is permanent,

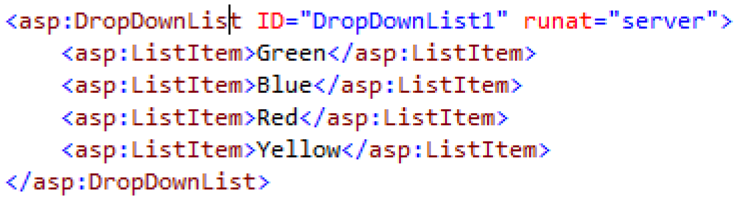
Setting temporary cookies

Design:

AspControl: DropDownList,button



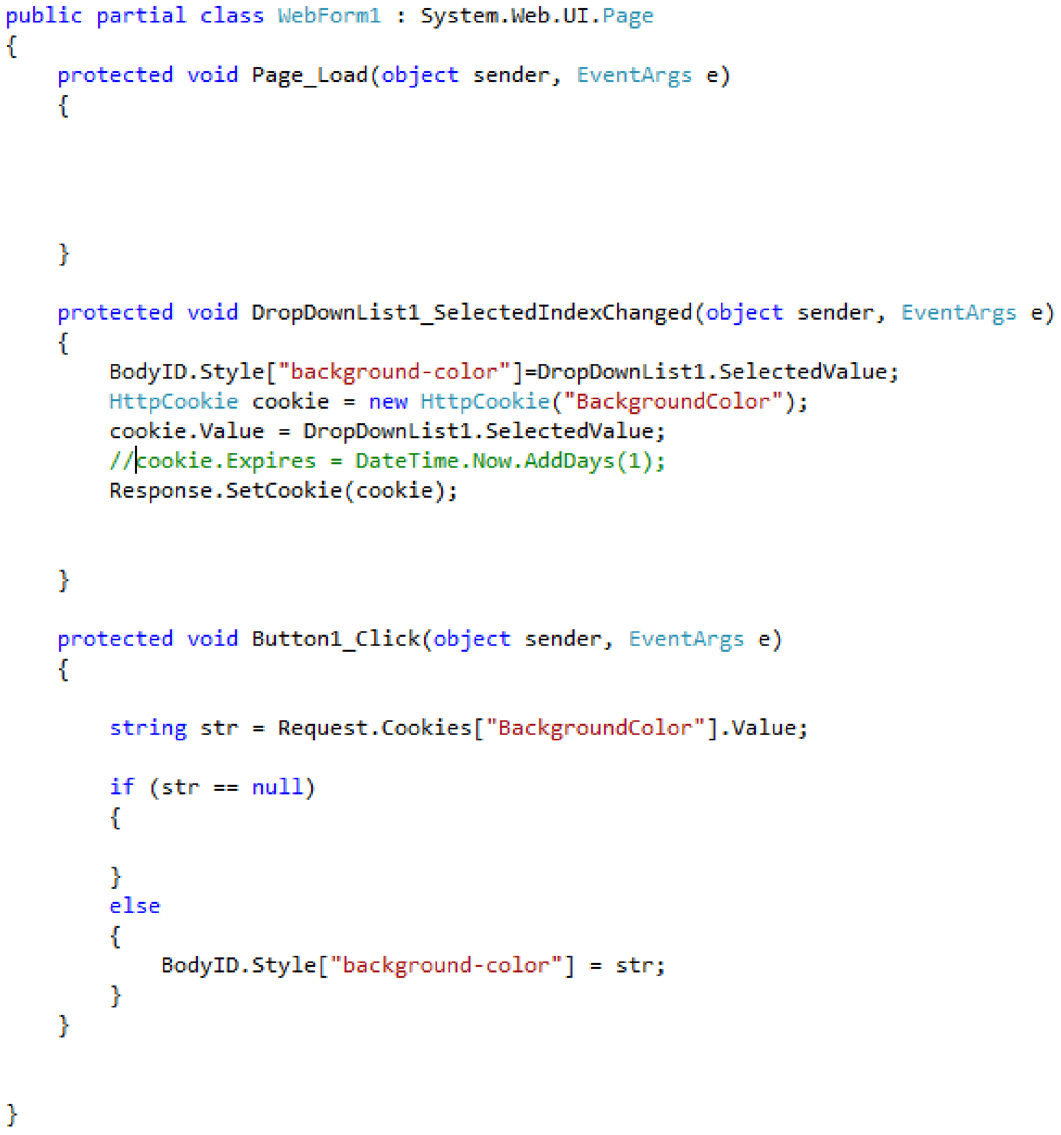
Adding items in dropdownlist:



```
<asp:DropDownList ID="DropDownList1" runat="server">
    <asp:ListItem>Green</asp:ListItem>
    <asp:ListItem>Blue</asp:ListItem>
    <asp:ListItem>Red</asp:ListItem>
    <asp:ListItem>Yellow</asp:ListItem>
</asp:DropDownList>
```

Now let’s add event handler to dropdownlist and button

.aspx.cs



```
public partial class WebForm1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {




    }

    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        BodyID.Style["background-color"]=DropDownList1.SelectedValue;
        HttpCookie cookie = new HttpCookie("BackgroundColor");
        cookie.Value = DropDownList1.SelectedValue;
        //cookie.Expires = DateTime.Now.AddDays(1);
        Response.SetCookie(cookie);


    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        string str = Request.Cookies["BackgroundColor"].Value;

        if (str == null)
        {

        }
        else
        {
            BodyID.Style["background-color"] = str;
        }
    }


}
```

In above file, in an dropdown event the body background is changed as shown.

Next we create an cookie using HttpCookie class and name it as Background-color, then give a value to it ,here we assign list value to it, and then set it with the response object using setCookie method.

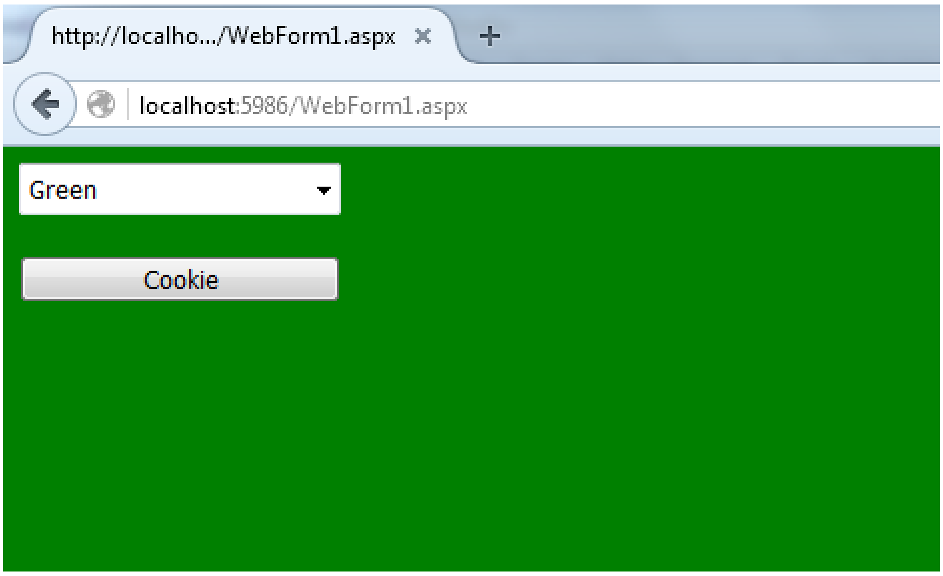
Next we use button event to fetch the existing cookie, if it exists then the body color can be set, else it will fire an exception.

This will only work in the current session , if you change the browser or restart the application , clicking on button won’t work.

Next for permanent cookie,

Just set the expires property then the cookie will remain for that time. And even if you restart the application this would the cookie would remain set.

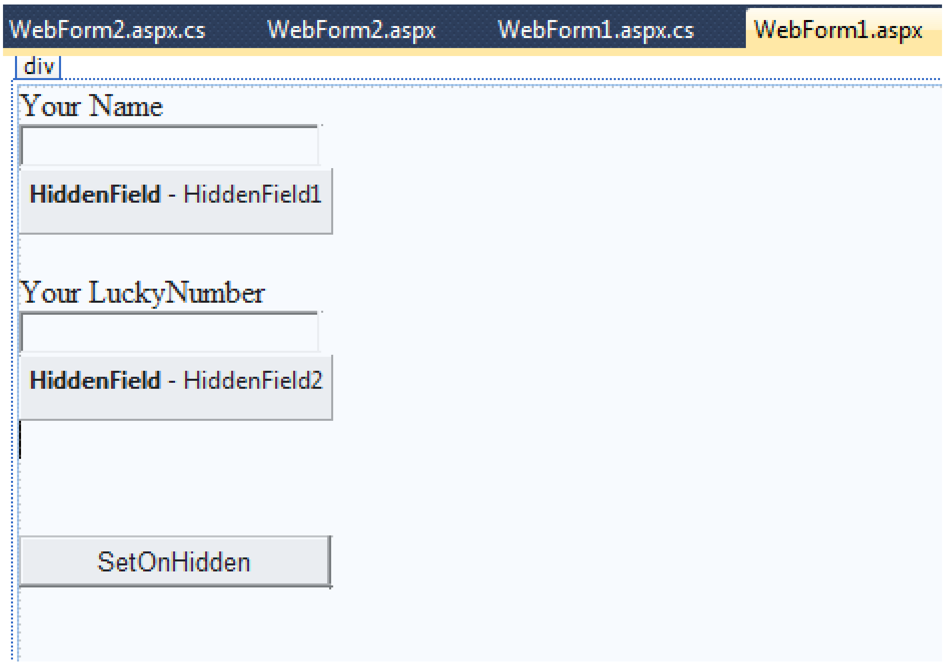
O/p


http://localho.../WebForm1.aspx
localhost:5986/WebForm1.aspx
Green
Cookie

3. Hidden Field and use Query String

Design

Asp Controls: Textboxes,hiddenfields and button.



Create an event handler on button and get the values from textboxes and set it to hidden fields , next it redirect it to next page using response object set the url using query string.

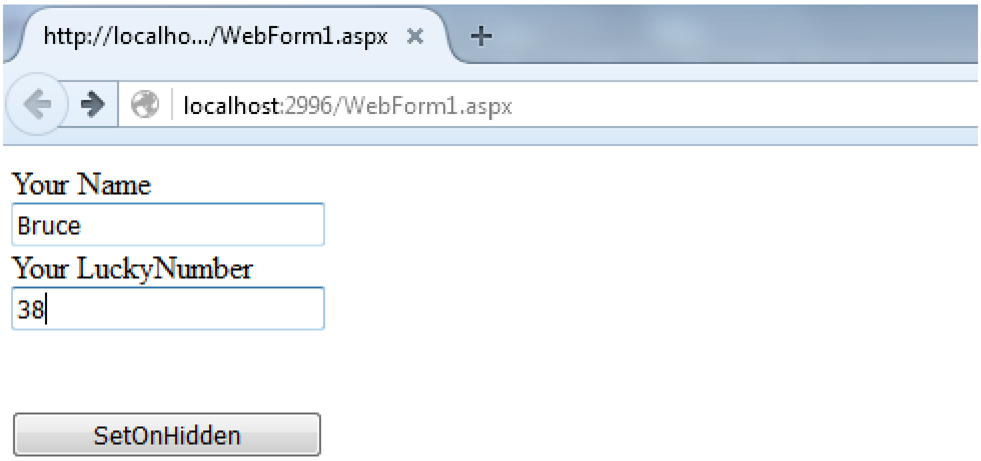
WebForm1.aspx.cs

```
public partial class WebForm1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        ///Setting the value of hidden field from textboxes
        HiddenField1.Value = TextBox1.Text;
        HiddenField2.Value = TextBox2.Text;
        ///Next just pass the value
        ///as response to pass on the values in next page
        Response.Redirect("WebForm2.aspx?name=" + HiddenField1.Value + "&num=" + HiddenField2.Value);
    }
}
```

WebForm2.aspx.cs

```
public partial class WebForm2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Response.Write("Hi " + Request.QueryString["name"] + " and LuckyNumber is " + Request.QueryString["num"]);
    }
}
```

O/p



http://localho.../WebForm1.aspx

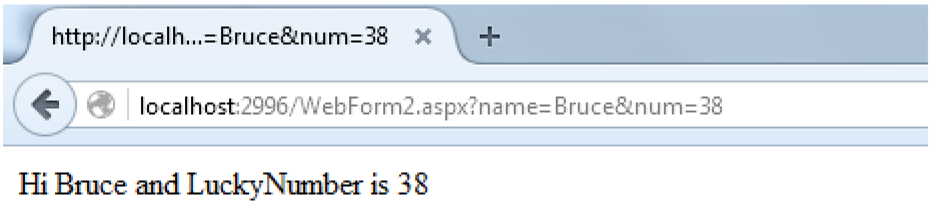
localhost:2996/WebForm1.aspx

Your Name

Bruce

Your LuckyNumber

38



http://localh...=Bruce&num=38

localhost:2996/WebForm2.aspx?name=Bruce&num=38

Hi Bruce and LuckyNumber is 38

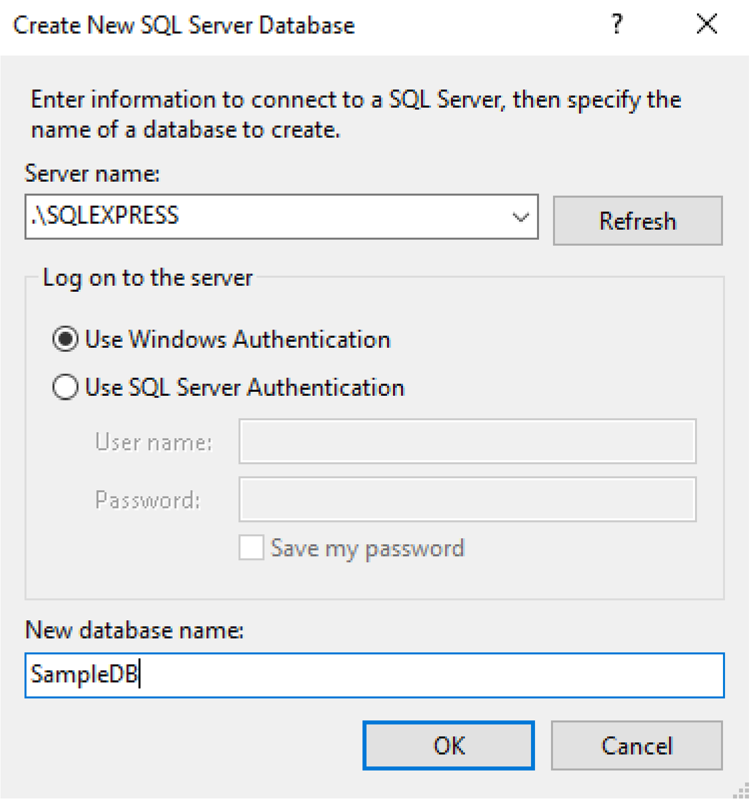
Practical 7

ADO.NET data access in asp.net

First we will create an database using sql server

Step 1: Go to server explorer

Step 2: Right click on data connections , create a new sql server database



Create New SQL Server Database

Enter information to connect to a SQL Server, then specify the name of a database to create.

Server name: .\SQLEXPRESS

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

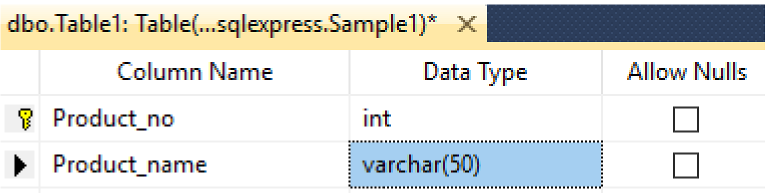
Password:

□ Save my password

New database name: SampleDB

Now we have created a database , lets add an table for that right click on tables option and add a new table

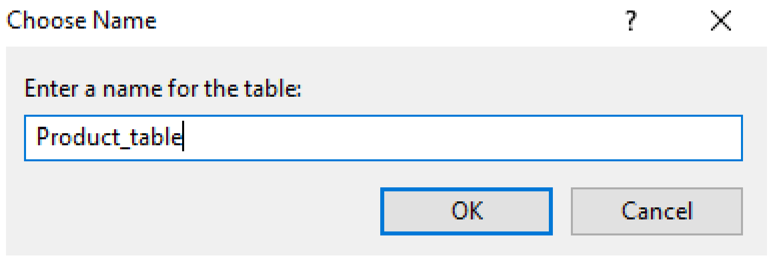
Define the table as shown ,



dbo.Table1: Table(...sqlexpress.Sample1)*

Column Name	Data Type	Allow Nulls
Product_no	int	□
Product_name	varchar(50)	□

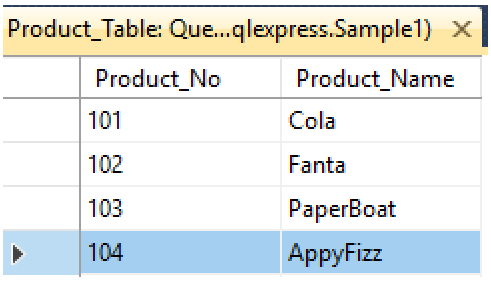
Now just make sure when you click the save option on top of the screen rename the default name to Product_table


Choose Name

Enter a name for the table:

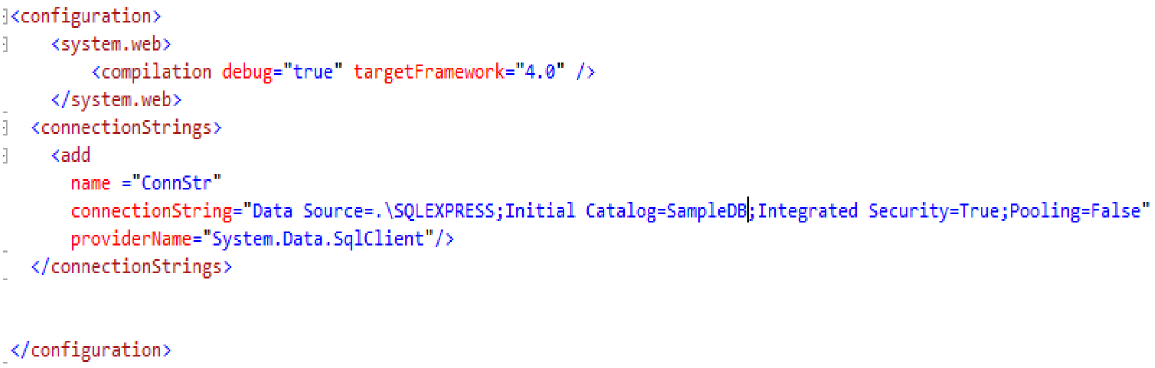
Product_table

Step 3 : Next to fill the table ,again right click on the table you just saved an select the option show table data ,fill it as follows and save it.


Product_Table: Que...qlexpress.Sample1)

Product_No	Product_Name
101	Cola
102	Fanta
103	PaperBoat
104	AppyFizz

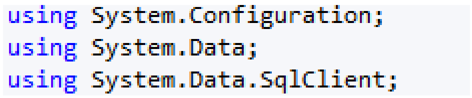
Next we create an web application, firstly add the connection string to your web.config file (obtain connection string from created database properties)

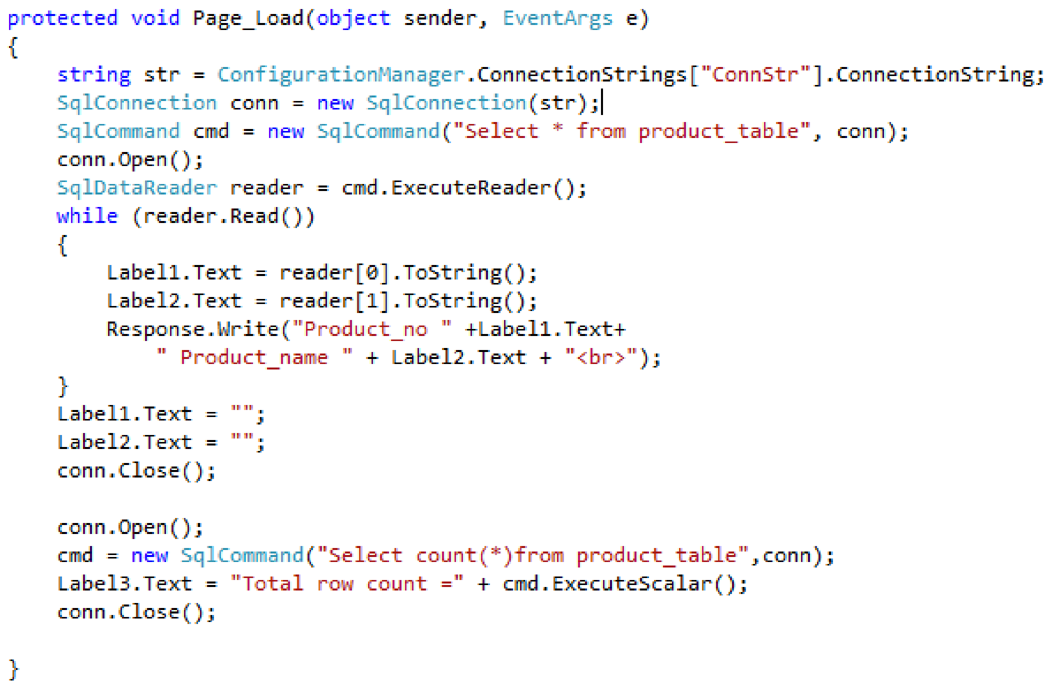

```
<configuration>
    <system.web>
        <compilation debug="true" targetFramework="4.0" />
    </system.web>
  <connectionStrings>
    <add
      name ="ConnStr"
      connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=SampleDB;Integrated Security=True;Pooling=False"
      providerName="System.Data.SqlClient"/>
  </connectionStrings>


</configuration>
```

1.Single Value data binding using ado.net data access

Create a web form add 3 labels

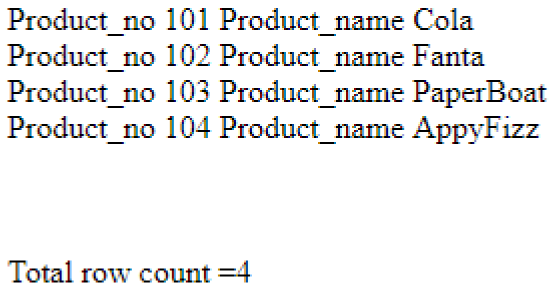
Now in webform1.aspx.cs write the code as follows, don’t forget to import namespaces as follows


```
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
```



```
protected void Page_Load(object sender, EventArgs e)
{
    string str = ConfigurationManager.ConnectionStrings["ConnStr"].ConnectionString;
    SqlConnection conn = new SqlConnection(str);
    SqlCommand cmd = new SqlCommand("Select * from product_table", conn);
    conn.Open();
    SqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        Label1.Text = reader[0].ToString();
        Label2.Text = reader[1].ToString();
        Response.Write("Product_no " +Label1.Text+
            " Product_name " + Label2.Text + "<br>");
    }
    Label1.Text = "";
    Label2.Text = "";
    conn.Close();

    conn.Open();
    cmd = new SqlCommand("Select count(*)from product_table",conn);
    Label3.Text = "Total row count =" + cmd.ExecuteScalar();
    conn.Close();

}
```

O/p



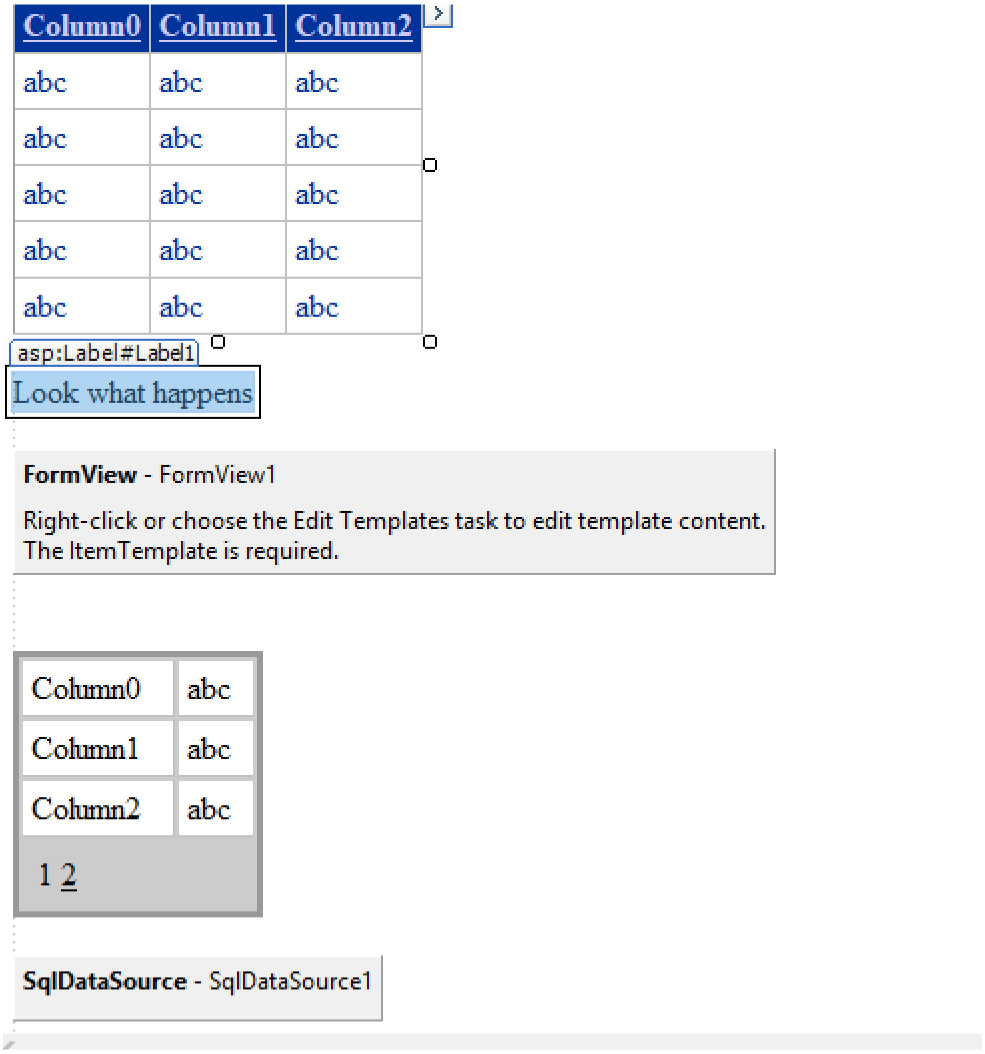
```
Product_no 101 Product_name Cola
Product_no 102 Product_name Fanta
Product_no 103 Product_name PaperBoat
Product_no 104 Product_name AppyFizz



Total row count =4
```

2. Multi value data binding

Create a new web form and add a data controls as follows

Grid View , Form View , Details View , SqlDataSource ,label from data options in toolbox.

.aspx



Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Look what happens

FormView - FormView1

Right-click or choose the Edit Templates task to edit template content.
The ItemTemplate is required.

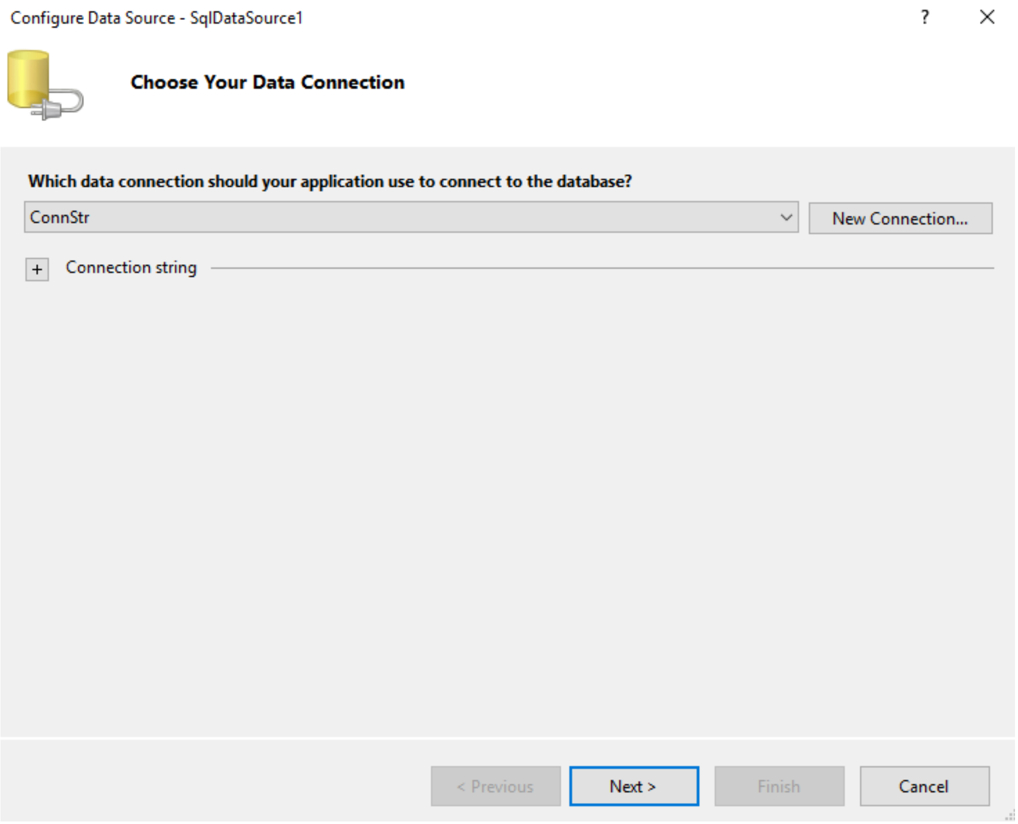
Column0	abc
Column1	abc
Column2	abc

1 2

SqlDataSource - SqlDataSource1

Now You need to configure the SqlDataSource ,for that select the on the top right corner of this control ,select option configure data source

Next select the connection string for the database the sane one we specified earlier on the web.config file

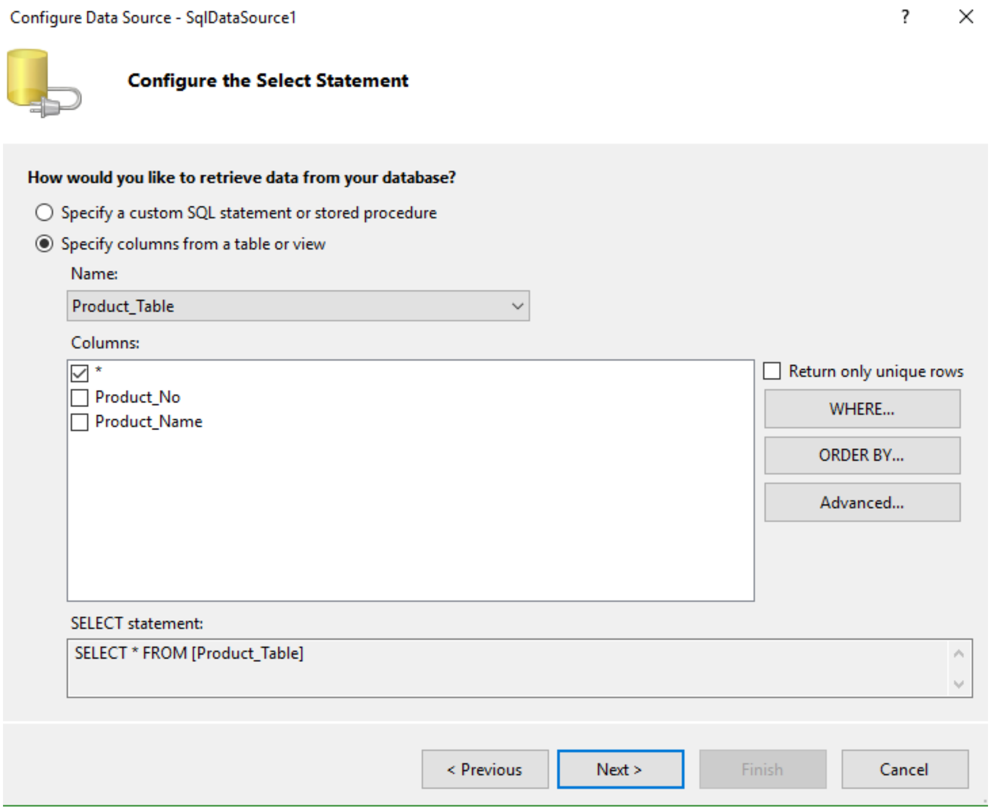


Configure Data Source - SqlDataSource1
?
Choose Your Data Connection

Which data connection should your application use to connect to the database?

ConnStr

New Connection...
Connection string
< Previous
Next >
Finish
Cancel



Configure Data Source - SqlDataSource1
?
Configure the Select Statement

How would you like to retrieve data from your database?

○ Specify a custom SQL statement or stored procedure

● Specify columns from a table or view

Name: Product_Table

Columns:

☑ *

□ Product_No

□ Product_Name

□ Return only unique rows

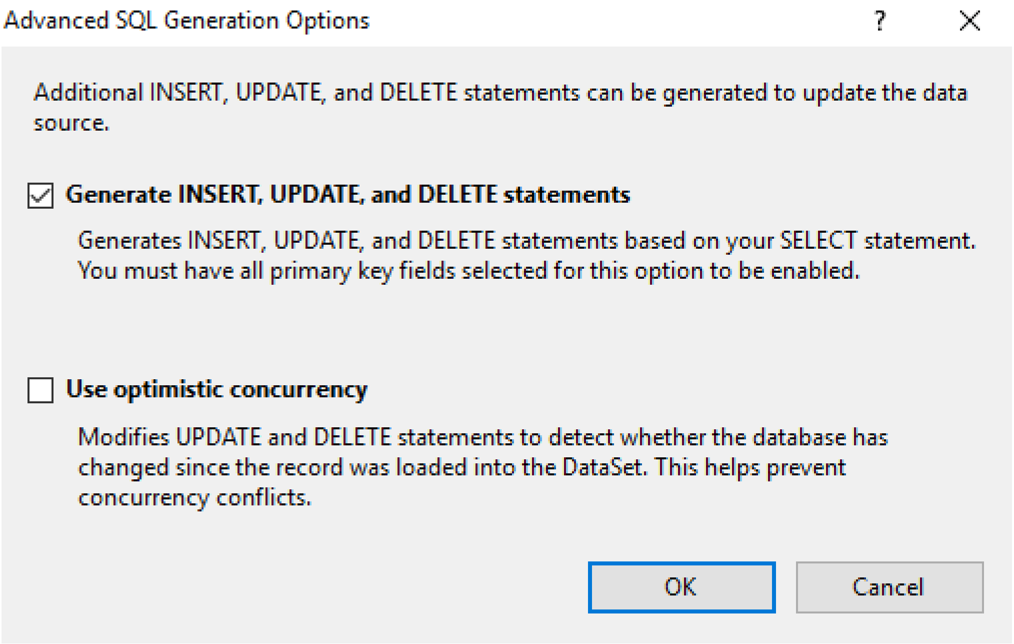
WHERE...
ORDER BY...
Advanced...

SELECT statement:

```
SELECT * FROM [Product_Table]
```

< Previous
Next >
Finish
Cancel

Select the Advanced option



Advanced SQL Generation Options

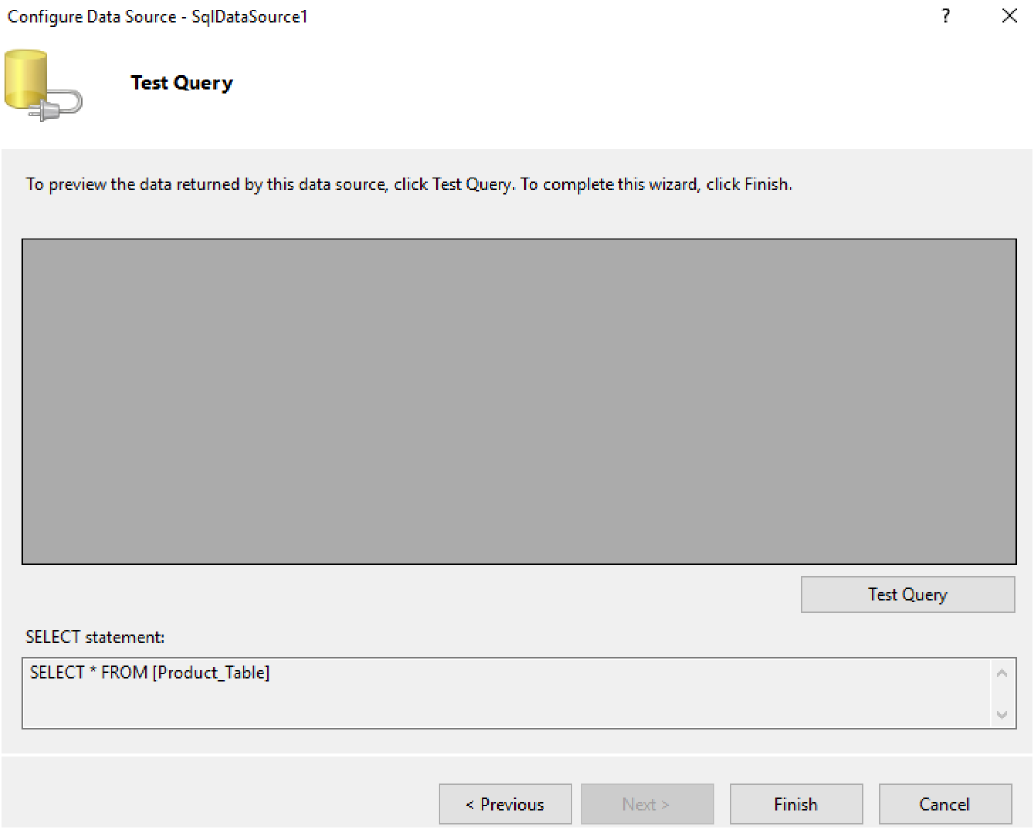
Additional INSERT, UPDATE, and DELETE statements can be generated to update the data source.

☑ **Generate INSERT, UPDATE, and DELETE statements**

Generates INSERT, UPDATE, and DELETE statements based on your SELECT statement. You must have all primary key fields selected for this option to be enabled.

☐ **Use optimistic concurrency**

Modifies UPDATE and DELETE statements to detect whether the database has changed since the record was loaded into the DataSet. This helps prevent concurrency conflicts.



Configure Data Source - SqlDataSource1

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

SELECT statement:

```
SELECT * FROM [Product_Table]
```

Click on Finish

Now just simply attach the data source to the data control by select it from top right corner.

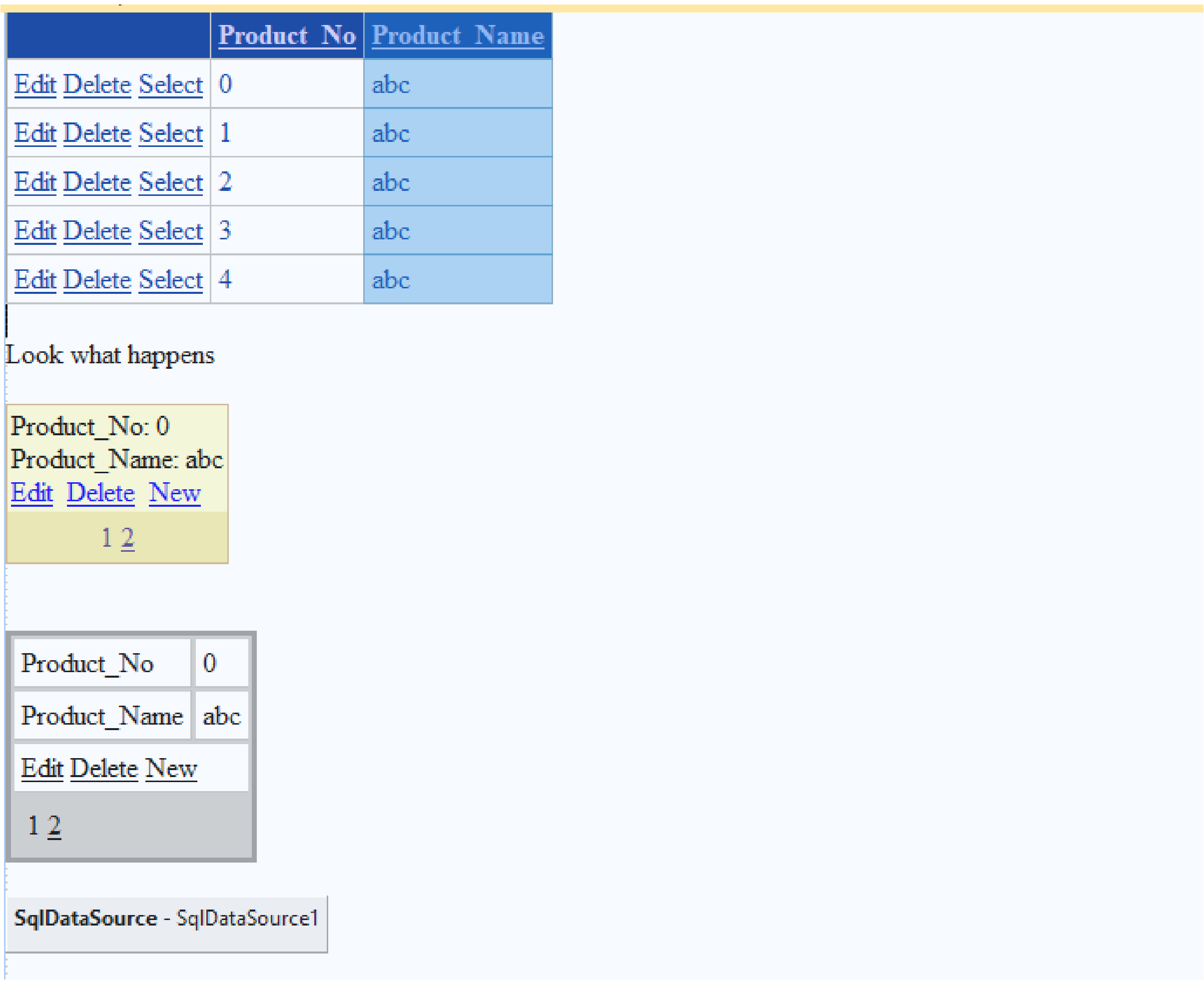
Make sure you enable following for each control ,

Grid View : enable editing, deleting ,sorting ,selecting

Form View : enable paging.

Details View: enable paging , editing, deleting,inserting.

The end result should look something like this,



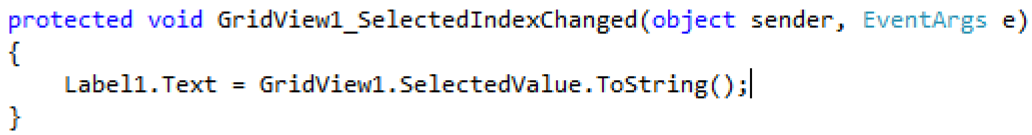
	Product_No	Product_Name
Edit Delete Select	0	abc
Edit Delete Select	1	abc
Edit Delete Select	2	abc
Edit Delete Select	3	abc
Edit Delete Select	4	abc

Look what happens

Product_No: 0
Product_Name: abc
Edit Delete New
1 2

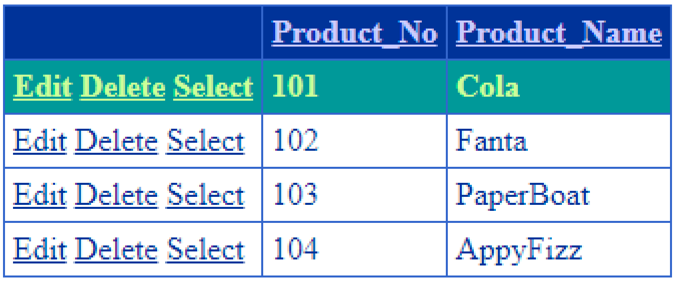
Product_No	0
Product_Name	abc
Edit Delete New	
1 2	

Finally attach a an event to grid view by double clicking on it ,



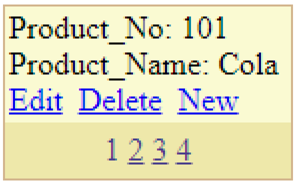
```
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
        Label1.Text = GridView1.SelectedValue.ToString();
    }
```

O/P



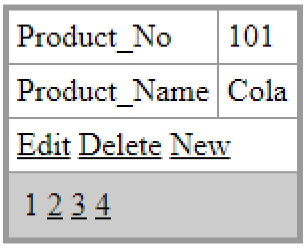
	Product_No	Product_Name
Edit Delete Select	101	Cola
Edit Delete Select	102	Fanta
Edit Delete Select	103	PaperBoat
Edit Delete Select	104	AppyFizz

101



Product_No: 101
Product_Name: Cola
Edit Delete New

1 2 3 4



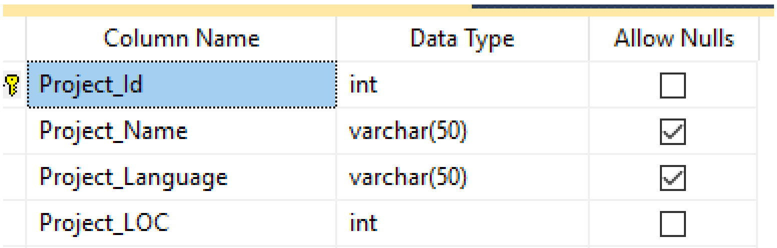
Product_No	101
Product_Name	Cola
Edit Delete New	
1 2 3 4	

Practical 8

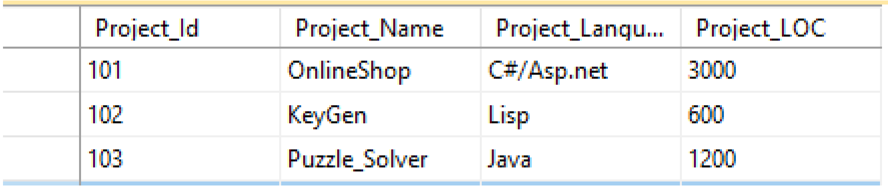
Using LINQ to Query Database

First create two tables named Projects, P_Tech, follow the steps to create it from previous practical’s, the column and row values of tables should be something as shown

For Projects ,

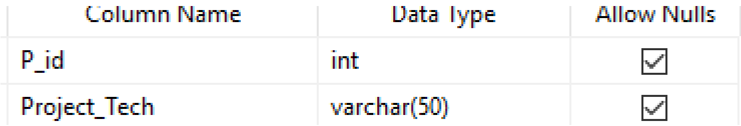


Column Name	Data Type	Allow Nulls
🔑 Project_Id	int	☐
Project_Name	varchar(50)	☑
Project_Language	varchar(50)	☑
Project_LOC	int	☐

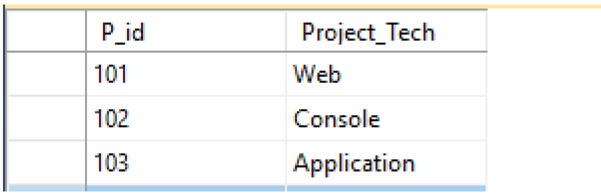


Project_Id	Project_Name	Project_Langu...	Project_LOC
101	OnlineShop	C#/Asp.net	3000
102	KeyGen	Lisp	600
103	Puzzle_Solver	Java	1200

For P_Tech ,



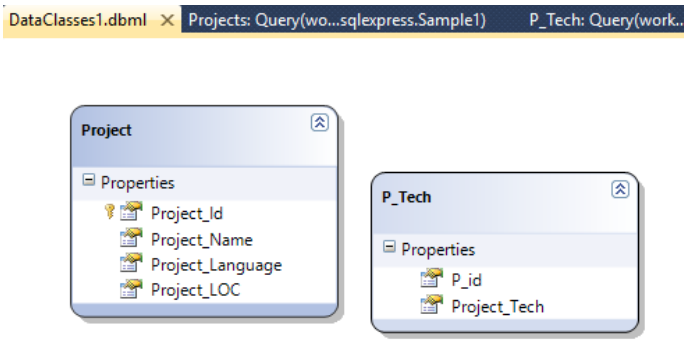
Column Name	Data Type	Allow Nulls
P_id	int	☑
Project_Tech	varchar(50)	☑



P_id	Project_Tech
101	Web
102	Console
103	Application

Now create a new web application add a webform and 4 grid view controls, then add a Linq to sql classes (.dbml) file.

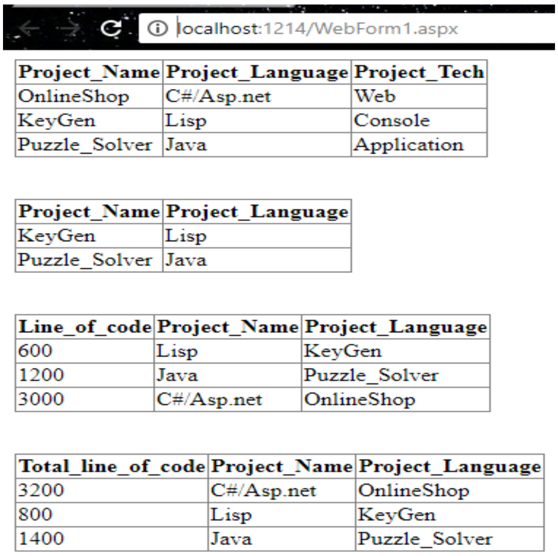
Now open up your .dbml file and simply drag and drop the tables which we created in the database it should look as shown,



Finally make the changes to .aspx.cs file,

```
protected void Page_Load(object sender, EventArgs e)
{
    DataClasses1DataContext d = new DataClasses1DataContext();

    GridView1.DataSource = from project in d.Projects
                           join tech in d.P_Teches
                           on project.Project_Id equals tech.P_id
                           select new
                           {
                               Project_Name = project.Project_Name,
                               Project_Language = project.Project_Language,
                               Project_Tech = tech.Project_Tech
                           };
    GridView1.DataBind();


    GridView2.DataSource=from project in d.Projects where project.Project_Id>101
                         select new
                         {
                         Project_Name =project.Project_Name,
                         Project_Language=project.Project_Language
                         };
    GridView2.DataBind();

    GridView3.DataSource = from project in d.Projects
                           orderby project.Project_LOC
                           select new
                           {
                               Line_of_code = project.Project_LOC,
                               Project_Name = project.Project_Language,
                               Project_Language = project.Project_Name
                           };
    GridView3.DataBind();

    GridView4.DataSource = from project in d.Projects
                           let total_lines =project.Project_LOC+200
                           select new
                           {
                               Total_line_of_code = total_lines,
                               Project_Name = project.Project_Language,
                               Project_Language = project.Project_Name
                           };
    GridView4.DataBind();


}
```

O/p



Project_Name	Project_Language	Project_Tech
OnlineShop	C#/Asp.net	Web
KeyGen	Lisp	Console
Puzzle_Solver	Java	Application

Project_Name	Project_Language
KeyGen	Lisp
Puzzle_Solver	Java

Line_of_code	Project_Name	Project_Language
600	Lisp	KeyGen
1200	Java	Puzzle_Solver
3000	C#/Asp.net	OnlineShop

Total_line_of_code	Project_Name	Project_Language
3200	C#/Asp.net	OnlineShop
800	Lisp	KeyGen
1400	Java	Puzzle_Solver

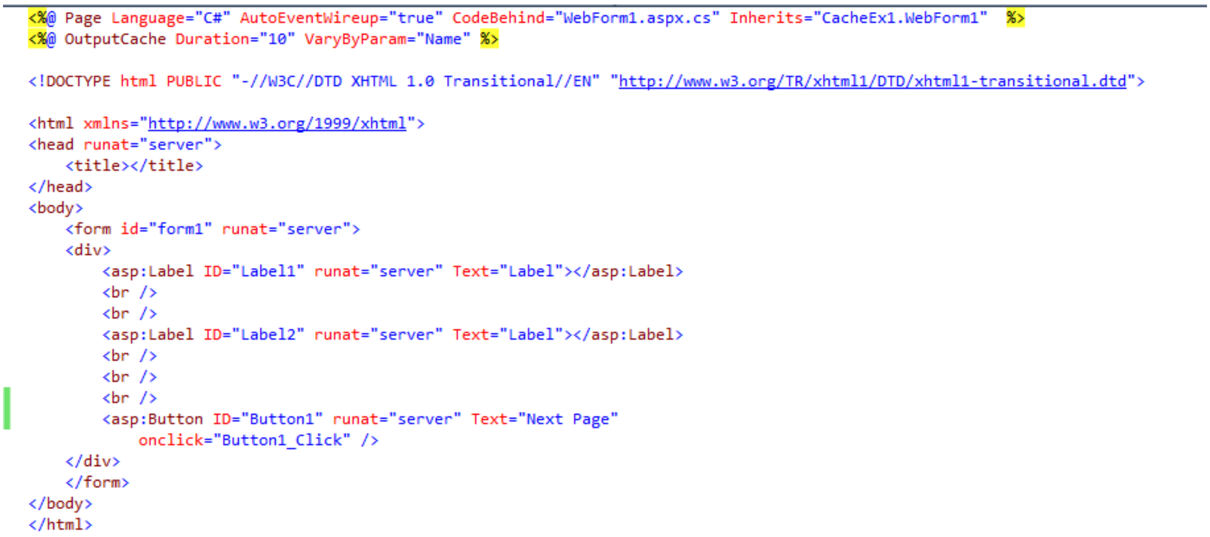
Practical 9

Working with cache

Demo of output and object caching

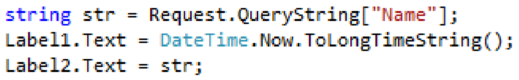
First create 3 web forms ,then add two asp labels and one button to it. Make sure you declare the OutputCache directive as shown in below.

WebForm1.aspx



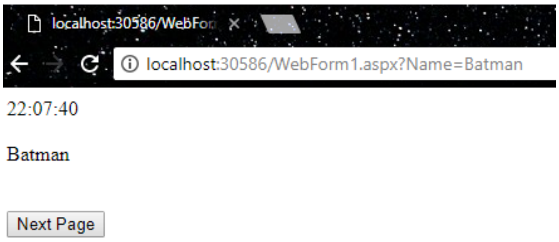
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="CacheEx1.WebForm1"  %>
<%@ OutputCache Duration="10" VaryByParam="Name" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<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:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <br />
        <asp:Button ID="Button1" runat="server" Text="Next Page"
            onclick="Button1_Click" />
    </div>
    </form>
</body>
</html>
```

Write the following code in WebForm1.aspx.cs page load event (we will continue to add more code in this event in our next example)



```
string str = Request.QueryString["Name"];
Label1.Text = DateTime.Now.ToLongTimeString();
Label2.Text = str;
```

Now see the output and make sure you pass on the query in the url as shown ,now try to refresh it the time won’t change and and also the name parameter would be saved in cache.



```
22:07:40

Batman
```

Ok now lets show how object caching works, for this you need to add any connection string to your web.config file (refer previous sections)

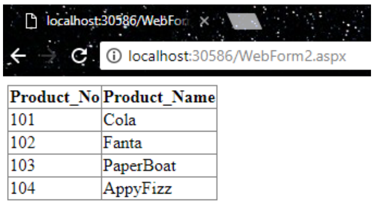
Now continue editing the WebForm1.aspx.cs file

```
SqlConnection conn;
SqlDataAdapter da;
DataSet ds;
protected void Page_Load(object sender, EventArgs e)
{
    string str = Request.QueryString["Name"];
    Label1.Text = DateTime.Now.ToLongTimeString();
    Label2.Text = str;


    if (!IsPostBack) {
        conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnStr"].ToString());
        da = new SqlDataAdapter("Select * from Product_table",conn);
        ds = new DataSet();
        da.Fill(ds);
        ViewState["data"] = ds;

    }

}

protected void Button1_Click(object sender, EventArgs e)
{
    Cache["data"] = ViewState["data"];
    Response.Redirect("WebForm2.aspx");
}
```

Now add a grid view control in Webform2 and write the following in its WebForm2.aspx.cs file

```
protected void Page_Load(object sender, EventArgs e)
{
    DataSet ds = (DataSet)Cache["data"];
    GridView1.DataSource = ds;
    GridView1.DataBind();
}
```

Further run the WebForm1 and click the next page button you should be able to see your table in web form 2 as below



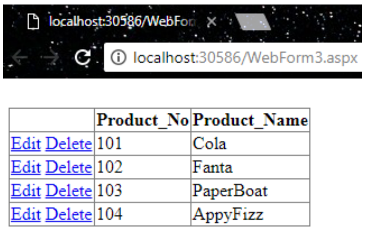
Product_No	Product_Name
101	Cola
102	Fanta
103	PaperBoat
104	AppyFizz

Finally we will demonstrate data caching, in WebForm3 add grid view and sql data source ,then attach data source to grid view(refer previous section if confused)

Just make few changes in Sql data source as shown below(make it from property window)

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnStr %>"
    SelectCommand="SELECT * FROM [Rules]" CacheDuration="60"
    EnableCaching="True"></asp:SqlDataSource>
```

O/p: Now if you edit or delete it should not reflect changes for about 60 seconds in database

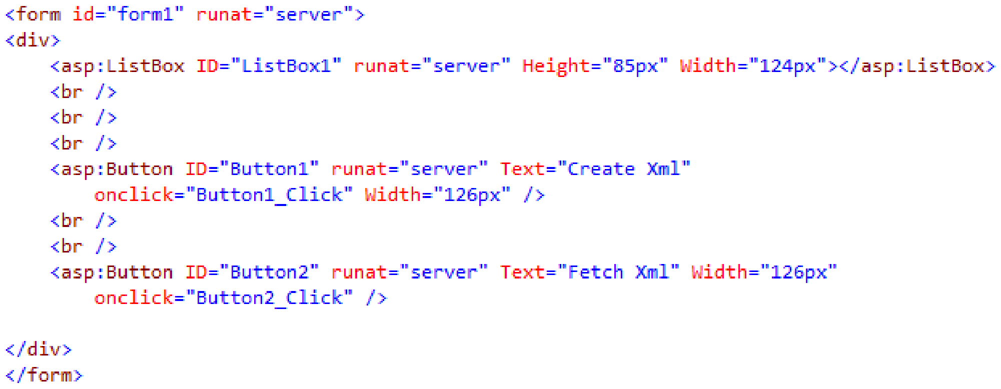


	Product_No	Product_Name
Edit Delete	101	Cola
Edit Delete	102	Fanta
Edit Delete	103	PaperBoat
Edit Delete	104	AppyFizz

Working with XML Reader/Writer.

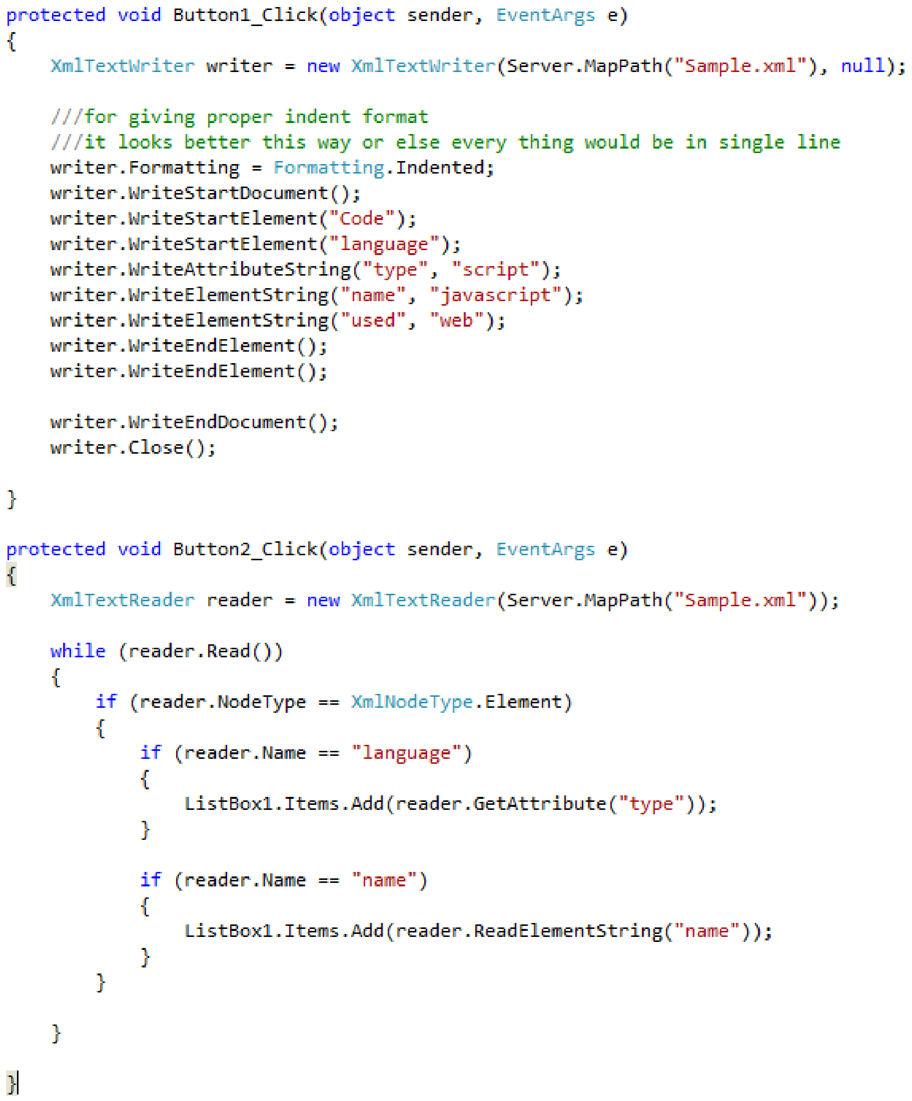
Create a new web form add a asp List box control and two asp buttons

.aspx file



```
<form id="form1" runat="server">
<div>
    <asp:ListBox ID="ListBox1" runat="server" Height="85px" Width="124px"></asp:ListBox>
    <br />
    <br />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Create Xml"
        onclick="Button1_Click" Width="126px" />
    <br />
    <br />
    <asp:Button ID="Button2" runat="server" Text="Fetch Xml" Width="126px"
        onclick="Button2_Click" />

</div>
</form>
```

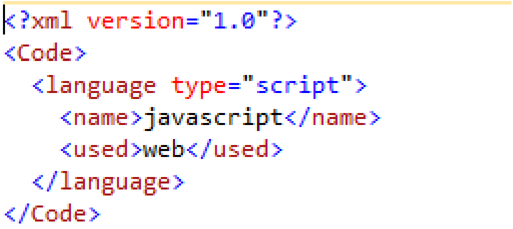
Add events to each button and write the following code to work with it,



```
protected void Button1_Click(object sender, EventArgs e)
{
    XmlTextWriter writer = new XmlTextWriter(Server.MapPath("Sample.xml"), null);

    ///for giving proper indent format
    ///it looks better this way or else every thing would be in single line
    writer.Formatting = Formatting.Indented;
    writer.WriteStartDocument();
    writer.WriteStartElement("Code");
    writer.WriteStartElement("language");
    writer.WriteAttributeString("type", "script");
    writer.WriteElementString("name", "javascript");
    writer.WriteElementString("used", "web");
    writer.WriteEndElement();
    writer.WriteEndElement();

    writer.WriteEndDocument();
    writer.Close();

}

protected void Button2_Click(object sender, EventArgs e)
{
    XmlTextReader reader = new XmlTextReader(Server.MapPath("Sample.xml"));

    while (reader.Read())
    {
        if (reader.NodeType == XmlNodeType.Element)
        {
            if (reader.Name == "language")
            {
                ListBox1.Items.Add(reader.GetAttribute("type"));
            }

            if (reader.Name == "name")
            {
                ListBox1.Items.Add(reader.ReadElementString("name"));
            }
        }

    }

}
```

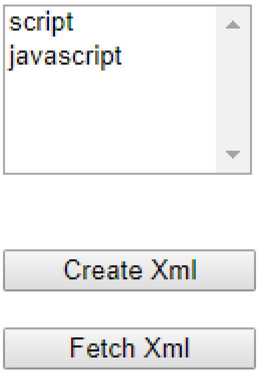
O/p

First it creates an xml file when you click on create Xml



```xml
<?xml version="1.0"?>
<Code>
  <language type="script">
    <name>javascript</name>
    <used>web</used>
  </language>
</Code>
```

And then fetch it



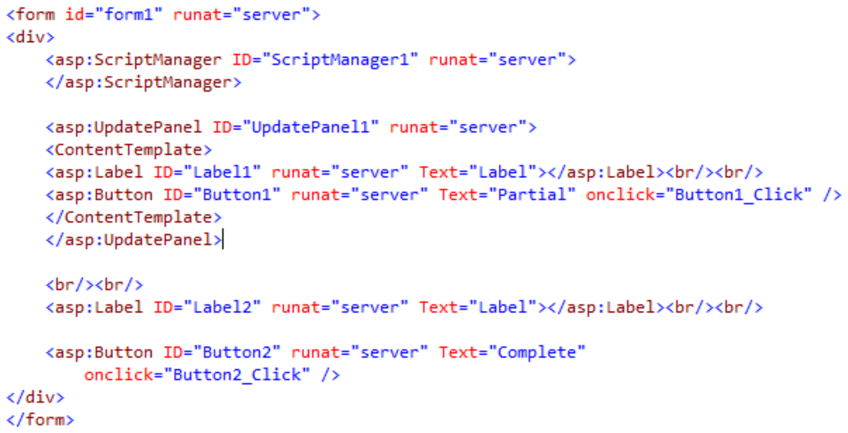
```
script
javascript
```

Practical 10

Use of Ajax

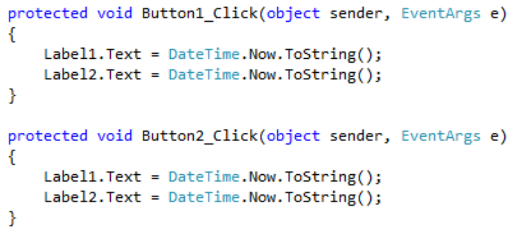
Create 3 web forms and add elements as shown below

Webform1.aspx



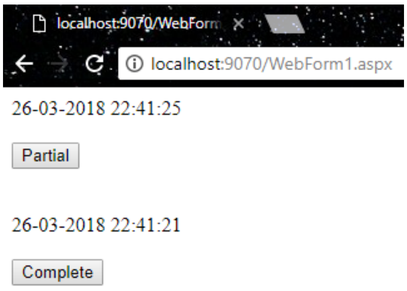
```
<form id="form1" runat="server">
<div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>

    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label><br/><br/>
    <asp:Button ID="Button1" runat="server" Text="Partial" onclick="Button1_Click" />
    </ContentTemplate>
    </asp:UpdatePanel>

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

    <asp:Button ID="Button2" runat="server" Text="Complete"
        onclick="Button2_Click" />
</div>
</form>
```

Attach events to both buttons WebForm2.aspx.cs



```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = DateTime.Now.ToString();
    Label2.Text = DateTime.Now.ToString();
}

protected void Button2_Click(object sender, EventArgs e)
{
    Label1.Text = DateTime.Now.ToString();
    Label2.Text = DateTime.Now.ToString();
}
```

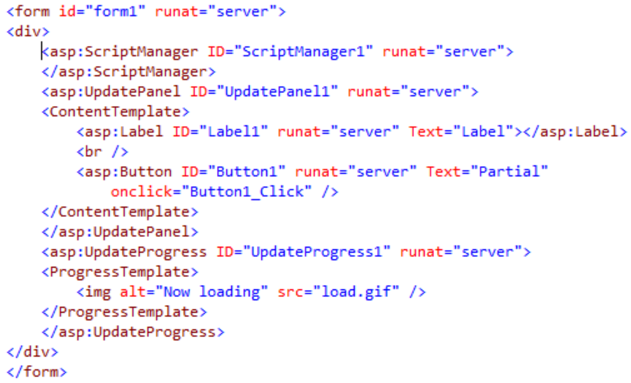
O/p : you could observe how partial and complete refresh works



26-03-2018 22:41:25

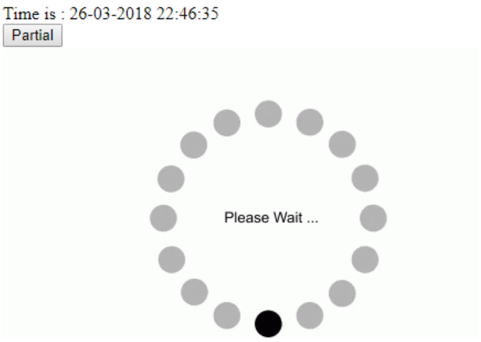
26-03-2018 22:41:21

Now move on to WebForm2.aspx(note that we need to copy an image to our root directory)



```
<form id="form1" runat="server">
<div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <asp:Button ID="Button1" runat="server" Text="Partial"
            onclick="Button1_Click" />
    </ContentTemplate>
    </asp:UpdatePanel>
    <asp:UpdateProgress ID="UpdateProgress1" runat="server">
    <ProgressTemplate>
        <img alt="Now loading" src="load.gif" />
    </ProgressTemplate>
    </asp:UpdateProgress>
</div>
</form>
```

O/p:

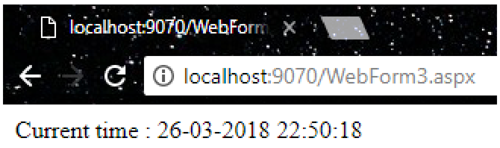


Time is : 26-03-2018 22:46:35

Next for WebForm3.aspx

```
<form id="form1" runat="server">
<div>
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
    <asp:Timer ID="Timer1" runat="server" Interval="5000" ontick="Timer1_Tick">
    </asp:Timer>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="Timer1" EventName="Tick" />
    </Triggers>
    </asp:UpdatePanel>
```

O/p



Current time : 26-03-2018 22:50:18