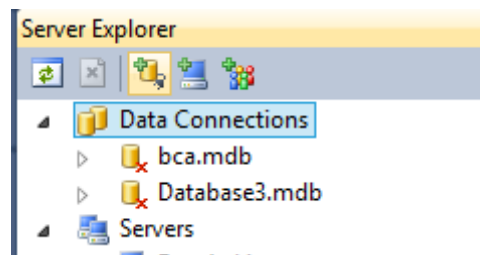


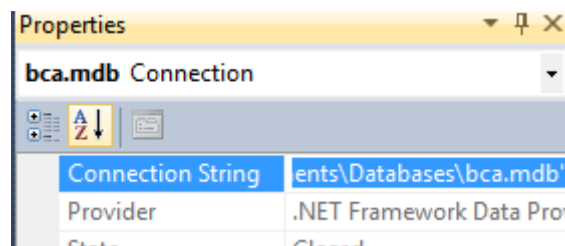
ASP.NET PRACTICALS (24 TO 40)

24) Create ASP.NET Webpage that will test Database connecton.

- Create Database in Microsoft Access Or SQLServer.
- Open Visual Studio and Create new website.
- Go to Server Explorer and Add new Database Connection.



Right click on your database and select properties. find connection string property and copy it.



.aspx

```
<div><asp:GridView ID="GridView1" runat="server"></asp:GridView> </div>
```

.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    SqlConnection con = new
    SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString
    (Sem_5_E)"].ConnectionString);

    con.Open();

    if (con != null)
    {
```

```

Response.Write(" conncted");

SqlDataAdapter ol = new SqlDataAdapter("SELECT * FROM T1", con);

DataSet ds = new DataSet();

ol.Fill(ds);

GridView1.DataSource = ds;

GridView1.DataBind();

}

else { Response.Write("Not conncted"); }

con.Close();

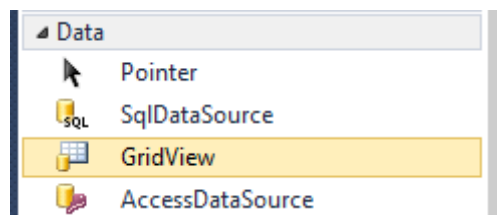
}

}

```

25) Create ASP.NET webpage that will display data in Gridview Control using SqlDataSource.

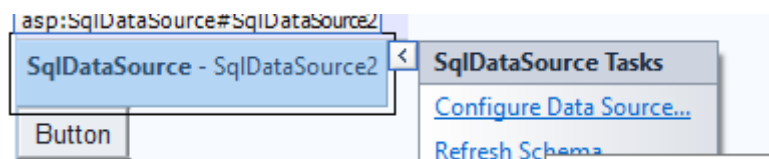
- Add SqlDataSurce – goto Toolbox > Data > SqlDataSource then add GridView .

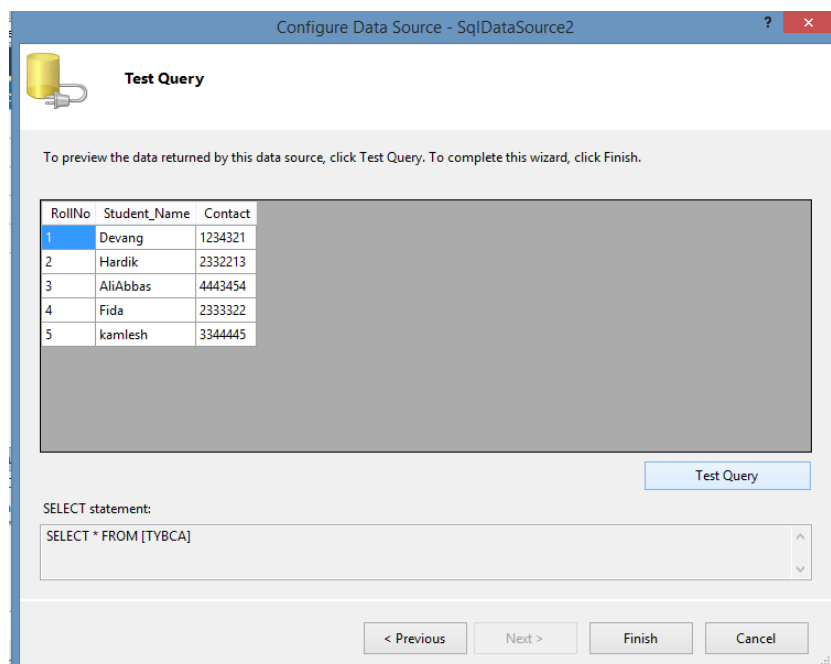
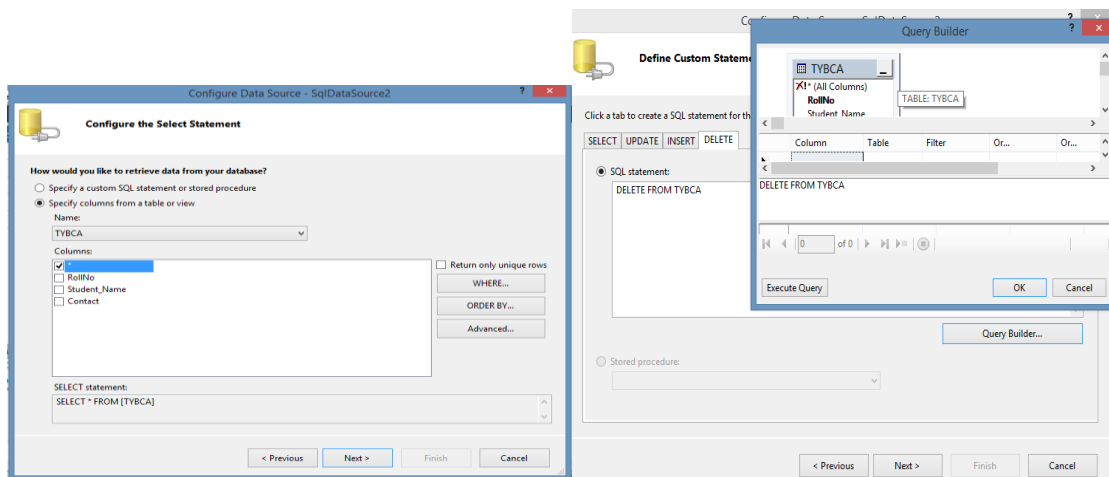
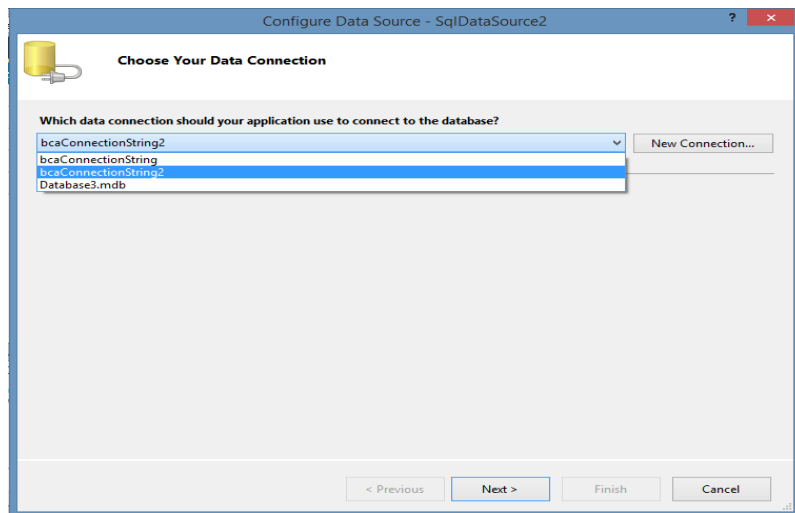


Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

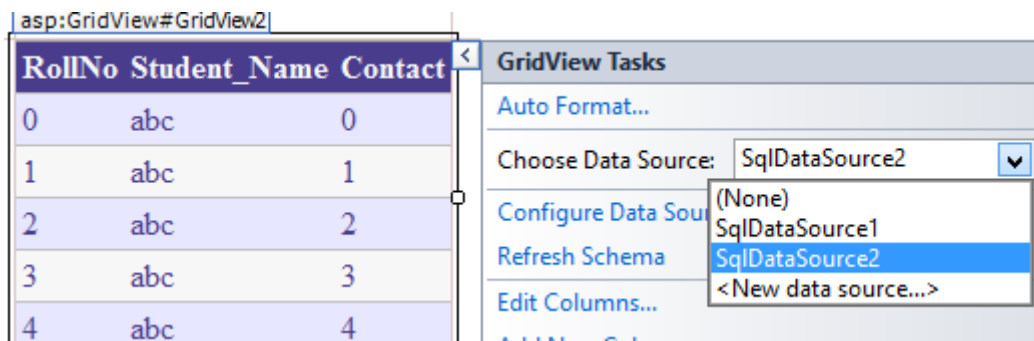
SqlDataSource - SqlDataSource2

- Now configure SQLDataSource.





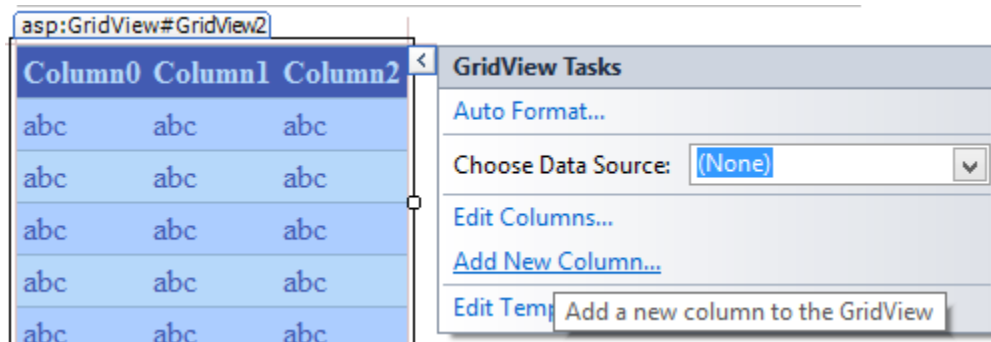
- Now set these datasource into gridview



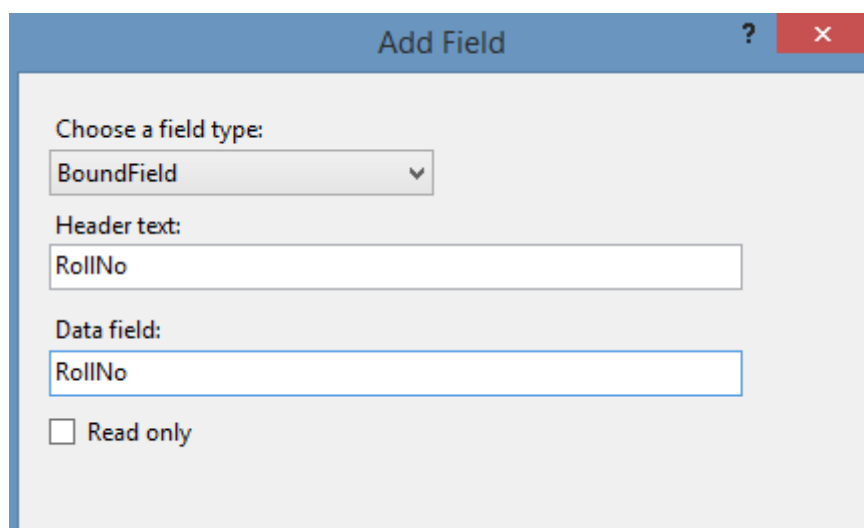
26) Create ASP.NET webpage that will display data in Gridview Control using C# code

.aspx

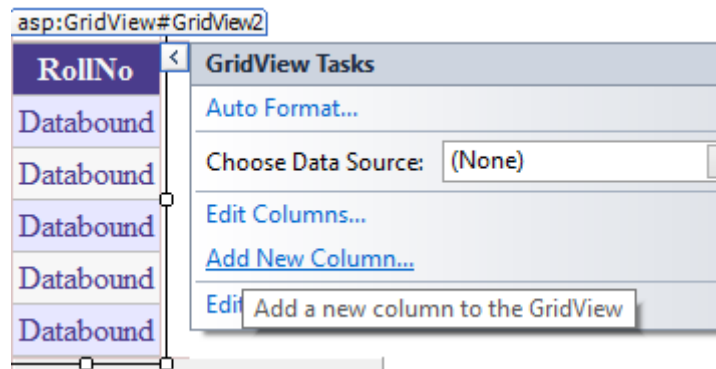
- Add Gridview control from toolbox > Data.
- Bind datafield to the gridview control.



- Add the DataField which you have passed in your SQL Statement.



You can see data is bound to gridview now repeat above steps and add the files into gridview.

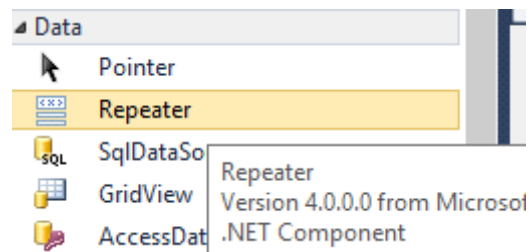


.aspx.cs

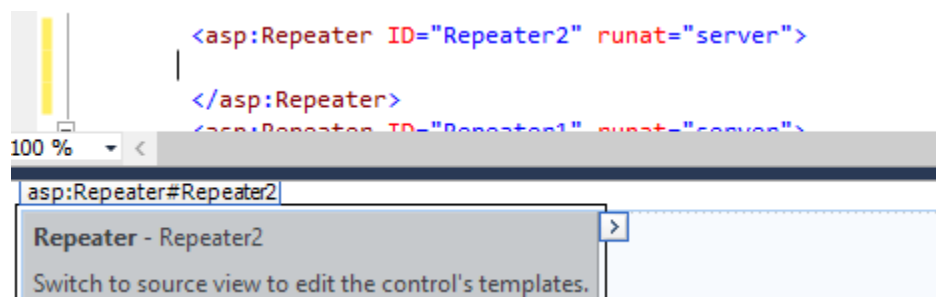
```
protected void Page_Load(object sender, EventArgs e)
{
    SqlConnection con = new
    SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString
    (Sem_5_E)"].ConnectionString);
    con.Open();
    if (con != null)
    {
        Response.Write(" conncted");
        SqlDataAdapter ol = new SqlDataAdapter("SELECT * FROM T1", con);
        DataSet ds = new DataSet();
        ol.Fill(ds);
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
    else { Response.Write("Not conncted"); }
    con.Close();
}
}
```

27) Create ASP.NET Webpage that will display data in to Repeater control.

- Add Repeater Control : Toolbox > Data > Repeater



- Now write down code in to repeater control.



.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    SqlConnection con = new
    SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString
    (Sem_5_E)"].ConnectionString);

    con.Open();

    if (con != null)
    {
        Response.Write(" conncted");

        SqlDataAdapter ol = new SqlDataAdapter("SELECT * FROM T2", con);

        DataSet ds = new DataSet();

        ol.Fill(ds);

        Repeater1.DataSource = ds;

        Repeater1.DataBind();
    }
}
```

```

    }
}
}

```

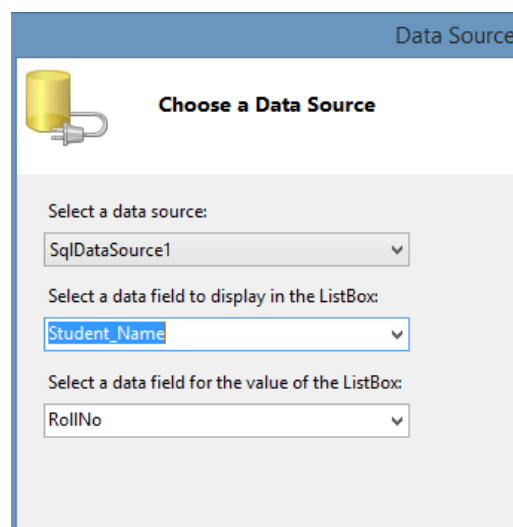
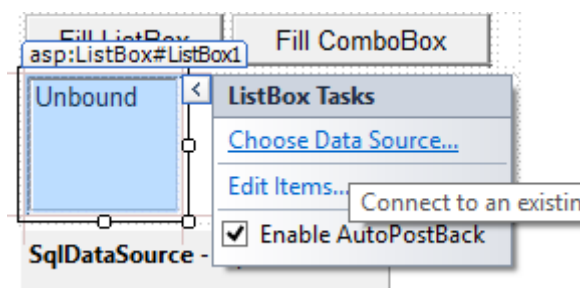
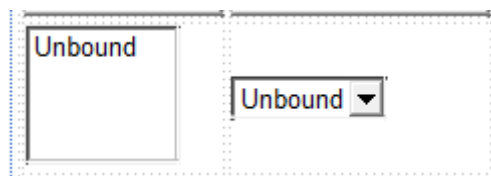
28) Create ASP.NET Webpage that will display data in to Listbox and dropdownlist control using SqlDataSource.

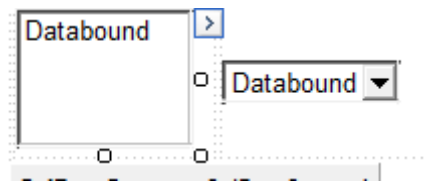
.aspx

Add SqlDatasource and configure with database.

Now see the datasource properties for listbox and dropdownlist.

Select sqldatasource and select dataTextField and Datavalue Fields for controls.

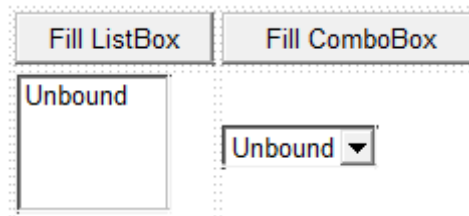




29) Create ASP.NET Webpage that will display data in to Listbox and dropdownlist control on button click event using c# cod.

.aspx

Add button controls and create button click event.



.aspx.cs

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString
(Sem_5_E)"].ConnectionString);
    con.Open();
    if (con != null)
    {
        Response.Write(" conncceted");
        SqlDataAdapter ol = new SqlDataAdapter("SELECT * FROM T2", con);
        DataSet ds = new DataSet();
        ol.Fill(ds);

        ListBox1.DataSource = ds;
        ListBox1.DataTextField = "Stu_Name";
```



```

        ListBox1.DataBind();
        con.Close();
    }
}

protected void Unnamed_SelectedIndexChanged(object sender, EventArgs e)
{
    Response.Write(DropDownList1.SelectedValue.ToString());
}

protected void Button2_Click(object sender, EventArgs e)
{
    SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString
(Sem_5_E)"].ConnectionString);
    con.Open();
    if (con != null)
    {
        Response.Write(" conncted");
        SqlDataAdapter ol = new SqlDataAdapter("SELECT * FROM T2", con);
        DataSet ds = new DataSet();
        ol.Fill(ds);

        DropDownList1.DataSource = ds;
        DropDownList1.DataTextField = "Id";
        DropDownList1.DataValueField = "Stu_Name";
        DropDownList1.DataBind();
    }
}
}

```

30) Create a ASP.NET webpage that will allow user to add product details to product master table and upload product image to server.

.aspx

Add 3 TextBoxes, 1 FileUpload and 2 buttons.

.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("Welcome !!!" + Convert.ToString(Session["user"]));

}

protected void Button1_Click(object sender, EventArgs e)
{
    if (FileUpload1.HasFile)
    {
        string filename = Path.GetFileName(FileUpload1.FileName);
        string filepath = filename;
        FileUpload1.SaveAs(Server.MapPath("~/\" + filename));

        SqlConnection con = new
        SqlConnection(ConfigurationManager.ConnectionStrings["DR"].ConnectionString);

        con.Open();

        SqlCommand cmd = new SqlCommand("Insert into Product Values('\" + TextBox1.Text + "\", '\" +
        filepath + "\", '\" + TextBox2.Text + "\", '\" + TextBox3.Text + \"')", con);

        cmd.ExecuteNonQuery();

        Response.Write("Inserted");
        TextBox1.Text = "";
        TextBox2.Text = "";
        TextBox3.Text = "";

    }

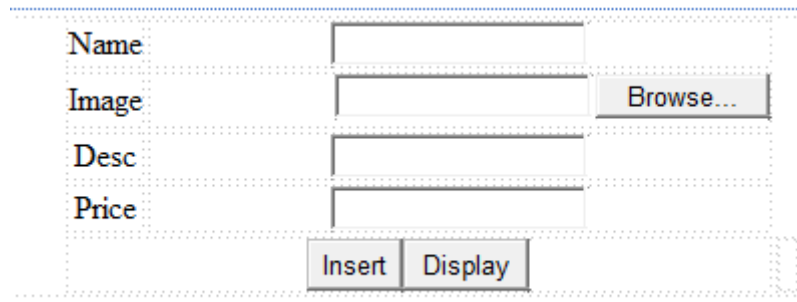
    else
```

```

    {
        Response.Write("error");
    }
}

```

OUTPUT:



The screenshot shows a web form with a dotted border. It contains four text input fields labeled 'Name', 'Image', 'Desc', and 'Price'. To the right of the 'Image' field is a 'Browse...' button. Below the input fields are two buttons labeled 'Insert' and 'Display'.

31) Create a ASP.NET web page that will display products from product_master table.

.aspx

```

<body>

    <form id="form1" runat="server">

        <div id="pr" runat="server"></div>

    </form>

</body>

```

.aspx.cs

```

protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("Welcome !!!" + Convert.ToString(Session["user"]));
    string htmlstring = "";
    int i = 0;
    SqlConnection con = new SqlConnection("Your Database Path");
    con.Open();
    SqlCommand cmd = new SqlCommand("select Name,image from Product", con);
    SqlDataReader rdr = cmd.ExecuteReader();
    while (rdr.Read())
    {
        ++i;
        if (i >= 4)
        {
            htmlstring += "<br>";
            i = 1;
        }
    }
}

```

```

    }
    htmlstring += "<a style='padding:5px;margin:10px'
href='ProductDetails.aspx?nm=" + rdr[0].ToString() + "'><img height='100px'
width='100' src='" + rdr[1].ToString() + "'></a>";
    }
    pr.InnerHtml = htmlstring;
}

```

OUTPUT:



32) Create a webpage that will show full details of the product on click .

.aspx

```

<body>
    <form id="form1" runat="server">
        <div id="pd" runat="server">

            </div>
        </form>
    </body>

```

.aspx.cs

```

protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("Welcome !!!"+Convert.ToString(Session["user"]));
    string name=Request.Params["nm"];
    string htmlstring = "";
    int i = 0;
    SqlConnection con = new SqlConnection("Your Database Path");
    con.Open();

```

```

        SqlCommand cmd = new SqlCommand("select Name,image,Desc,Price from Product
where Name='"+name+"'", con);
        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            ++i;
            if (i >= 4)
            {
                htmlstring += "<br>";
                i = 1;
            }
            htmlstring += "<center><img style='height:300px;' src='" +
rdr[1].ToString() + "'><br>" + rdr[0].ToString() + "<br>Desc
:" + rdr[2].ToString() + "<br>Price:" + rdr[3].ToString() + "</center>";
        }
        pd.InnerHtml = htmlstring;
    }
}

```

OUTPUT:



33 to 36) Create ASP.NET webpage that will display all the records into gridview control. Create Employee Table EmpID,EmpName,Bdate,Contact.

In above ASP.NET webpage add web controls that will insert data into gridview control.

In above ASP.NET webpage add edit button into gridview control that will edit/update data of gridview control.

In above ASP.NET webpage add edit button into gridview control that will delete data of gridview control.

.aspx

```

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    AutoGenerateDeleteButton="True"
    DataKeyNames="EmpID" DataSourceID="SqlDataSource1">
    <Columns>

```

```

<asp:CommandField ShowEditButton="True" />
<asp:BoundField DataField="EmpID" HeaderText="EmpID" InsertVisible="False"
    ReadOnly="True" SortExpression="EmpID" />
<asp:BoundField DataField="EmpName" HeaderText="EmpName"
    SortExpression="EmpName" />
<asp:BoundField DataField="Bdate" HeaderText="Bdate" SortExpression="Bdate" />
<asp:BoundField DataField="Contact" HeaderText="Contact"
    SortExpression="Contact" />
</Columns>
</asp:GridView>

```

.aspx.cs

```

protected void Button1_Click(object sender, EventArgs e)
{
    if (Button1.Text.Equals("New"))
    {
        Label1.Visible = true;
        TextBox1.Visible = true;
        Label2.Visible = true;
        TextBox2.Visible = true;
        Label3.Visible = true;
        TextBox3.Visible = true;

        Button1.Text = "Insert";
    }
    else if (Button1.Text.Equals("Insert"))
    {
        if(TextBox1.Text.Equals(string.Empty) ||
        TextBox1.Text.Equals(string.Empty) || TextBox3.Text.Equals(string.Empty))
        {
            Label4.Visible = true;
            Label4.Text="Pls Enter all the values";
        }
        else
        {
            SqlDataSource1.InsertCommand=string.Format("INSERT INTO [Employee]
([EmpName],[Bdate],[Contact])
Values('{0}','{1}','{2}]", TextBox1.Text, TextBox2.Text, TextBox3.Text);
            SqlDataSource1.Insert();
            GridView1.DataBind();

            TextBox3.Text = string.Empty;
            TextBox2.Text = string.Empty;
            TextBox1.Text = string.Empty;
            Label1.Visible = false;
            TextBox1.Visible = false;
            Label2.Visible = false;
            TextBox2.Visible = false;
            Label3.Visible = false;
            TextBox3.Visible = false;
            Button1.Text = "New";
        }
    }
}

```

Steps: add gridview and create dynamic fields into gridview and set enable editing & enable deleting =true then add user interface for insert data and set all controls property visible=false. Add sqldatasource ,configure it and write update command,delete command code in design code.

OUTPUT:

The screenshot shows a web form with a 'New' button and three input fields labeled 'Name', 'Bdate', and 'Contact'. Below these is a 'Label' and a GridView. The GridView has columns for 'EmpID', 'EmpName', 'Bdate', and 'Contact'. It contains 5 rows of data, each with 'Delete' and 'Edit' links. At the bottom, a 'SqlDataSource - SqlDataSource1' control is visible.

	EmpID	EmpName	Bdate	Contact
Delete Edit	0	abc	30-08-14 00:00:00	0
Delete Edit	1	abc	30-08-14 00:00:00	1
Delete Edit	2	abc	30-08-14 00:00:00	2
Delete Edit	3	abc	30-08-14 00:00:00	3
Delete Edit	4	abc	30-08-14 00:00:00	4

37) Create ASP.NET web page that will display the current date & time and create an output cache of the webpage and create a cache by param. Create ASP.NET webpage that will display the current date & time from user control and create a partial cache of the webpage.

Design code:

```
<%@ OutputCache Duration="15" VaryByParam="none" %>
```

C# code:

```
Response.Write(DateTime.Now.ToString());
```

(2) By param –

Design code

```
<%@ OutputCache Duration="15" VaryByParam="id" %>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<a href="Default.aspx?id=1">Click here for id = 1 </a>
```

```
<a href="Default.aspx?id=2">Click here for id = 2</a>
```

```
</div>
```

```
</form>
```

C# code:

```
Response.Write(DateTime.Now.ToString());
```

Create User Control : Add new item>Web User Control

Design Code of web user control:

```
<%@ OutputCache Duration="10" VaryByParam="none" %>
```

```
<asp:Label ID="Label1" runat="server" ForeColor="#CC00CC"
```

```
Text="Label"></asp:Label>
```

C# Code of web user control:

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

```
{
```

```
Label1.Text = DateTime.Now.ToString();
```

```
}
```

Design code for webpage :

```
<%@ Register Src="~/WebUserControl.ascx" TagPrefix="uc"
TagName="timecontrol" %>
<uc:timecontrol runat="server"></uc:timecontrol>
```

OUTPUT:

03-09-2025 15:01:04

38) Create ASP.NET web pages that will display the current date & time using absolute expiration cache. 37. Create ASP.NET a web page that will display data into grid view control and create data caching.

```
using System.Web.Caching;

protected void Page_Load(object sender, EventArgs e)
{
    if (Cache["AbsoluteCacheKey"] == null)
    {
        string cacheData = DateTime.Now.ToString();
        Cache.Insert("AbsoluteCacheKey", cacheData, null,
DateTime.Now.AddSeconds(10), System.Web.Caching.Cache.NoSlidingExpiration);
        Response.Write("Cache created");
    }
    else
    {
        string value = Cache["AbsoluteCacheKey"].ToString();
        Response.Write(value);
    }
}
}
```

OUTPUT:

03-09-2025 15:01:04

39) Create an XML file that will store student information. Create ASP.NET Web page that will read XML file and display XML data into GridView control.

.aspx

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

.aspx.cs

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

```
// Create DataSet and Read XML
```

```
DataSet ds = new DataSet();
```

```
ds.ReadXml(Server.MapPath("student.xml")); // student.xml should be inside project folder
```

```
// Bind to GridView
```

```
GridView1.DataSource = ds;
```

```
GridView1.DataBind();
```

```
}
```

```
}
```

OUTPUT:

Button		
ID	Name	Course
1	Rahul Sharma	B.Tech
2	Priya Verma	MBA
3	Amit Kumar	B.Sc

40) Create ASP.NET Webpage that will read data from database and write on XML on button click.

.aspx

```
<form id="form1" runat="server">
<div>
  <asp:Button ID="Button1" runat="server" Text="WriteXML"
    onclick="Button1_Click" />
  <br />
  <br />
</div>
</form>
```

.aspx.cs

```
using System.Data.Sql;
    using System.Data;

    SqlConnection con = new SqlConnection("Your Database Path");
    con.Open();
    SqlDataAdapter adp = new SqlDataAdapter("Select * from TYBCA", con);
    DataSet ds = new DataSet();
    adp.Fill(ds);

    ds.WriteXml(MapPath("xyz.xml"));
    con.Close();
    Response.Write("XML is created");
```

OUTPUT

