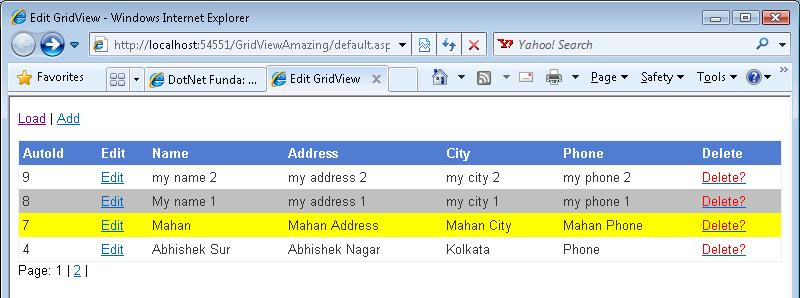
1. Default.aspx - to display the recods and fire necessary jQuery command for CRUD perration.
2. GridViewData.aspx - to bind the GridView according to the command received using jQuery.

Our final result will look something similar to shown below. Where Yellow row is the highlighted row on mouse over, gray row is the selected row by clicking it.



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

<div id="divGridView">

<asp:gridview id="GridView1"

runat="server"

enableviewstate="False"

width="100%"

BorderWidth="1"

cellpadding="4"

forecolor="#333333"

gridlines="None"

autogeneratecolumns="false"

datakeynames="AutoId"

autogeneratedeletebutton="false"

EmptyDataText="No records found" >

<HeaderStyle BackColor="#507CD1" HorizontalAlign="Left" Font-Bold="True" ForeColor="White" />

<Columns>

<asp:BoundField DataField="AutoId" HeaderText="AutoId" />

<asp:TemplateField HeaderText="Edit">

<ItemTemplate>

<a href="javascript:void(0)"

onclick="ShowEditBox(<%# Eval("AutoId") %>)"

title="Edit">Edit</a>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Name">

<ItemTemplate>

<span id="PName<%# Eval("AutoId") %>"><%# Eval("Name") %></span>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField HeaderText="Address" DataField="Address" />

<asp:BoundField HeaderText="City" DataField="City" />

<asp:BoundField HeaderText="Phone" DataField="Phone" />

<asp:TemplateField HeaderText="Delete">

<ItemTemplate>

<span onclick="return confirm('Are you sure?')">

<a href="javascript:DeleteRecord(<%# Eval("AutoId") %>)"

title="Delete"><font

color="red">Delete?</font></a>

</span>

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:gridview>

<asp:Literal runat="server" ID="litPaging" EnableViewState="False" />

<p>

<asp:label id="lblMessage" runat="server" enableviewstate="false" />

</p>

</div>

</form>

<script language="javascript" type="text/javascript">

// highlight the row when clicked

$(document).ready(function () {

$("#divGridView table tbody **tr**").**mouseover**(function () {

$(this).addClass("highlightRow");

}).mouseout(function () { $(this).removeClass('highlightRow'); })

});

// highlight row by clicking it

$(document).ready(function () {

$("#divGridView table tbody **tr**").**click**(function () {

$(this).addClass("select");

})

});

// double click delete rows

$(document).ready(function () {

$("#divGridView table tbody **tr**").**dblclick**(function () {

// find the id first

var v = confirm(**'Are you sure to delete?**');

if (v) {

var autoId = $(this).find("td:first").text();

// remove the row from server side (the database)

**DeleteRecord(autoId);**

// remove from the clien side

$(this).remove();

}

})

});

</script>

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class GridViewData : System.Web.UI.Page

{

string \_connStr = ConfigurationManager.ConnectionStrings["ConnStr"].ConnectionString;

int \_startRowIndex = 0;

int \_pageSize = 4;

int \_thisPage = 1;

protected void Page\_Load(object sender, EventArgs e)

{

HandleRequestObjects();

try

{

if (!IsPostBack) BindMyGrid();

}

catch (Exception ee)

{

throw ee;

}

}

/// <summary>

/// Handles the request objects.

private void HandleRequestObjects()

{

try

{

// check for paging

if (Request.Form["startRowIndex"] != null && Request.Form["thisPage"] != null)

{

\_startRowIndex = int.Parse(Request.Form["startRowIndex"].ToString());

\_thisPage = int.Parse(Request.Form["thisPage"].ToString());

}

// check for edit

if (Request.Form["editId"] != null)

{

UpdateInsertData(Request.Form["editId"]);

}

// check for deletion

if (Request.Form["deleteId"] != null)

{

DeleteRecord(Request.Form["deleteId"]);

}

}

catch (Exception ee)

{

throw ee;

}

}

/// Updates the data.

private void UpdateInsertData(string editId)

{

string sql = string.Empty;

string message = "Added";

if (editId.EndsWith("0"))

{

sql = "insert into SampleForTutorials (Name, Address, Phone, City) values " +

" (@Name, @Address, @Phone, @City)";

}

else

{

message = "Update";

sql = "Update SampleForTutorials set Name = @Name, Address = @Address, " +

" Phone = @Phone, City = @City WHERE AutoId = @AutoId";

}

// get the data now

using (SqlConnection conn = new SqlConnection(\_connStr))

{

using (SqlCommand cmd = new SqlCommand(sql, conn))

{

cmd.CommandType = CommandType.Text;

SqlParameter p = new SqlParameter("@Name", SqlDbType.VarChar, 50);

p.Value = Request.Form["pName"];

cmd.Parameters.Add(p);

p = new SqlParameter("@Address", SqlDbType.VarChar, 150);

p.Value = Request.Form["pAddress"];

cmd.Parameters.Add(p);

p = new SqlParameter("@Phone", SqlDbType.VarChar, 50);

p.Value = Request.Form["pPhone"];

cmd.Parameters.Add(p);

p = new SqlParameter("@City", SqlDbType.VarChar, 50);

p.Value = Request.Form["pCity"];

cmd.Parameters.Add(p);

p = new SqlParameter("@AutoId", SqlDbType.Int);

p.Value = int.Parse(editId);

cmd.Parameters.Add(p);

conn.Open();

cmd.ExecuteNonQuery();

conn.Close();

}

}

lblMessage.Text = "Selected record " + message + " successfully !";

// rebind the data again

BindMyGrid();

}

/// Deletes the record.

private void DeleteRecord(string id)

{

int productId = int.Parse(id);

string sql = "delete SampleForTutorials where AutoId = @AutoId";

using (SqlConnection conn = new SqlConnection(\_connStr))

{

using (SqlCommand cmd = new SqlCommand(sql, conn))

{

cmd.Parameters.AddWithValue("@AutoId", productId);

conn.Open();

cmd.ExecuteNonQuery();

conn.Close();

}

}

lblMessage.Text = "Selected record deleted successfully !";

// rebind the data again

BindMyGrid();

}

/// Binds my grid.

private void BindMyGrid()

{

// sql for paging. In production write this in the Stored Procedure

string sql = "SELECT \* FROM ( " +

" Select SampleForTutorials.\*, ROW\_NUMBER() OVER (ORDER BY AutoId DESC) as RowNum " +

" FROM SampleForTutorials) as AddressList " +

" WHERE RowNum BETWEEN @startRowIndex AND (@startRowIndex + @pageSize) - 1 " +

"ORDER BY AutoId DESC";

DataTable table = new DataTable();

int totalCount = 0;

// get the data now

using (SqlConnection conn = new SqlConnection(\_connStr))

{

using (SqlCommand cmd = new SqlCommand(sql, conn))

{

cmd.CommandType = CommandType.Text;

SqlParameter p = new SqlParameter("@startRowIndex", SqlDbType.Int);

p.Value = \_startRowIndex + 1;

cmd.Parameters.Add(p);

p = new SqlParameter("@pageSize", SqlDbType.Int);

p.Value = \_pageSize;

cmd.Parameters.Add(p);

conn.Open();

// get the data first

using (SqlDataAdapter ad = new SqlDataAdapter(cmd))

{

ad.Fill(table);

}

// get the total count of the records now

sql = "select count(AutoId) from SampleForTutorials";

cmd.Parameters.Clear();

cmd.CommandText = sql;

object obj = cmd.ExecuteScalar();

totalCount = Convert.ToInt32(obj);

conn.Close();

}

}

// do the paging now

litPaging.Text = DoPaging(\_thisPage, totalCount, \_pageSize);

// bind the data to the grid

GridView1.DataSource = table;

GridView1.DataBind();

}

/// Do the paging now

/// <param name="thisPageNo"></param>

/// <param name="totalCount"></param>

/// <param name="pageSize"></param>

/// <returns></returns>

private string DoPaging(int thisPageNo, int totalCount, int pageSize)

{

if (totalCount.Equals(0)) return "";

int pageno = 0;

int start = 0;

int loop = totalCount / pageSize;

int remainder = totalCount % pageSize;

int startPageNoFrom = thisPageNo - 6;

int endPageNoTo = thisPageNo + 6;

int lastRenderedPageNo = 0;

StringBuilder strB = new StringBuilder("<div>Page: ", 500);

// write 1st if required

if (startPageNoFrom >= 1)

{

strB.Append("<a href=\"javascript:LoadGridViewData(0, 1)\" title=\"Page 1\">1</a> | ");

if (!startPageNoFrom.Equals(1))

{

strB.Append(" ... | ");

}

}

for (int i = 0; i < loop; i++)

{

pageno = i + 1;

if (pageno > startPageNoFrom && pageno < endPageNoTo)

{

if (pageno.Equals(thisPageNo))

strB.Append("<span>" + pageno + "</span>&nbsp;| ");

else

strB.Append("<a href=\"javascript:LoadGridViewData(" + start + ", " + pageno + ")\" title=\"Page " + pageno + "\">" + pageno + "</a> | ");

lastRenderedPageNo = pageno;

}

start += pageSize;

}

// write ... if required just before end

if (!pageno.Equals(lastRenderedPageNo))

{

strB.Append(" ... | ");

}

if (remainder > 0)

{

pageno++;

if (pageno.Equals(thisPageNo))

strB.Append("<span>" + pageno + "</span>&nbsp;| ");

else

strB.Append("<a href=\"javascript:LoadGridViewData(" + start + ", " + pageno + ")\" title=\"Page " + pageno + "\">" + pageno + "</a> | ");

}

else // write last page number

{

if (loop >= endPageNoTo)

{

if (loop.Equals(thisPageNo))

strB.Append("<span>" + loop + "</span>&nbsp;| ");

else

strB.Append("<a href=\"javascript:LoadGridViewData(" + start + ", " + pageno + ")\" title=\"Page " + loop + "\">" + loop + "</a> | ");

}

}

return strB.ToString() + "</div>";

}

}