**Connection string:**

Making connection to database in C#:

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Irfan Ullah\Documents\Visual Studio 2012\WebSites\MyBlog\App\_Data\blog\_db.mdf;Integrated Security=True");

**Inserting record into database table:**

if (con.State == ConnectionState.Closed)

{

con.Open();

String query = "INSERT INTO posts(post\_title,post\_description) VALUES('" + title + "','" + Description + "')";

SqlCommand com = new SqlCommand(query, con);

com.ExecuteNonQuery();

Response.Write("<script>alert('Post Published');</script>");

Response.Redirect("Default.aspx");

}

else

{

Response.Write("<script>alert('Error not close');</script>");

}

**Redirecting to another page:**

Response.Redirect("../AddPost.aspx");

**GridView:**

<asp:GridView CssClass="table table-hover table-condensed table-bordered table-responsive" ID="GridView1" runat="server" AllowPaging="True" AllowSorting="True" AutoGenerateColumns="False" DataSourceID="SqlDataSource1" OnRowUpdated="updateRowRecord" DataKeyNames="Id">

<Columns>

<asp:BoundField DataField="Id" HeaderText="Id" SortExpression="Id" InsertVisible="False" ReadOnly="True" Visible="false" />

<asp:BoundField DataField="post\_title" HeaderText="POST TITLE" SortExpression="post\_title" />

<asp:BoundField DataField="post\_description" HeaderText="DESCRIPTION" SortExpression="post\_description" />

<asp:BoundField DataField="post\_date" HeaderText="DATE" SortExpression="post\_date" />

<asp:CommandField ShowEditButton="true" />

<asp:CommandField ShowDeleteButton="true" />

</Columns>

</asp:GridView>

<asp:Button CssClass="btn btn-success" ID="btnAddRow" runat="server" OnClick="btnAddRow\_Click" Text="Add Row" />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:ConnectionString %>"

SelectCommand="SELECT [Id], [post\_title], [post\_description], [post\_date] FROM [posts]"

UpdateCommand="UPDATE posts SET post\_title = @post\_title, post\_description = @post\_description WHERE Id = @Id"

DeleteCommand="DELETE FROM posts WHERE Id = @Id"

OnUpdating="SqlDataSource1\_Updating"

></asp:SqlDataSource>

**When OnRowUpdated event is occurred. The following method will be called.**

public void updateRowRecord(object sender, GridViewUpdatedEventArgs e)

{

Response.Write("<script>alert('Record Updated Successfully.');</script>");

}

The following are the different events that can be added to gridView

OnRowCancelEditing

OnRowCommand

OnRowCreate

OnRowDeleted

OnRowDataBound

OnRowDeleting

OnRowEditing

OnRowUpdating

OnSorting

OnSorted

OnLoad

**Fetching Records and saving to dataset and setting that dataset to GridView, ListView or Datalist.**

con.Open();

string pid = Request.QueryString["id"];

String query = "SELECT \* FROM posts WHERE Id = '"+pid+"'";

SqlDataAdapter da = new SqlDataAdapter(query, con);

DataSet ds = new DataSet();

da.Fill(ds);

DataList1.DataSource = ds;

DataList1.DataBind();

con.Close();

<asp:DataList ID="DataList1" runat="server">

<ItemTemplate>

<asp:Label ID="Label1" runat="server" Text='<%# Eval("Id") %>'></asp:Label>

<asp:Label ID="Label2" runat="server" Text='<%# Eval("post\_title") %>'></asp:Label>

<asp:Label ID="pd" runat="server" Text='<%# Eval("post\_description") %>'></asp:Label>

</ItemTemplate>

</asp:DataList>

**Sending and Getting the value of parameters passed from one page to another in link.**

NavigateUrl='<%# string.Format("ViewPost.aspx?id={0}", Eval("Id")) %>'

//the above tag should be included in ASP.NET HyperLink

<asp:HyperLink runat="server" NavigateUrl='<%# string.Format("ViewPost.aspx?id={0}", Eval("Id")) %>' ID="post\_title" ><%# Eval("post\_title") %></asp:HyperLink>

string pid = Request.QueryString["id"];

**Echoing/printing value in tags:**

<%# Eval("Id") %>

**SETTING SESSIONS:**

Session.Add("key", "this is session value");

**GETTING SESSION VALUE:**

string v = Session["key"].ToString();

**SQL DATA SOURCE COMMANDS**

SelectCommand="SELECT [Id], [post\_title], [post\_description], [post\_date] FROM [posts]"

UpdateCommand="UPDATE posts SET post\_title = @post\_title, post\_description = @post\_description WHERE Id = @Id"

DeleteCommand="DELETE FROM posts WHERE Id = @Id"

//post\_title, post\_description and post\_date are the database fields

In the update command @post\_title, @post\_description are pointing to TEXT FIELDS. When edit is clicked. So it will take values from the fields and the update command will be executed.

**/////////////// database joins – has good use in asp.net and C#**

For example, we have two tables 1) country 2) provinces.

The country table has a child table provinces.

The provinces table has the foreign key of the country table.

Now we will use inner join sql statement to fetch countries having provinces in the provinces table.

countries

|  |  |
| --- | --- |
| Country\_id (pk) | Country\_name |
| 1 | Pakistan |
| 2 | India |
| 3 | Afghanistan |

**provinces**

|  |  |  |
| --- | --- | --- |
| Province\_id | **Province\_name** | **Country\_id (FK)** |
| **1** | **KP** | **1** |
| **2** | **Punjab** | **2** |
| **3** | **kundoz** | **3** |
| **4** | **Katmandu** |  |

**INNER JOIN:**

**SQL = “SELECT \* FROM countries INNER JOIN provinces ON countries.Country\_id = provinces.Country\_id”**

**But it has a disadvantage because the countries having no provinces will not be displayed.**

**LEFT JOIN:**

SQL = “SELECT \* FROM countries LEFT JOIN provinces ON countries.Country\_id = provinces.Country\_id”

It is best choice as compared to INNER JOIN because in LEFT JOIN, the names of the countries will be displayed irrespective of having provinces or not.

**RIGHT JOIN:**

SQL = “SELECT \* FROM countries RIGHT JOIN provinces ON countries.Country\_id = provinces.Country\_id”

Will display all the provinces and countries, but won’t display a province having no record.

**FULL OUTER JOIN:**

SQL = “SELECT \* FROM countries FULL OUTER JOIN provinces ON countries.Country\_id = provinces.Country\_id”

Will display every thing regardless of relationship. It is the combination of INNER, LEFT and RIGHT JOINS.