UNIVERSITY OF MUMBAI

Teacher's Reference Manual

Subject: Advanced Web Programming

with effect from the academic year

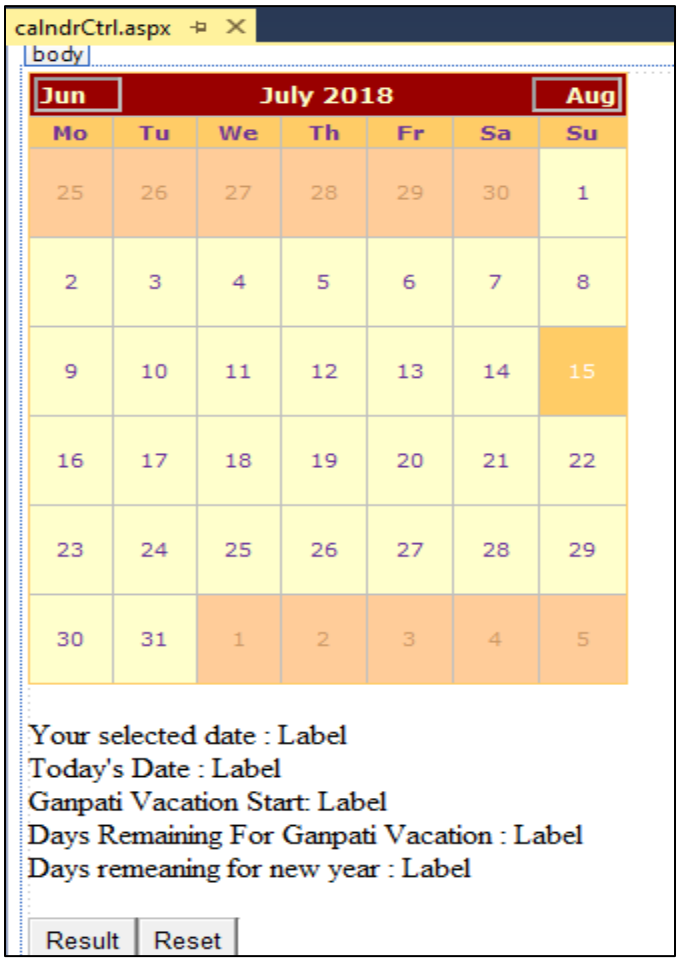
2018 – 2019

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

a) Display messages in a calendar control **b) Display vacation in a calendar control**

c) Selected day in a calendar control using style **d) Difference between two calendar dates**

<u>calndrCtrl.aspx</u>



Calender properties set for this example:

```
<asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC"
BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" Height="200px"
NextPrevFormat="ShortMonth" OnDayRender="Calendar1_DayRender"
ShowGridLines="True" Width="300px"
OnSelectionChanged="Calendar1_SelectionChanged" >

    <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
    <NextPrevStyle BorderStyle="Solid" BorderWidth="2px" Font-Size="9pt"
     ForeColor="#FFFFCC" />
    <OtherMonthDayStyle BackColor="#FFCC99" BorderStyle="Solid"
```

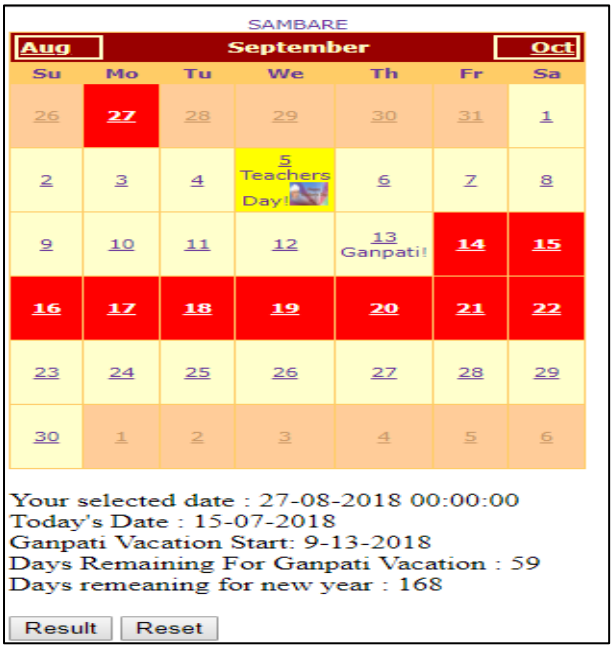
```
        ForeColor="#CC9966" />
      <SelectedDayStyle BackColor="Red" Font-Bold="True" />
      <SelectorStyle BackColor="#FFCC66" />
      <TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt"
                                  ForeColor="#FFFFCC" />
      <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
      <WeekendDayStyle Height="50px" />
    </asp:Calendar>
```

calndrCtrl.aspx.cs

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


    Label2.Text = "Todays Date"+Calendar1.TodaysDate.ToShortDateString();
    Label3.Text = "Ganpati Vacation Start: 9-13-2018";
    TimeSpan d = new DateTime(2018, 9, 13) - DateTime.Now;
    Label4.Text = "Days Remaining For Ganpati Vacation:"+d.Days.ToString();
    TimeSpan d1 = new DateTime(2018, 12, 31) - DateTime.Now;
    Label5.Text = "Days Remaining for New Year:"+d1.Days.ToString();
      if (Calendar1.SelectedDate.ToShortDateString() == "9-13-2018")
       Label3.Text = "<b>Ganpati Festival Start</b>";
      if (Calendar1.SelectedDate.ToShortDateString() == "9-23-2018")
       Label3.Text = "<b>Ganpati Festival End</b>";
  }


  protected void Calendar1_DayRender(object sender,
                System.Web.UI.WebControls.DayRenderEventArgs e)
  {
    if (e.Day.Date.Day == 5 && e.Day.Date.Month == 9)
    {
      e.Cell.BackColor = System.Drawing.Color.Yellow;
      Label lbl = new Label();
      lbl.Text = "<br>Teachers Day!";
      e.Cell.Controls.Add(lbl);
      Image g1 = new Image();
      g1.ImageUrl = "td.jpg";
      g1.Height = 20;
      g1.Width = 20;
      e.Cell.Controls.Add(g1);
```

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


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

OUTPUT



Practical 3(c).Demonstrate the use of Treeview control perform following operations.
a) Treeview control and datalist **b) Treeview operations**

Add XML File

Website -> Add -> XML File and Name it 'stdetail'.

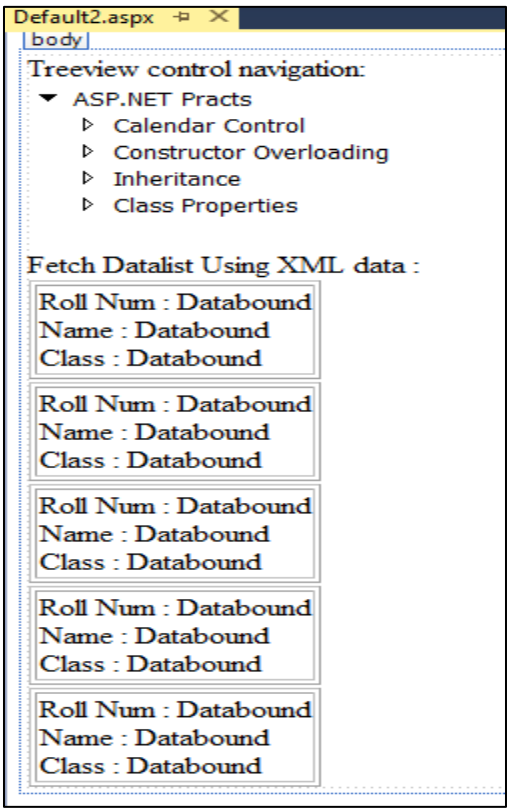
stdetail.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<studentdetail>
 <student>
  <sid>1</sid>
  <sname>Tushar</sname>
  <sclass>TYIT</sclass>
 </student>
 <student>
  <sid>2</sid>
  <sname>Sonali</sname>
  <sclass>TYCS</sclass>
 </student>
<student>
  <sid>3</sid>
  <sname>Yashashree</sname>
  <sclass>TYIT</sclass>
 </student>
<student>
  <sid>4</sid>
  <sname>Vedshree</sname>
  <sclass>TYCS</sclass>
 </student>
</studentdetail>
```

Default2.aspx

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

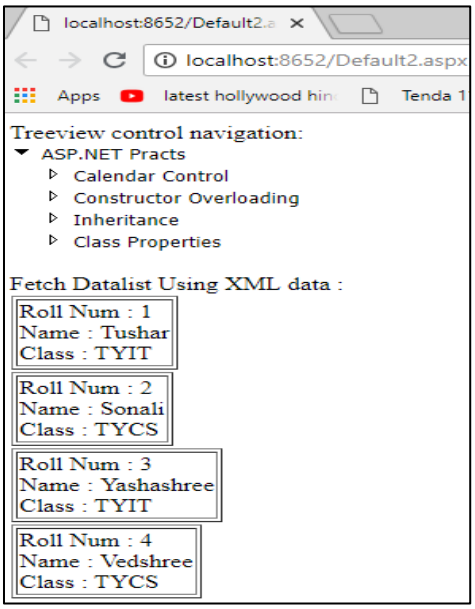
```
</asp:TreeNode>
</Nodes>
    <NodeStyle Font-Names="Tahoma" Font-Size="10pt" ForeColor="Black"
HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />
    <ParentNodeStyle Font-Bold="False" />
    <SelectedNodeStyle Font-Underline="True" ForeColor="#5555DD"
HorizontalPadding="0px" VerticalPadding="0px"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



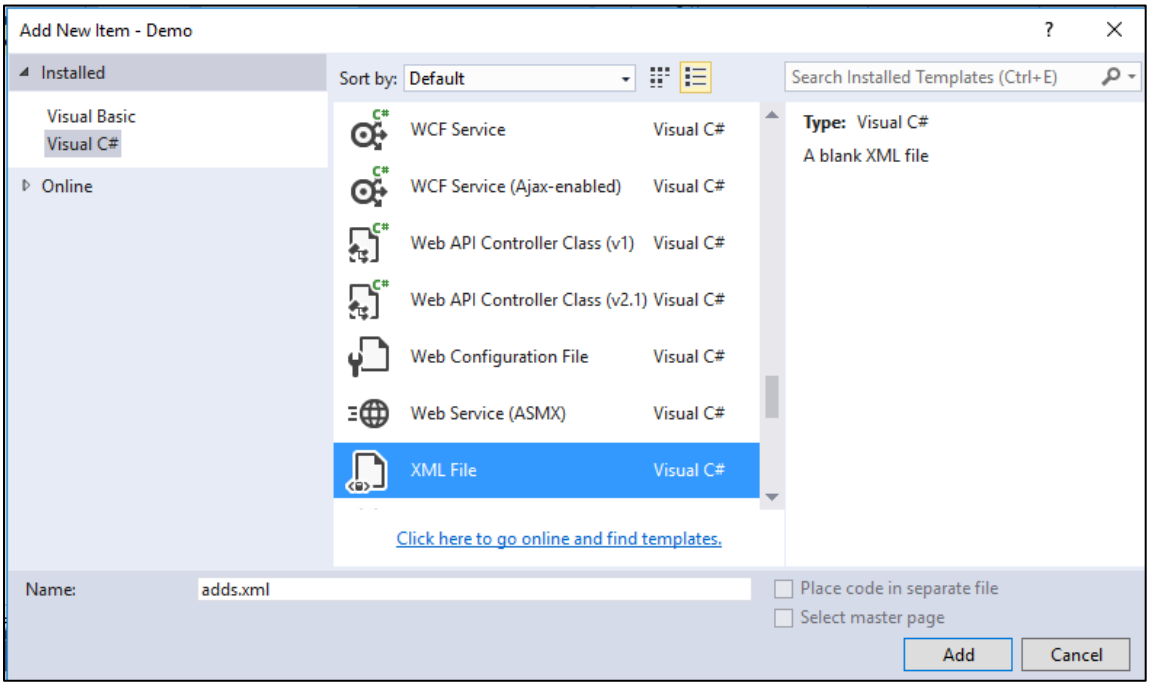
Default2.aspx.cs

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

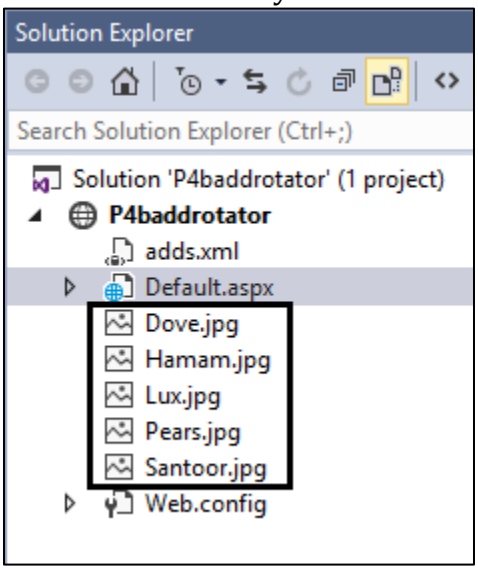
OUTPUT



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



Add images to test out the adrotator functionality.



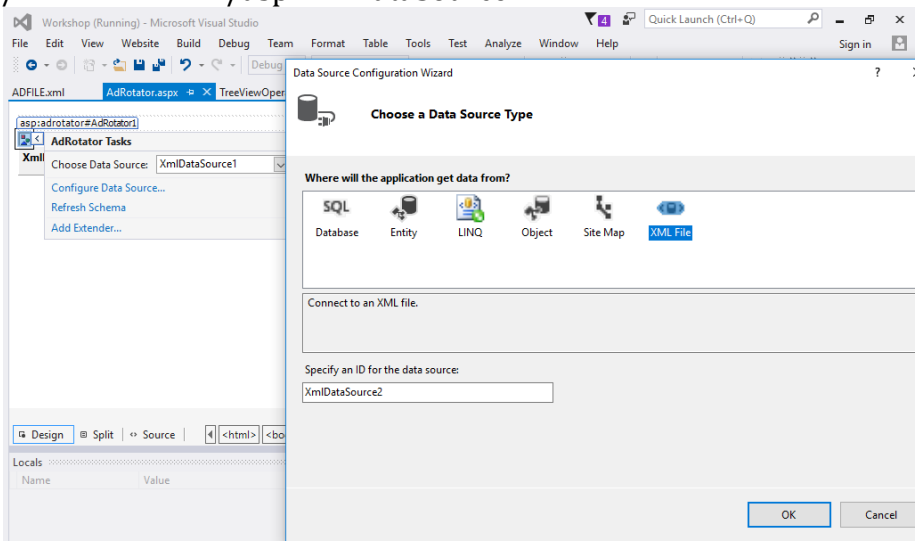
XML File

```
<Advertisements>
<Ad>
 <ImageUrl>rose1.jpg</ImageUrl>
 <NavigateUrl>http://www.1800flowers.com</NavigateUrl>
 <AlternateText>
```

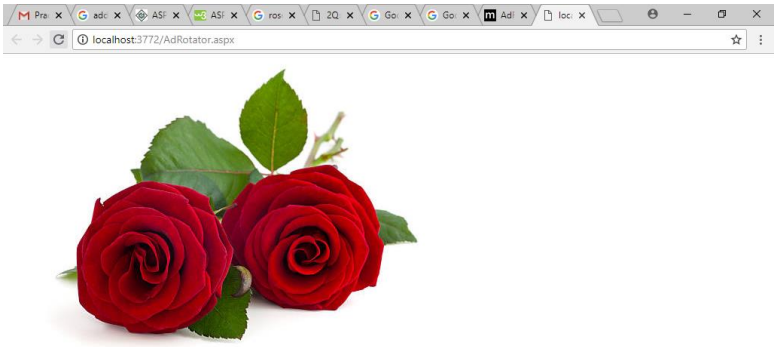
```
   Order flowers, roses, gifts and more
  </AlternateText>
  <Impressions>20</Impressions>
  <Keyword>flowers</Keyword>
</Ad>

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

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

</Advertisements>
```

Default.aspx

```
<asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1" />
      <asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/ADFILE.xml"></asp:XmlDataSource>
```



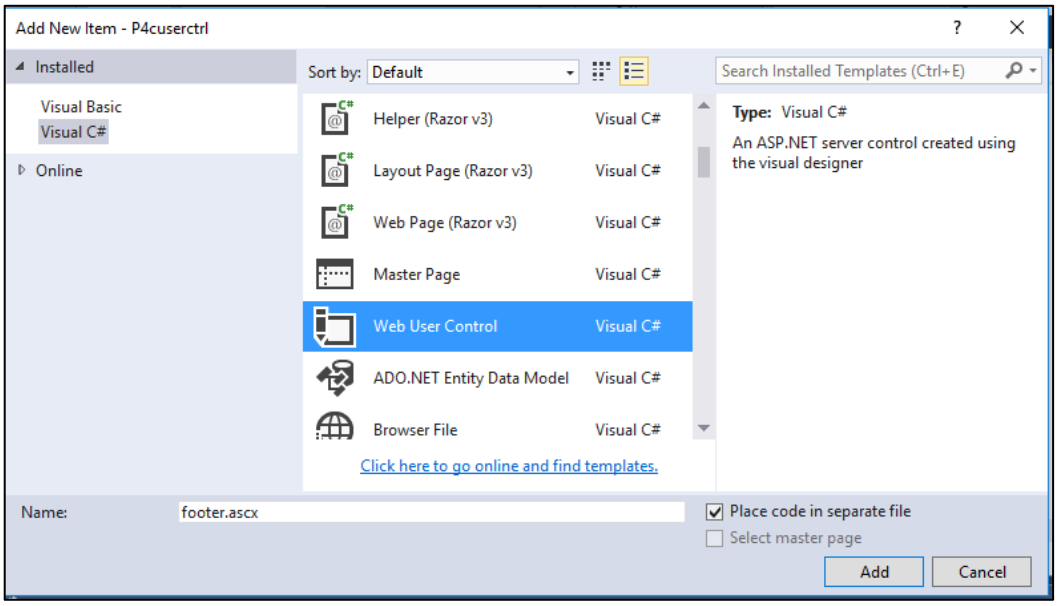
OUTPUT:



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

Add Web User Control

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



MyUserControl.ascx

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

<tr>
<td>Name</td>
<td>

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

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

<td>
  <asp:Button ID="txtSave" runat="server" Text="Save" onclick="txtSave_Click" />
</td>
</tr>
</table><br />
 <asp:Label ID="Label1" runat="server" ForeColor="White" Text=" "></asp:Label>
```

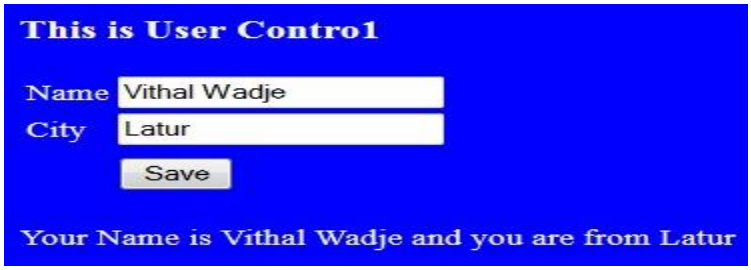
MyUserControl.ascx.cs

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

UserControlDisplay**.aspx**

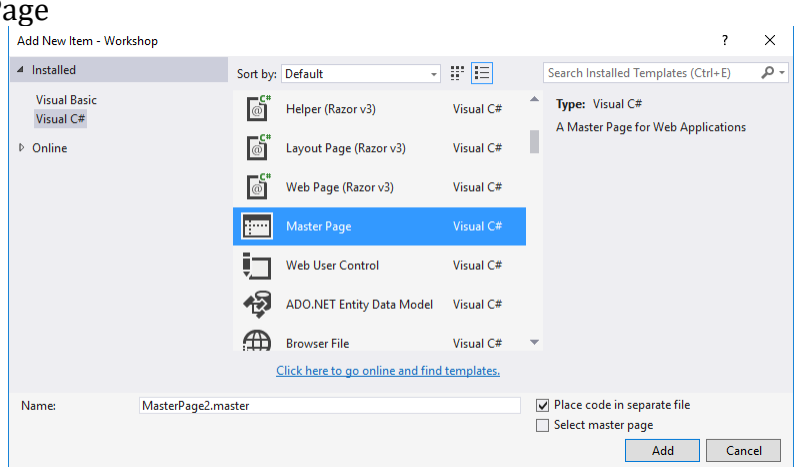
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UserControlDisplay.aspx.cs" Inherits="UserControlDisplay" %>
<%@ Register Src="~/MyUserControl.ascx" TagPrefix="uc"
TagName="Student"%>
<!DOCTYPE html>

<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 :

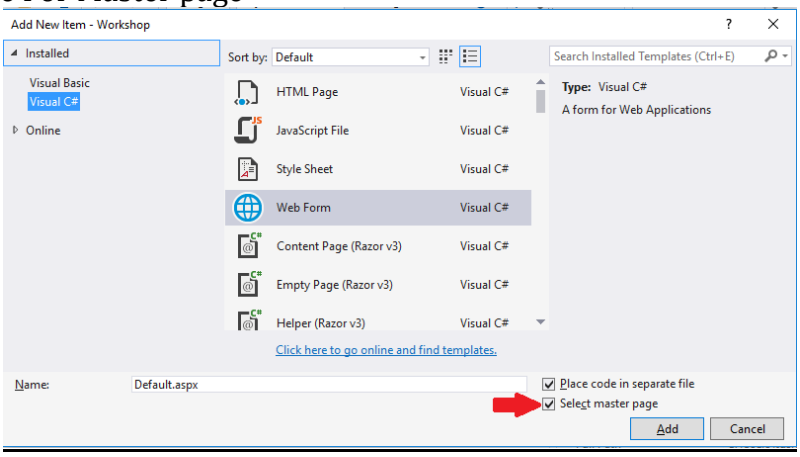


Practical 5(b).Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification.

Adding Master Page



Adding Web page For Master page



MasterPage.master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```
    <title>Master Page Demo</title>
    <link href="css/my.css" rel="stylesheet" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <style type="text/css">
        .auto-style1 {
            position: absolute;
            top: 373px;
            left: 1028px;
            bottom: 303px;
        }
        .auto-style2 {
            position: absolute;
            top: 537px;
            left: 1016px;
            z-index: 1;
        }
    </style>
</head>
<body>
    <!DOCTYPE html>
    <form id="form1" runat="server">

<html>
<head>
    <title>Master</title>
    <link rel="stylesheet" type="text/css" href="StyleSheet.css">
</head>
<body>
<header id="header">
<h1>Demo Of Master Page</h1>
</header>
<nav id="nav">
    <ul>
        <li><a href="home.aspx">Insight</a></li>
        <li><a href="#">Products</a></li>
        <li><a href="#">Downloads</a></li>
        <li><a href="#">Contact Us</a></li>
    </ul>
</nav>
<aside id="side">
    <h1>Info</h1>
    <a href="#"><p>Product Type 1</p></a>
    <a href="#"><p>Product Type 2</p></a>
    <a href="#"><p>Product Type 3<a href="#"><asp:ScriptManager ID="ScriptManager1"
runat="server">
        </asp:ScriptManager>
        </a>
    </p>
    <asp:Button ID="Button2" runat="server" CssClass="auto-style1" style="z-index: 1"
Text="Button" />
    <asp:Button ID="Button1" runat="server" CssClass="auto-style2" Text="Button" />


</aside>


    <div id="con">
```

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

        </asp:ContentPlaceHolder>
    </div>


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

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

MasterDisplay.aspx

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

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <h1>Home page</h1>
</asp:Content>
```

StyleSheet.css

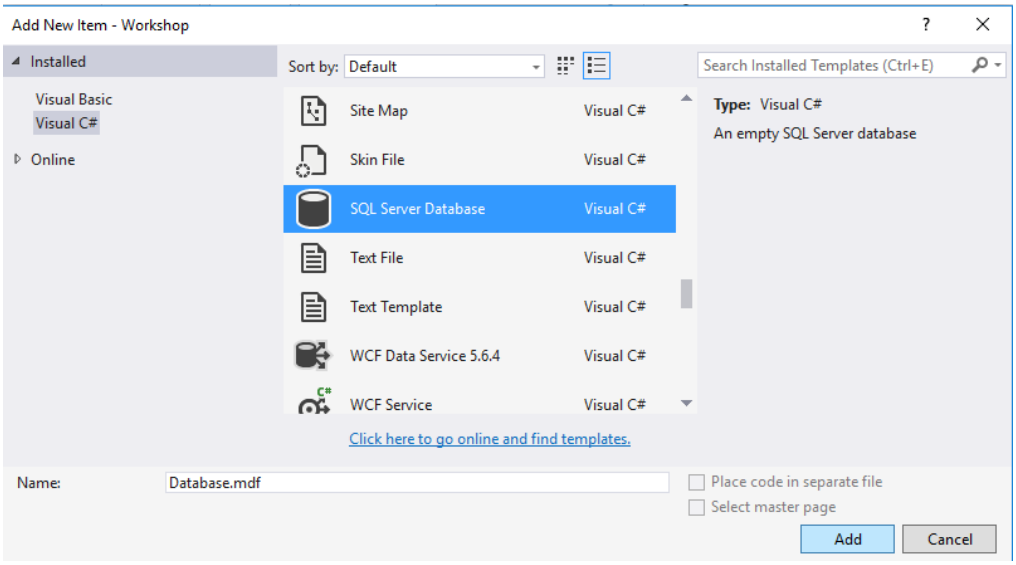
```
#header{
    color: blueviolet;
    text-align: center;
    font-size: 20px;
}
#nav{
    background-color:darkseagreen;
    padding: 5px;
}
ul{

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

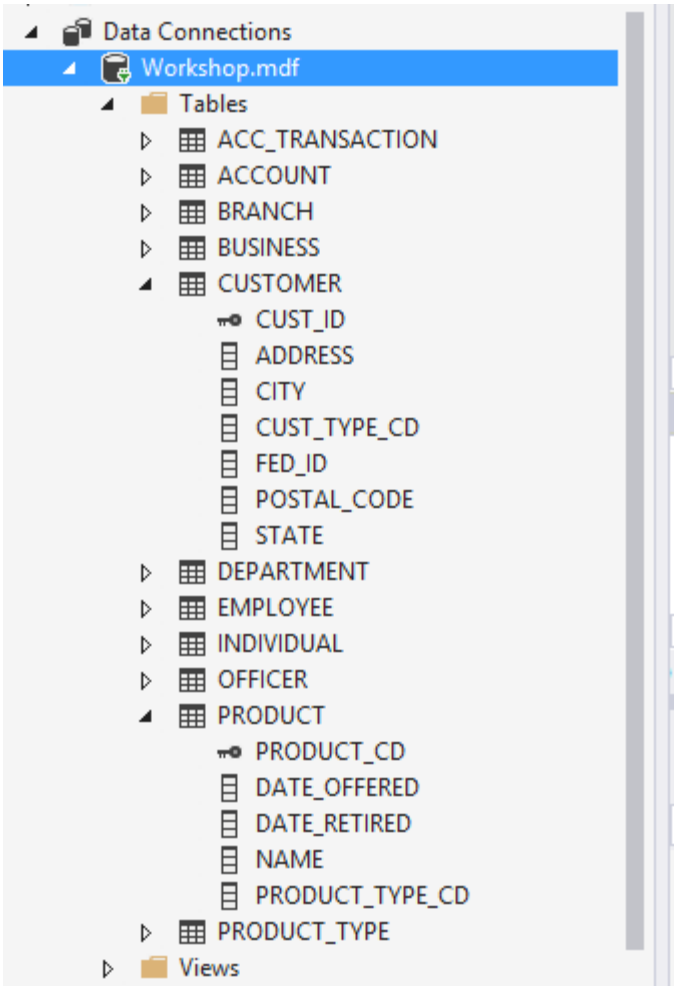
```
    li a:hover{
     background-color: aqua;
     color:coral ;
     padding:1%;
    }
    #side{
     text-align: center;
     float: right;
     width: 15%;
     padding-bottom: 79%;
     background-color: #F1FAEE;
    }
    #article{
     background-color: burlywood;
     padding: 10px;
     padding-bottom: 75%;
    }
    #footer{
     background-color: #C7EFCF;
     text-align:center;
     padding-bottom: 5%;
     font-size: 20px;
}
    #con{
        border:double;
        border-color:burlywood;
    }
```

Database Practicals*******

Note: For Database practical's we have used SQL Server 2014 version.

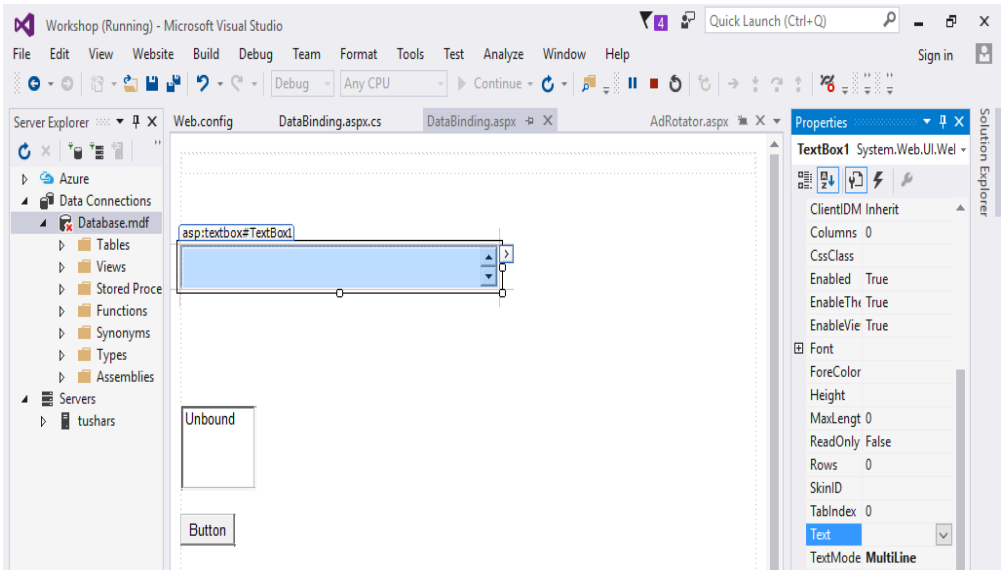
Here we to add new database in our website, as shown below. Add this database inside App_Data folder.





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

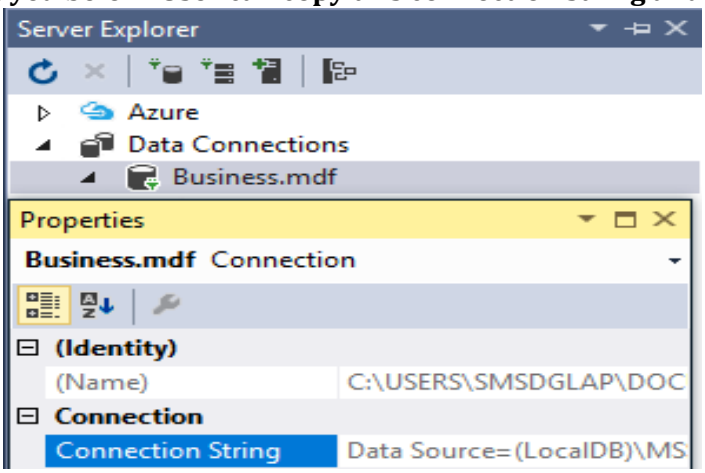
1. Create a webpage with one **Button,** one Multiline TextBox and one list box with setting **TextMode** Property of text box to **Multiline** as shown below.



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

Note: **The users have to use their own system connection string in place of connection string given in following code.**

The connection string is available in Server Explorer (Right click on Database Name and Select Properties) as displayed below. User can copy this connection string and can use in code.



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

Web.confing

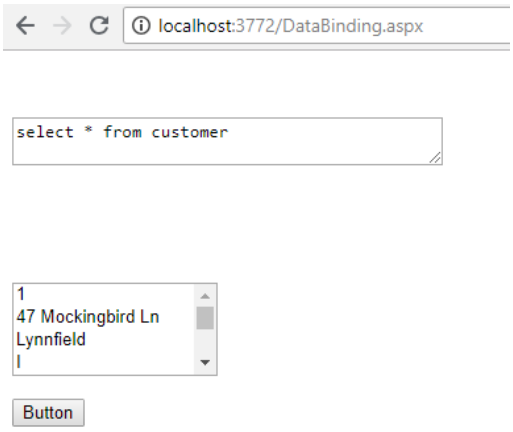
```
<configuration>

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

```
    </connectionStrings>

</configuration>
```

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

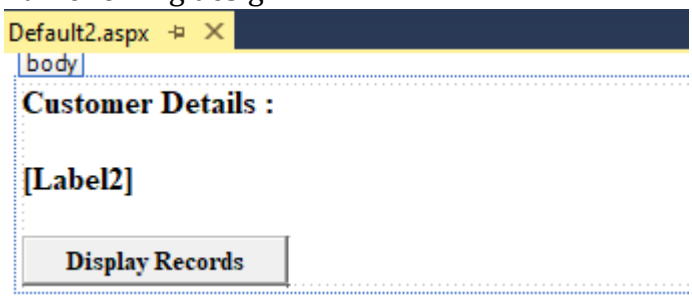
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;


public partial class DataBinding : System.Web.UI.Page
{
    protected void 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();

    }
}
```

Output:



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

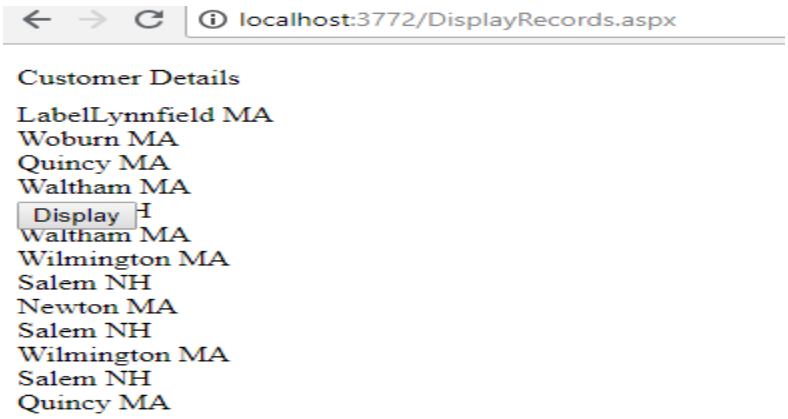
Create a web page with following design:



Add the following code on Button click event in C# Code behind file.

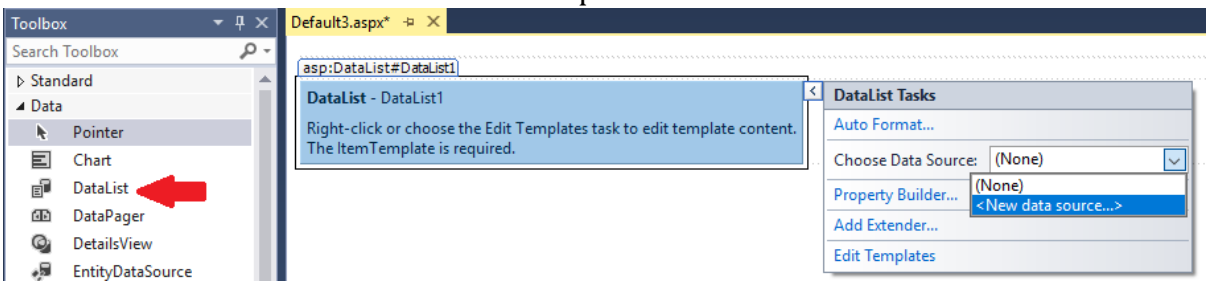
```
protected void Button1_Click(object sender, EventArgs e)
    {
        string connStr =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
        SqlConnection con = new SqlConnection(connStr);
        SqlCommand cmd = new SqlCommand("Select City, State from Customer", con);
  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

        while (reader.Read())
        {


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

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

    }
```

Output:

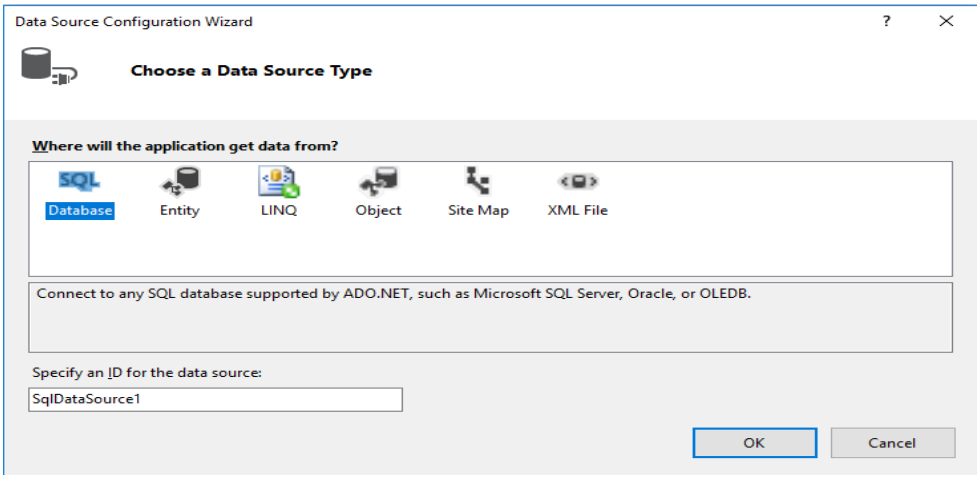


Practical 6 (c): Demonstrate the use of Datalist link control.

1. Drag the Datalist control to our web page form toolbox->Data-> Datalist.
2. Then select **Choose Data Source** Option and select **<New Data Source>.**

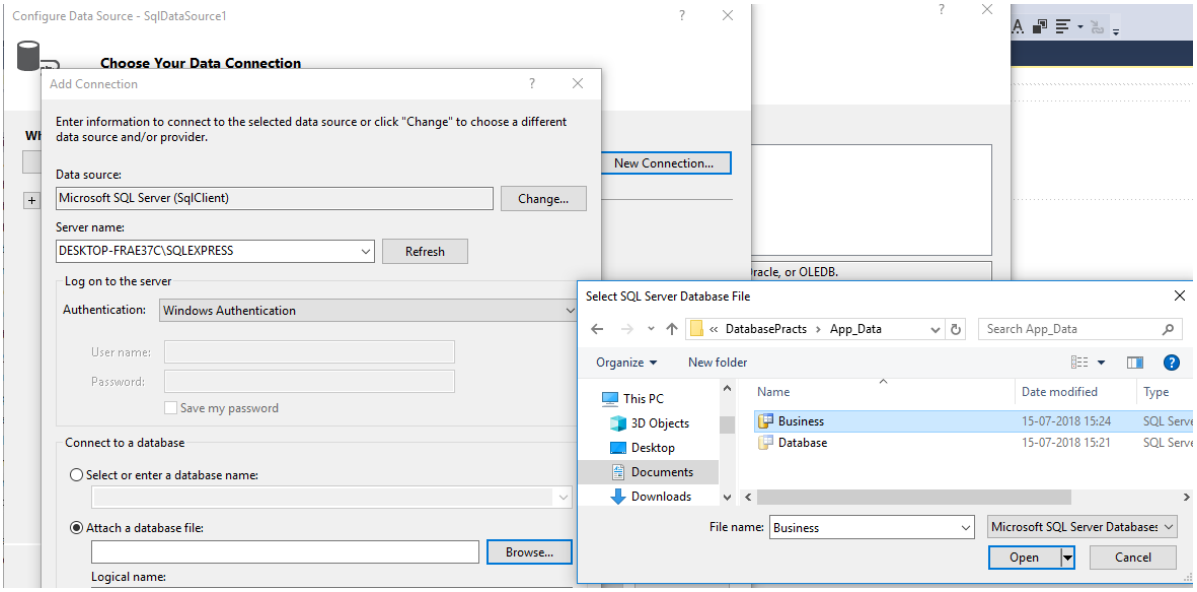


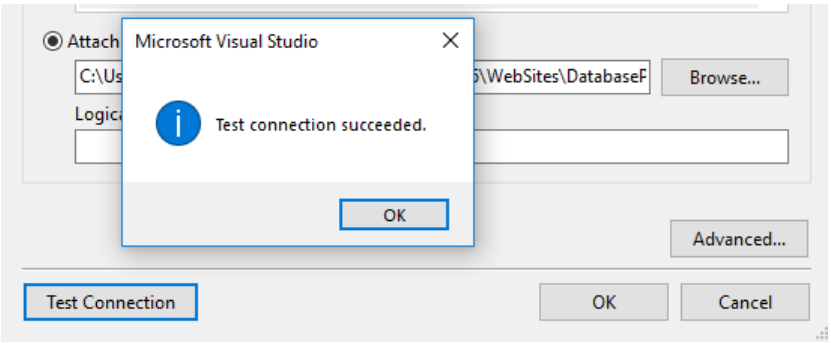
3. Now Select SQL Database from options and Click Ok button.



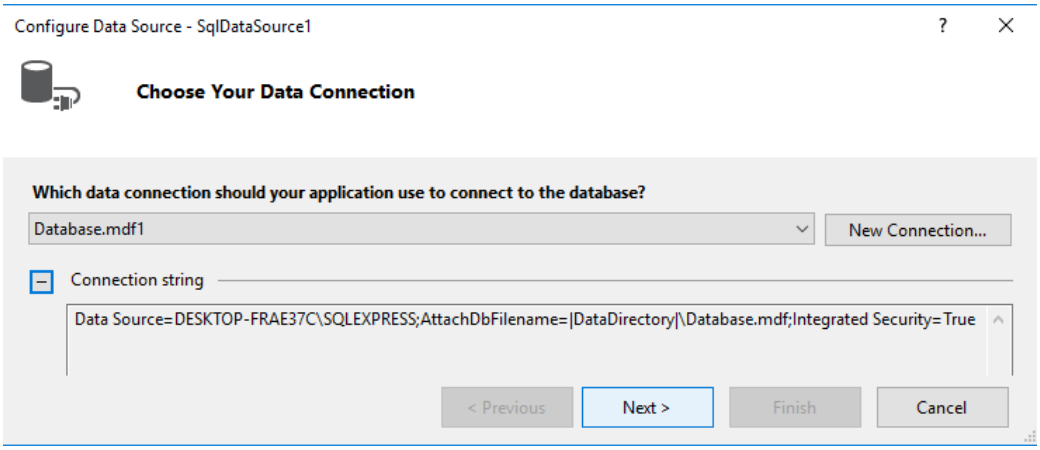
4. In next window click on New Connection button.
5. In add connection window Select the available SQL Server Name
6. Keep the Authentication as Windows Authentication.

7. After that select Attach a Database file radio button. Here we have to select the database that we have created in our application. (Usually it will be in Documents folder under Visual Studio 2015/ Websites).
8. After selection of Database file. We can also Test the connection.
9. Then Click on OK button.

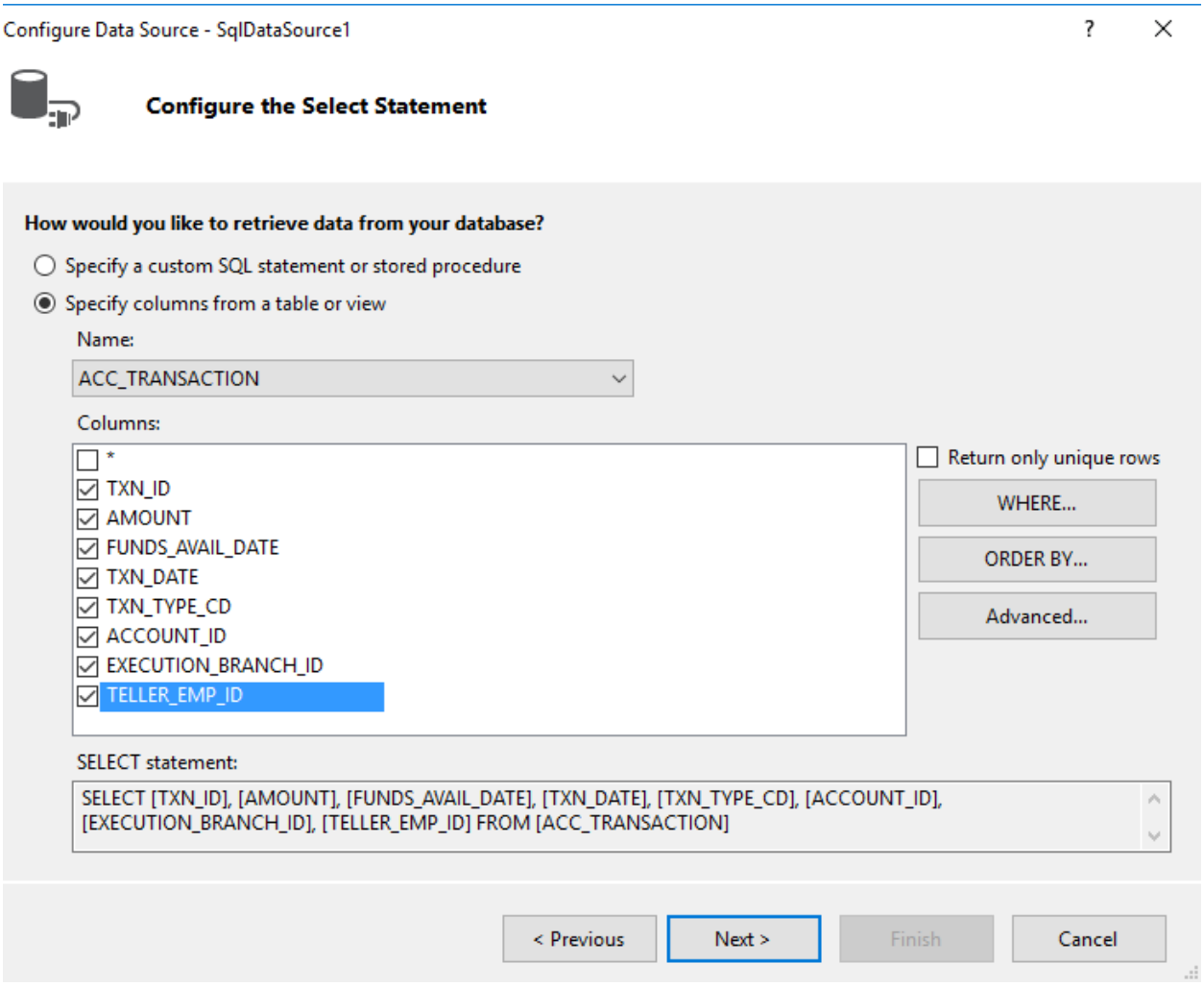




10. Once the Connection is made then click on Next button from Data Source Wizard.

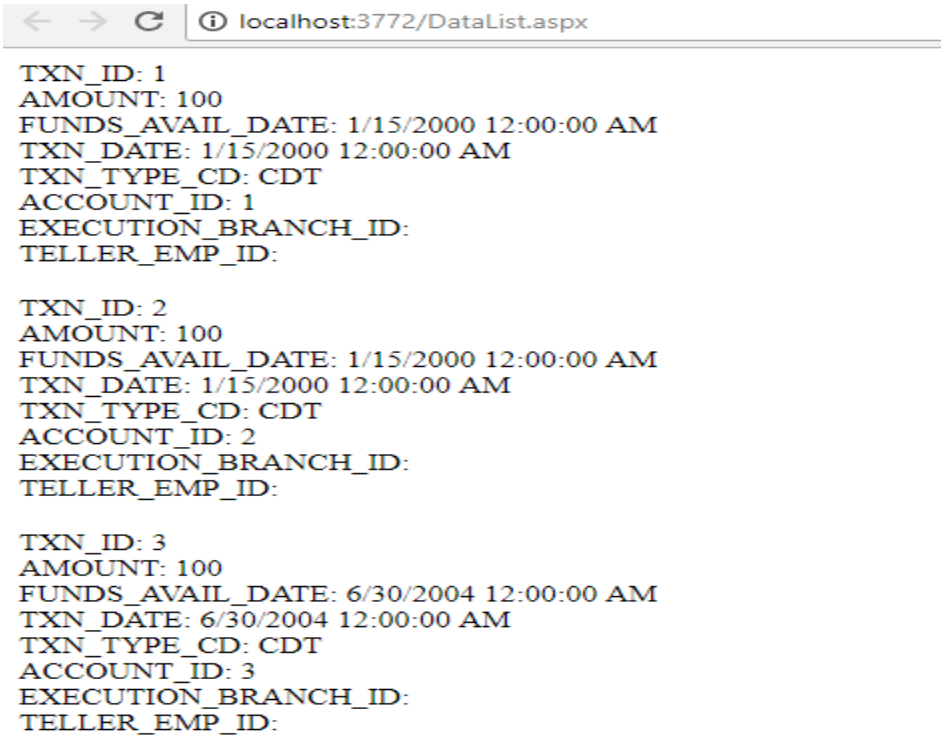


11. Then wizard ask for saving the connection string in configuration file. If you already stored it web.config file then uncheck check box, if you haven't, then select the checkbook. Then click on next button.
12. The next screen gives option to configure the select statement. Here we can choose the table as well as configure the select statement as we need to display the data on web page.



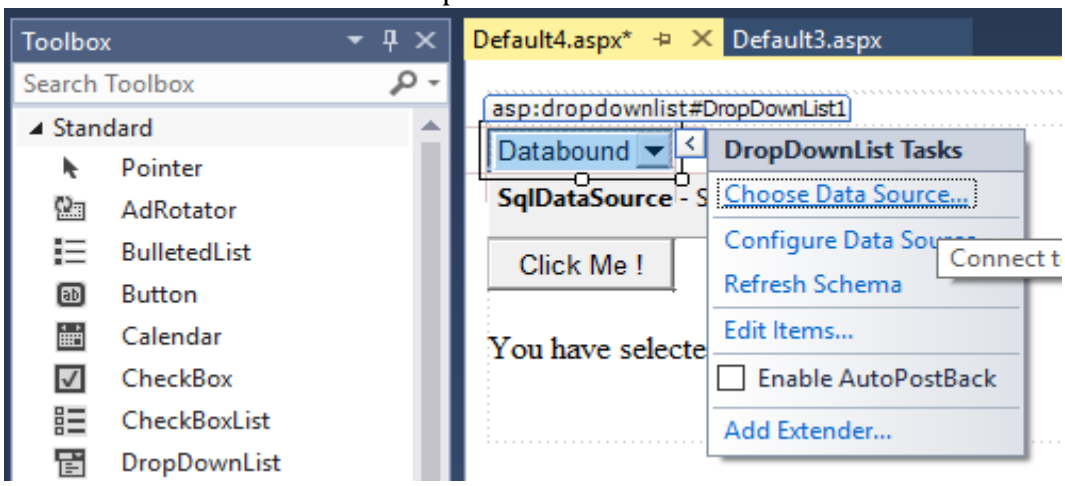
13. In next screen we can test our query to check the output. Then Click on finish.

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

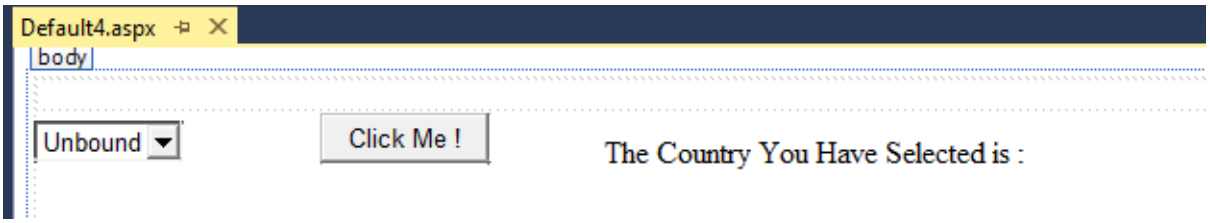


Practical 7 (a): Create a web application to display Databinding using Dropdownlist control.

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



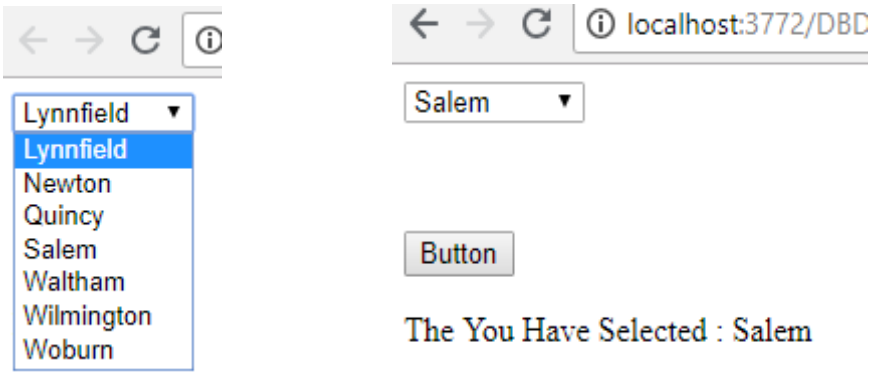
Or with Code also we can achieve the same thing.



Code of C# Code behind file

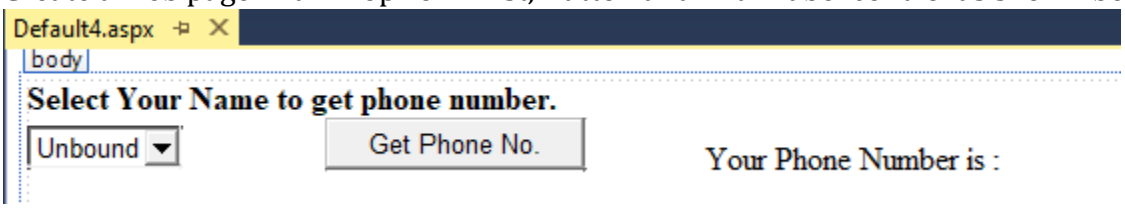
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;


public partial class DBDropDown : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            string connStr =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
            SqlConnection con = new SqlConnection(connStr);
            SqlCommand cmd = new SqlCommand("Select Distinct City from Customer", con);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            DropDownList1.DataSource = reader;
            DropDownList1.DataTextField = "City";
            DropDownList1.DataBind();
            reader.Close();
            con.Close();
        }

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = "The You Have Selected : " + DropDownList1.SelectedValue;
    }
}
```

Output:



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

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

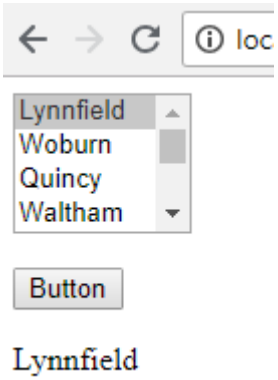


Code of C# Code behind file

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

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


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

    }

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

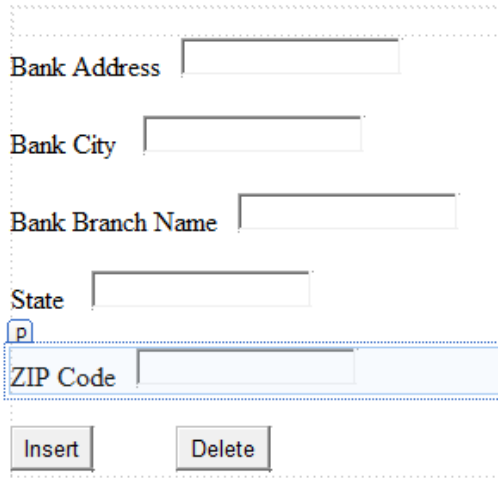
```
            SqlDataReader reader = cmd.ExecuteReader();
            ListBox1.DataSource = reader;
            ListBox1.DataTextField = "City";
            ListBox1.DataValueField = "POSTAL_CODE";
            ListBox1.DataBind();

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

        }
    }
}
```

Output:



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

Create a web page with TextBox, and Two Button and one Label control as shown below. And follow the database related steps same as it is in previous examples.



Code of C# Code behind file

```
using System;
using System.Collections.Generic;
```

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


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

   }
}
```