Chapter No.7

ASP.Net

To

Database

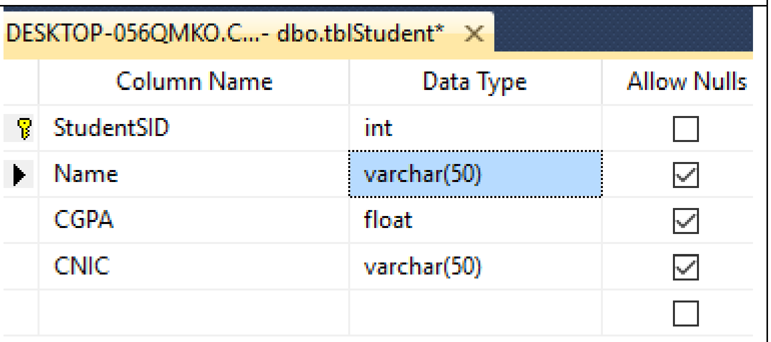
DAY7

Today we will learn,

- ✓ Filling the GridView Control using C# Code
- ✓ Filling the Dropdown List Control
- ✓ Using Stored Procedures to
    - Add New Record to Database
    - Deleting an Existing Database Record
    - Updating an Existing Database Record

1.1 Database Details

In order to complete the today’s Tasks, we assume the following database Detials

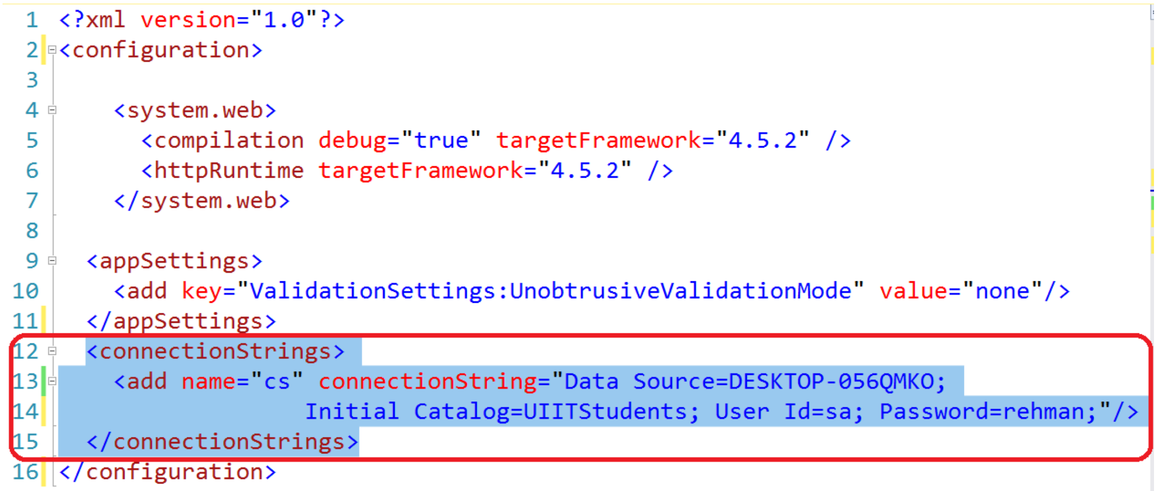
<table>
<tr><td><b>Server Name</b></td><td>DESKTOP-056QMKO</td></tr>
<tr><td><b>User Id</b></td><td>Sa</td></tr>
<tr><td><b>Password</b></td><td>Rehman</td></tr>
<tr><td><b>Database Name</b></td><td>UIITStudents</td></tr>
<tr><td><b>Table Name</b></td><td>tblStudents</td></tr>
<tr><td><b>Table Structure</b></td><td>
DESKTOP-056QMKO.C...- dbo.tblStudent* ✕
<table>
<tr><th></th><th>Column Name</th><th>Data Type</th><th>Allow Nulls</th></tr>
<tr><td>🔑</td><td>StudentSID</td><td>int</td><td>☐</td></tr>
<tr><td>▶</td><td>Name</td><td>varchar(50)</td><td>☑</td></tr>
<tr><td></td><td>CGPA</td><td>float</td><td>☑</td></tr>
<tr><td></td><td>CNIC</td><td>varchar(50)</td><td>☑</td></tr>
<tr><td></td><td></td><td></td><td>☐</td></tr>
</table></td></tr>
<tr><td colspan="2"><b>StudentSID</b> is a Primary key and it is auto generated</td></tr>
</table>

1.2 C# DataAccessLayer Class

In most of the application, developers use a generic class, in which they put the commonly used methods which they need almost in all web pages. These methods are included open database connection, filling the dataset, filling the dropdownlist, executing the DML queries etc. Once the methods are added to the data access layer class, these methods are then called from all the web pages within which these methods are needed. In addition to the commonly used methods, connection string is also handled in the data access layer.

Here are the steps you need to follow to design a C# class called **DataAccessLayer.CS**.

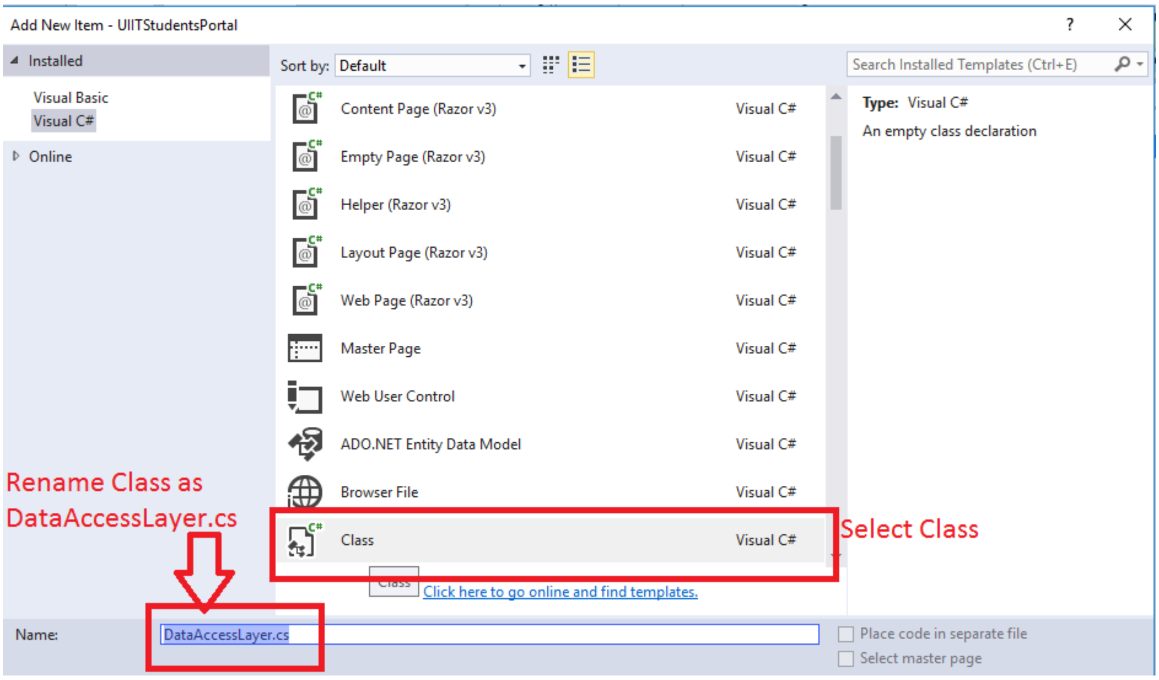
1. Start with new ASP.Net Empty Web Site, Rename it as UIITPortal
2. Open the web.config file and put the Connection String in web.config file. The connection string in the web.config file is shown here



```
<?xml version="1.0"?>
<configuration>

    <system.web>
      <compilation debug="true" targetFramework="4.5.2" />
      <httpRuntime targetFramework="4.5.2" />
    </system.web>

  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="none"/>
  </appSettings>
  <connectionStrings>
    <add name="cs" connectionString="Data Source=DESKTOP-056QMKO;
                  Initial Catalog=UIITStudents; User Id=sa; Password=rehman;"/>
  </connectionStrings>
</configuration>
```

3. Right click on the Web Site name, then Choose Add ➔ Add New Item

4. Select Class, rename it as DataAccessLayer.cs and press the Add button.



Add New Item - UIITStudentsPortal
?
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
Content Page (Razor v3) Visual C#
Empty Page (Razor v3) Visual C#
Helper (Razor v3) Visual C#
Layout Page (Razor v3) Visual C#
Web Page (Razor v3) Visual C#
Master Page Visual C#
Web User Control Visual C#
ADO.NET Entity Data Model Visual C#
Browser File Visual C#
Class Visual C#
Click here to go online and find templates.
□ Place code in separate file
□ Select master page

Type: Visual C#

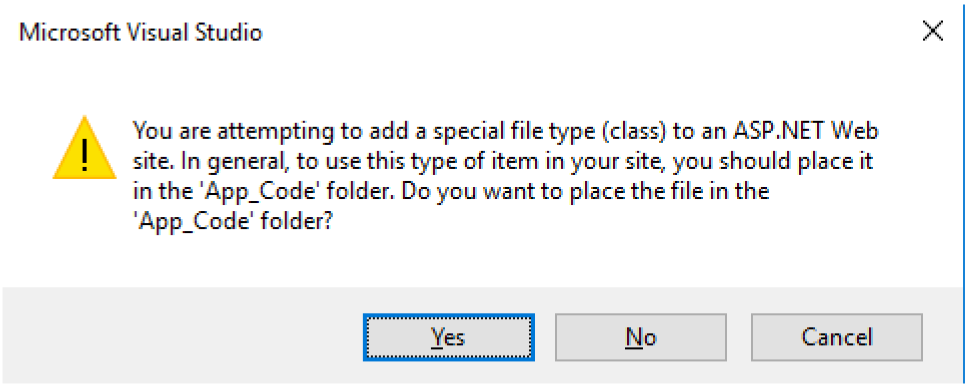
An empty class declaration

Name: DataAccessLayer.cs

Rename Class as DataAccessLayer.cs

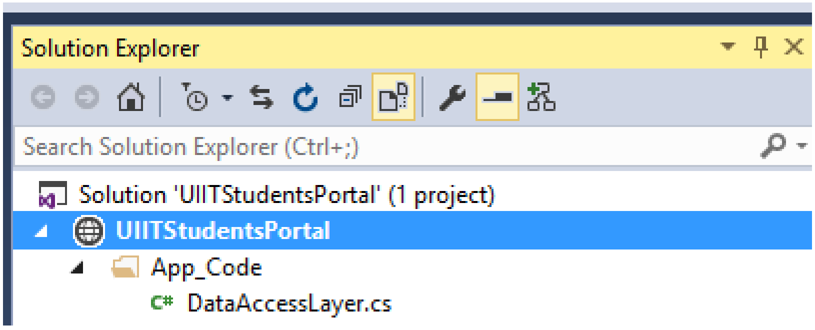
Select Class

5. Here, you will notice that Visual Studio prompt you to add the Class to App_Code folder. Press the Yes button



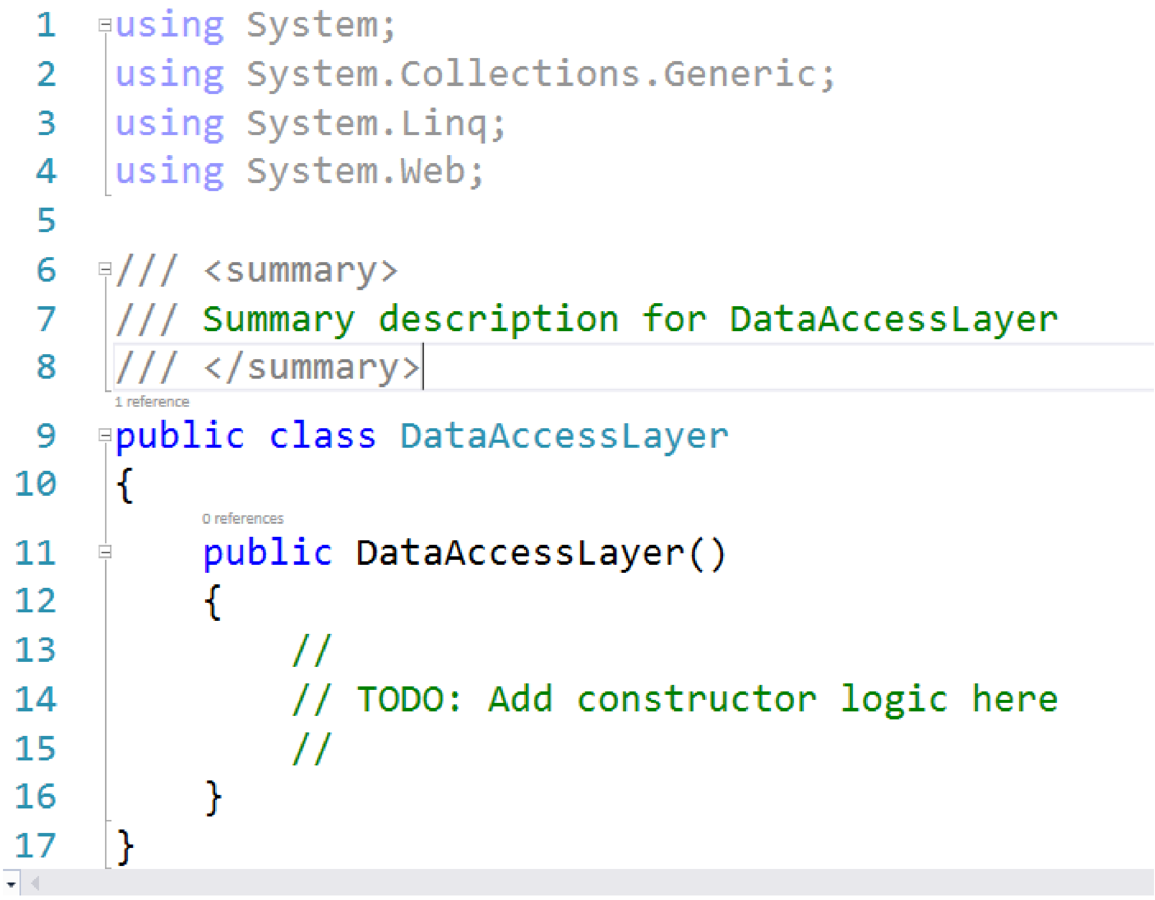
You are attempting to add a special file type (class) to an ASP.NET Web site. In general, to use this type of item in your site, you should place it in the 'App_Code' folder. Do you want to place the file in the 'App_Code' folder?

6. You will see that a C# class with name DataAccessLayer.cs is added to your web site under the App_Code folder as shown



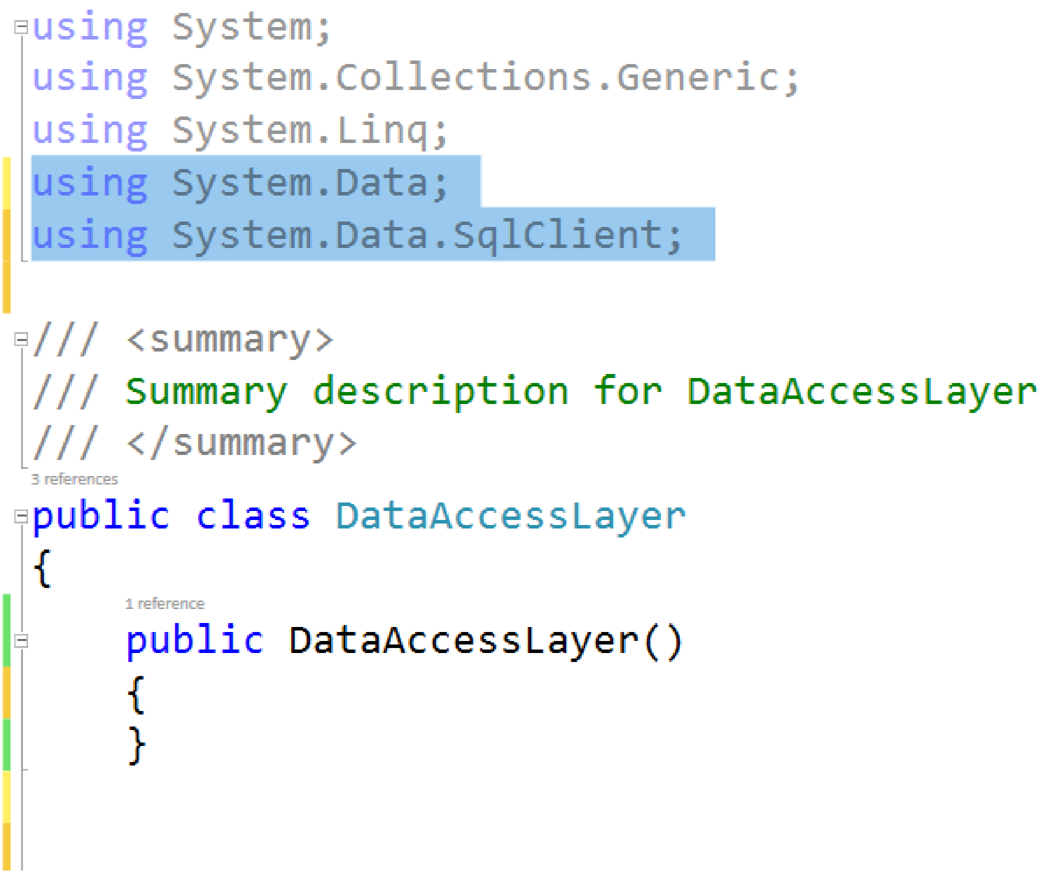
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'UIITStudentsPortal' (1 project)
UIITStudentsPortal
App_Code
DataAccessLayer.cs

7. Open the DataAccessLayer class code



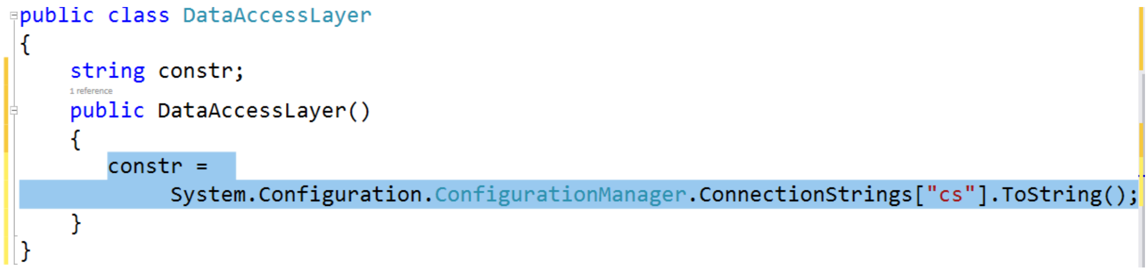
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for DataAccessLayer
/// </summary>
public class DataAccessLayer
{
    public DataAccessLayer()
    {
        //
        // TODO: Add constructor logic here
        //
    }
}
```

8. Now add the following two class libraries to the DataAccessLayer class as shown



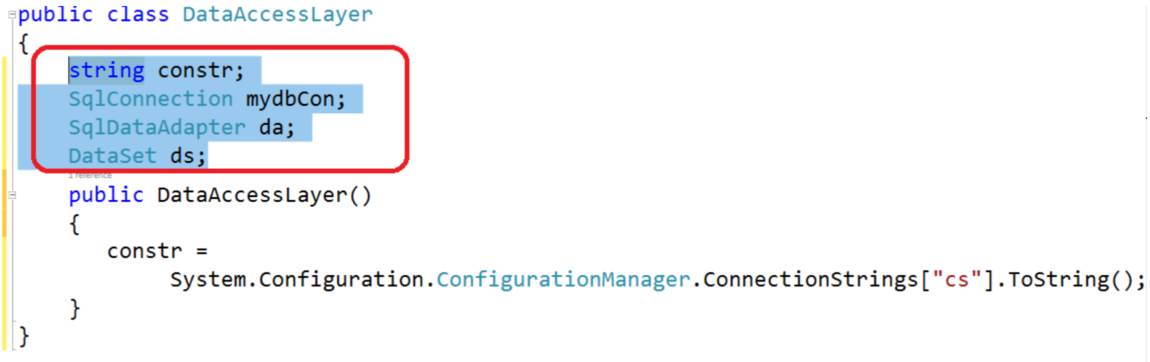
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data;
using System.Data.SqlClient;

/// <summary>
/// Summary description for DataAccessLayer
/// </summary>
public class DataAccessLayer
{
    public DataAccessLayer()
    {
    }
```

9. Next, declare a constr variable of type string and retrieve the value of connection string from the web.config file by the following code



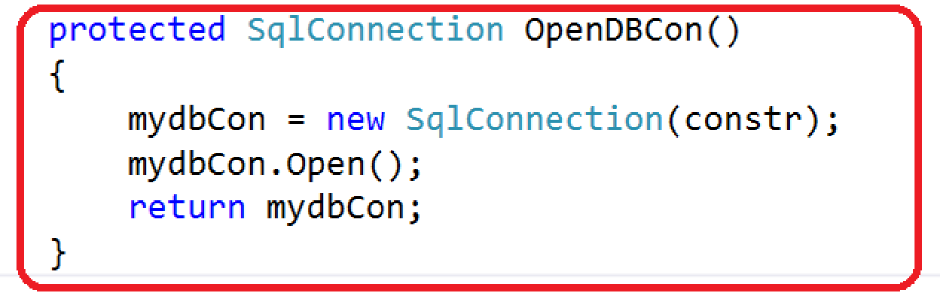
```
public class DataAccessLayer
{
    string constr;
    public DataAccessLayer()
    {
        constr =
            System.Configuration.ConfigurationManager.ConnectionStrings["cs"].ToString();
    }
}
```

10. Open database connection is very commonly used method. Next, we add the open database connection method, name the method as **<u>OpenDBCon()</u>** with protected scope and SQLConnection as return type. Before defining Open database connection method, declare the SQLConneciton, SQLDataAdapter, Dataset object above at class level as shown here



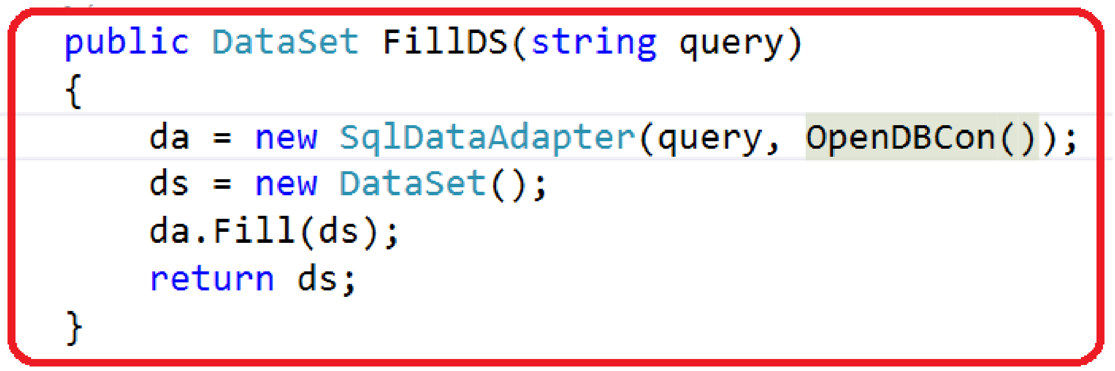
```
public class DataAccessLayer
{
    string constr;
    SqlConnection mydbCon;
    SqlDataAdapter da;
    DataSet ds;
    public DataAccessLayer()
    {
        constr =
            System.Configuration.ConfigurationManager.ConnectionStrings["cs"].ToString();
    }
}
```

Thus, the OpenDBCon() is given by, this method is added after the constructor , which is shown in the above figure



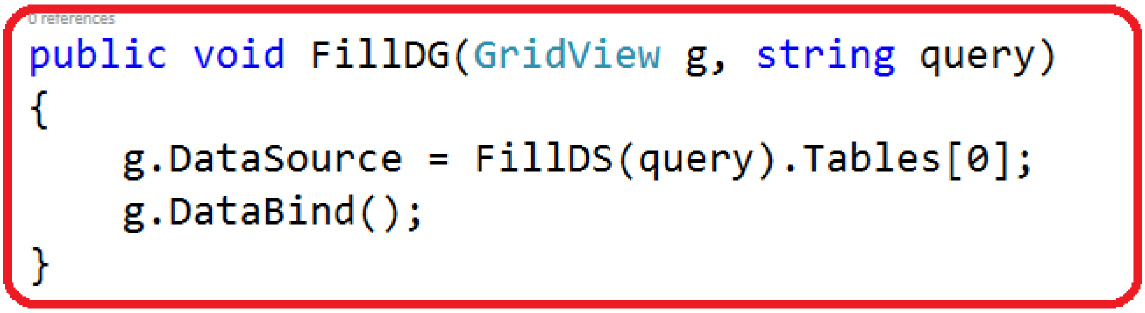
```
protected SqlConnection OpenDBCon()
{
    mydbCon = new SqlConnection(constr);
    mydbCon.Open();
    return mydbCon;
}
```

11. Next, most commonly used method is for filling the dataset. Define method **FillDS(string query)** with protected scope and Dataset as its return type. This method will take query as string type parameter, which represents the SQL Select Query which will be forward to it to fill in the dataset. Put the code of the FillDS() method well after the OpenDBCon() method. The method is shown here.



```
public DataSet FillDS(string query)
{
    da = new SqlDataAdapter(query, OpenDBCon());
    ds = new DataSet();
    da.Fill(ds);
    return ds;
}
```

12. The other method used frequently in ASP.Net application is the filling the GridView control. To fill a GridView control, we need the filled Dataset. Here, we design a **FillDG(string query, GridView gv)** method. We will pass SQL select query as string to FillDG() method along with the GridView name to be filled with. The query passed to FillDG() will be further passed to FillDS() method which will return the FillDG() method filled Dataset. Complete Code of FillDG() is shown here



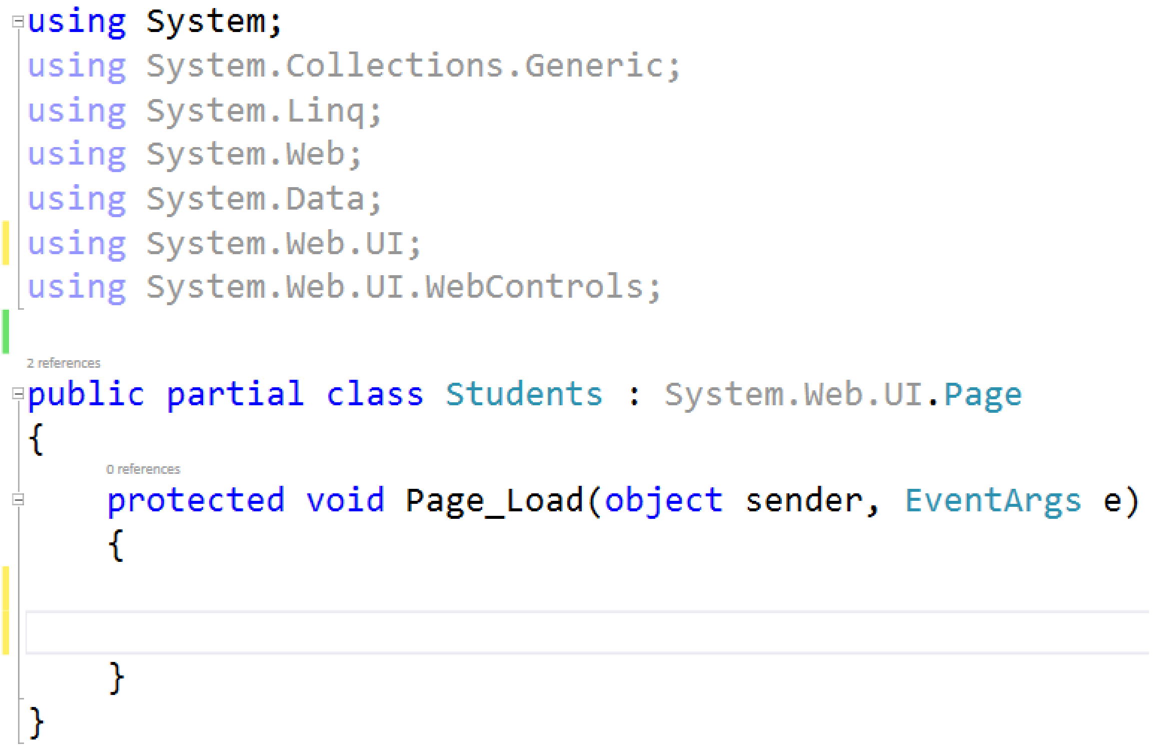
```
public void FillDG(GridView g, string query)
{
    g.DataSource = FillDS(query).Tables[0];
    g.DataBind();
}
```

13. We have completed the DataAccessLayer.CS for our first task, in which we will fill the Gridview Contol. Complete code of DataAccessLayer.CS class is shown here

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

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

using System.Data;
using System.Data.SqlClient;

public class DataAccessLayer
{
    string constr;
    SqlConnection mydbCon;
    SqlDataAdapter da;
    DataSet ds;
    public DataAccessLayer()
    {
        constr =
System.Configuration.ConfigurationManager.ConnectionStrings["cs"].ToString();
    }
    protected SqlConnection OpenDBCon()
    {
        mydbCon = new SqlConnection(constr);
        mydbCon.Open();
        return mydbCon;
    }
    public DataSet FillDS(string query)
    {
        da = new SqlDataAdapter(query, OpenDBCon());
        ds = new DataSet();
        da.Fill(ds);
        return ds;
    }

    public void FillDG(GridView g, string query)
    {
        g.DataSource = FillDS(query).Tables[0];
        g.DataBind();
    }

}
```

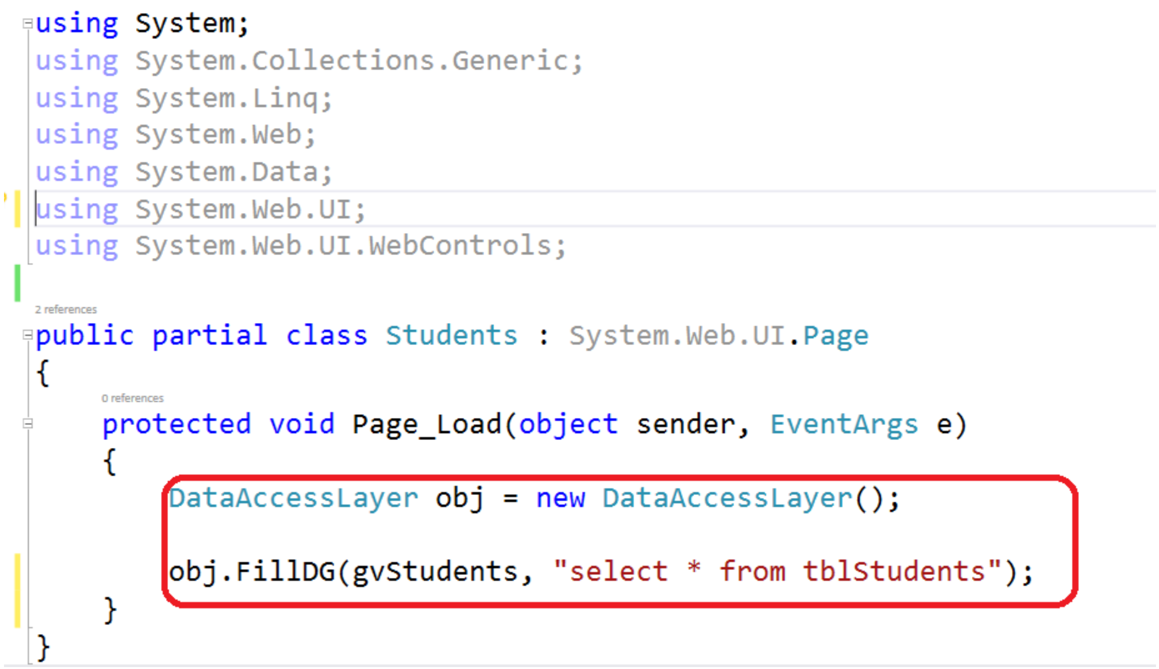
1.3 Binding GridView Using DataAccessLayer.CS class

1. Add a new ASP.Net Web Page in the UIITPortal Web Site, which we have created in Section 7.2 of this Chapter. Rename the Web Page as Students.aspx
2. Drag and Drop a GridView on Students.aspx page. Rename the GridView as **gvStudents**
3. Open the Code Behind File (.CS file of the Students.aspx) of the Students.aspx Page, as shown here



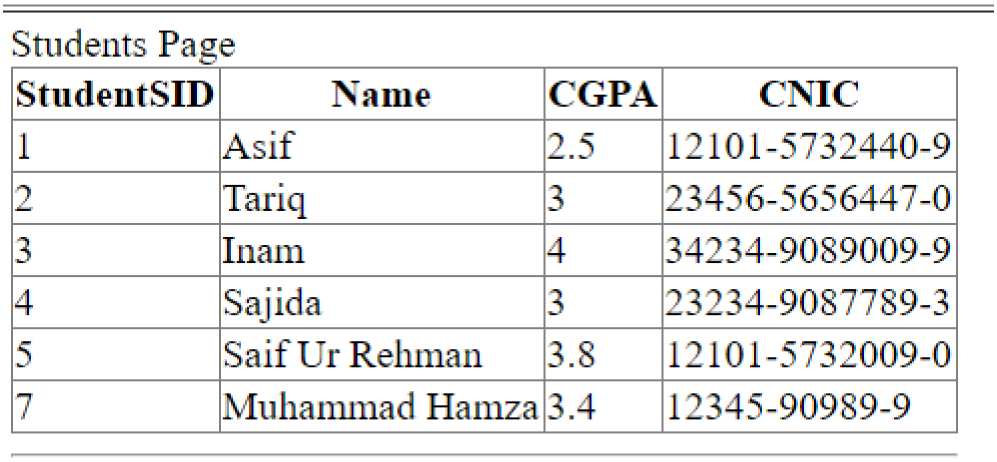
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Web.UI;
using System.Web.UI.WebControls;

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


    }
}
```

4. In the Page_Load() event add the code as shown here



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

public partial class Students : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        DataAccessLayer obj = new DataAccessLayer();

        obj.FillDG(gvStudents, "select * from tblStudents");
    }
}
```

5. Save and Run the Web Site, you will see the following output

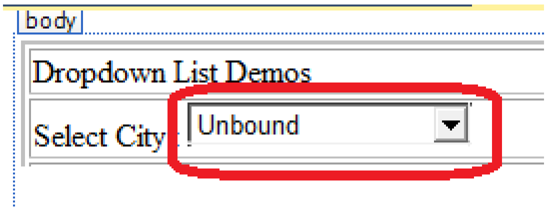


Students Page

StudentSID	Name	CGPA	CNIC
1	Asif	2.5	12101-5732440-9
2	Tariq	3	23456-5656447-0
3	Inam	4	34234-9089009-9
4	Sajida	3	23234-9087789-3
5	Saif Ur Rehman	3.8	12101-5732009-0
7	Muhammad Hamza	3.4	12345-90989-9

1.4 Filling the Dropdown List Control

Dropdown List are most commonly used web controls in the ASP.Net Web Applications. DropDownList Control is shown here



body

Dropdown List Demos

Select City: Unbound

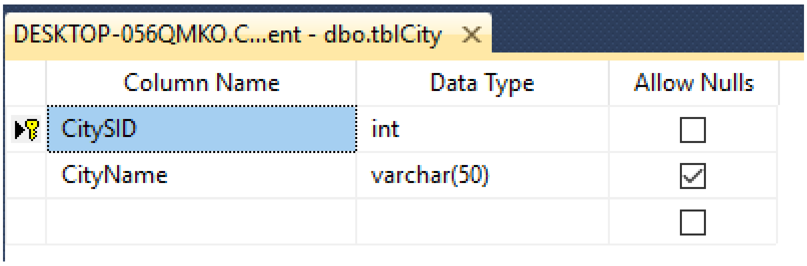
This control is used to display the multiple choices to users and user can select one choice Normally, when there are more than five choices then developers/programmers prefer the dropdown list then using the Radio Buttons.

The Steps to fill the DropDownList control are

- Fill the Dataset Object
- Assign the Dataset Object to DropDownList, as we did with the GridView Control
- Choose the Field to be displayed in the Web Page
- Choose the Field which will be used as reference (This field is normally the Primary Key of the Table, which will be used to save in the Referring Table. Referring Table mean where this Field is used as Foreign Key).
-

Here, first we will shown the simple code to fill in the DropDownList and then we will put the code in the DataAccessLayer.CS class.

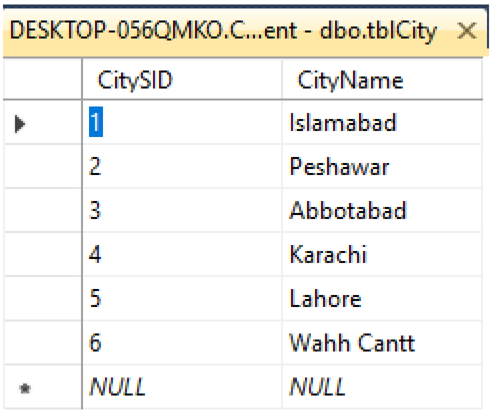
- Design a table in the Database and name it as tblCity. Structure of the table is shown here



DESKTOP-056QMKO.C...ent - dbo.tblCity

Column Name	Data Type	Allow Nulls
CitySID	int	□
CityName	varchar(50)	☑
		□

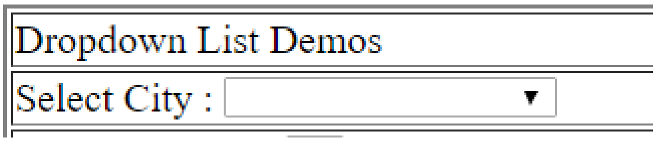
- Add few Records to the Table



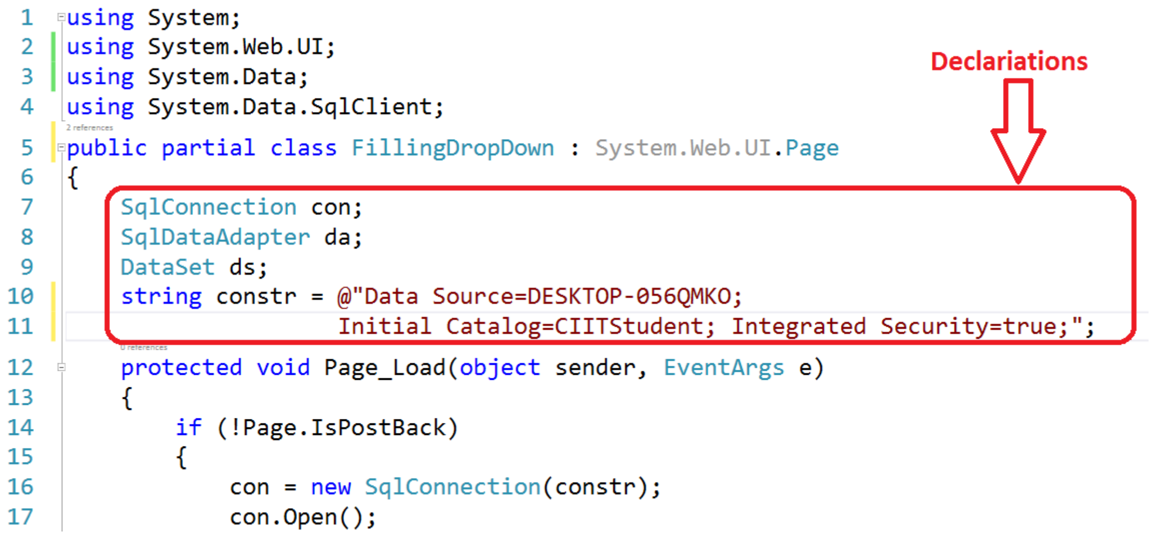
DESKTOP-056QMKO.C...ent - dbo.tblCity

	CitySID	CityName
▶	1	Islamabad
	2	Peshawar
	3	Abbotabad
	4	Karachi
	5	Lahore
	6	Wahh Cantt
*	NULL	NULL

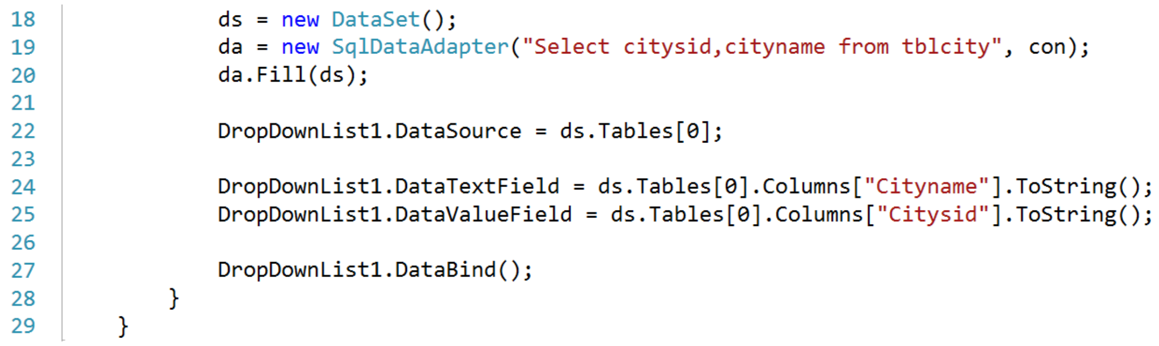
- Now, Open the Web Site **UIITStudentsPortal** and Design an ASP.Net Web Form. Rename the Web Form as **FillingDropDown.aspx**. Drag and drop a DropDownList from ToolBox, rename DropDownList as ddCity



- Open the Code behind file (C# Code of the FillingDropDown.aspx Page). Add the following code in the Page_Load() event



```
1  using System;
2  using System.Web.UI;
3  using System.Data;
4  using System.Data.SqlClient;
5  public partial class FillingDropDown : System.Web.UI.Page
6  {
7      SqlConnection con;
8      SqlDataAdapter da;
9      DataSet ds;
10     string constr = @"Data Source=DESKTOP-056QMKO;
11                     Initial Catalog=CIITStudent; Integrated Security=true;";
12     protected void Page_Load(object sender, EventArgs e)
13     {
14         if (!Page.IsPostBack)
15         {
16             con = new SqlConnection(constr);
17             con.Open();
```



```
18             ds = new DataSet();
19             da = new SqlDataAdapter("Select citysid,cityname from tblcity", con);
20             da.Fill(ds);
21 
22             DropDownList1.DataSource = ds.Tables[0];
23 
24             DropDownList1.DataTextField = ds.Tables[0].Columns["Cityname"].ToString();
25             DropDownList1.DataValueField = ds.Tables[0].Columns["Citysid"].ToString();
26 
27             DropDownList1.DataBind();
28         }
29     }
```

Line 14

```
if (!Page.IsPostBack)
```

This line applies ***if statement*** to avoid the code execution each time the Web page is refreshed. If we delete the line No. 14, then the code inside the Page_Load() event will be executed each time the page is refreshed (It get executed when ant button on the page is clicked, when the dropdown list index is changed etc). Therefore, to avoid the again and again execution of the code inside the Page_Load() event, developers/programmers enclose the code in

```
if (!Page.IsPostBack)
{
```

//Put Code here, which you want to execute just once

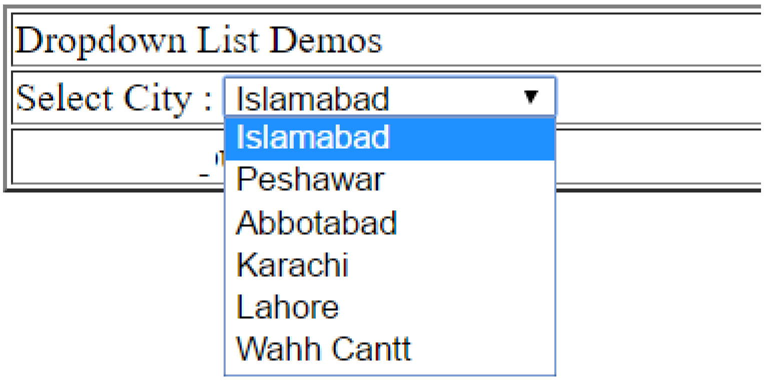
}

Line 24 & 25

```
        DropDownList1.DataTextField = ds.Tables[0].Columns["Cityname"].ToString();
        DropDownList1.DataValueField = ds.Tables[0].Columns["Citysid"].ToString();
```

Line No. 24, specifies the column to be displayed in the dropdown list at run time, and **Line No. 25** specifies the ID of the value selected in the dropdown list control at run time. The ID value is normally used to be Inserted in the table where this ID is used as a Foreign Key. We will show this in practical in the upcoming days.

Finally, Save the code and run the web site,

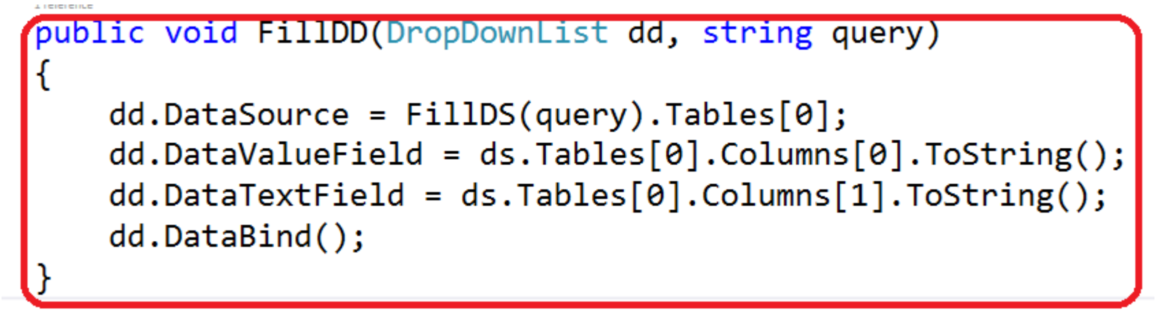

Dropdown List Demos

Select City : Islamabad

1.5 Filling the DropdownList Method from DataAccessLayer.CS

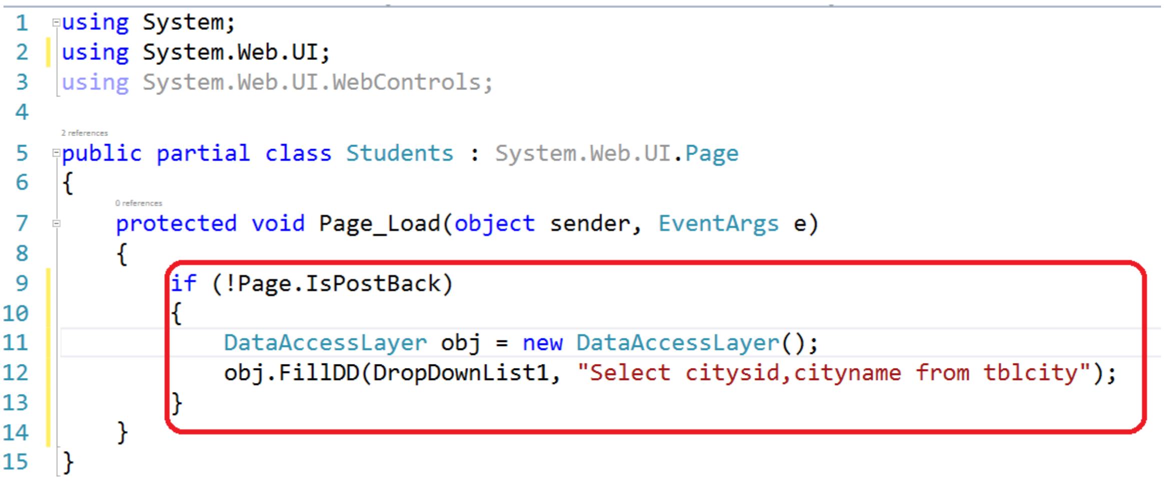
Next, we put the code of filling the dropdown list in the **DataAcessLayer.CS** file, so that we can call this method from all of other ASP.Net web forms where we need to use the dropdown list. As we did with GridView , add the FillDD() method to DataAccessLayer.CS file. Here we need to pass the two parameters (1)- SELECT Query and (2)- name of the DropDownList to be filled to the FillDD() method.

- Open the **<u>DataAcessLayer.CS</u>**
- Add the following code in this file as shown here



```
public void FillDD(DropDownList dd, string query)
{
    dd.DataSource = FillDS(query).Tables[0];
    dd.DataValueField = ds.Tables[0].Columns[0].ToString();
    dd.DataTextField = ds.Tables[0].Columns[1].ToString();
    dd.DataBind();
}
```

- Next, call the FillDD() from the code given in the last section 7.4, by removing all the code and just calling the method FillDD() with appropriate parameters, as shown here



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

public partial class Students : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            DataAccessLayer obj = new DataAccessLayer();
            obj.FillDD(DropDownList1, "Select citysid,cityname from tblcity");
        }
    }
}
```

- You can notice that code is optimized and now everything is done DataAccessLayer.cs file. We just call the method.

Note:

<u>It is important to note that while filling the dropdown list control using the DataAccessLAyer.CS class, always keep keep the ID column first in the query and the name field as second column. This is because we are using column 0 as DataValueField and column 1 as DataTextField in the DataAccessLayer.cs file</u>

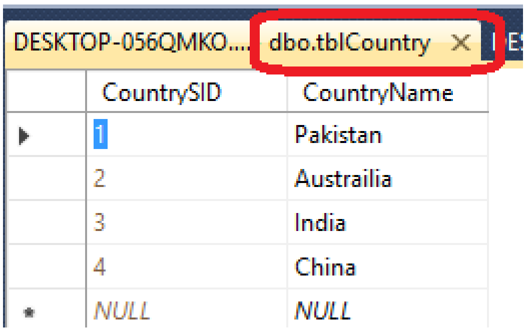
1.6 Cascading DropDownLists in ASP.Net

Cascading DropDownList means a series of dependent DropDownLists where one DropDownList is dependent on the parent or previous DropDownList and is populated based on the item selected by the user. On many occasions we need to make use of Cascading DropDownLists.

Follow the steps to design the cascading DopdownLists

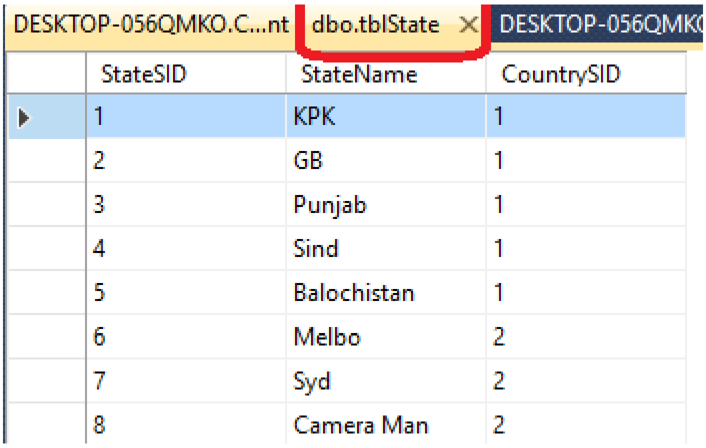
- Consider the following three Tables with sample data in the **CIITStudent** database

  Table 1: tblCountry



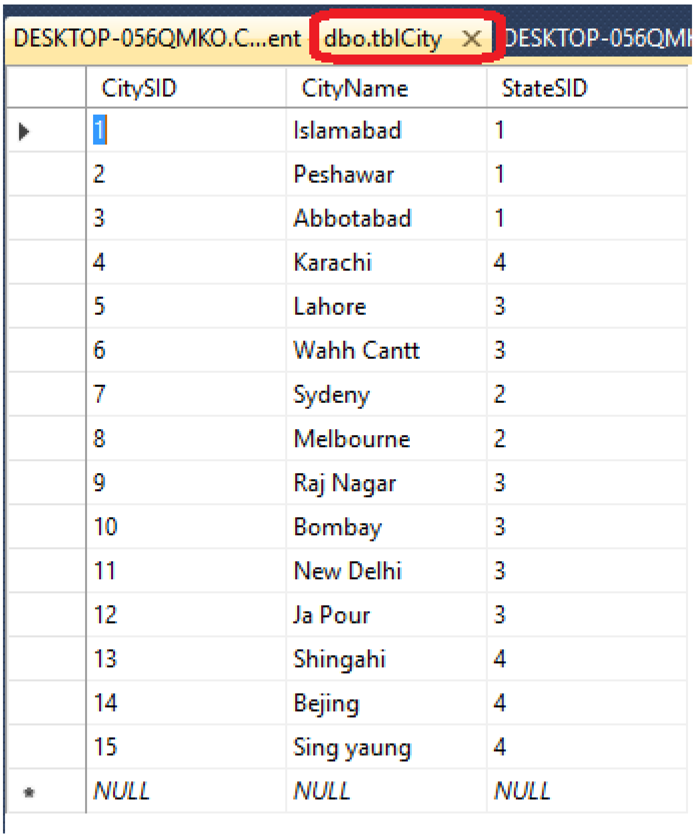
CountrySID	CountryName
1	Pakistan
2	Austrailia
3	India
4	China
NULL	NULL

Table 2: tblState CountrySID is the Foreign Key



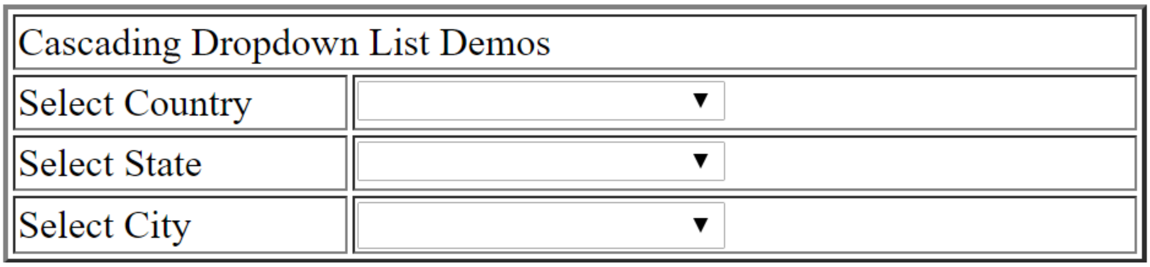
StateSID	StateName	CountrySID
1	KPK	1
2	GB	1
3	Punjab	1
4	Sind	1
5	Balochistan	1
6	Melbo	2
7	Syd	2
8	Camera Man	2

Table 3 : tblCity StateSID is the Foreign Key



	CitySID	CityName	StateSID
▶	1	Islamabad	1
	2	Peshawar	1
	3	Abbotabad	1
	4	Karachi	4
	5	Lahore	3
	6	Wahh Cantt	3
	7	Sydeny	2
	8	Melbourne	2
	9	Raj Nagar	3
	10	Bombay	3
	11	New Delhi	3
	12	Ja Pour	3
	13	Shingahi	4
	14	Bejing	4
	15	Sing yaung	4
*	NULL	NULL	NULL

- Design the ASP.Net form as shown

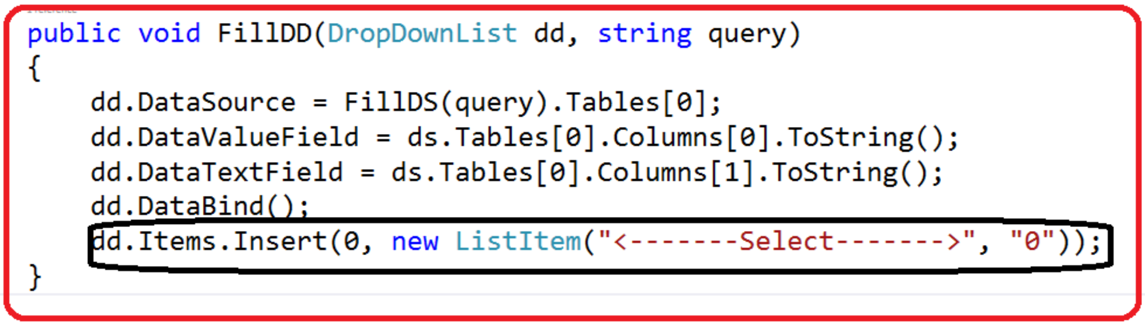


Cascading Dropdown List Demos	
Select Country	▼
Select State	▼
Select City	▼

Set the properties as follow

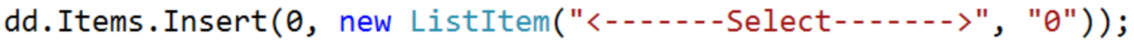
Control Name	**Property**	**Property Value**
DropDownList1	Name AutoPostBack	ddCountry True
DropDownList2	Name AutoPostBack	ddState True
DropDownList3	Name	ddCity

- For this demo, we will use the **DataAccessLayer.CS** file code. **DataAccessLayer** class contains a method called **FillDD()**, this is shown here. We have added this method in the last pages.



```
public void FillDD(DropDownList dd, string query)
{
    dd.DataSource = FillDS(query).Tables[0];
    dd.DataValueField = ds.Tables[0].Columns[0].ToString();
    dd.DataTextField = ds.Tables[0].Columns[1].ToString();
    dd.DataBind();
    dd.Items.Insert(0, new ListItem("<-------Select------->", "0"));
}
```

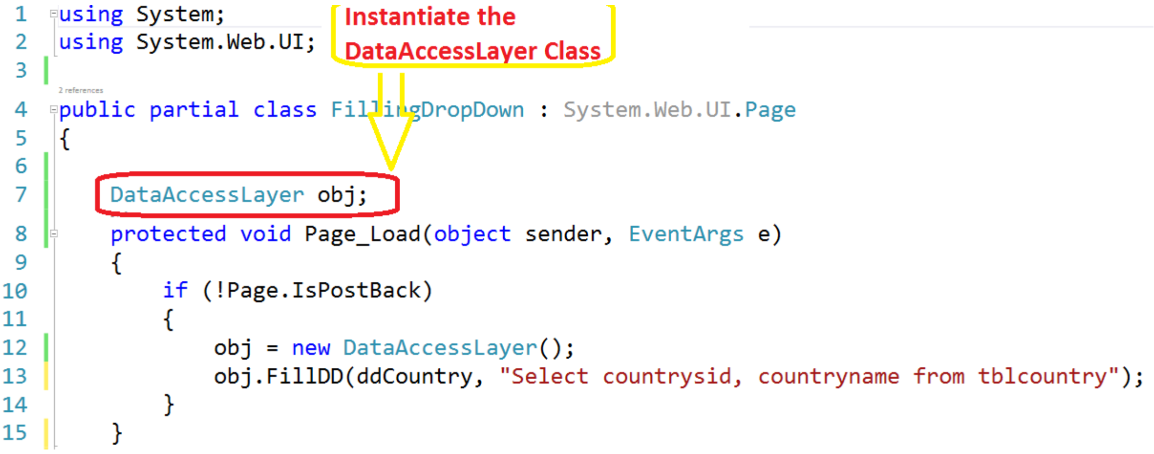
We have added an additional line in the FillDD() method, which is



```
dd.Items.Insert(0, new ListItem("<-------Select------->", "0"));
```

This line will add < -----Select ---- > text at the 0 index.

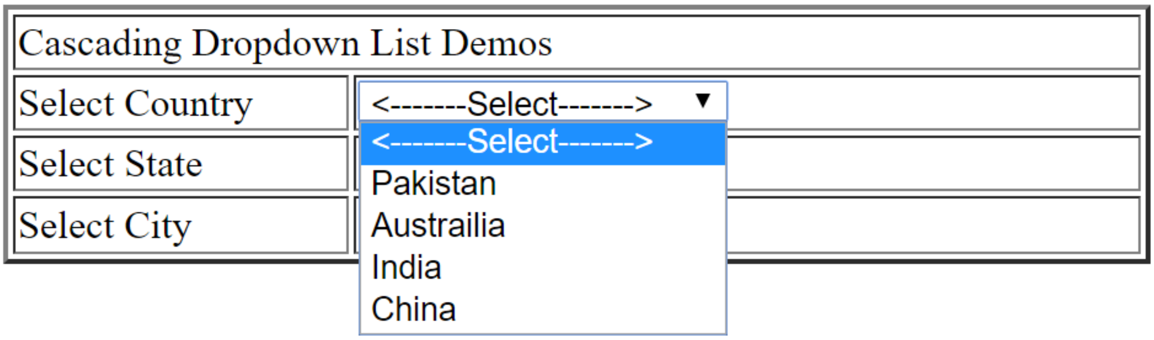
- Now, open the C# code. Go to **Page_Load()** event and call FillDD() with Query and dropdownlist name as shown



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

2 references
public partial class FillingDropDown : System.Web.UI.Page
{

    DataAccessLayer obj;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            obj = new DataAccessLayer();
            obj.FillDD(ddCountry, "Select countrysid, countryname from tblcountry");
        }
    }
```

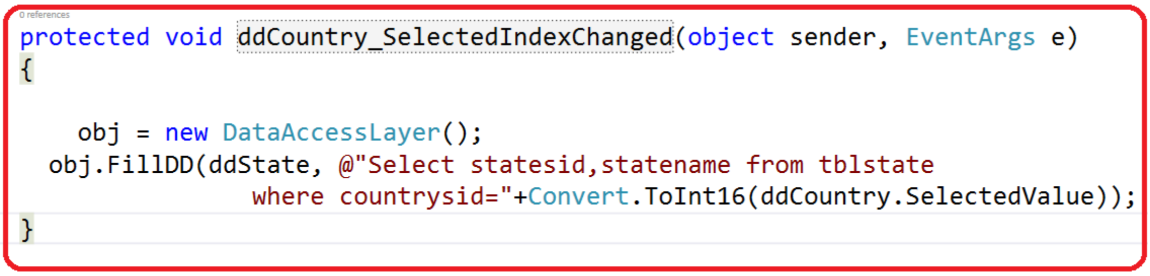
Instantiate the DataAccessLayer Class

Save and browse the web site.



Cascading Dropdown List Demos	
Select Country	<-------Select-------> ▼
Select State	
Select City	

- Next, step is to fill the State dropdownlist. First select the **ddCountry** dropdownlist, change its **AutoPostBack** Property to **TRUE.** Double Click on the ddCountry in the design view, this will create the following ddCountry event. This event is called every time when the index or selection of the country name is changed.



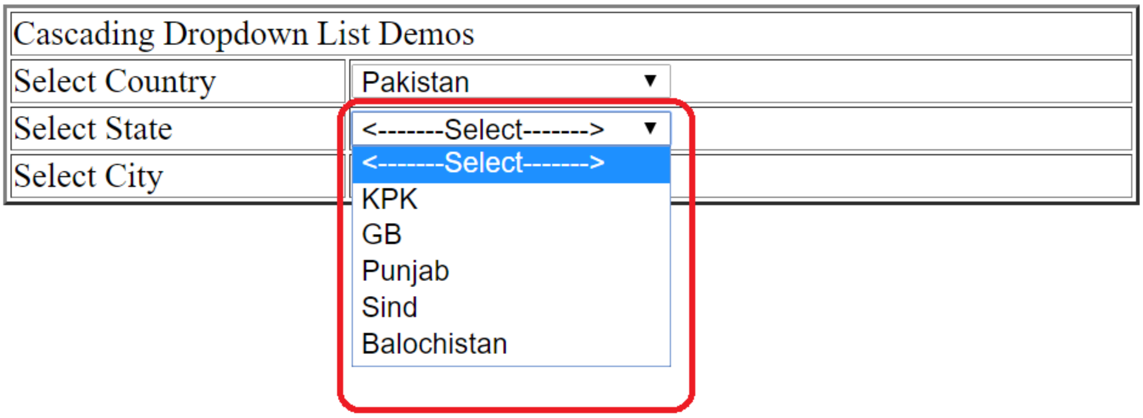
```
0 references
protected void ddCountry_SelectedIndexChanged(object sender, EventArgs e)
{

    obj = new DataAccessLayer();
  obj.FillDD(ddState, @"Select statesid,statename from tblstate
                where countrysid="+Convert.ToInt16(ddCountry.SelectedValue));
}
```

- The query to fill the ddState dropdownlist is

```
      Select statesid,statename from tblstate
where countrysid="+Convert.ToInt16(ddCountry.SelectedValue)
```

*Here, query contains the **Where clause** to check which country is selected. Based on the selected country the **CountrySID** is retrieved and used in the **Where clause** of the query to fill only those states against the selected country.*

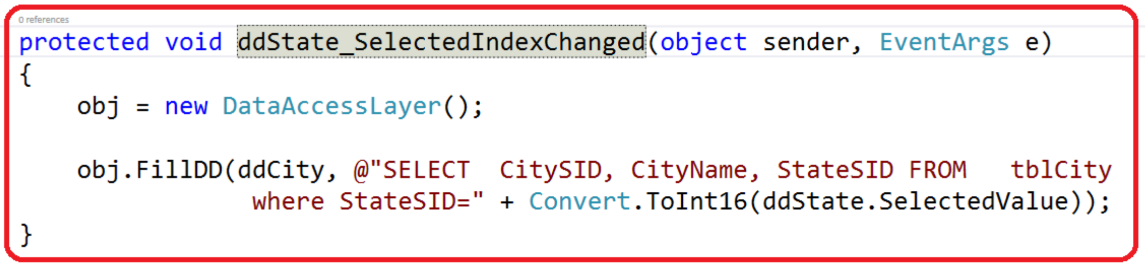
- Be remember, that only the ddCountry_SelectedIndexChanged() event is executed when the **AutoPostBack** property of the ddCountry is set to **TRUE**
- Save and browse the Web site again



Cascading Dropdown List Demos	
Select Country	Pakistan ▼
Select State	<-------Select-------> ▼
Select City	

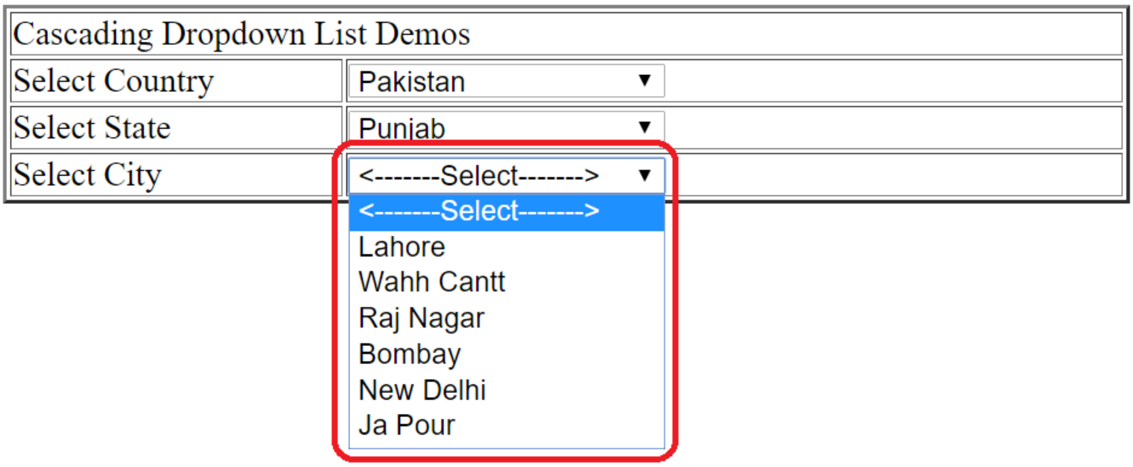
<-------Select------->
KPK
GB
Punjab
Sind
Balochistan

- Similarly, select the ddState dropdwonlist. Change its **AutoPostBack** Property to **TRUE** and double click on the ddState dropdwonlist to generate its Selected_IndexChanging() event.



```
protected void ddState_SelectedIndexChanged(object sender, EventArgs e)
{
    obj = new DataAccessLayer();

    obj.FillDD(ddCity, @"SELECT  CitySID, CityName, StateSID FROM   tblCity
                where StateSID=" + Convert.ToInt16(ddState.SelectedValue));
}
```

- Save and Run the web site



Cascading Dropdown List Demos	
Select Country	Pakistan ▼
Select State	Punjab ▼
Select City	<-------Select-------> ▼

<-------Select------->
Lahore
Wahh Cantt
Raj Nagar
Bombay
New Delhi
Ja Pour

Congratulations! You have successfully Bind the FormView Control to Database Table

These ASP.Net Tutorials are prepared only for the purpose of learning ASP.Net with C#. Anyone can copy, print or reuse it in any format

Saif Ur Rehman Saifi
Assistant Professor, UIIT
PMAS Arid Agriculture University, Rawalpindi

If you found this tutorial helpful or you want to give us any valuable suggestions contact at www.ASPUIIT.blogspot.com