**Dal/sqlexecutelayer.cs**

public class SqlExecuteLayer

{

#region connectionstring

private string \_sqlconn;

public string sqlconn

{

get { return \_sqlconn = Convert.ToString(ConfigurationManager.AppSettings["conn"]); }

set { \_sqlconn = value; }

}

#endregion

#region param to SQL command

public void AddParamToSQLCmd(SqlCommand cmd, string param, SqlDbType sqlDbType, int size, ParameterDirection parameterDirection, object paramvalue)

{

try

{

SqlParameter p = new SqlParameter();

p.ParameterName = param;

p.SqlDbType = sqlDbType;

if (size > 0)

{

p.Size = size;

}

p.Direction = parameterDirection;

if (paramvalue != null)

{

p.Value = paramvalue;

}

cmd.Parameters.Add(p);

}

catch(SqlException ex)

{

throw ex;

}

}

#endregion

#region DML operations

public int ExecuteBatchIUD(SqlCommand cmd)

{

int i = 0;

using (SqlConnection conn = new SqlConnection(sqlconn))

{

SqlTransaction t = null;

cmd.Connection = conn;

try

{

conn.Open();

t = conn.BeginTransaction();

cmd.Transaction = t;

i = cmd.ExecuteNonQuery();

t.Commit();

}

catch(SqlException ex)

{

t.Rollback();

throw ex;

}

finally

{

conn.Close();

conn.Dispose();

}

return i;

}

}

#endregion

}

**Dal/sqldataaccesslayer.cs**

public class SqlDataAccessLayerlass1:SqlExecuteLayer

{

#region IUD

public int studIUD(clsstudreg obj,string action)

{

int i = 0;

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "studIUD";

AddParamToSQLCmd(cmd, "@studid", SqlDbType.Int, 50, ParameterDirection.Input, obj.studid);

AddParamToSQLCmd(cmd, "@studnm", SqlDbType.NVarChar, 50, ParameterDirection.Input, obj.studnm);

AddParamToSQLCmd(cmd, "@studgrade", SqlDbType.Int, 50, ParameterDirection.Input, obj.studgrade);

AddParamToSQLCmd(cmd, "@studclass", SqlDbType.NVarChar, 50, ParameterDirection.Input, obj.studclass);

AddParamToSQLCmd(cmd, "@action", SqlDbType.NVarChar, 20, ParameterDirection.Input, action);

return i=ExecuteBatchIUD(cmd);

}

#endregion

}

**Bal/datamanagerstud.cs**

public class DataManagerStud

{

SqlDataAccessLayerlass1 dal = new SqlDataAccessLayerlass1();

public int studdel(clsstudreg eobj, string action)

{

int i = 0;

return i = dal.studIUD(eobj, action);

}

}

**Global.asax**

public class Global : System.Web.HttpApplication

{

private static string m\_Errorpagereturn;

private static string m\_errorinfo;

public static string errorpagereturn

{

get { return m\_Errorpagereturn; }

set { m\_Errorpagereturn = value; }

}

public static string errorinfo

{

get { return m\_errorinfo; }

set { m\_errorinfo = value; }

}

protected void Application\_Start(object sender, EventArgs e)

{

}

protected void Session\_Start(object sender, EventArgs e)

{

}

protected void Application\_BeginRequest(object sender, EventArgs e)

{

}

protected void Application\_AuthenticateRequest(object sender, EventArgs e)

{

}

protected void Application\_Error(object sender, EventArgs e)

{

Exception ex = Server.GetLastError().GetBaseException();

string err = "<br/><b>error page source:</b>" + Request.Url.ToString() +

"<br/><b>error message:</b>" + ex.Message.ToString() +

"<br/><b>stack track:"+ex.StackTrace.ToString();

errorpagereturn = "studreg1.aspx";

errorinfo = err;

Server.ClearError();

Response.Redirect("globalerror.aspx");

}

protected void Session\_End(object sender, EventArgs e)

{

}

protected void Application\_End(object sender, EventArgs e)

{

}

}

**Before insert pro.**

protected void grd\_RowCommand(object sender, GridViewCommandEventArgs e)

{

if (e.CommandName == "edit")

{

int sid = Convert.ToInt32(e.CommandArgument.ToString());

ViewState["sid"] = sid;

DataTable d = (DataTable)Session["temp"];

DataRow[] r = d.Select("studid=" + sid);

foreach (DataRow dr in r)

{

txtnm.Text = dr[1].ToString();

txtgrade.Text = dr[2].ToString();

txtclass.Text = dr[3].ToString();

}

//using (datacontextDataContext dc = new datacontextDataContext())

//{

// var m = (from tbl in dc.tblstuds where tbl.studid == sid select new { tbl.studnm, tbl.studgrade, tbl.studclass }).FirstOrDefault();

// if (m != null)

// {

// txtclass.Text = m.studclass;

// txtgrade.Text = Convert.ToString(m.studgrade);

// txtnm.Text = m.studnm;

// }

//}

}

else if (e.CommandName == "delete")

{

int sid = Convert.ToInt32(e.CommandArgument.ToString());

eobj.studid = sid; eobj.studclass = ""; eobj.studgrade = 0; eobj.studnm = "";

int i = dms.studdel(eobj, "del");

if (i > 0)

{

lblerr.Text = "Record deleted successfully!";

}

fillgrid();

}

}

protected void grd\_RowEditing(object sender, GridViewEditEventArgs e)

{

}

protected void grd\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

}

protected void btnmanual\_Click(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt.Columns.Add("studid", typeof(int));

dt.Columns.Add("studnm", typeof(string));

dt.Columns.Add("studgrade", typeof(int));

dt.Columns.Add("studclass", typeof(string));

int sid;

if (Session["temp"] != null)

{

dt = (DataTable)Session["temp"];

sid = Convert.ToInt32(Session["sid"].ToString()) + 1;

}

else

{

sid = 1;

}

DataRow dr = dt.NewRow();

dr["studid"] = sid; Session["sid"] = sid;

dr["studnm"] = txtnm.Text;

dr["studgrade"] = txtgrade.Text;

dr["studclass"] = txtclass.Text;

dt.Rows.Add(dr);

Session["temp"] = dt;

fillgrid();

clearAll();

}