

Unit -2 Information Passing and Standard Controls**2.1 PASSING INFORMATION FROM ONE PAGE TO ANOTHER:**

ASP.NET is a server-side technology. Here, browser (client) and server are communicating with each other using HTTP (HyperText Transfer Protocol). HTTP is a stateless protocol. Server is not maintaining persistence connectivity with the Client (Browser). Whenever browser is making a request for a particular page, server behave like a stranger as it do not maintain persistent connectivity with the browser.

http server needs not keep track of any state information so that's why its stateless protocol

As server is not maintaining persistence connectivity with the browser, it is difficult to pass the information from one page to another page. There are couple of ways in the ASP.NET, by using that we can pass the information from one page to another.

[1] Previouspage.FindControl() method:

We can pass the information from one page to another page using Findcontrol() method of the Previouspage object. In your website add two pages called default.aspx and default2.aspx. In the default.aspx page add a textbox control and a button control. Set the PostbackUrl property of the button to 'Default2.aspx'. Now in the second page (Default2.aspx) place a label control. Write the following code on the load event of the Default2.aspx.

```
TextBox t1;
t1 =
(TextBox)PreviousPage.FindControl("Textbox1");
Label1.Text = t1.Text;
```

Run the Default.aspx page, type your name in the Textbox and click on the button. You will be redirected to the Default2.aspx, and can notice your name will appear in the Label control. In this example we have used a method called Findcontrol () of the Previouspage object.

[2] Cookies:

We can also pass the information from one page to another page using cookies. A cookie is information that a Web site puts on your hard disk so that it can remember something about you at a later time. (More technically, it is information for future use that is stored by the server on the client side of a client/server communication).

The following code will create the cookies called ProductName, and store the details entered by the user in the textbox.

```
HttpCookie ck = new HttpCookie("ProductName");
ck.Value = TextBox1.Text;
ck.Expires = DateTime.Now.AddDays(2);
Response.Cookies.Add(ck);
Response.Redirect("Default2.aspx");
```

In the code mentioned above, we set the expiry of the cookies for 2 days. That means the ProductName cookies created by this code will be automatically deleted after 2 days. The last line of the code will redirect the user to the Default2.aspx webpage. To fetch the data from the ProductName cookies on default2.aspx page write the following code on the load event of the Default2.aspx page.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Request.Cookies["ProductName"] == null)
    {
        Label1.Text =
Request.Cookies["ProductName"].Value.ToString();
    }
    else
    {
        Response.Write("Cookies Not Found");
    }
}
```

In this code we check the "ProductName" cookie exists or not. If it exists then we are reading the cookie, fetching the details stored in it and displaying it in the label. If cookie with the name 'ProductName' doesn't exist then we put the message that the "Cookies Not Found".

[3] QueryString:

QueryString is another method to pass the information from one page to another page. It is the simplest method where information of one page will be passed to another page using URL (Uniform Resource Locator) of the browser. In the World Wide Web, query string is the part of a URL that contains data to be passed to web application such as CGI programs. To pass the information from one page to another page using computer generated imagery QueryString method you need to write following code on the click event of the button.

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{  
    Response.Redirect("Default2.aspx?ProductName=" +  
    TextBox1.Text);  
}
```

This code will put the value entered by the user in variable ProductName in the URL and redirect user to the Default2.aspx. On the load event of the Default2.aspx page write the following code to fetch value from the ProductName variable in the URL and print it into the Label control.

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Label1.Text = Request.QueryString["ProductName"];  
}
```

The limitations of this methods are:

- The information passed from one page to another is visible to the user in the URL.
- This approach is not suitable to pass longer information from one page to another as the size of the URL should not be exceeded by 255 characters.

[4] Session variables:

Session variable is another method to pass the information from one page to another page. The information stored in the session variables will remain as it is till the session is active. Once the session is terminated by the server (when user logout) the information stored in the session variables will be destroyed.

To create the session variable named ProductName write the following code on the click event of the button.

```
protected void Button1_Click(object sender, EventArgs e)  
{  
    Session["ProductName"] = TextBox1.Text;  
    Response.Redirect("Default2.aspx");  
}
```

The code written above will store the information entered by the user in the Textbox1 in the session variable named ProductName and redirect the user to the Default2.aspx. On the load event of Default2.aspx write the following code to fetch the value stored in the ProductName variable of the session and display it in the Label control.

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Label1.Text = (string)Session["ProductName"];  
}
```

[5] Application variables:

Application variable is similar to the session variable, just the difference is session variables are separate for each session (user). One user cannot see the information stored in the session of other users. Whereas application variables are used to share the information among user. Application variables do not create its user wise copies. There is a single copy of the application variable is created, which is shared among all the users. The life of the application variable is longer than the session variables. They are created when application starts and will remain in the memory of the server till the application is live. While the session variables are created when the session starts and destroyed automatically when the session ends.

To store the information into the Application variable you need to write the following code:

```
Application["ProductName"] = TextBox1.Text;
```

And to read the information stored in the application variables, you need to write the following code:

```
Label1.Text = (string)Application["ProductName"];
```

2.2 Web Server Controls Button:

There three button controls are there in the ASP.NET, Button, LinkButton and ImageButton. The difference between Button and LinkButton is Button looks like a push button whereas LinkButton looks like a Hypelink at runtime. All three controls share the common property as described below:

- 1. OnClick:** This attribute takes the name of the subroutine created in either C#.NET language or in VB.NET language, and execute it when the user clicks on the button at runtime. As we know the code written in the C#.NET or VB.NET is executed by the server, the name specified of the subroutine in the Click property of the button will be executed by the server.
- 2. OnClientClick:** This attribute takes the name of the JavaScript function as a value. As we know that the JavaScript is client-side scripting language, the JavaScript function specified with this property will be executed by the browser, when user click on the button at runtime. At time of page rendering the JavaScript function will be given to the browser by the server.
- 3. PostBackUrl:** This property takes the name of the webpage. When user clicks on the button at runtime, the user will be redirected to the page specified in this property.

Compare to the Button control, ImageButton control has additional capability to show the image on the button. Because of ImageButton has additional feature to show the images, there are few more attributes are there to ImageButton control than Button and LinkButton controls. The additional attributes of the ImageButton are discussed below:

1. **ImageUrl:** ImageUrl property takes the path of the Image, which has to be shown on the ImageButton control.
2. **AlternateText:** AlternateText property takes the text (string), and it will be shown at runtime if the browser is not be able to show the image specified in the ImageUrl property.

2.3 Hyperlink control:

Hyperlink control is used to show hyperlink which provides a link to the other webpages. Hyperlink control does not belong to Button family, therefore in the hyperlink control does not have PostBackUrl property. The important properties of the Hyperlink control are:

1. **NavigateUrl:** This property takes the path (URL) of the destination page. At runtime when user clicks on the Hyperlink control, user will be redirected to the page specified in the NavigateUrl property.
2. **Target:** The possible values of this property are _blank, _parent, _search, _self, and _top. This property specifies where the destination page has to be opened. For example, if you set the Target property of the Hyperlink control to _blank, then the destination page will be opened in the new window / new tab of the browser.

2.4 Imagemap control:

The ImageMap controls enables you to create a client-side image map. When you click different areas of the image, different things happen. An ImageMap control is composed out of instances of the HotSpot class. A HotSpot defines the clickable regions in an image map. Imagemap control has following important properties:

1. **ImageUrl:** This property takes the path of the image, which is to be used to show on the webpage.
2. **Hotspots:** This property takes collection of the Hotspot regions. Hotspot are the clickable region in the image which navigate user to specific webpage. There are three types of Hotspot we can create in the Imagemap control of ASP.NET, these are:
 - CircleHotSpot
 - RectangleHotSpot
 - PolygonHotSpot

2.5 Checkbox control:

Checkbox control is used to take Boolean (TRUE / FALSE) value from the user. Usually it is used to allow multiple selections by the user. For Example, if you want to take hobbies from the user, you cannot restrict the user to input only one value. User can have multiple interests; in such scenarios we can use checkbox server control. Important attributes of the checkbox server controls are listed below:

- AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is 'FALSE'. If you set this attribute to be 'TRUE' then the page will be immediately transfer to the server, when user check or uncheck the checkbox.
- Text:** Text property accepts text (string) from the user. The string given in this property will be appear on the checkbox.
- TextAlign:** The possible values of this attributes are Right and Left. The default value for this attribute is Right, that means the text will appear at right hand side and checkbox will appear at left hand side. You may change it Left which shows text at left side and checkbox at right.
- Checked:** This property gets / set the checkbox is checked or not. It takes Boolean value from the user. The default value is FALSE. If the checkbox is checked then this attribute returns TRUE value.

2.6 RadioButton Control:

RadioButton control is similar to the Checkbox control, except it is used to allow user to select only one option from the given options. RadioButton control has following attributes:

- AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is 'FALSE'. If you set this attribute to be 'TRUE' then the page will be immediately transfer to the server, when user check or uncheck the checkbox.
- Text:** Text property accepts text (string) from the user. The string given in this property will be appear on the checkbox.
- TextAlign:** The possible values of this attributes are Right and Left. The default value for this attribute is Right, that means the text will appear at right hand side and checkbox will appear at left hand side. You may change it Left which shows text at left side and checkbox at right.
- Checked:** This property gets / set the checkbox is checked or not. It takes Boolean value from the user. The default value is FALSE. If the checkbox is checked then this attribute returns TRUE value.
- GroupName:** This property takes string from the user. User can select only one RadioButtons from more than one RadioButtons having same group name. For

Example, user can select only one RadioButton from two RadioButtons (Male, FeMale) if their GroupName is same called ‘Gender’.

2.7 CheckBoxList Control:

CheckBoxList server control is a collection of various checkboxes. It is a group of multiple checkboxes. User is allowed to check multiple checkboxes from the CheckBoxList server control. CheckBoxList server control allow to insert the items statically (click on edit item and then add items), dynamically (by writing the code) and from Database. Some important attributes of CheckBoxList server control are listed below.

1. **AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is ‘FALSE’. If you set this attribute to be ‘TRUE’ then the page will be immediately transfer to the server, when user check or uncheck any checkbox from CheckBoxList control.
2. **DataSource:** DataSource property takes name of the data source, which will be used to fetch the data from the database. DataSource property is used to bind CheckBoxList server control with any database table or field.
3. **RepeatColumns:** RepeatColumns get or sets integer value. The default value for this attribute is 0. By default, all the checkboxes listed in the CheckBoxList server controls are belongs to a single column. If you set RepeatColumns attribute to 2 then all checkboxes are listed in 2 different columns.
4. **RepeatDirection:** If the values of RepeatColumns are 2 or more then this property provides a way of arrangement of checkboxes into the CheckBoxList control. The possible values are ‘Vertical’ and ‘Horizontal’. The default value for this attribute is ‘Vertical’.
5. **Items:** Items property represents collection of list items. Using this property, you can insert any ne list item into the CheckBoxList control or remove any list item from it.
6. **SelectedIndex:** This property gives Index number of the item selected by the user. It returns integer value.
7. **SelectedItem:** This property returns the object of ListItem, which is checked (selected) by the user.
8. **SelectedItems:** This property returns a collection (Array) of the objects of ListItem, which are checked (selected) by the user.

2.8 RadioButtonList Control:

RadioButtonList server control is a collection of various RadioButtons. It is a group of multiple RadioButtons. User is allowed to check single RadioButton from the

RadioButtonList server control. RadioButtonList server control allow to insert the items statically (click on edit item and then add items), dynamically (by writing the code) and from Database. Some important attributes of RadioButtonList server control are listed below.

1. **AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is 'FALSE'. If you set this attribute to be 'TRUE' then the page will be immediately transfer to the server, when user check or uncheck any RadioButton from the RadioButtonList control.
2. **DataSource:** DataSource property takes name of the data source, which will be used to fetch the data from the database. DataSource property is used to bind RadioButtonList server control with any database table or field.
3. **RepeatColumns:** RepeatColumns get or sets integer value. The default value for this attribute is 0. By default, all the RadioButtons listed in the RadioButtonList server controls are belongs to a single column. If you set RepeatColumns attribute to 2 then all RadioButtons are listed in 2 different columns.
4. **RepeatDirection:** If the values of RepeatColumns are 2 or more then this property provides a way of arrangement of RadioButtons into the RadioButtonList control. The possible values are 'Vertical' and 'Horizontal'. The default value for this attribute is 'Vertical'.
5. **Items:** Items property represents collection of list items. Using this property, you can insert any ne list item into the RadioButtonList control or remove any list item from it.
6. **SelectedIndex:** This property gives Index number of the item selected by the user. It returns integer value.
7. **SelectedItem:** This property returns the object of ListItem, which is checked (selected) by the user.
8. **SelectedItems:** This property returns a collection (Array) of the objects of ListItem, which are checked (selected) by the user.

2.9 ListBox Control:

ListBox server control is a collection of various ListItem objects. It is a group of multiple ListItems. User is allowed to check single item or multiple items (by setting property SelectionMode to 'Multiple') from the ListBox server control. ListBox server control allow to insert the items Statically (click on edit item and then add items), Dynamically (by writing the code) and from Database. ListBox shows multiple option at a time. If the number of items is more than its width then it provides vertical scrollbar. Some important attributes of ListBox server control are listed below.

1. **AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is 'FALSE'. If you set this attribute to be 'TRUE'

then the page will be immediately transfer to the server, when user select or unselect any item from the ListBox control.

2. **DataSource:** DataSource property takes name of the data source, which will be used to fetch the data from the database. DataSource property is used to bind ListBox server control with any database table or field.
3. **Items:** Items property represents collection of list items. Using this property, you can insert any ne list item into the ListBox control or remove any list item from it.
4. **SelectionMode:** The possible values for this property are 'Single' and 'Multiple'. Default value for this attribute is 'Single', which means user is allowed to select only one item from the ListBox, if you change it to 'Multiple' then user can select multiple items from the ListBox.

2.10 DropDownList Control:

DropDownList server control is a collection of various **ListItem objects**. It is a group of multiple ListItems. User is allowed to select only single item from the DropDownList server control. DropDownList server control allow to insert the items statically (click on edit item and then add items), dynamically (by writing the code) and from Database. DropDownList shows single item at a time, to view multiple items user needs to pull down it. Some important attributes of DropDownList server control are listed below.

1. **AutopostBack:** AutopostBack property takes Boolean value from the user. The default value for this attribute is 'FALSE'. If you set this attribute to be 'TRUE' then the page will be immediately transfer to the server, when user change item from the DropDownList control.
2. **DataSource:** DataSource property takes name of the data source, which will be used to fetch the data from the database. DataSource property is used to bind DropDownList server control with any database table or field.
3. **Items:** Items property represents collection of list items. Using this property, you can insert any ne list item into the DropDownList control or remove any list item from it.

2.11 FileUpload Control:

FileUpload control allow user to upload any file available in the client computer to the server. With the FileUpload control user can upload any type of file to the server. To allow user so that, user can upload the file you need to place a FileUpload control and a Button. You need to write the following code on the click event of the Button.

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (FileUpload1.HasFile)
```

```
{  
FileUpload1.SaveAs(Server.MapPath("~/UploadImages/" +  
FileUpload1.FileName));  
}  
}
```

- **Properties:**

1. **HasFile:** This property of the FileUpload control returns Boolean value. If the particular file is selected by the user by clicking on the 'Browse' button then it will return TRUE, otherwise it will return FALSE.
2. **FileName:** FileName property returns the name of the file selected by the user, using FileUpload control.

- **Methods:**

SaveAs: This method will upload the file from client machine to server. In the code described above, the file will be uploaded in the UploadImages directory created in the website on the server. SaveAs method needs absolute path the file to be uploaded, so to convert the relative path to the Absolute path we have used a function called Server.MapPath().

2.12 AdRotator Control:

AdRotator control is used to show advertisements on the webpage (.aspx page). We need to provide multiple advertisements to the AdRotator control by giving XML file or Database as a source of advertisements. From the collection of multiple advertisements AdRotator control will pick up any one random advertisement and shows to the user at time of loading of a webpage. As discussed, there are two methods are there to provide advertisements to the AdRotator control, first method is Database. Create a table called Advertisements with the fields called ImageUrl, NavigateUrl, Keyword and AlternateText. Do the sufficient entry into the table and using sqlDataSource fetch the details to the table? In this case we need to set DataSource property to sqlDataSource. In the second approach, you need to make an XML file to store the advertisements as shown below:

```
<?xml version="1.0" encoding="utf-8" ?>  
<Advertisements>  
  <Ad>  
    <ImageUrl>~/AdImages/BannerAd1.gif</ImageUrl>  
    <NavigateUrl>http://www.msn.com</NavigateUrl>  
    <AlternateText>msn site</AlternateText>  
    <Keyword>MSN</Keyword>
```

</Ad>

```

<Ad>
  <ImageUrl>~/AdImages/BannerAd2.gif</ImageUrl>
  <NavigateUrl>http://www.yahoo.com</NavigateUrl>
  <AlternateText>yahoo site</AlternateText>
  <Keyword>YAHOO</Keyword>
</Ad>

```

```

<Ad>
  <ImageUrl>~/AdImages/BannerAd3.gif</ImageUrl>
  <NavigateUrl>http://www.indiatimes.com</NavigateUrl>
  <AlternateText>Indiatimes site</AlternateText>
  <Keyword>INDIATIMES</Keyword>
</Ad>

```

</Advertisements>

Now to fetch the advertisements from the XML file take XMLDataSource control, and give a path of XMLfile to the XMLDataSource control. To bind AdRotator control to the XMLDataSource control use DataSource property of the AdRotator control.

2.13 Calendar Control:

ASP.NET provides a Calendar control that is used to display a calendar on a Web page. ASP.NET Calendar control displays a month calendar that allows user to select dates and move to the next and previous months.

By default, this control displays the name of the current month, day headings for the days of the weeks, days of the month and arrow characters for navigation to the previous or next month. The important property of the calendar control is SelectionMode property.

- SelectionMode:** SelectionMode property allows user to select one or multiple dates from the calendar control. The possible values for this property are: None, Day, DayWeek, and DayWeekMonth. Default value for this property is Day, means user can select any one date from the Calendar control. If you change it to DayWeek then user can either select any one date or entire week. If the value of this property is DayWeekMonth then calendar allow user to select any one date, entire week or whole month. If the value of this property is none then the calendar control does not allow selecting any date from the calendar.

2. **SelectedDate:** SelectedDate property returns date (object of DateTime class) which is selected by the user from the calendar control.
3. **SelectedDates:** SelectedDates property returns collection (Array) of multiple dates (objects of DateTime class), which are selected by the user.



Exercises

Question : 1 Answer in short:

- 1) What is QueryString?
- 2) What is Cookie?
- 3) Differentiate Application and Session variables.
- 4) Explain FindControl() method of PreviousPage object.
- 5) Differentiate ListBox and DropDownList controls.

Question : 2 Explain following properties:

- 1) AlternateText (ImageButton)
- 2) ImageUrl (ImageButton)
- 3) NavigateUrl (HyperLink)
- 4) PostBackUrl (LinkButton)
- 5) HotSpots (ImageMap)
- 6) AutopostBack (CheckBox)
- 7) GroupName (RadioButton)
- 8) SelectedIndex (CheckBoxList)
- 9) RepeatDirection (RadioButton)
- 10) DataSource (DropdownList)
- 11) SelectionMode (ListBox)
- 12) HasFile (FileUpload)

Question : 3 Answer in brief:

- 1) List and explain different methods of passing information from one page to another.
- 2) Write a short note on AdRotator control.
- 3) Explain FileUpload control in brief.

- 4) Explain Calendar control in detail.
- 5) Write a short note on CheckBoxList server control.

Question: 4 Answer the following MCQs:

Q:1 _____ Control is used to show random advertisements on the web page.

(A) Advertisements (B) Ads
✓(C) AdRotator (D) Banner

Q:2 _____ function will be used to convert relative path into absolute path.

✓(A) Server.MapPath() (B) Server.ConverPath()
(C) Response.MapPath() (D) Request.MapPath()

Q:3 To give different hyperlink on the different areas of the same image control is used.

(A) Image (B) ImageMap
(C) LinkButton (D) HyperLink

Q:4 _____ property is useful for the control ImageButton if browser is not be able to show the images.

(A) PostBackUrl (B) AutoPostBack
✓(C) AlternateText (D) ImageUrl

Answers :

1. C [AdRotator]
2. A [Server.MapPath]
3. B [ImageMap]
4. C AlternateText



CC-304 Unit – 2 Web Application Development-I Practicals

Program : 1

Write a program containing the following controls:

- A DropDownList
- A Button
- A Label

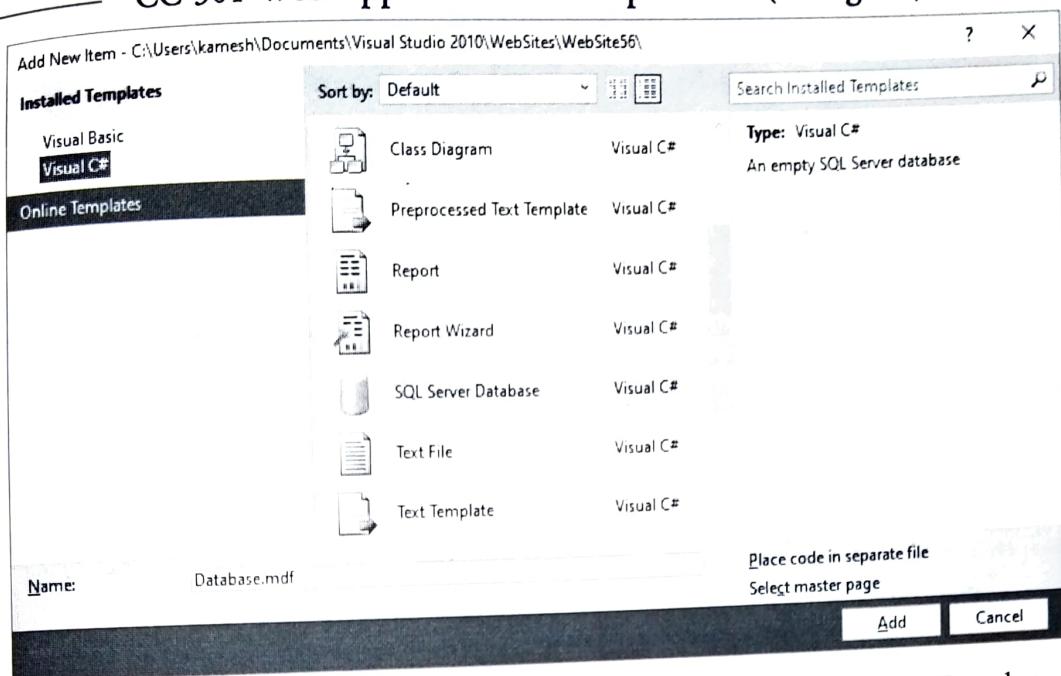
The DropDownList is used to list items available in a store. When the user clicks on an item in the DropDownList, the cost of the selected item is displayed in the label control. The Form title must be ASP.NET. A button must be in the center of a form. Add the following more controls:

- Two labels
- A TextBox
- A Button

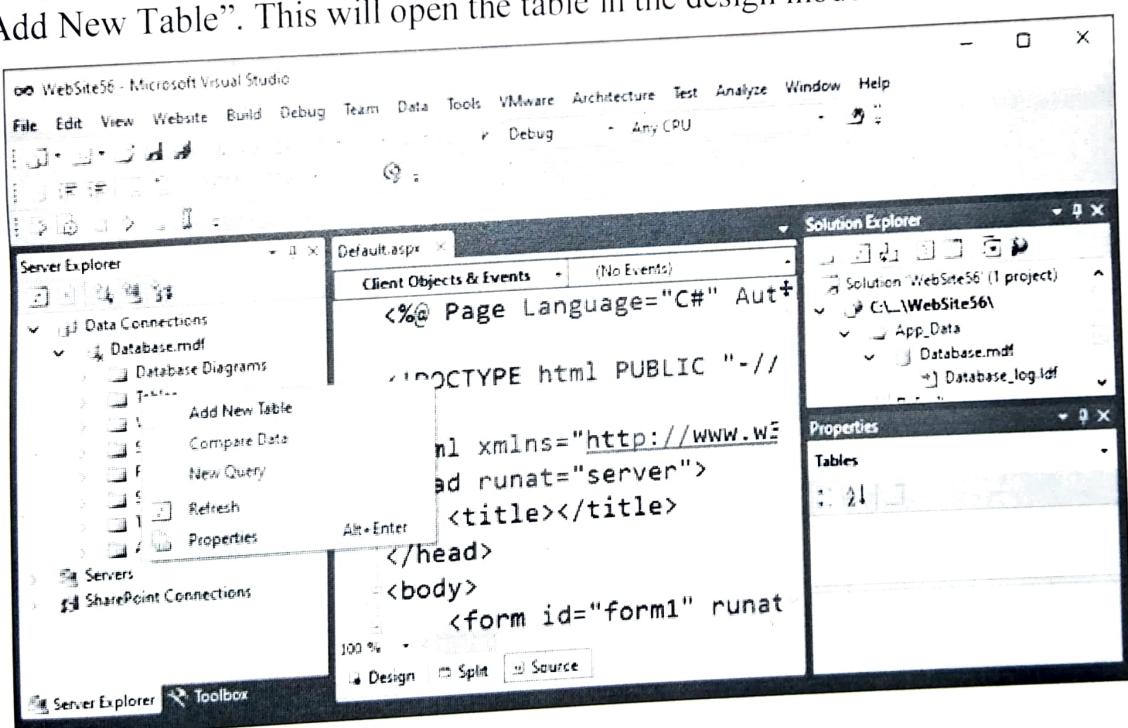
One of the labels is displayed adjacent to the textbox, displaying the message “Enter the quantity:” When the user enters the quantity in the textbox and clicks the button, the total cost is evaluated and displayed in another label.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button.
- Now, Right click on the website, in the Solution Explorer and choose, option “Add ASP.NET Folder” and select App_Data.
- This action will add a folder App_Data to your website. Usually this folder is used to store database files. Now, Right click on the App_Data folder, and choose option “Add New Item”. This will open a dialog “Add New Item” dialog box.
- From this dialog box select option “SQL Server Database” and Click on Add button.

CC-301 Web Application Development- I (Using C#)



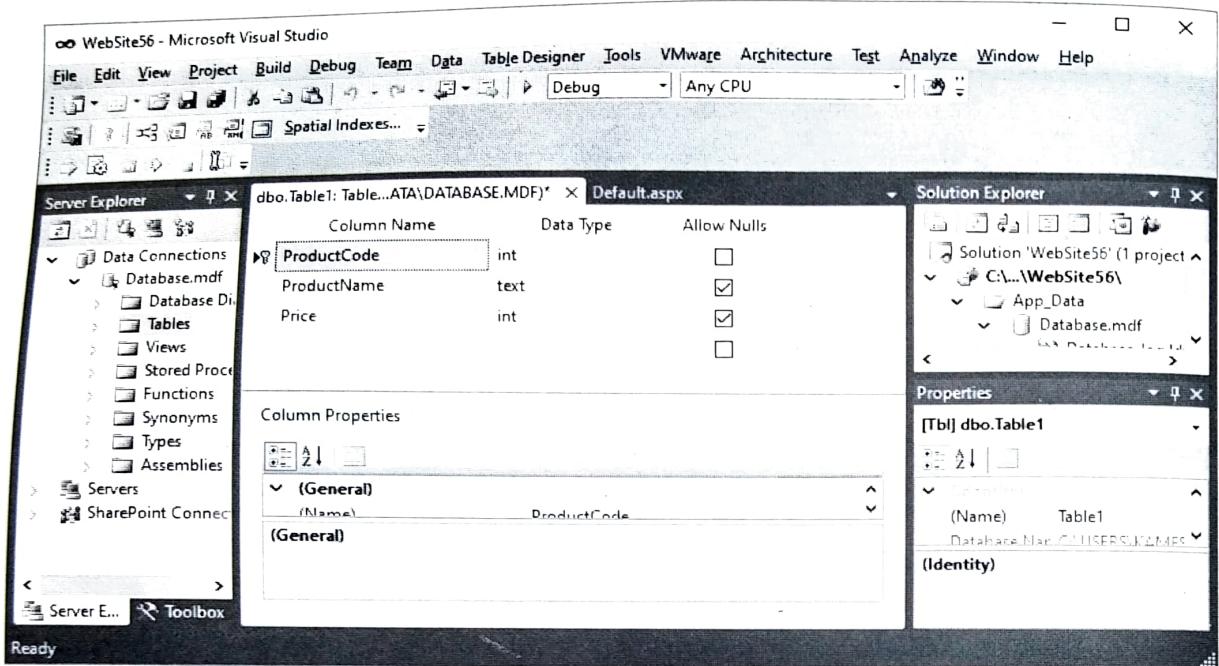
- This action will add a database file into App_Data directory called "Database1.mdf". It will automatically open the Server Explorer Tab at the left side of the screen.
- Now, from the Server Explorer Right click on the "Table" option and choose option "Add New Table". This will open the table in the design mode.



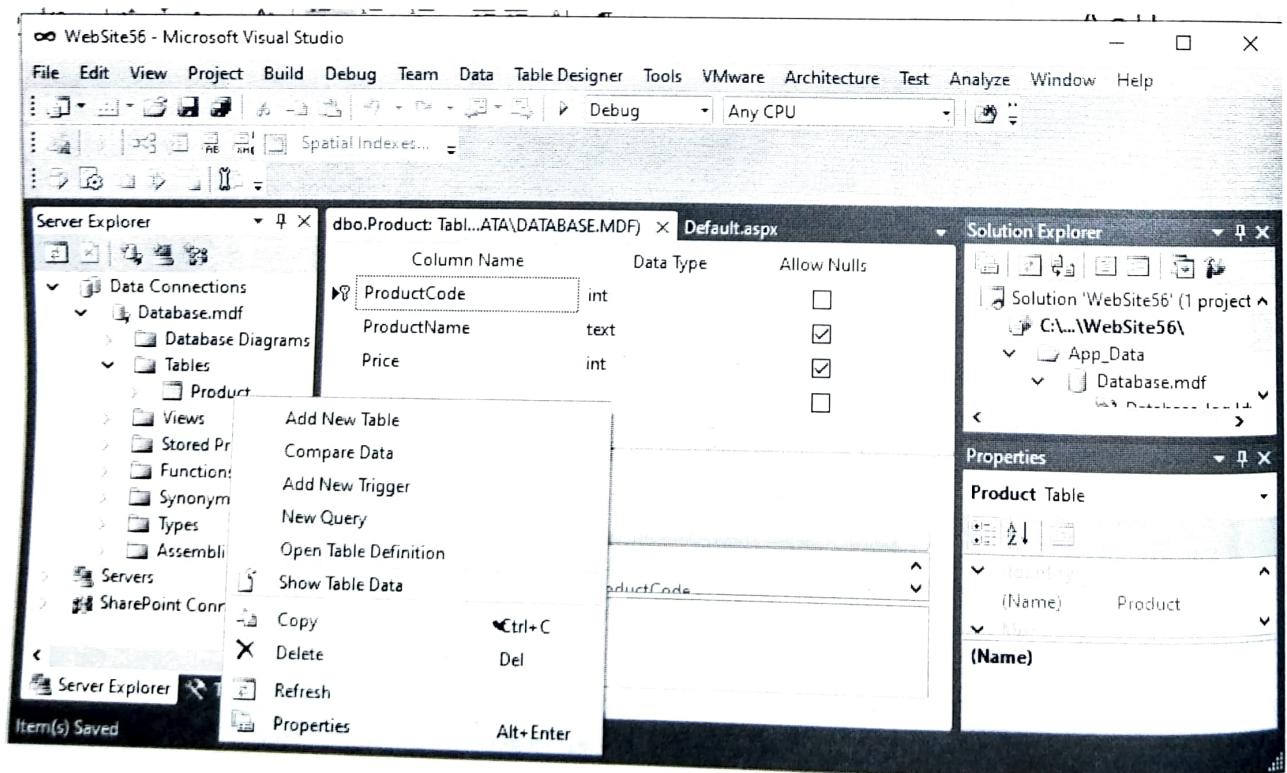
- Now create the following Table.

Field Name	Data Type	Constraint
ProductCode	Int	Primary Key
ProductName	Text	
Price	Int	

- Now Click on the save button from the Toolbar, It will ask for Table name. Save the table with “Product” name.

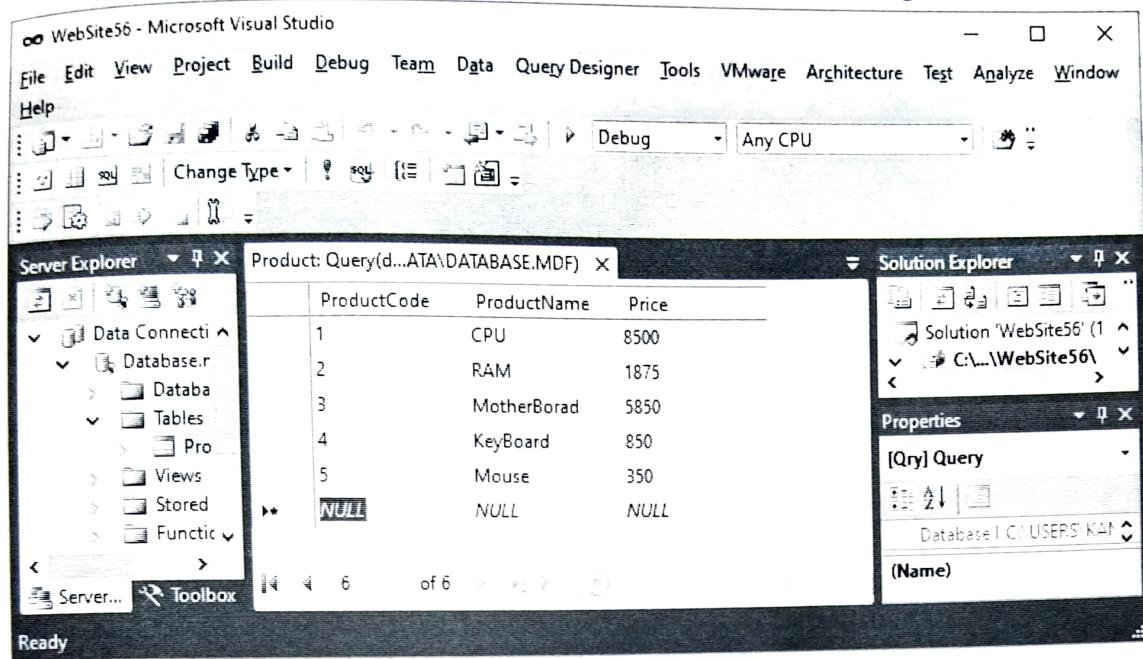


- Now Right click on the “Product” table from the Tables, and choose option “Show Table Data”.

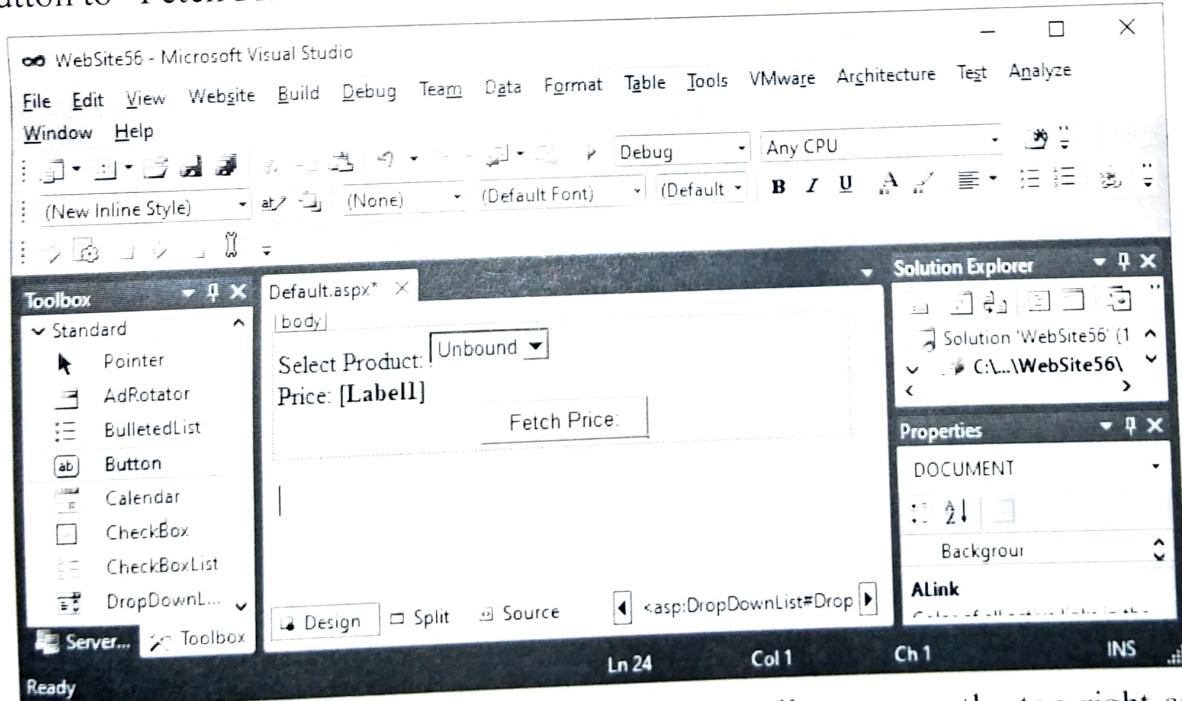


- This action will open the table into the data entry mode. Insert the following records in it.

CC-301 Web Application Development- I (Using C#)

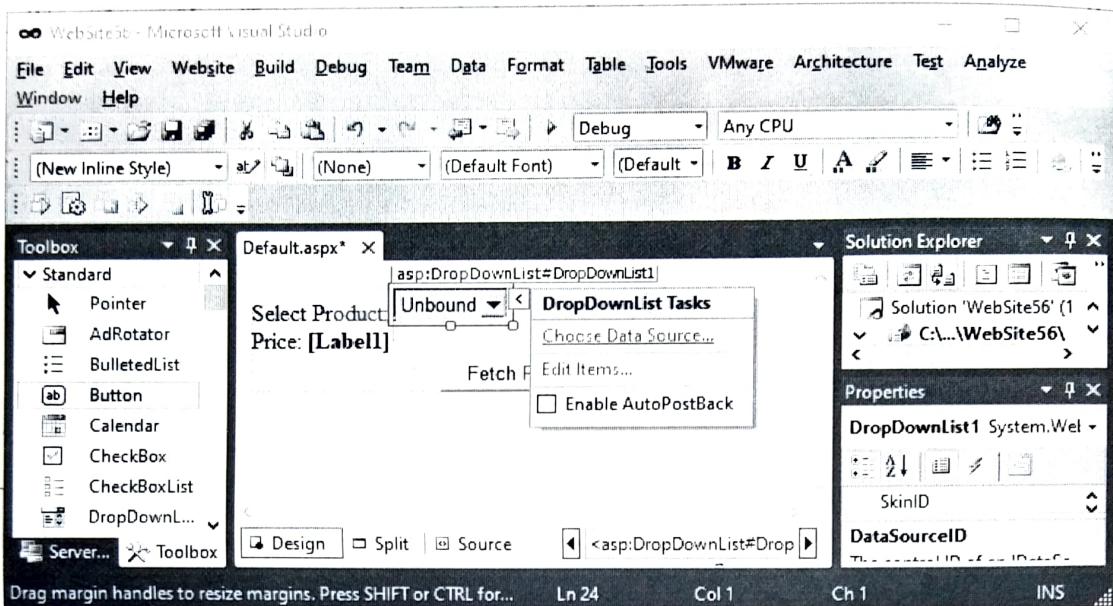


- Now close the table, and open the default.aspx page by double click on it from the Solution Explorer.
 - Design the Interface as shown in the default.aspx page using a DropDownList, Button and a Label control. Set the Text property of the Label is NULL. Select the Label and Press Ctrl+B, to show label in the Bold face. Put the button in the `<center> </center>` tag, so it will be shown in the centre of the form. Change the Text property of the Button to “Fetch Price”.

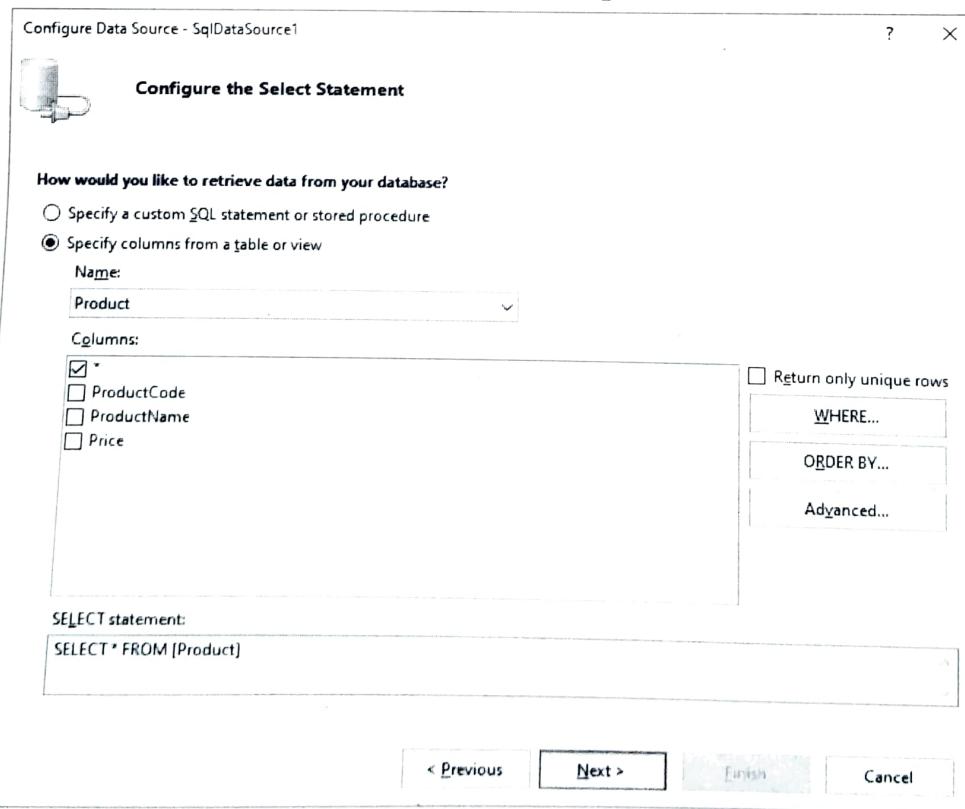


- Ready

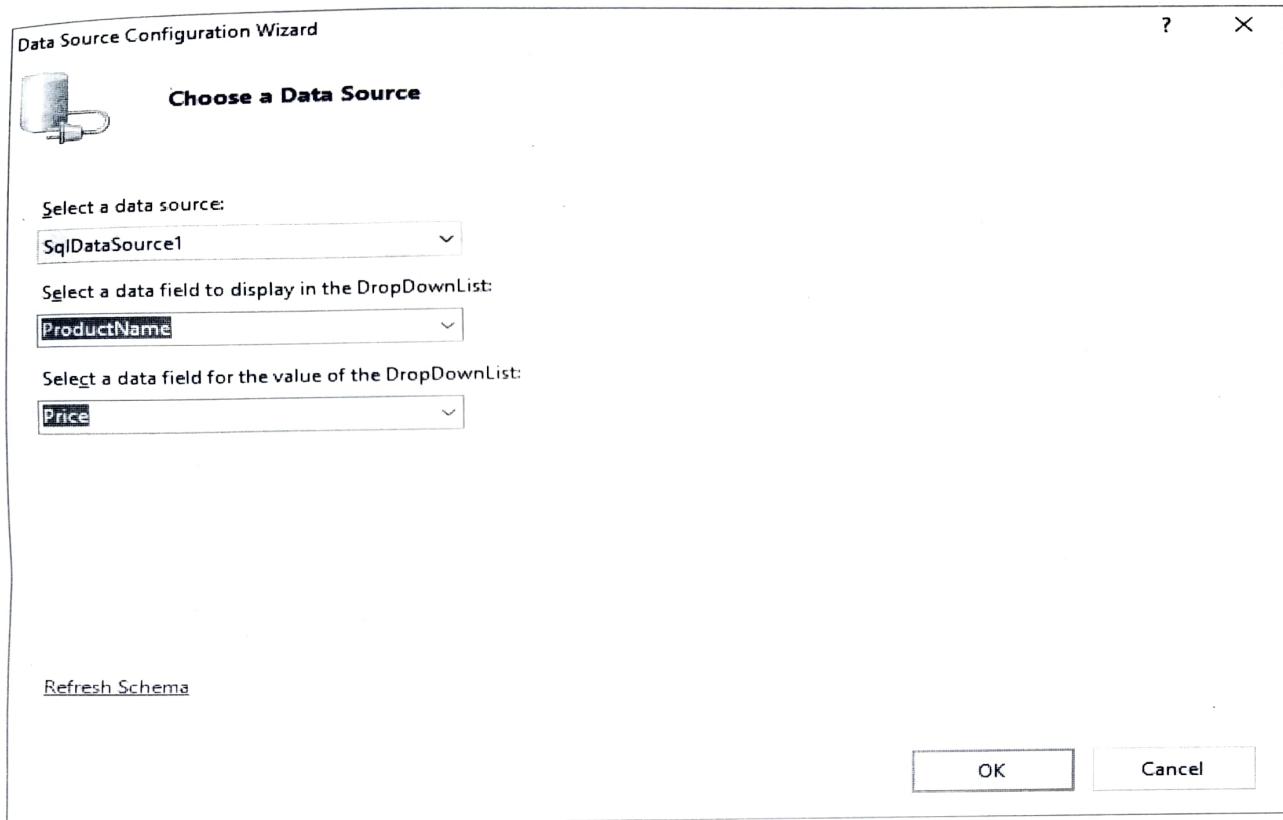
 - Now select the DropDownList, you can find a small arrow on the top-right side of it. Click on it, it will open a smart menu. Select option “Choose DataSource” from it. One Wizard will be opened.



- Now, from the wizard Select option “Select a data source” and choose “<New data source...>” option. In the Next step select “Database” (SQL) option and click on OK button.
- In the Next screen of the wizard there is a DropDownList is there. Open the DropDownList and select “Database.mdf” and then click on “Next” Button. In the Next step of the Wizard you don’t have to do anything simply click on “Next” button again.
- The next screen will be shown in the figure given below. Click on “Next” button and finally click on the “Finish” button at the last step of the wizard.



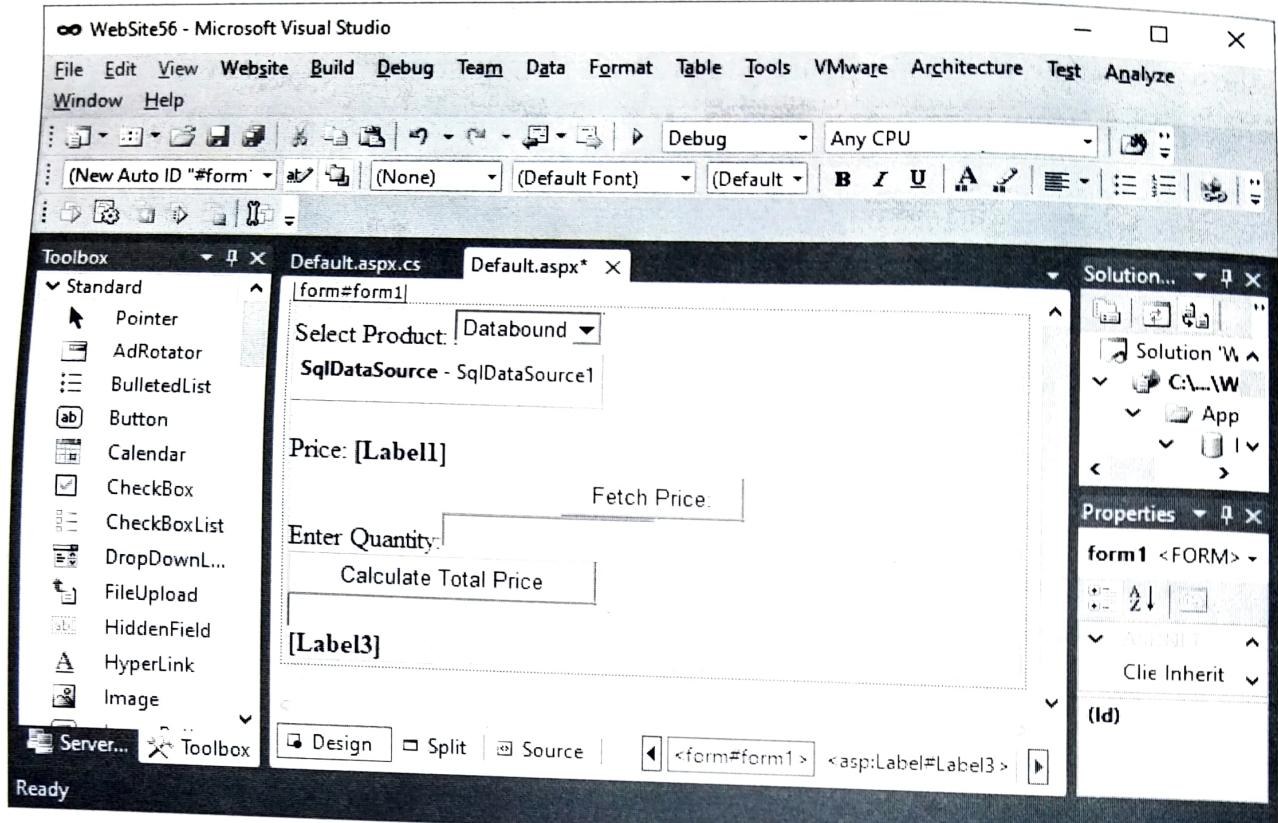
- Now you will be coming back to “Choose Data source” screen. Select the options as shown from the following figure and click on OK Button.



- Make sure in the first DropDownList you have to select “SqlDataSource1”, in the second DropDownList you need to select “ProductName”, and from the third “DropDownList” you need to select “Price”.
- After clicking on the OK button, do Double click on the button you have place on default.aspx page (Fetch Price). It will open subroutine “button1_Click”. Write the following code in it.

Label1.Text = DropDownList1.SelectedValue;

- Run the default.aspx page in the browser by pressing Ctrl+F5 button. Select the product from the DropDownList and click on the button called “Fetch Price”, you can see that the price of that product will automatically come in the Label.
- Now add 2 Labels, 1 Textbox and 1 Button and change their text property as shown below.



- Double click on the button2 (Calculate Total Price), and write the following code on the click event of that button.

```
int tprice;
tprice = Convert.ToInt16(Label1.Text) *
Convert.ToInt16(textBox1.Text);
Label3.Text = "Total Price is:" +
Convert.ToString(tprice);
```

- The complete designer code is given below:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!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>

            Select Product:
            <asp:DropDownList ID="DropDownList1" runat="server"
                DataSourceID="SqlDataSource1"
                DataTextField="ProductName"
                DataValueField="Price">
            </asp:DropDownList>
            <asp:SqlDataSource ID="SqlDataSource1" runat="server"
                ConnectionString=<%$ConnectionString:ConnectionString %>">
                SelectCommand="SELECT * FROM [Product]"></asp:SqlDataSource>
            <br />
            Price:
            <asp:Label ID="Label1" runat="server" style="font-weight: 700"></asp:Label>
            <br />
            <center>    <asp:Button ID="Button1" runat="server" Text="Fetch Price:" onclick="Button1_Click" /></center>
            </div>
            <asp:Label ID="Label2" runat="server" Text="Enter Quantity:></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="Button2" runat="server" onclick="Button2_Click" Text="Calculate Total Price" />
```

```
<br />
<br />
<asp:Label ID="Label3" runat="server" style="font-
weight: 700"></asp:Label>
</form>
</body>
</html>
```

- And complete code in C#.NET is as given below.

```
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)
    {

    }

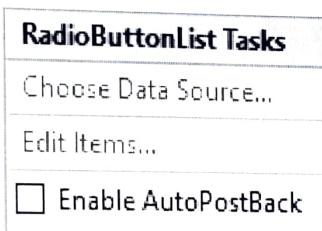
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = DropDownList1.SelectedValue;
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        int tprice;
        tprice = Convert.ToInt16(Label1.Text) *
Convert.ToInt16(textBox1.Text);
        Label3.Text = "Total Price is:" +
Convert.ToString(tprice);
    }
}
```

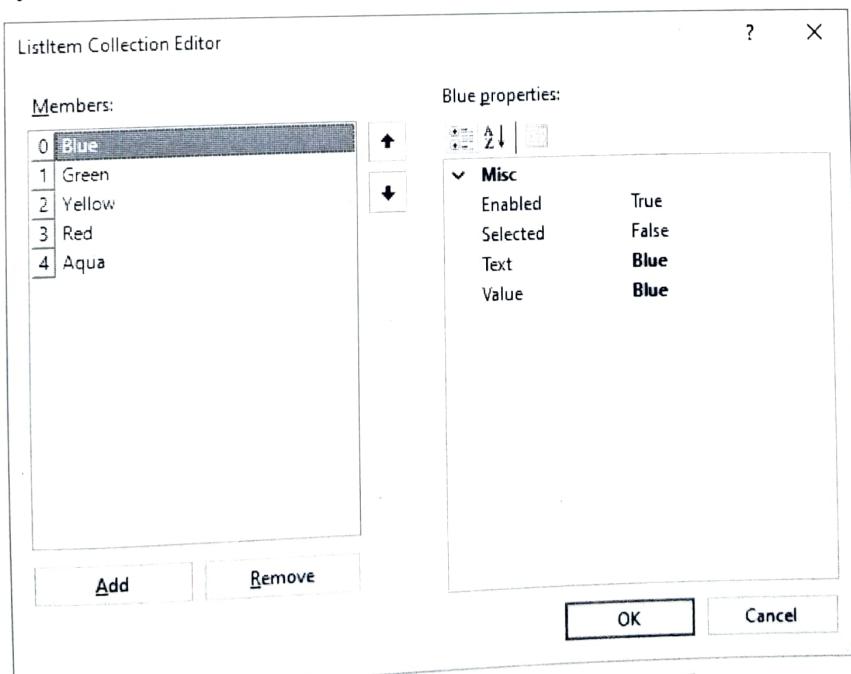
Program : 2

Create a RadioButtonList that displays the names of some colors in two columns. Add a button to the Web Form which when clicked changes the color of the Form to the color selected from the list.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Now add a RadioButtonList control to the default.aspx from the ToolBox.
- Select the RadioButtonList control and click on the Arrow shown on the top-right side to open the smart menu.



- Now, select option “Edit Items...”.
- It will open the dialog box, click on “Add” button to add new item. In each item type the unique color name in the Text property displayed at the right side in the dialog box.
- Create a list of colors as shown in the following figure and then click on OK button.
- Now to “Source” of your default.aspx page and change <body> tag to <body id=”MyBody” runat=”server>. We mean add two attributes to body tag that are id=”MyBody” and runat=”server”.



- Now place a button on the web page and change its Text property to "Submit". Double click on the "Submit" button, which will open the subroutine called "button_Click". Add following code to the subroutine on the click event of the button.

```
String Clr="White";
foreach (ListItem i in RadioButtonList1.Items)
{
    if (i.Selected == true)
        Clr = i.Text;
}
MyBody.Style[HtmlTextWriterStyle.BackgroundColor]
= Clr;
```

- The complete source code of the web page is given below:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!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 id= "MyBody" runat ="server" >
    <form id="form1" runat="server">
        <div>

            <asp:RadioButtonList ID="RadioButtonList1"
runat="server">
                <asp:ListItem>Blue</asp:ListItem>
                <asp:ListItem>Green</asp:ListItem>
                <asp:ListItem>Yellow</asp:ListItem>
                <asp:ListItem>Red</asp:ListItem>
                <asp:ListItem>Aqua</asp:ListItem>

```

```
</asp:RadioButtonList>
<br />
<asp:Button ID="Button1" runat="server"
onclick="Button1_Click" Text="Submit" />

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

- The complete C#.NET code of the web page is given below.

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

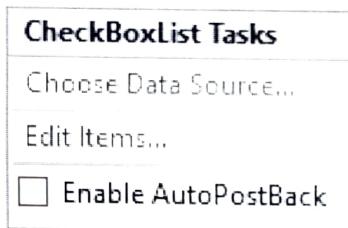
public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        }
    protected void Button1_Click(object sender, EventArgs e)
    {
        String Clr="White";
        foreach (ListItem i in RadioButtonList1.Items)
        {
            if (i.Selected == true)
                Clr = i.Text;
        }
        MyBody.Style[HtmlTextWriterStyle.BackgroundColor]
= Clr;
    }
}
```

- Run the web page in the browser, select any color from the RadioButtonList and click on the “Submit” button. You will notice that the background color of the web page gets changed.

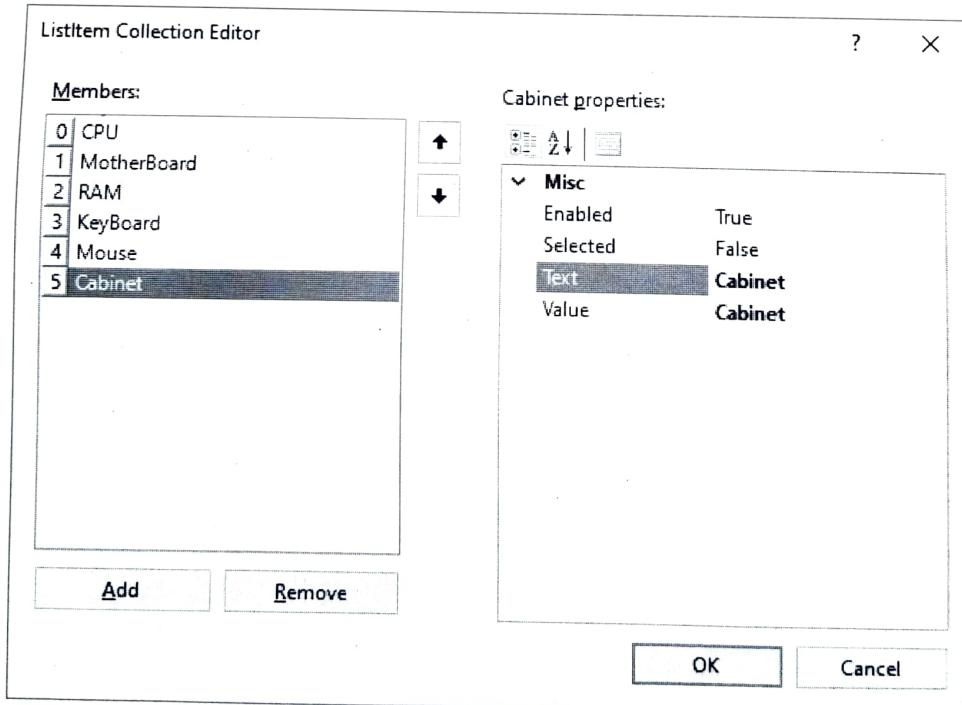
Program : 3

Create a web page having checkboxlist control shows different products. Web page should have a button and a label. On the click event of the button shows the message “Thank You for placing the order of following items” and then list of all products selected by the user in the checkboxlist server control. Each selected product should be displayed in the new line.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Place a CheckBoxList control on the default.aspx page from the ToolBox. Select option “Edit Items...” from the smart menu.



- Add the name of the products as shown in the following figure, by clicking on the Add buttons and set the “Text” property of each list items as shown below.

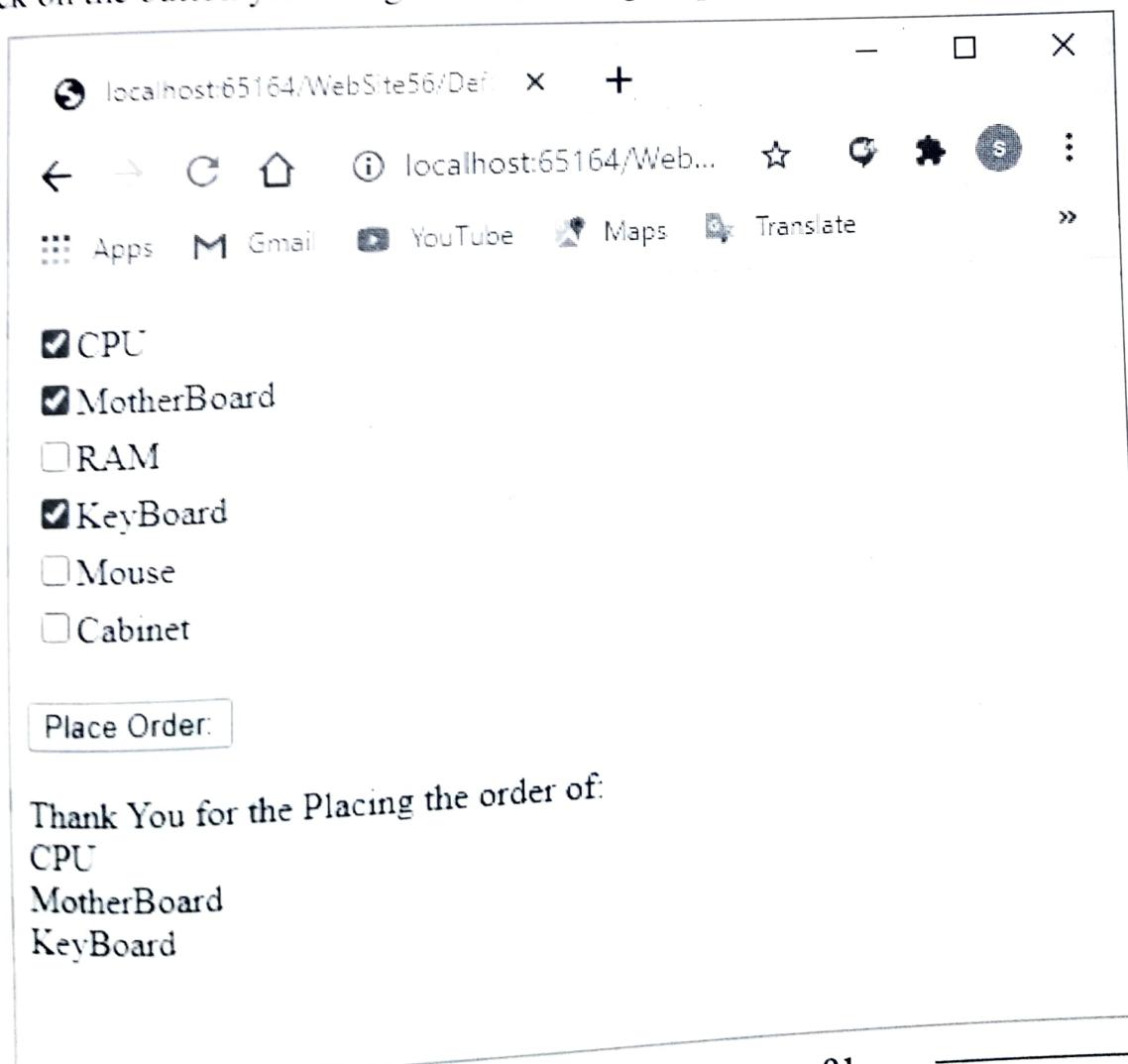


Click on the "OK" button after adding items to the CheckBoxList control. Now Place a button to your web page and set its text property to "Place Order". Place a Label control to the web page and set its Text property to NULL.

Double click on the button and write the following code in the click event of the button1.

```
Label1.Text = "Thank You for the Placing the order  
<br />";  
foreach(ListItem l in CheckBoxList1.Items)  
{  
    if(l.Selected == true)  
        Label1.Text = Label1.Text + l.Text +  
"<br />";  
}
```

- Run the web page in the Browser select some items from the CheckBoxList control. Click on the button you will get the following output.



Program : 4

Write a simple Web application which keeps track of the number of times a user has visited the page from the same machine. The application keeps track of this information by storing this counter value in a persistent cookie at the client's machine.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Place a Label control on it. Now, double click on the web page, this action will open a subroutine called “Page_Load”. Write the following code on the load event of the page.

```

int cntr = 1;
if (Request.Cookies["myCookies"] != null)
{
    cntr =
Convert.ToInt16(Request.Cookies["myCookies"].Value);
    cntr++;
}
HttpCookie ck = new HttpCookie("myCookies");
ck.Value = Convert.ToString(cntr);
Response.Cookies.Add(ck);
Label1.Text = "<B> This Page is visited for " +
Convert.ToString(cntr) + " times on this Computer</B>";

```

- Run the web page for 2 or more times. You will see that the counter variable will be increase each time.

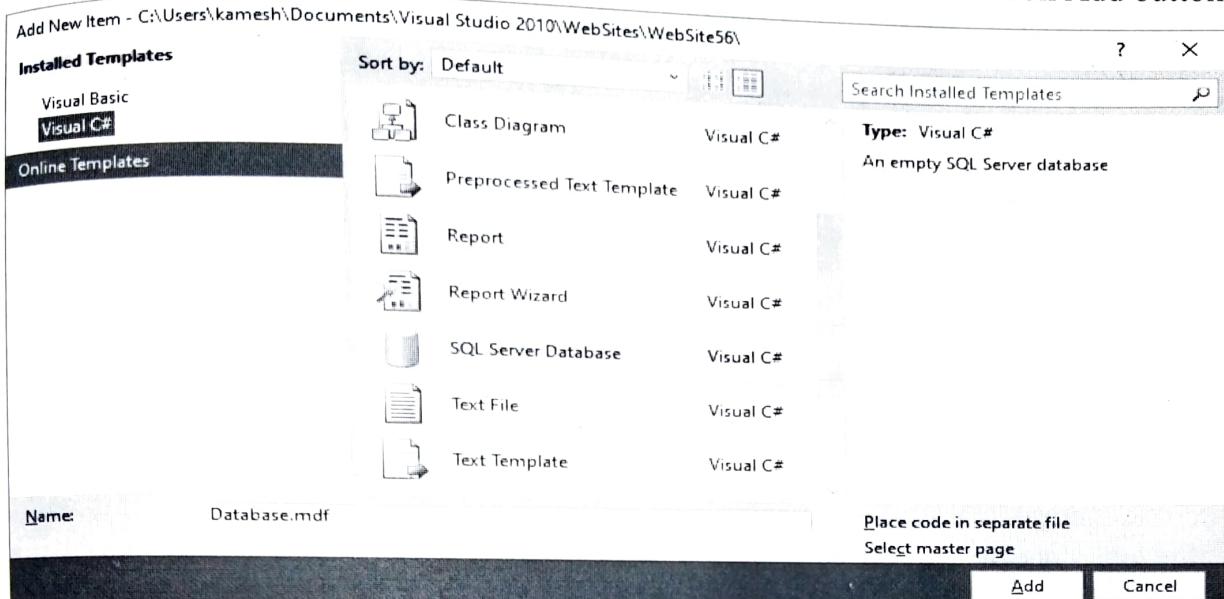
Program : 5

Display name of country in dropdown list when page is loaded. Allow the user to select the country and display the name of states of that country selected by user in another dropdown list. (Also perform through datareader).

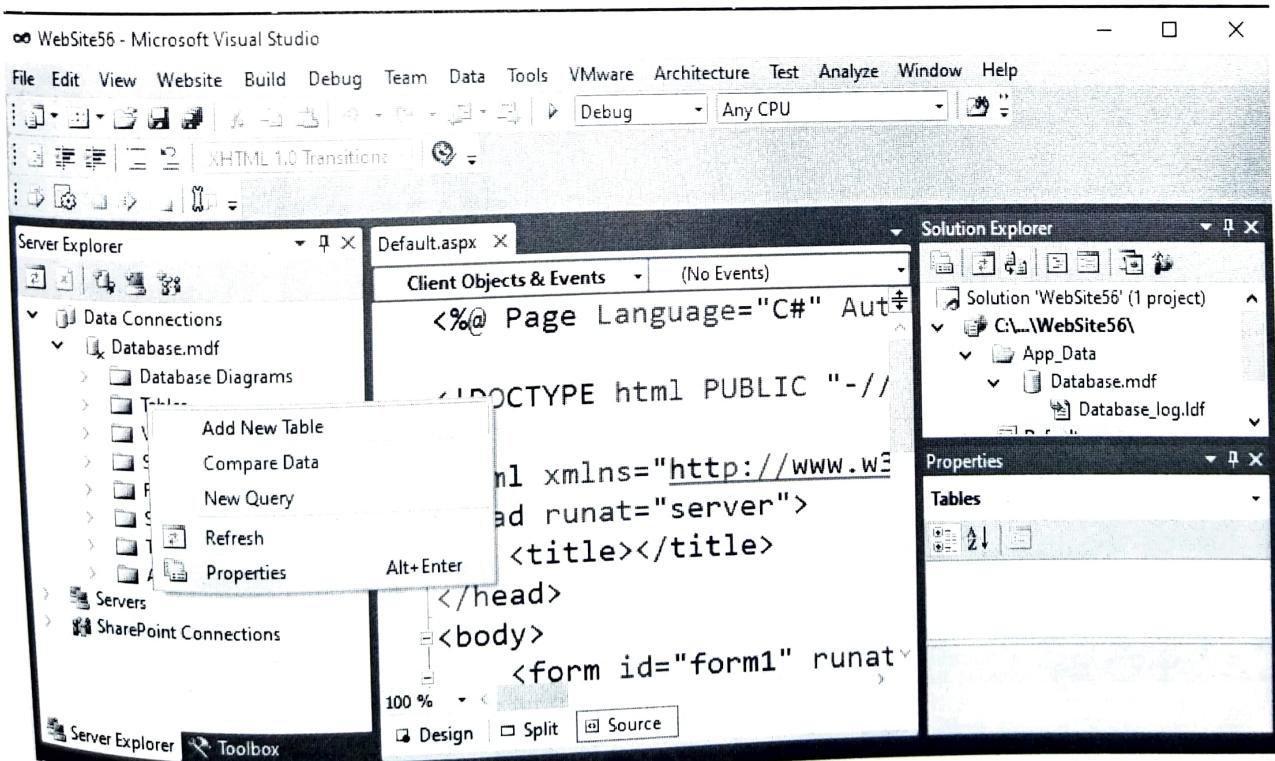
- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Now, Right click on the website, in the Solution Explorer and choose, option “Add ASP.NET Folder” and select App_Data.

CC-301 Web Application Development- I (Using C#)

- This action will add a folder App_Data to your website. Usually this folder is used to store database files. Now, Right click on the App_Data folder, and choose option “Add New Item”. This will open a dialog “Add New Item” dialog box.
- From this dialog box select option “SQL Server Database” and Click on Add button.



- This action will add a database file into App_Data directory called “Database1.mdf”. It will automatically open the Server Explorer Tab at the left side of the screen.
- Now, from the Server Explorer Right click on the “Table” option and choose option “Add New Table”. This will open the table in the design mode.



- Now create the following Tables.

Table:1 State

Field Name	Data Type	Constraint
StateCode	Int	Primary Key
StateName	Text	

Table:2 City

Field Name	Data Type	Constraint
CityCode	Int	Primary Key
CityName	Text	
StateCode	Int	

- Now, Right-Click on the State table choose option “Show Table Data”. This will open the table into the data sheet mode. Enter the data of various states as shown in the figure given below.
- Similarly, Right click on the City table, choose option “Show Table Data”, and add the names of various cities as shown below.

WebSite55 - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Query Designer Tools VMware Architecture Test Analyze Window Help

Server Explorer Solution Explorer

State: Query(desk...ATA\DATABASE.MDF) Default2.aspx.cs

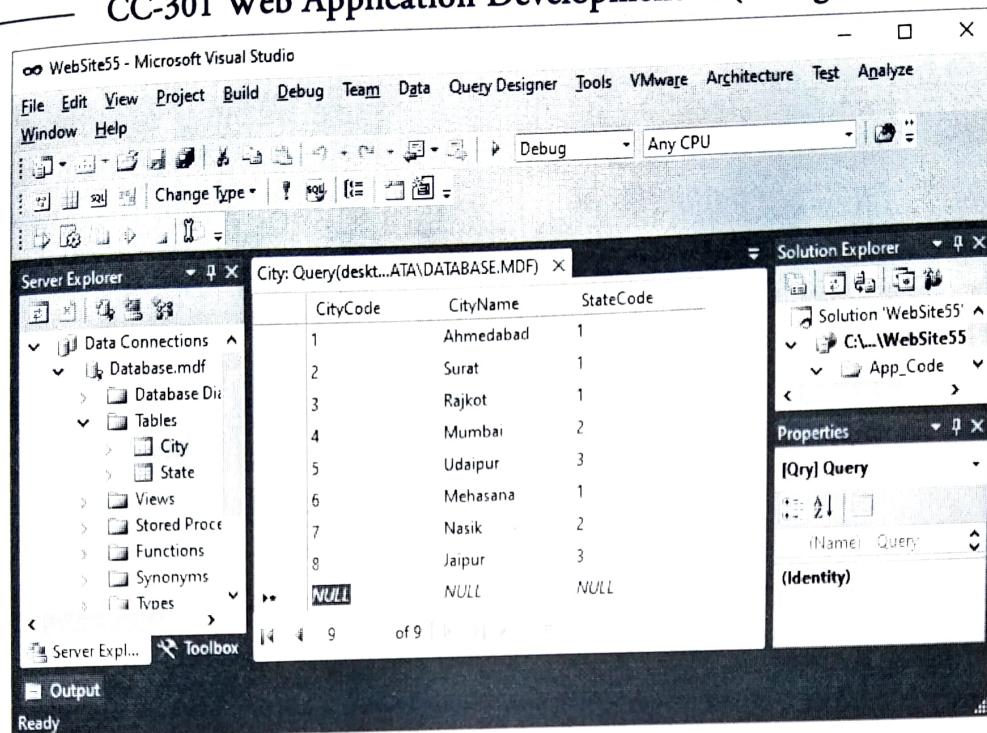
Properties

StateCode	StateName
1	Gujarat
2	Maharastra
3	Rajasthan
4	Punjab
5	Haryana
NULL	NULL

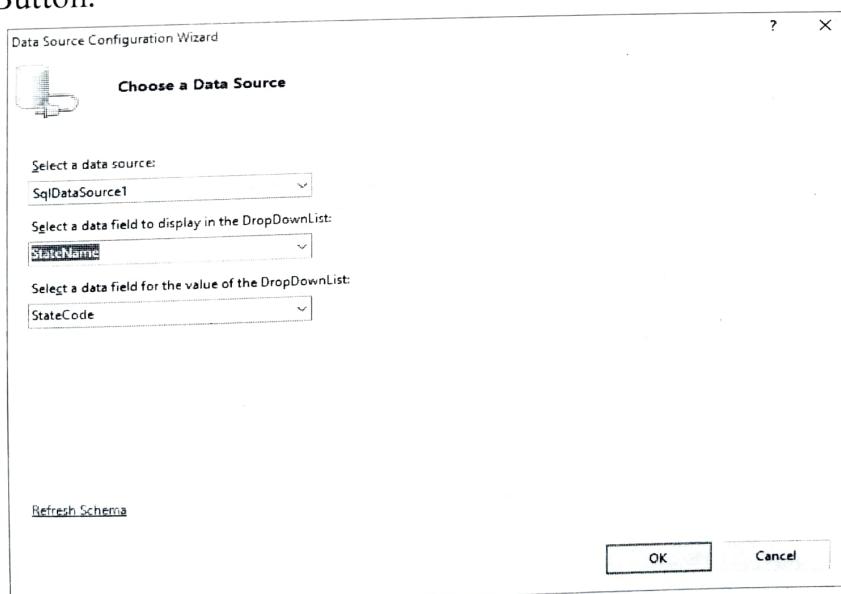
(Name) Query
(Identity)

Output

Ready

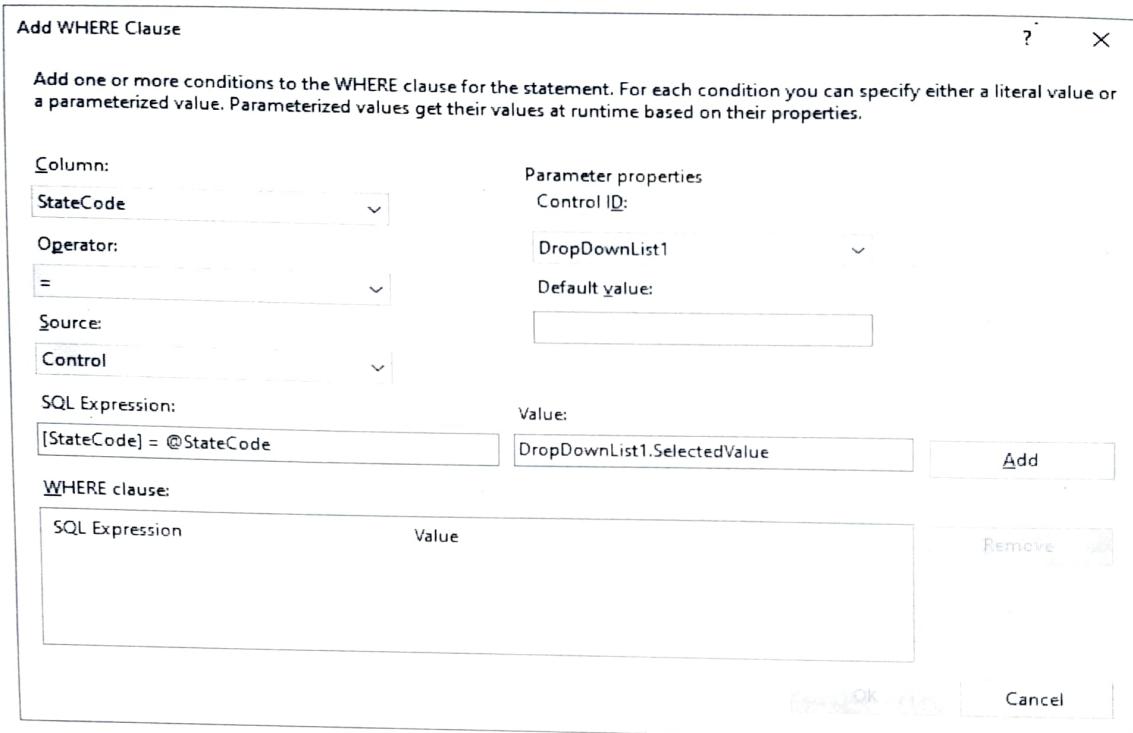


- Now place 2 drop down lists' controls on the default.aspx webpage as shown below.
- Click on the first DropDownList and click on the Top-Right arrow to open the smart menu. First check the option “Enable AutopostBack”.
- Now click on “Choose Data Source” option. From the “Select Data Source” option Select “<New Data Source...>” option. From the next screen select SQL Database option and click on Next Button. In the Next Screen select “ConnectionString” option from the DropDownList, and again click on the Next button.
- Now select “State” table from the DropDownList and click on the next Button. Now in the next screen click on “Finish” button.
- Select following options from the “Choose Data Source” wizard step screen, and then click on OK Button.

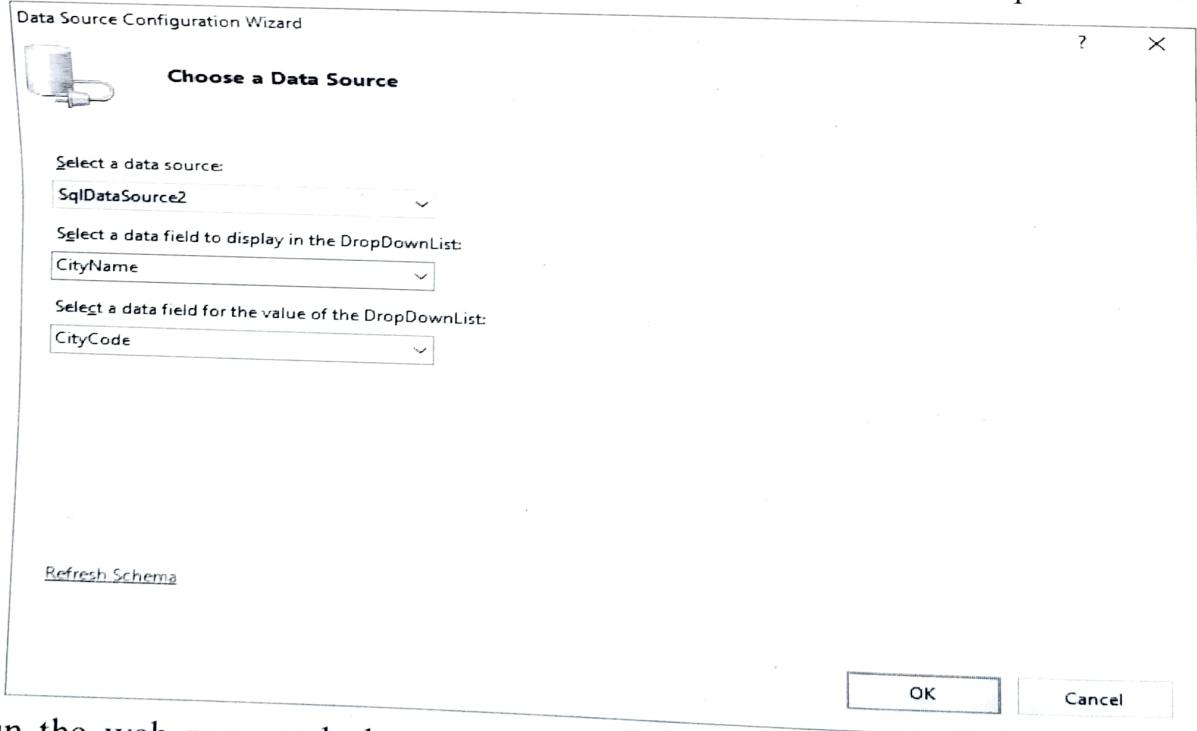


CC-301 Web Application Development- I (Using C#)

- Repeat the same steps for “DropDownList2”, This time instead of State you Choose “City” table. Now Click on Where button as set following things and then click on “Add” and then OK button.



- Select the following options from choose data source screen for DropDownList2.

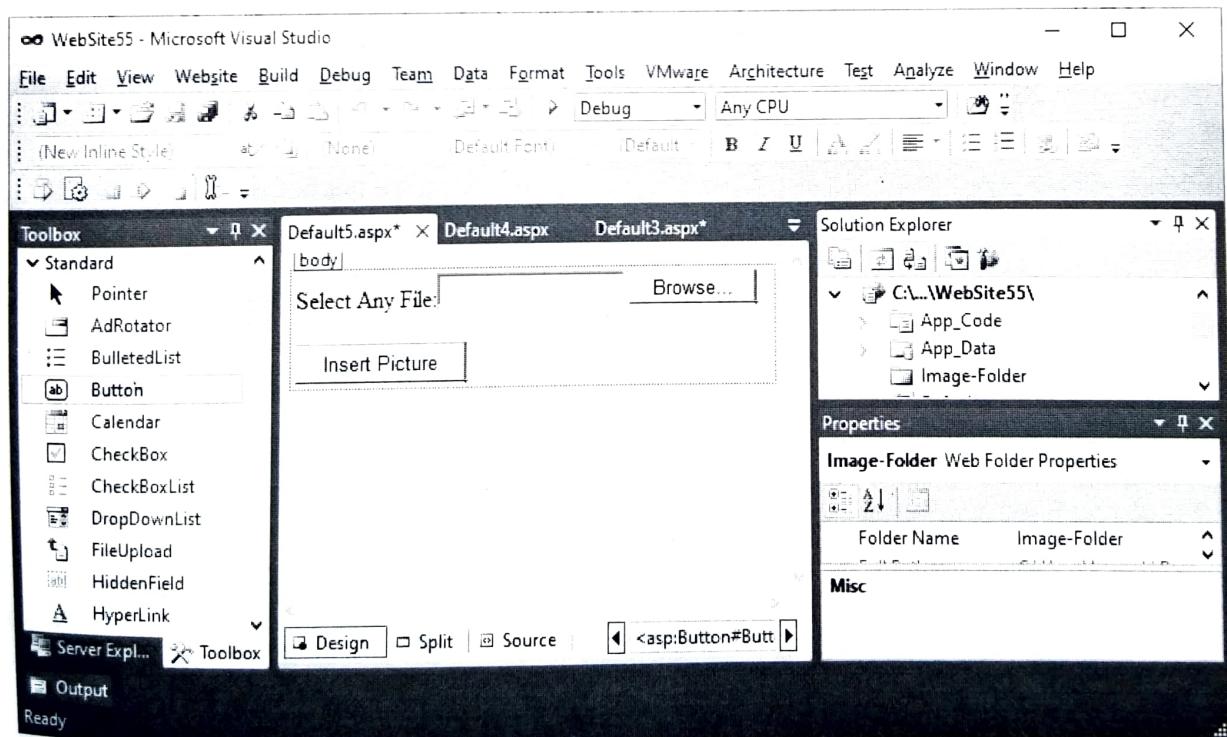


- Run the web page and change the state from the DropDownList1, you can see that according to state name, cities also changed from the DropDownList2.

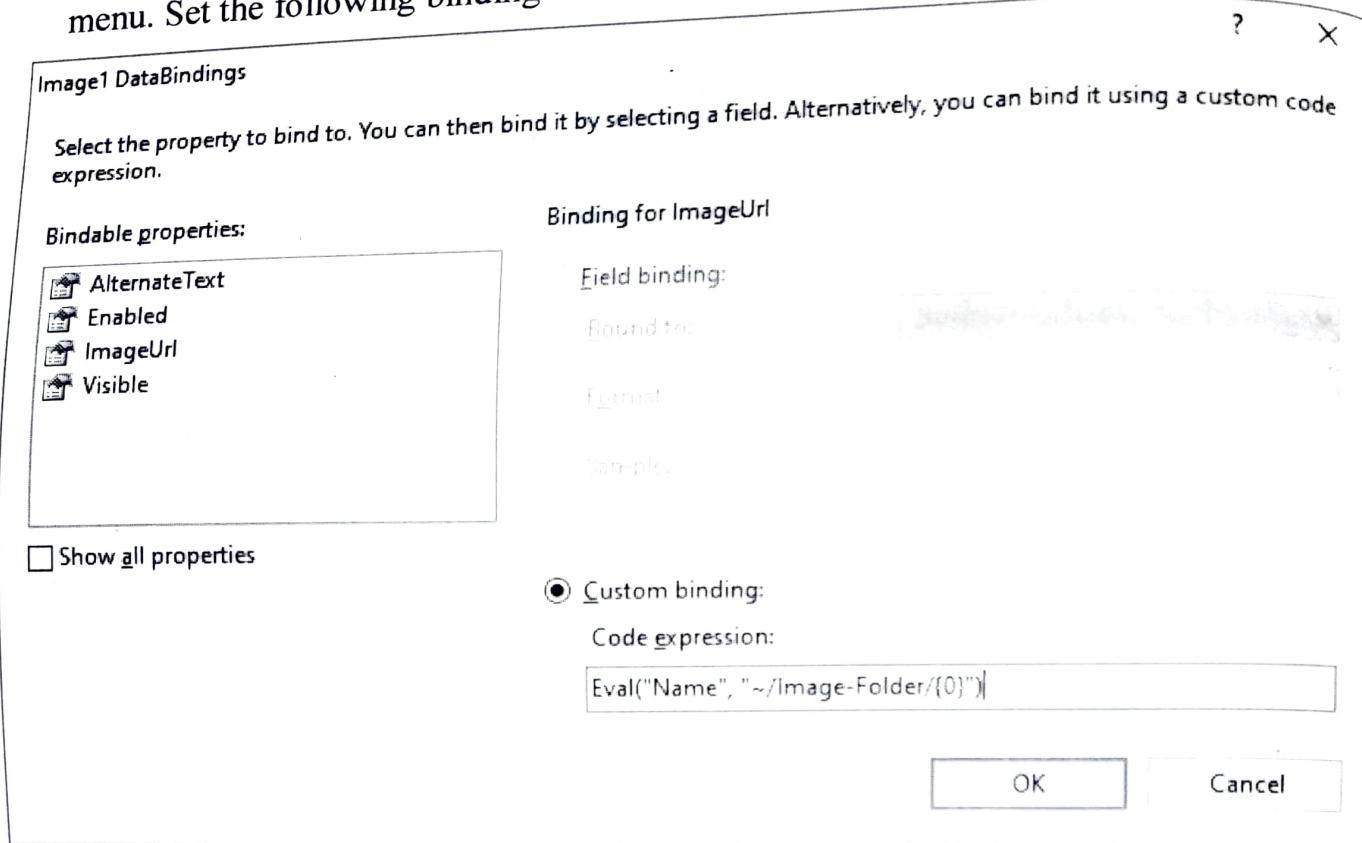
Program : 6

Write code to upload only image files (.bmp, .jpg, .gif) and less than 1 kb in folder “Image-Folder”. Also display uploaded image files on the same web page using datalist control.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Right click on the website in the Solution Explorer and choose option “New Folder”. Give name “image-folder” to the newly created folder.
- Design the Default.aspx page as shown below using a FileUpload control and a Button. Set the Text property of the button to “Insert Picture”.
- Add a DataList control from the Data category of the ToolBox. Now from the smart menu option select “Edit Templet” option.
- Place an Image control in it, and set its Hight and Width property to 250 and 225.



- Select the Image control and click on the “Edit DataBinding...” option from the smart menu. Set the following binding as shown in the following figure.



- Select ImageUrl option and type code expression to: “Eval("Name", “~/Image-Folder/{0}”)” and click on the OK Button. Click on “End Template Editing”.
- Now Select the DataList control and choose option “Property Builder”. Enter Column to 5 and click on OK Button.
- Now write the following code.

```

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

public partial class Default5 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
}

```

```

        }
        protected void Page_PreRenderComplete(object sender,
EventArgs e)
        {
            System.IO.DirectoryInfo di = new
System.IO.DirectoryInfo(Server.MapPath("~/Image-Folder"));
            DataList1.DataSource = di.GetFiles();
            DataList1.DataBind();
        }

protected void Button1_Click(object sender, EventArgs e)
{
    if (FileUpload1.HasFile)
    {

        if (FileUpload1.FileContent.Length < 10000)
        {

            FileUpload1.SaveAs(Server.MapPath("~/Image-Folder/" +
FileUpload1.FileName));
        }
    }
}
}

```

- Now instead of 1KB, we have considered a file of 10KB (10,000Bytes), as 1KB File is too small. The complete source of the Default.aspx is given below:

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default5.aspx.cs" Inherits="Default5" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

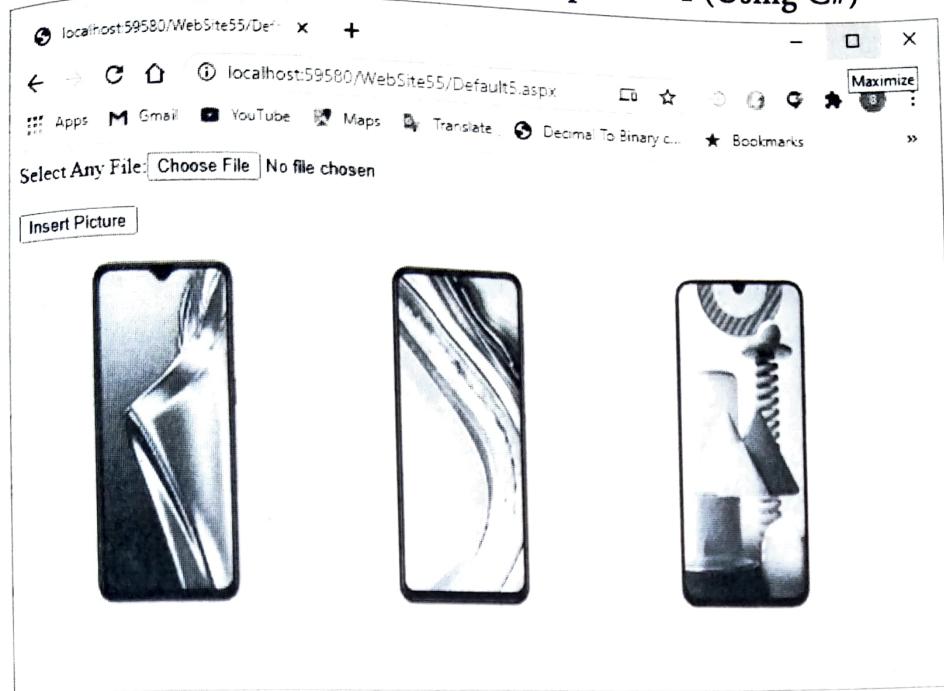
```

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

            Select Any File:<asp:FileUpload ID="FileUpload1"
runat="server" />
            <br />
            <br />
            <asp:Button ID="Button1" runat="server"
onclick="Button1_Click"
                Text="Insert Picture" />
            <br />
            <br />
            <asp:DataList ID="DataList1" runat="server"
RepeatColumns="5">
                <ItemTemplate>
                    <asp:Image ID="Image1" runat="server"
Height="250px"
                        ImageUrl='<%# Eval("Name", "~/Image-
Folder/{0}") %>' Width="225px" />
                </ItemTemplate>
            </asp:DataList>

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

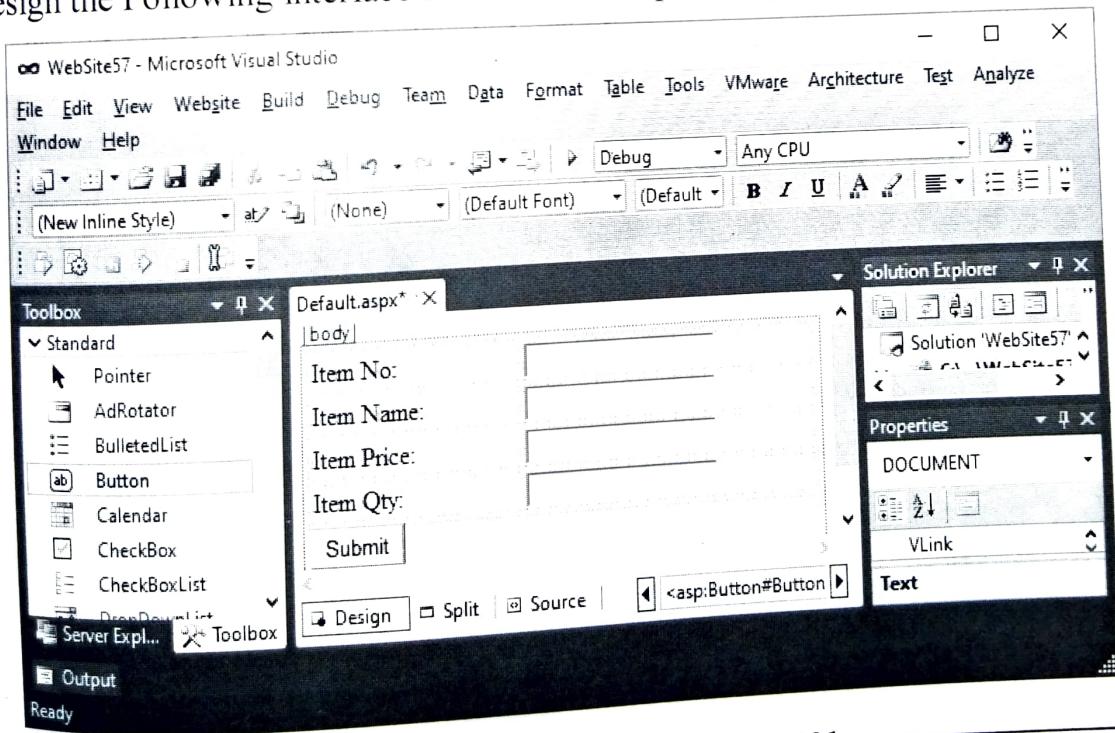
- Run the web page into the Browser Upload, some images and View the web page. It looks like follows:



Program : 7

Accept Item No, Item Name, Item Price, Item Quantity. Store information in cookie. Display stored information in next page.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Design the Following interface in the Default.aspx web page.



- Write the following code on the click event of the "Submit" button.

```

HttpCookie cookie = new HttpCookie("myMultiValcookie");
cookie.Values.Add("I_No", TextBox1.Text);
cookie.Values.Add("I_Name", TextBox2.Text);
cookie.Values.Add("I_Price", TextBox3.Text);
cookie.Values.Add("I_Qty", TextBox4.Text);
Response.Cookies.Add(cookie);
Response.Redirect("~/Default2.aspx");

```

- Now add another web page to your website called Default2.aspx, and place 4 labels in 4 different rows. Write the following code on Page_Load event.

```

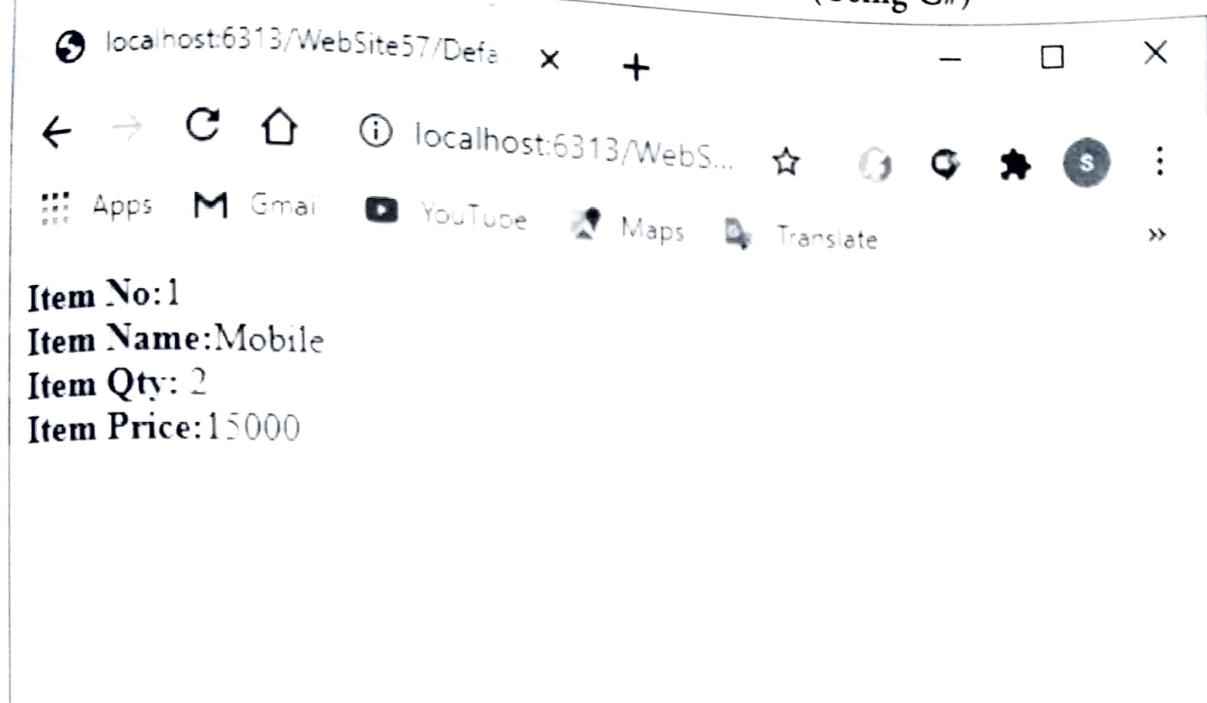
if (Request.Cookies["myMultiValcookie"] != null)
{
    Label1.Text = "<B>Item No:</B>" +
    Request.Cookies["myMultiValcookie"]["I_No"];
    Label2.Text = "<B>Item Name:</B>" +
    Request.Cookies["myMultiValcookie"]["I_Name"];
    Label3.Text = "<B>Item Qty:</B>" +
    Request.Cookies["myMultiValcookie"]["I_Qty"];
    Label4.Text = "<B>Item Price:</B>" +
    Request.Cookies["myMultiValcookie"]["I_Price"];
}

```

- Run Default.aspx into the browser fill the details as given in the figure and click on the "Submit" button. The values will be stored in the cookies and you will be redirected to the Default2.aspx. This page will fetch the values from the cookies and shows it in the various Labels.

Item No:	<input type="text" value="1"/>
Item Name:	<input type="text" value="Mobile"/>
Item Price:	<input type="text" value="15000"/>
Item Qty:	<input type="text" value="2"/>

Default.aspx



Default2.aspx

Program : 8

Take single image having 3 rectangle shapes horizontally having text “Home”, “Product” and “Services” written in the boxes. When user clicks on the first rectangle Home.aspx page should be opened. Similarly, when user clicks on the Product rectangle the product.aspx and Service rectangle then service.aspx should be opened. Use ImageMap control.

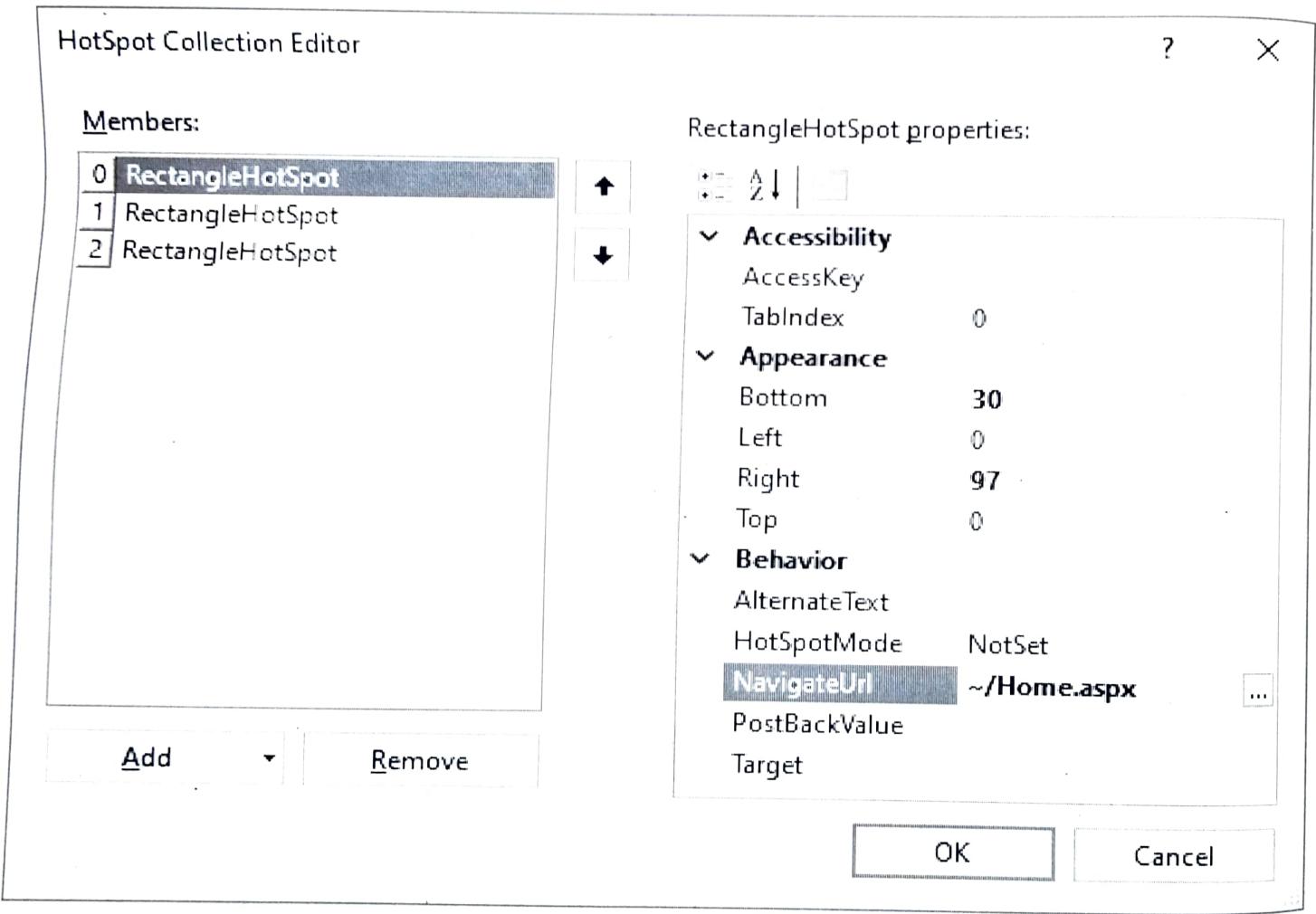
- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Place a ImageMap Control on the Default.aspx page from the ToolBox. We have created an image in the MS-Paint as shown below.



- Copy and paste the image in the website using Solution Explorer. Select ImageMap control and first set the ImageURL property to “ImageBar.jpg” (it nothing but the name of the Image we have shown above).
- Now Select ImageMap control and click on the ellipse (...) in the HotSpot property. It will open following dialog box. Now Expand Add Button by clicking on the small arrow in it and click on “RectangleHotSpot”. In the same way add 3 Rectangle HotSpots.

- Set the following Properties for each HotSpot.

Property	RectangleHotSpot 1	RectangleHotSpot 2	RectangleHotSpot 3
Bottom	30	30	30
Left	0	100	200
Right	97	197	297
Top	0	0	0
NavigateUrl	~/Home.aspx	~/Product.aspx	~/Services.aspx
I			



- Depends on your image, coordinates can defer. Use proper image editor or MS-Paint to know coordinates.
- Create 3 web pages called Home.aspx, Product.aspx and Services.aspx and design it. Run the default.aspx and click on the Different areas of the image like Home, Product or Service, it will redirect you to specific web page you have designed.

Program : 9

Using AdRotator control, display 3 images of car and when user click on it, open website of it. Load the advertisement details from the XML file as well as database.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Now to show images for rotating banner advertisement, we have taken 3 Images called BannerAd1.gif, BannerAd2.gif and BannerAd3.gif in the Directory “AdImages”. The whole directory we have copied in the website from Server Explorer. You do the same thing.
- Go to Default.aspx webpage and place an AdRotator control to it from ToolBox. Now Right click on the Solution Explorer and choose option “Add New Item”. From the “Add New Item”, dialog box select XML file and click on the “Add” button.
- This action will add a file called “XMLFile.xml” to your website. Write the following code to it.

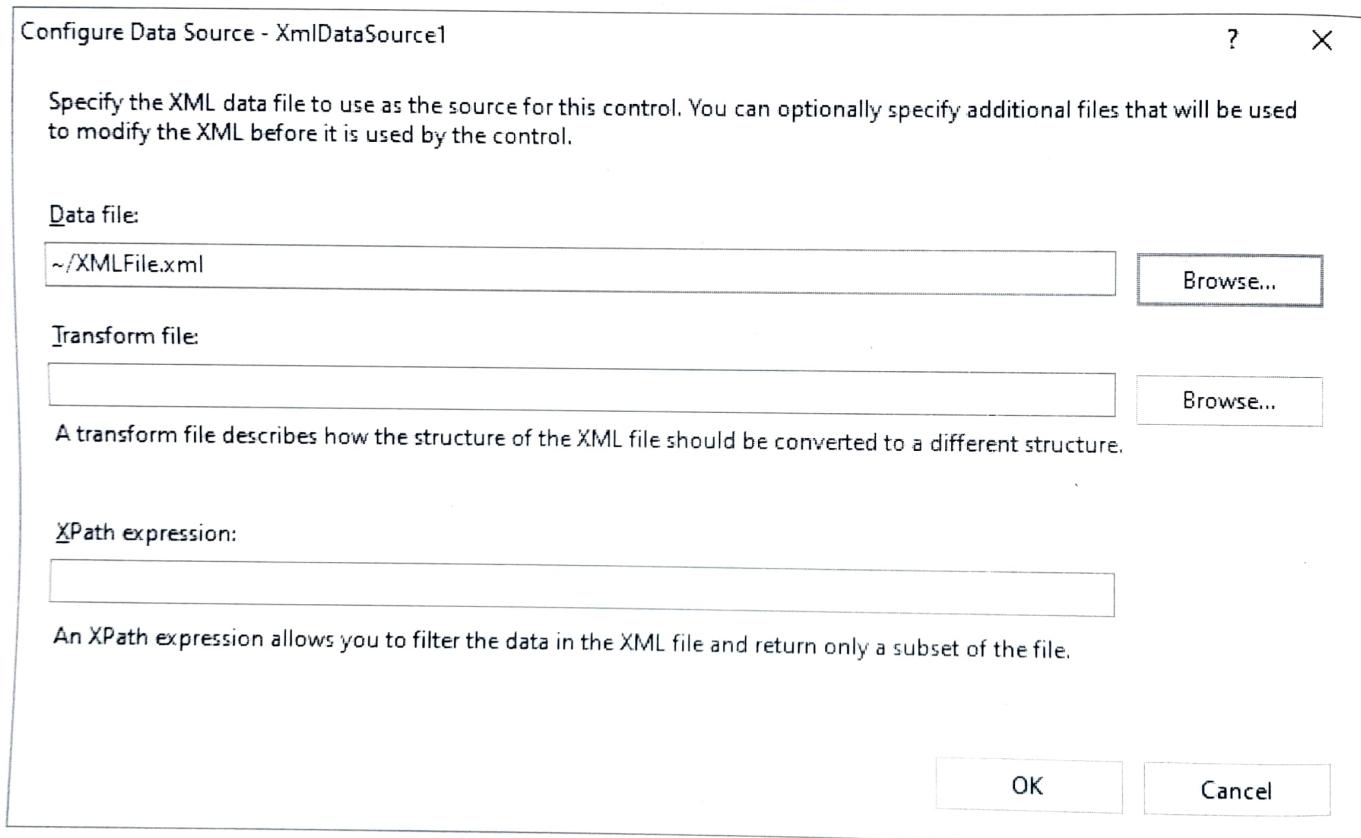
```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
    <Ad>
        <ImageUrl>~/AdImages/BannerAd1.gif</ImageUrl>
        <NavigateUrl>http://www.msn.com</NavigateUrl>
        <AlternateText>msn site</AlternateText>
        <Keyword>MSN</Keyword>
    </Ad>

    <Ad>
        <ImageUrl>~/AdImages/BannerAd2.gif</ImageUrl>
        <NavigateUrl>http://www.yahoo.com</NavigateUrl>
        <AlternateText>yahoo site</AlternateText>
        <Keyword>YAHOO</Keyword>
    </Ad>

    <Ad>
        <ImageUrl>~/AdImages/BannerAd3.gif</ImageUrl>
```

```
<NavigateUrl>http://www.indiatimes.com</NavigateUrl>
<AlternateText>Indiatimes site</AlternateText>
<Keyword>INDIATIMES</Keyword>
</Ad>
</Advertisements>
```

- Give the proper path in the <ImageUrl> tag. Now again come back to Default.aspx page and add “XMLDataSource” control to your webpage from the Data category of ToolBox.
- Now, click on the XMLDataSource1 control and click on the top-right arrow to open smart menu and click on “Configure Data Source” option. Select your XMLFile.xml in the Data file: section.

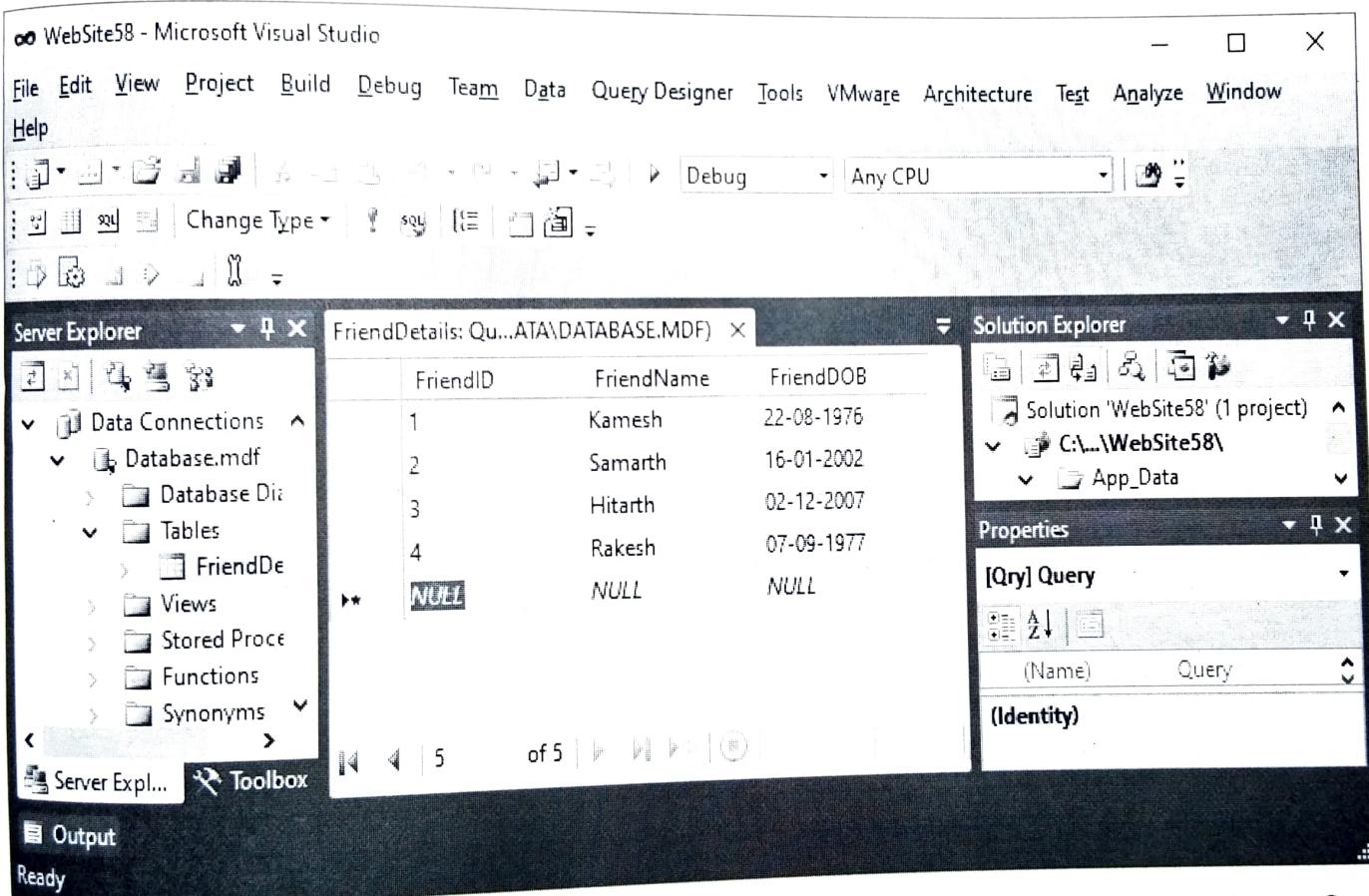


- Click on OK Button. Now click on the AdRotator control and open the smart menu of it. Choose option “Select Data Source” and assign “XMLDataSource1” to it.
- Run the we page in the browser multiple times. You will notice that every time one random image will be displayed. AdRotator control will take any one random advertisement from the file and show it on the webpage. If you click on the image you will be redirected to the site URL, you have written in the <NavigateUrl> tag of XML file. You can refresh the webpage by pressing F5 button to change the advertisement.

Program : 10

Using calendar control, allow user to select date from that. Display students whose birthday falls on that date (use database).

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button. This will add default.aspx page to your website.
- Add a Database to your website and Create one table called “FriendsDetails” having 3 fields:[1] FriendID int (Primary Key) [2] FriendName Text and [3] FriendDOB Date.
- Right click on the table and choose option “Show Table Data”. This will open the table in the datasheet mode where you can do necessary data entry.
- Do the following data entry in the table.



- Now go to Default.aspx control, and place a calendar control from the ToolBox. Open the smart menu choose the “AutoFormat” option and select proper Format for Calendar control.
- Write the following code on the DayRender event of the Calendar control.

```
System.Data.SqlClient.SqlConnection cn = new
System.Data.SqlClient.SqlConnection();
System.Data.SqlClient.SqlConnection();
```

```
cn.ConnectionString = "Data  
Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Dat  
abase.mdf;Integrated Security=True;User Instance=True";  
cn.Open();  
System.Data.SqlClient.SqlCommand cmd = new  
System.Data.SqlClient.SqlCommand();  
cmd.Connection = cn;  
cmd.CommandText = "Select * from FriendDetails";  
System.Data.SqlClient.SqlDataReader dr =  
cmd.ExecuteReader();  
while (dr.Read())  
{  
    DateTime dt = (DateTime)dr.GetValue(2);  
    DateTime dt1 = e.Day.Date;  
    if (dt.Month == dt1.Month)  
    {  
        if (Convert.ToString(dt.Day) ==  
e.Day.DayNumberText)  
        {  
            int n=(int)dr.GetValue(0);  
            LiteralControl l = new  
LiteralControl("<br /><a href=Default2.aspx?EventID=" + n  
+ ">BirthDay</a>");  
            e.Cell.Controls.Add(l);  
        }  
    }  
}
```

- Now add another page called Default2.aspx to your website and add SQLDataSource Control to the webpage from “Data” category of the ToolBox.
- Open the smart menu and click on the “Configure Data Source” option. Configure the data source as shown in the following figure.

CC-301 Web Application Development- I (Using C#)

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: FriendDetails

Columns:

*
 FriendID
 FriendName
 FriendDOB

Return only unique rows
WHERE...
ORDER BY...
Advanced...

SELECT statement:

```
SELECT * FROM [FriendDetails]
```

< Previous Next > **Finish** **Cancel**

- Click on WHERE button and set following things as shown in the figure given below. Click on the Add button and finally click on the OK button to complete the Wizard.

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column:	Parameter properties
FriendID	QueryString field:
Operator:	EventID
=	Default value:
Source:	
QueryString	
SQL Expression:	Value:
[FriendID] = @FriendID	Request.QueryString("EventID")
Add	
WHERE clause:	
SQL Expression	Value
Remove	
OK	
Cancel	

CC-301 Web Application Development- I (Using C#)

- Now Place a GirdView control or DataList control and set its “DataSource” property to SQLDataSource1.
- Run the webpage and View the calendar control. This month’s events will be displayed in the calendar.
- Navigate calendar control to Next month or Previous month to view the events of next month or previous month.
- Click on any Event you will be redirected to another page, which will give complete details of the event in the GridView or DataList control.

