Practical No 3

**RollNo:46 Name:Rode Anil Nanaso**

# Design an ASP.NET web Applications using Connected Database architecture to add, update, delete and search customer information from database table. Consider following schema for customer\_info table customer\_info(cid primary key, cname, cadd)

**Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="connectedDb.aspx.cs" Inherits="dbDemo1.WebForm1" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="Label1" runat="server" Text="Enter Roll No. :"></asp:Label>

<asp:TextBox ID="txtRollNo" runat="server"></asp:TextBox>

<br />

<asp:Label ID="Label2" runat="server" Text="Enter Name :"></asp:Label>

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<br />

<asp:Label ID="Label3" runat="server" Text="Enter Address :"></asp:Label>

<asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>

<br />

<br />

<asp:GridView ID="gvStudDetails" runat="server">

</asp:GridView>

<br />

<asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnDelete" runat="server" Text="Delete" OnClick="btnDelete\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnUpdate" runat="server" Text="Update" OnClick="btnUpdate\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSearch" runat="server" Text="Search" OnClick="btnSearch\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSort" runat="server" Text="Sort" OnClick="btnSort\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtSort" runat="server"></asp:TextBox>

<br />

<asp:Button ID="btnAddSP" runat="server" Text="Add using Stored Proc" OnClick="btnAddSP\_Click" />&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSearchSP" runat="server" Text="Search using Stored Proc" OnClick="btnSearchSP\_Click" />

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

using System.Collections.Generic; using System.Configuration; using System.Data;

using System.Data.SqlClient using System.Linq;

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

public partial class WebForm1 : System.Web.UI.Page

{

static string conStr = ConfigurationManager.ConnectionStrings["studConnString"].ToString(); SqlConnection conn = null;

SqlCommand cmd = null;

SqlDataReader dr = null;

DataTable dt = null;

public void clearTextBox()

{

txtRollNo.Text = ""; txtName.Text = ""; txtAddress.Text = "";

}

public void showData()

{

try

{

cmd = new SqlCommand("SELECT \* FROM student", conn);

//cmd.Connection = conn;

//cmd.CommandText = "SELECT \* FROM stud57";

//cmd.CommandType = CommandType.Text; if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

dt = new DataTable();

dr = cmd.ExecuteReader(); dt.Load(dr); gvStudDetails.DataSource = dt; gvStudDetails.DataBind();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

conn = new SqlConnection(conStr); showData();

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand("INSERT INTO student(rollno,name,address)VALUES(@rn, @nm, @add)",

conn);

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text); cmd.Parameters.AddWithValue("@nm", txtName.Text); cmd.Parameters.AddWithValue("@add", txtAddress.Text); int r = cmd.ExecuteNonQuery();

if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record inserted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not inserted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter all details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

Response.Write("<script type=\"text/javascript\">alert('btn clicked');</script>"); try

{

if (txtRollNo.Text != "" && Convert.ToUInt16(txtRollNo.Text) > 0)

{

cmd = new SqlCommand("DELETE student WHERE RollNo=@rn", conn); if (conn.State == ConnectionState.Closed)

{ conn.Open(); } cmd.Parameters.AddWithValue("@rn", txtRollNo.Text); int r = cmd.ExecuteNonQuery();

if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record deleted Successfully!');</script>"); } else

{

Response.Write("<script type=\"text/javascript\">alert('Record not deleted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter correct RollNo!');</script>"); }

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand("UPDATE student SET name=@nm,address=@add WHERE rollno = @rn",

conn);

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text); cmd.Parameters.AddWithValue("@nm", txtName.Text);

cmd.Parameters.AddWithValue("@add", txtAddress.Text);

int r = cmd.ExecuteNonQuery(); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record updated successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not updated!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

try

{

cmd = new SqlCommand("SELECT \* FROM student WHERE RollNo=@rn", conn);

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text);

dr = cmd.ExecuteReader(); if (dr.Read())

{

Response.Write("<script type=\"text/javascript\">alert('Record found Successfully!');</script>"); clearTextBox();

txtRollNo.Text = dr["rollno"].ToString(); txtName.Text = dr["name"].ToString(); txtAddress.Text = dr["address"].ToString();

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not found!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

try

{

DataView dv = new DataView(dt);

//dv.Sort = txtSort.Text; // "name ASC"; dv.RowFilter = txtSort.Text; //"address = 'Ratnagiri'"; gvStudDetails.DataSource = dv; gvStudDetails.DataBind();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

}

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

{

try

{

if (txtRollNo.Text != "" && Convert.ToInt32(txtRollNo.Text) > 0)

{

cmd = new SqlCommand();

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Connection = conn;

//set command type as stored procedure cmd.CommandType = CommandType.StoredProcedure;

//pass the stored procedure name cmd.CommandText = "searchData";

cmd.Parameters.Add(new SqlParameter("@rollno", SqlDbType.SmallInt)).Value = Convert.ToInt16(txtRollNo.Text);

dr = cmd.ExecuteReader(); if (dr.HasRows)

{

Response.Write("<script type=\"text/javascript\">alert('Record found Successfully!');</script>");

while (dr.Read())

{

clearTextBox();

txtRollNo.Text = dr["rollno"].ToString(); txtName.Text = dr["name"].ToString();

txtAddress.Text = dr["address"].ToString();

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not found!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter roll number!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand();

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Connection = conn;

//set command type as stored procedure cmd.CommandType = CommandType.StoredProcedure;

//pass the stored procedure name cmd.CommandText = "insertData";

cmd.Parameters.Add(new SqlParameter("@rollno", SqlDbType.SmallInt)).Value = Convert.ToInt16(txtRollNo.Text);

cmd.Parameters.Add(new SqlParameter("@Name", SqlDbType.VarChar)).Value = txtName.Text;

cmd.Parameters.Add(new SqlParameter("@add", SqlDbType.VarChar)).Value = txtAddress.Text;

int r = cmd.ExecuteNonQuery(); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record inserted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not inserted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter all details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

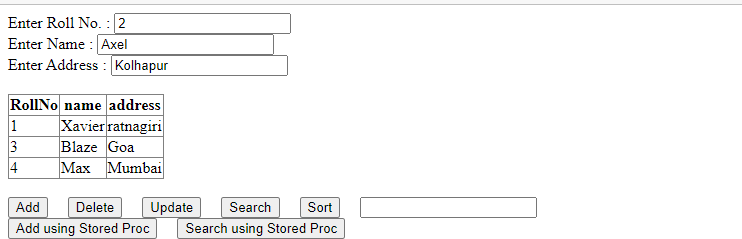
}

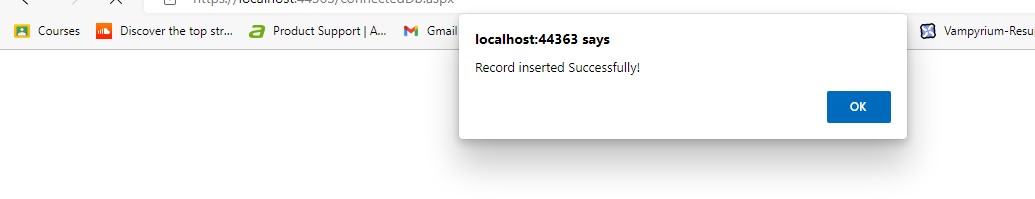
}

}

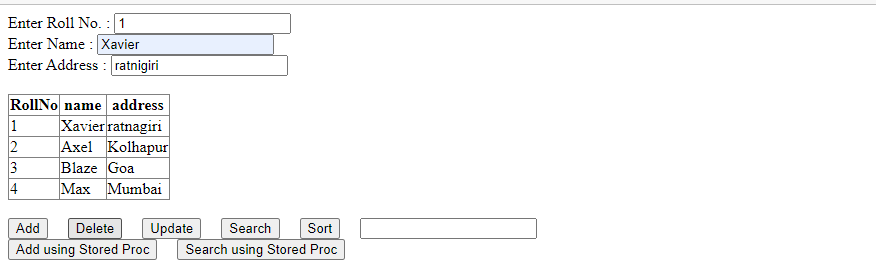
}

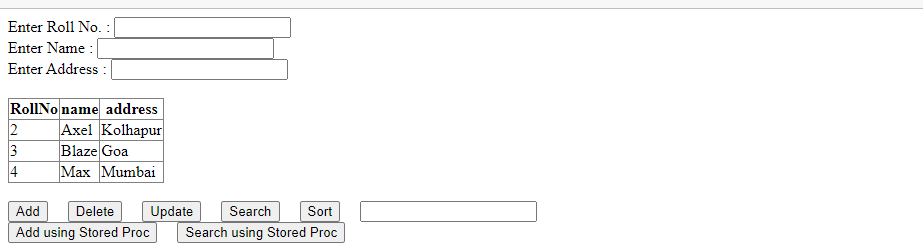
# Output: Insert:





**Delete:**

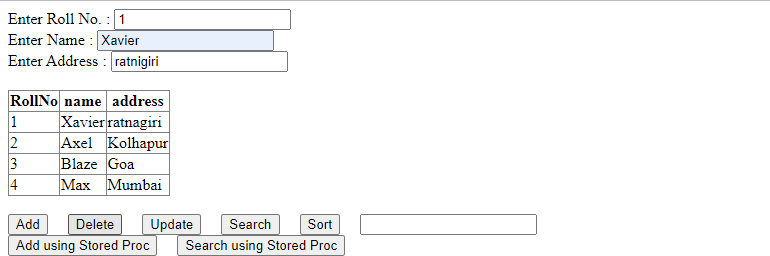


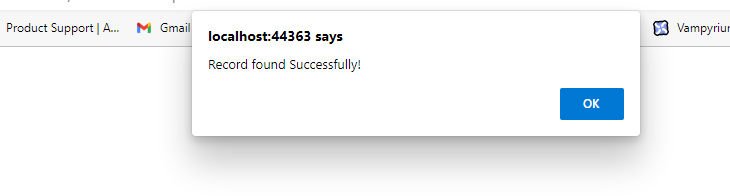


# Update:



**Search:**





# Design an ASP.NET web Applications using Disconnected Database architecture to to add, update, delete and search employee information from database table. Consider following schema for emp\_info table emp\_info(eid primary key, ename, designation, salary)

**Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="disconnectedDb.aspx.cs" Inherits="dbDemo1.WebForm2" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="LblEid" runat="server" Text="employee ID:"></asp:Label>

<asp:TextBox ID="TbEid" runat="server"></asp:TextBox>

<br />

<asp:Label ID="LblName" runat="server" Text="Employee Name:"></asp:Label>

<asp:TextBox ID="TbEname" runat="server"></asp:TextBox>

<br />

<asp:Label ID="LblDesignation" runat="server" Text="Designation/Post:"></asp:Label>

<asp:TextBox ID="TbDesignation" runat="server"></asp:TextBox>

<br />

<asp:Label ID="LblSalary" runat="server" Text="Salary:"></asp:Label>

<asp:TextBox ID="TbSalary" runat="server"></asp:TextBox>

<br />

<br />

<asp:Button ID="BtnAddInfo" runat="server" Text="Add Info" OnClick="Add\_Info"

/>&nbsp;<asp:Button ID="BtnDeleteInfo" runat="server" Text="Delete Info" OnClick="delete\_info" />&nbsp;<asp:Button ID="BtnUpdateInfo" runat="server" Text="Update Info" OnClick="update\_info" />&nbsp;<asp:Button ID="BtnSearchInfo" runat="server" Text="Search Info" OnClick="search\_info" />

<br />

<br />

<br />

<asp:GridView ID="GvEmployee" runat="server" AutoGenerateColumns="False" DataKeyNames="eid">

<Columns>

<asp:BoundField DataField="eid" HeaderText="eid" ReadOnly="True" SortExpression="eid" />

<asp:BoundField DataField="ename" HeaderText="ename" SortExpression="ename" />

<asp:BoundField DataField="designation" HeaderText="designation" SortExpression="designation" />

<asp:BoundField DataField="salary" HeaderText="salary" SortExpression="salary" />

</Columns>

</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:empConnString %>" SelectCommand="SELECT \* FROM [emp\_info]"></asp:SqlDataSource>

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

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

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

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

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

namespace dbDemo1

{

public partial class WebForm2 : System.Web.UI.Page

{

static string connString = ConfigurationManager.ConnectionStrings["empConnString"].ToString();

SqlConnection conn = null; SqlCommand cmd = null; SqlDataAdapter da = null; SqlCommandBuilder builder = null; DataSet ds = null;

public void clearTextBox()

{

TbEid.Text = ""; TbEname.Text = ""; TbDesignation.Text = ""; TbSalary.Text = "";

}

public void showData()

{

try

{

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

da = new SqlDataAdapter(); ds = new DataSet();

da.MissingSchemaAction = MissingSchemaAction.AddWithKey; cmd = new SqlCommand("Select \* from emp\_info", conn); cmd.CommandType = CommandType.Text;

da.SelectCommand = cmd; da.Fill(ds, "emp\_info");

GvEmployee.DataSource = ds.Tables["emp\_info"]; GvEmployee.DataBind();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

conn = new SqlConnection(connString); showData();

}

//Adding info in database

protected void Add\_Info(object sender, EventArgs e)

{

try

{

if (TbEid.Text != "" && TbEname.Text != "" && TbDesignation.Text != "" &&

TbSalary.Text != "")

{

builder = new SqlCommandBuilder(da);

DataRow drNew = ds.Tables["emp\_info"].NewRow(); drNew[0] = TbEid.Text;

drNew[1] = TbEname.Text; drNew[2] = TbDesignation.Text; drNew[3] = TbSalary.Text;

ds.Tables["emp\_info"].Rows.Add(drNew); builder.GetInsertCommand();

int r = da.Update(ds, "emp\_info"); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record Inserted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record Not Inserted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter

all details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

//Deleting row from database

protected void delete\_info(object sender, EventArgs e)

{

try

{

if (TbEid.Text != "")

{

builder = new SqlCommandBuilder(da); DataRow drDelete =

ds.Tables["emp\_info"].Rows.Find(Convert.ToInt32(TbEid.Text));

drDelete.Delete(); builder.GetDeleteCommand();

int r = da.Update(ds, "emp\_info"); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record Deleted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record Not Deleted!');</script>");

}

}

else

{

No.!');</script>");

}

Response.Write("<script type=\"text/javascript\">alert('Enter Roll

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

//Updated info in database

protected void update\_info(object sender, EventArgs e)

{

try

{

if (TbEid.Text != "" && TbEname.Text != "" && TbDesignation.Text != "" && TbSalary.Text != "")

{

builder = new SqlCommandBuilder(da); DataRow drUpdate =

ds.Tables[0].Rows.Find(Convert.ToInt32(TbEid.Text));

drUpdate[1] = TbEname.Text; drUpdate[2] = TbDesignation.Text; drUpdate[3] = TbSalary.Text; builder.GetUpdateCommand();

int r = da.Update(ds, "emp\_info"); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record updated Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not updated!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter

details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

protected void search\_info(object sender, EventArgs e)

{

try

{

if (TbEid.Text != "")

{

DataRow drSearch = ds.Tables[0].Rows.Find(Convert.ToInt32(TbEid.Text));

if (drSearch != null)

{

Response.Write("<script type=\"text/javascript\">alert('Record

Found Successfully!');</script>");

clearTextBox();

TbEid.Text = drSearch[0].ToString(); TbEname.Text = drSearch[1].ToString();

TbDesignation.Text = drSearch[2].ToString(); TbSalary.Text = "\*\*\*\*\*\*\*\*\*\*";

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not Found!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter

Roll No.!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

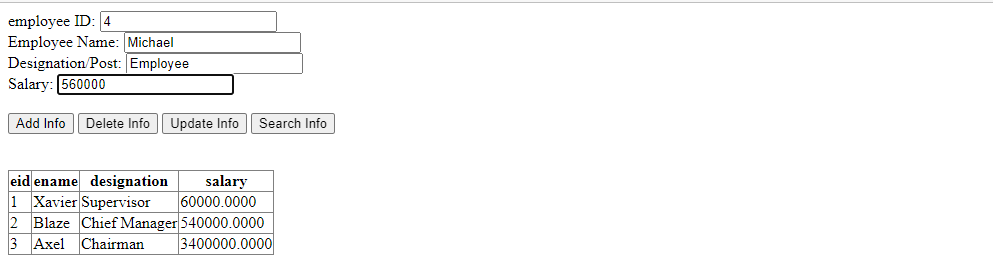
}

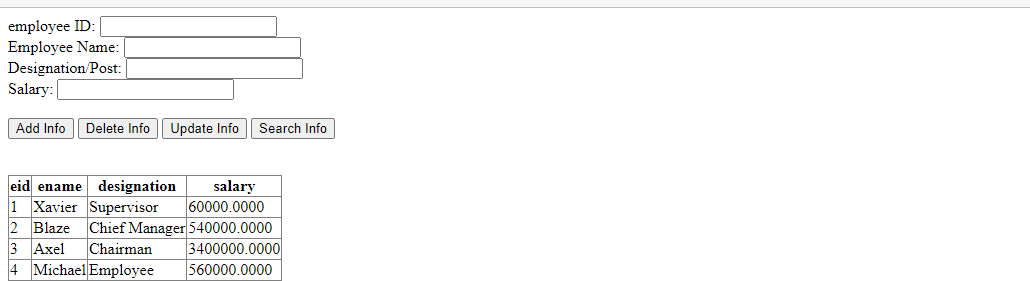
}

}

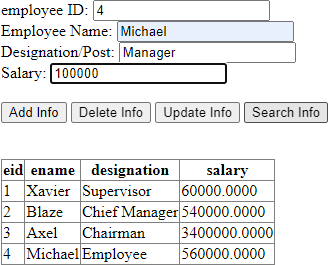
}

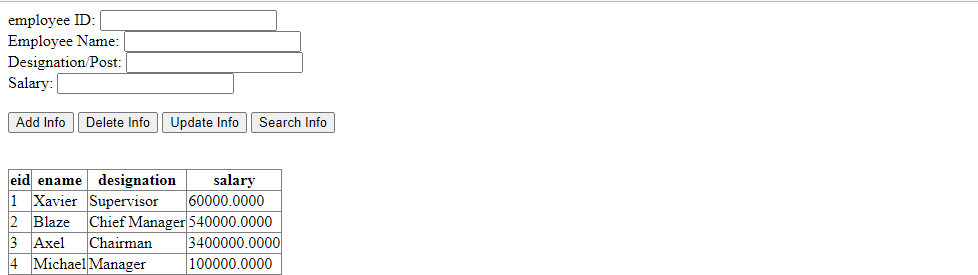
# Output: Insert



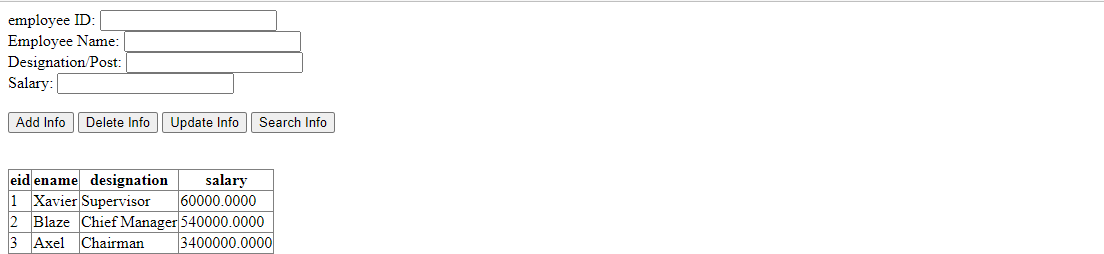


**Update:**

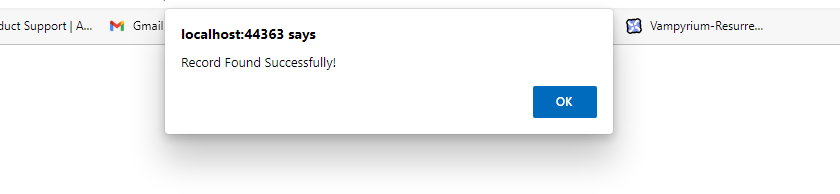




# Delete:



**Search:**



# Create stored procedures to add and search employee information from emp\_info table and design asp.net web form to call both procedures using connected architecture.

**Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="connectedDb.aspx.cs" Inherits="dbDemo1.WebForm1" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="Label1" runat="server" Text="Enter Roll No. :"></asp:Label>

<asp:TextBox ID="txtRollNo" runat="server"></asp:TextBox>

<br />

<asp:Label ID="Label2" runat="server" Text="Enter Name :"></asp:Label>

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<br />

<asp:Label ID="Label3" runat="server" Text="Enter Address :"></asp:Label>

<asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>

<br />

<br />

<asp:GridView ID="gvStudDetails" runat="server">

</asp:GridView>

<br />

<asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnDelete" runat="server" Text="Delete" OnClick="btnDelete\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnUpdate" runat="server" Text="Update" OnClick="btnUpdate\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSearch" runat="server" Text="Search" OnClick="btnSearch\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSort" runat="server" Text="Sort" OnClick="btnSort\_Click"

/>&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtSort" runat="server"></asp:TextBox>

<br />

<asp:Button ID="btnAddSP" runat="server" Text="Add using Stored Proc" OnClick="btnAddSP\_Click" />&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSearchSP" runat="server" Text="Search using Stored Proc" OnClick="btnSearchSP\_Click" />

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

using System.Collections.Generic; using System.Configuration; using System.Data;

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

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

public partial class WebForm1 : System.Web.UI.Page

{

static string conStr = ConfigurationManager.ConnectionStrings["studConnString"].ToString(); SqlConnection conn = null;

SqlCommand cmd = null;

SqlDataReader dr = null; DataTable dt = null;

public void clearTextBox()

{

txtRollNo.Text = ""; txtName.Text = ""; txtAddress.Text = "";

}

public void showData()

{

try

{

cmd = new SqlCommand("SELECT \* FROM student", conn);

//cmd.Connection = conn;

//cmd.CommandText = "SELECT \* FROM stud57";

//cmd.CommandType = CommandType.Text; if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

dt = new DataTable();

dr = cmd.ExecuteReader(); dt.Load(dr); gvStudDetails.DataSource = dt; gvStudDetails.DataBind();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

conn = new SqlConnection(conStr); showData();

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand("INSERT INTO student(rollno,name,address)VALUES(@rn, @nm, @add)", conn); if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text); cmd.Parameters.AddWithValue("@nm", txtName.Text); cmd.Parameters.AddWithValue("@add", txtAddress.Text); int r = cmd.ExecuteNonQuery();

if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record inserted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not inserted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter all details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

Response.Write("<script type=\"text/javascript\">alert('btn clicked');</script>"); try

{

if (txtRollNo.Text != "" && Convert.ToUInt16(txtRollNo.Text) > 0)

{

cmd = new SqlCommand("DELETE student WHERE RollNo=@rn", conn); if (conn.State == ConnectionState.Closed)

{ conn.Open(); } cmd.Parameters.AddWithValue("@rn", txtRollNo.Text);

int r = cmd.ExecuteNonQuery(); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record deleted Successfully!');</script>"); } else

{

Response.Write("<script type=\"text/javascript\">alert('Record not deleted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter correct RollNo!');</script>"); }

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand("UPDATE student SET name=@nm,address=@add WHERE rollno = @rn", conn); if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text); cmd.Parameters.AddWithValue("@nm", txtName.Text); cmd.Parameters.AddWithValue("@add", txtAddress.Text);

int r = cmd.ExecuteNonQuery(); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record updated successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not updated!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

}

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

{

try

{

cmd = new SqlCommand("SELECT \* FROM student WHERE RollNo=@rn", conn);

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Parameters.AddWithValue("@rn", txtRollNo.Text);

dr = cmd.ExecuteReader(); if (dr.Read())

{

Response.Write("<script type=\"text/javascript\">alert('Record found Successfully!');</script>"); clearTextBox();

txtRollNo.Text = dr["rollno"].ToString(); txtName.Text = dr["name"].ToString(); txtAddress.Text = dr["address"].ToString();

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not found!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

try

{

DataView dv = new DataView(dt);

//dv.Sort = txtSort.Text; // "name ASC"; dv.RowFilter = txtSort.Text; //"address = 'Ratnagiri'"; gvStudDetails.DataSource = dv; gvStudDetails.DataBind();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

}

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

{

try

{

if (txtRollNo.Text != "" && Convert.ToInt32(txtRollNo.Text) > 0)

{

cmd = new SqlCommand();

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Connection = conn;

//set command type as stored procedure cmd.CommandType = CommandType.StoredProcedure;

//pass the stored procedure name cmd.CommandText = "searchData";

cmd.Parameters.Add(new SqlParameter("@rollno", SqlDbType.SmallInt)).Value = Convert.ToInt16(txtRollNo.Text);

dr = cmd.ExecuteReader(); if (dr.HasRows)

{

Response.Write("<script type=\"text/javascript\">alert('Record found Successfully!');</script>");

while (dr.Read())

{

clearTextBox();

txtRollNo.Text = dr["rollno"].ToString(); txtName.Text = dr["name"].ToString();

txtAddress.Text = dr["address"].ToString();

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not found!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter roll number!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close();

}

}

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

{

try

{

if (txtRollNo.Text != "" && txtName.Text != "" && txtAddress.Text != "")

{

cmd = new SqlCommand();

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

cmd.Connection = conn;

//set command type as stored procedure cmd.CommandType = CommandType.StoredProcedure;

//pass the stored procedure name cmd.CommandText = "insertData";

cmd.Parameters.Add(new SqlParameter("@rollno", SqlDbType.SmallInt)).Value = Convert.ToInt16(txtRollNo.Text);

cmd.Parameters.Add(new SqlParameter("@Name", SqlDbType.VarChar)).Value = txtName.Text;

cmd.Parameters.Add(new SqlParameter("@add", SqlDbType.VarChar)).Value = txtAddress.Text;

int r = cmd.ExecuteNonQuery(); if (r != 0)

{

Response.Write("<script type=\"text/javascript\">alert('Record inserted Successfully!');</script>");

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not inserted!');</script>");

}

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Please enter all details!');</script>");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</script>");

}

finally

{

conn.Close(); clearTextBox(); showData();

}

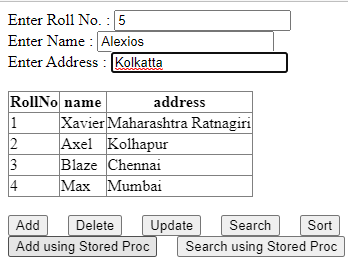
}

}

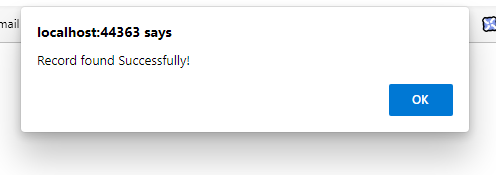
}

# Output:

**Stored Procedure to Add:**



# Stored Search Procedure



1. **Design asp.net web application to demonstrate DataList, DetailsView, FormView, GridView, ListView and Repeater Data Bound Controls. (Use ProductDetails table).**

# Code:

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="program4.aspx.cs" Inherits="dbDemo1.program4" %>

<!DOCTYPE html>

<html xmlns="<http://www.w3.org/1999/xhtml>">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="LblDataList" runat="server" Text="Data List" Font-Bold="True" Font-Names="Arial Black" Font- Size="Large"></asp:Label>

<asp:DataList ID="DataList1" runat="server" CellPadding="4" DataKeyField="eid" DataSourceID="SqlDataSourceDataList" ForeColor="#333333">

<AlternatingItemStyle BackColor="White" ForeColor="#284775" />

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<ItemStyle BackColor="#F7F6F3" ForeColor="#333333" />

<ItemTemplate> eid:

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

<br /> ename:

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

<br /> designation:

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

<br /> salary:

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

<br />

<br />

</ItemTemplate>

<SelectedItemStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />

</asp:DataList>

<asp:SqlDataSource ID="SqlDataSourceDataList" runat="server" ConnectionString="<%$ ConnectionStrings:empConnString %>" SelectCommand="SELECT \* FROM [emp\_info]"></asp:SqlDataSource>

<br />

<asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Names="Arial Black" Font-Size="Large" Text="Details View"></asp:Label>

<asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False" CellPadding="4" DataKeyNames="eid" DataSourceID="SqlDataSourceDataList" ForeColor="#333333" GridLines="None" Height="50px" Width="125px">

<AlternatingRowStyle BackColor="White" ForeColor="#284775" />

<CommandRowStyle BackColor="#E2DED6" Font-Bold="True" />

<EditRowStyle BackColor="#999999" />

<FieldHeaderStyle BackColor="#E9ECF1" Font-Bold="True" />

<Fields>

<asp:BoundField DataField="eid" HeaderText="eid" ReadOnly="True" SortExpression="eid" />

<asp:BoundField DataField="ename" HeaderText="ename" SortExpression="ename" />

<asp:BoundField DataField="designation" HeaderText="designation" SortExpression="designation" />

<asp:BoundField DataField="salary" HeaderText="salary" SortExpression="salary" />

</Fields>

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />

<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />

</asp:DetailsView>

<br />

<asp:Label ID="Label2" runat="server" Font-Bold="True" Font-Names="Arial Black" Font-Size="Large" Text="Form View"></asp:Label>

<asp:FormView ID="FormView1" runat="server" CellPadding="4" DataKeyNames="eid" DataSourceID="SqlDataSourceDataList" ForeColor="#333333">

<EditItemTemplate> eid:

<asp:Label ID="eidLabel1" runat="server" Text='<%# Eval("eid") %>' />

<br /> ename:

<asp:TextBox ID="enameTextBox" runat="server" Text='<%# Bind("ename") %>' />

<br /> designation:

<asp:TextBox ID="designationTextBox" runat="server" Text='<%# Bind("designation") %>' />

<br /> salary:

<asp:TextBox ID="salaryTextBox" runat="server" Text='<%# Bind("salary") %>' />

<br />

<asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True" CommandName="Update" Text="Update" />

&nbsp;<asp:LinkButton ID="UpdateCancelButton" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel" />

</EditItemTemplate>

<EditRowStyle BackColor="#999999" />

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<InsertItemTemplate> eid:

<asp:TextBox ID="eidTextBox" runat="server" Text='<%# Bind("eid") %>' />

<br /> ename:

<asp:TextBox ID="enameTextBox" runat="server" Text='<%# Bind("ename") %>' />

<br /> designation:

<asp:TextBox ID="designationTextBox" runat="server" Text='<%# Bind("designation") %>' />

<br /> salary:

<asp:TextBox ID="salaryTextBox" runat="server" Text='<%# Bind("salary") %>' />

<br />

<asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True" CommandName="Insert" Text="Insert"

/>

&nbsp;<asp:LinkButton ID="InsertCancelButton" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel" />

</InsertItemTemplate>

<ItemTemplate> eid:

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

<br /> ename:

<asp:Label ID="enameLabel" runat="server" Text='<%# Bind("ename") %>' />

<br /> designation:

<asp:Label ID="designationLabel" runat="server" Text='<%# Bind("designation") %>' />

<br /> salary:

<asp:Label ID="salaryLabel" runat="server" Text='<%# Bind("salary") %>' />

<br />

</ItemTemplate>

<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />

<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />

</asp:FormView>

<br />

<asp:Label ID="Label3" runat="server" Font-Bold="True" Font-Names="Arial Black" Font-Size="Large" Text="Grid View"></asp:Label>

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" CellPadding="4" DataKeyNames="eid" DataSourceID="SqlDataSourceDataList" ForeColor="#333333" GridLines="None">

<AlternatingRowStyle BackColor="White" ForeColor="#284775" />

<Columns>

<asp:BoundField DataField="eid" HeaderText="eid" ReadOnly="True" SortExpression="eid" />

<asp:BoundField DataField="ename" HeaderText="ename" SortExpression="ename" />

<asp:BoundField DataField="designation" HeaderText="designation" SortExpression="designation" />

<asp:BoundField DataField="salary" HeaderText="salary" SortExpression="salary" />

</Columns>

<EditRowStyle BackColor="#999999" />

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />

<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />

<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />

<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />

<SortedAscendingCellStyle BackColor="#E9E7E2" />

<SortedAscendingHeaderStyle BackColor="#506C8C" />

<SortedDescendingCellStyle BackColor="#FFFDF8" />

<SortedDescendingHeaderStyle BackColor="#6F8DAE" />

</asp:GridView>

<br />

<asp:Label ID="Label4" runat="server" Font-Bold="True" Font-Names="Arial Black" Font-Size="Large" Text="List View"></asp:Label>

<br />

<asp:ListView ID="ListView1" runat="server" DataKeyNames="eid" DataSourceID="SqlDataSourceDataList">

<AlternatingItemTemplate>

<span style="">eid:

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

<br /> ename:

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

<br /> designation:

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

<br /> salary:

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

<br />

<br />

</span>

</AlternatingItemTemplate>

<EditItemTemplate>

<span style="">eid:

<asp:Label ID="eidLabel1" runat="server" Text='<%# Eval("eid") %>' />

<br /> ename:

<asp:TextBox ID="enameTextBox" runat="server" Text='<%# Bind("ename") %>' />

<br /> designation:

<asp:TextBox ID="designationTextBox" runat="server" Text='<%# Bind("designation") %>' />

<br /> salary:

<asp:TextBox ID="salaryTextBox" runat="server" Text='<%# Bind("salary") %>' />

<br />

<asp:Button ID="UpdateButton" runat="server" CommandName="Update" Text="Update" />

<asp:Button ID="CancelButton" runat="server" CommandName="Cancel" Text="Cancel" />

<br />

<br />

</span>

</EditItemTemplate>

<EmptyDataTemplate>

<span>No data was returned.</span>

</EmptyDataTemplate>

<InsertItemTemplate>

<span style="">eid:

<asp:TextBox ID="eidTextBox" runat="server" Text='<%# Bind("eid") %>' />

<br /> ename:

<asp:TextBox ID="enameTextBox" runat="server" Text='<%# Bind("ename") %>' />

<br /> designation:

<asp:TextBox ID="designationTextBox" runat="server" Text='<%# Bind("designation") %>' />

<br /> salary:

<asp:TextBox ID="salaryTextBox" runat="server" Text='<%# Bind("salary") %>' />

<br />

<asp:Button ID="InsertButton" runat="server" CommandName="Insert" Text="Insert" />

<asp:Button ID="CancelButton" runat="server" CommandName="Cancel" Text="Clear" />

<br />

<br />

</span>

</InsertItemTemplate>

<ItemTemplate>

<span style="">eid:

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

<br /> ename:

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

<br /> designation:

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

<br /> salary:

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

<br />

<br />

</span>

</ItemTemplate>

<LayoutTemplate>

<div id="itemPlaceholderContainer" runat="server" style="">

<span runat="server" id="itemPlaceholder" />

</div>

<div style="">

</div>

</LayoutTemplate>

<SelectedItemTemplate>

<span style="">eid:

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

<br /> ename:

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

<br /> designation:

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

<br /> salary:

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

<br />

<br />

</span>

</SelectedItemTemplate>

</asp:ListView>

<br />

<asp:Label ID="Label5" runat="server" Font-Bold="True" Font-Names="Arial Black" Font-Size="Large" Text="Repeater"></asp:Label>

<br />

<asp:Repeater ID="Repeater1" runat="server" DataSourceID="SqlDataSourceDataList">

<ItemTemplate>

<span style="">eid:

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

<br /> ename:

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

<br /> designation:

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

<br />

salary:

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

<br />

<br />

</span>

</ItemTemplate>

</asp:Repeater>

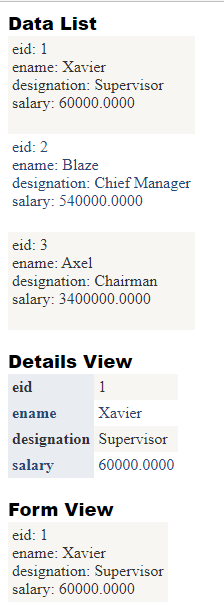
</div>

</form>

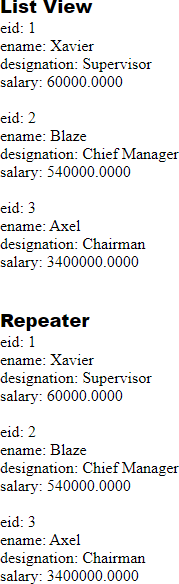
</body>

</html>

# Output:







1. **Design asp.net web form to search all words with length is less than five characters from string array using LINQ.**

# Code:

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="charLessThan5WithLinq.aspx.cs" Inherits="dbDemo1.charLessThan5WithLinq" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="lblHeading" runat="server" Text="LINQ program to find word less than 5 characters"></asp:Label>

<br />

<br />

<asp:Label ID="lblInputWords" runat="server" ></asp:Label>

<br />

<br />

<asp:Button ID="btnOutput" runat="server" OnClick="btn\_showSmallWords" Text="click for short words" />

<br />

<asp:Label ID="Label1" runat="server" Text="(Words less than 5 alphabets)"></asp:Label>

<br />

<br />

<asp:Label ID="lblOutputWords" runat="server" ></asp:Label>

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

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

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

public partial class charLessThan5WithLinq : System.Web.UI.Page

{

string[] words = { "I", "Am", "Wajahat", "Nizam", "Phansopkar" }; protected void Page\_Load(object sender, EventArgs e)

{

lblInputWords.Text = "Inputed Words: "; int count = 0;

foreach (string s in words)

{

if(count == 0)

{

lblInputWords.Text = lblInputWords.Text + "" + s;

}

else {

lblInputWords.Text = lblInputWords.Text + " , " + s;

}

count++;

}

}

protected void btn\_showSmallWords(object sender, EventArgs e)

{

//LINQ query to find out short words

var shortWords = from word in words where word.Length <= 5 select word;

//Loop to show short words lblOutputWords.Text = "Short Words: "; int count = 0;

foreach (var oneword in shortWords)

{

if (count == 0)

{

}

else

{

}

lblOutputWords.Text = lblOutputWords.Text + "" + oneword;

lblOutputWords.Text = lblOutputWords.Text + " , " + oneword;

count++;

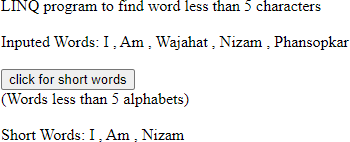
}

}

}

}

# Output:



1. **Design asp.net web form to find odd numbers from integer Array using LINQ Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="findOddNosWithLinq.aspx.cs" Inherits="dbDemo1.findEvenNosWithLinq" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="lblHeading" runat="server" Text="Finding odd numbers with LINQ"></asp:Label>

<br />

<br />

<asp:Label ID="lblInputNos" runat="server" ></asp:Label>

<br />

<br />

<asp:Button ID="btnFindEven" runat="server" OnClick="btnFindEven\_click" Text="Find Odd" />

<br />

<br />

<asp:Label ID="lblOutputNos" runat="server" ></asp:Label>

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

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

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

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

{

int[] numbers = new int[9] { 1, 2, 3, 4, 5, 6, 7,8,9, }; protected void Page\_Load(object sender, EventArgs e)

{

lblInputNos.Text = "Input Numbers : "; foreach (int n in numbers)

{

lblInputNos.Text = lblInputNos.Text + " " + n;

}

}

protected void btnFindEven\_click(object sender, EventArgs e)

{

IEnumerable<int> numQuery = from num in numbers

where (num % 2) != 0 select num;

// 3. Query execution. lblOutputNos.Text = "Odd Numbers: "; foreach (int n in numQuery)

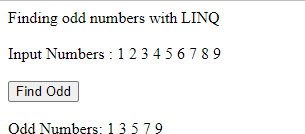
{ lblOutputNos.Text = lblOutputNos.Text + " " + n.ToString(); }

}

}

}

# Output:



1. **Design asp.net web application to perform add, update, delete and search operations on emp\_info Table using LINQ to SQL.**

# Code:

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="CRUDWithLinq.aspx.cs" Inherits="dbDemo1.CRUDWithLinq" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="LblHeading" runat="server" Text="CRUD operations using LINQ" Font-Bold="True" Font-Size="Large"></asp:Label>

<br />

<br />

<asp:Label ID="LblEid" runat="server" Text="employee ID:"></asp:Label>

<asp:TextBox ID="TbEid" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblName" runat="server" Text="Employee Name:"></asp:Label>

<asp:TextBox ID="TbEname" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblDesignation" runat="server" Text="Designation/Post:"></asp:Label>

<asp:TextBox ID="TbDesignation" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblSalary" runat="server" Text="Salary:"></asp:Label>

<asp:TextBox ID="TbSalary" runat="server"></asp:TextBox>

<br />

<br />

<asp:Button ID="BtnAddInfo" runat="server" Text="Add Info" OnClick="BtnAddInfo\_click"/>

&nbsp;<asp:Button ID="BtnDeleteInfo" runat="server" Text="Delete Info" OnClick="BtnDeleteInfo\_Click"/>

&nbsp;<asp:Button ID="BtnUpdateInfo" runat="server" Text="Update Info" OnClick="BtnUpdateInfo\_Click"/>

&nbsp;<asp:Button ID="BtnSearchInfo" runat="server" Text="Search Info" OnClick="BtnSearchInfo\_Click"/>

<br />

<br />

<br />

<asp:GridView ID="GvEmployee" runat="server" AutoGenerateColumns="False">

<Columns>

<asp:BoundField DataField="eid" HeaderText="eid" ReadOnly="True" SortExpression="eid" />

<asp:BoundField DataField="ename" HeaderText="ename" SortExpression="ename" />

<asp:BoundField DataField="designation" HeaderText="designation" SortExpression="designation" />

<asp:BoundField DataField="salary" HeaderText="salary" SortExpression="salary" />

</Columns>

</asp:GridView>

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

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

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

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

{

empInfoClassDataContext dc = new empInfoClassDataContext(); public void clearTextBox()

{

TbEid.Text = ""; TbEname.Text = ""; TbSalary.Text = ""; TbDesignation.Text = "";

}

public void showData()

{

try

{

}

var q = from a in dc.GetTable<emp\_info>() select a; GvEmployee.DataSource = q;

GvEmployee.DataBind();

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{ showData(); }

protected void BtnAddInfo\_click(object sender, EventArgs e)

{

try

{

//Data maping object to our database emp\_info objAdd = new emp\_info(); objAdd.eid = Convert.ToInt16(TbEid.Text);

added database

objAdd.ename = Convert.ToString(TbEname.Text); objAdd.salary = Convert.ToDecimal(TbSalary.Text); objAdd.designation = Convert.ToString(TbDesignation.Text);

//Adds an entity in a pending insert state to this

//System.Data.Linq.Table<TEntity>and parameter is the entity which to be

dc.emp\_infos.InsertOnSubmit(objAdd);

//executes the appropriate commands to implement the changes to the

dc.SubmitChanges(); clearTextBox();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

finally

{ showData(); }

}

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

{

try

{

//Get Single stud record which needs to Delete

emp\_info objDel = dc.emp\_infos.Single(emp\_info => emp\_info.eid ==

Convert.ToInt16(TbEid.Text));

//Puts an entity from this table into a pending delete state and parameter is the entity which to be deleted.

dc.emp\_infos.DeleteOnSubmit(objDel);

//executes the appropriate commands to implement the changes to the

database

dc.SubmitChanges(); clearTextBox();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

finally

{ showData(); }

}

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

{

try

{

emp\_info objSearch = dc.emp\_infos.Single(emp\_info => emp\_info.eid ==

Convert.ToInt16(TbEid.Text));

TbEid.Text = Convert.ToString(objSearch.eid); TbEname.Text = objSearch.ename;

TbSalary.Text = Convert.ToString(objSearch.salary); TbDesignation.Text = objSearch.designation;

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{

try

{

emp\_info objUpdate = dc.emp\_infos.Single(emp\_info => emp\_info.eid ==

Convert.ToInt16(TbEid.Text));

//Field which will be update

objUpdate.ename = Convert.ToString(TbEname.Text); objUpdate.salary = Convert.ToDecimal(TbSalary.Text); objUpdate.designation = Convert.ToString(TbDesignation.Text);

// executes the appropriate commands to implement the changes to the

database

dc.SubmitChanges(); clearTextBox();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

finally

{ showData(); }

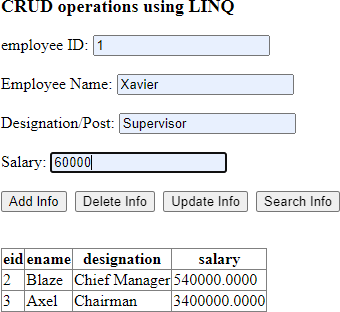
}

}

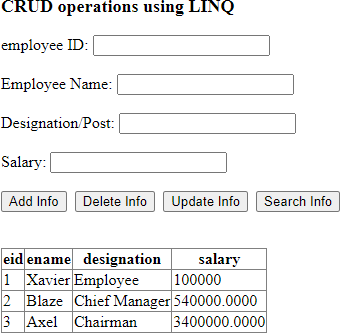
}

# Output: Insert:





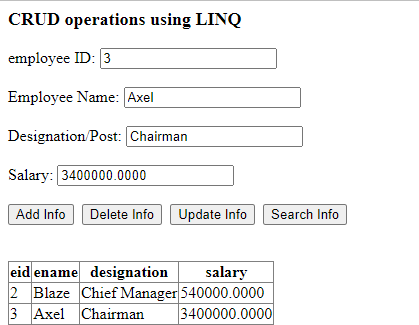
**Update:**



# Delete:



**Search:**



# Design asp.net web application to perform add, update, delete and search operations on ProductDetails Table using Entity Framework.

**Code:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="CRUDWithEntityFramework.aspx.cs" Inherits="dbDemo1.CRUDWithEntityFramework"

%>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xht](http://www.w3.org/1999/xhtml)ml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="LblHeading" runat="server" Text="CRUD operations using Entity framework" Font-Bold="True" Font-Size="Large"></asp:Label>

<br />

<br />

<asp:Label ID="LblEid" runat="server" Text="employee ID:"></asp:Label>

<asp:TextBox ID="TbEid" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblName" runat="server" Text="Employee Name:"></asp:Label>

<asp:TextBox ID="TbEname" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblDesignation" runat="server" Text="Designation/Post:"></asp:Label>

<asp:TextBox ID="TbDesignation" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="LblSalary" runat="server" Text="Salary:"></asp:Label>

<asp:TextBox ID="TbSalary" runat="server"></asp:TextBox>

<br />

<br />

<asp:Button ID="BtnAddInfo" runat="server" Text="Add Info" OnClick="BtnAddInfo\_Click"/>

&nbsp;<asp:Button ID="BtnDeleteInfo" runat="server" Text="Delete Info" OnClick="BtnDeleteInfo\_Click"/>

&nbsp;<asp:Button ID="BtnUpdateInfo" runat="server" Text="Update Info" OnClick="BtnUpdateInfo\_Click"/>

&nbsp;<asp:Button ID="BtnSearchInfo" runat="server" Text="Search Info" OnClick="BtnSearchInfo\_Click"/>

<br />

<br />

<br />

<asp:GridView ID="GvEmployee" runat="server" AutoGenerateColumns="False">

<Columns>

<asp:BoundField DataField="eid" HeaderText="eid" ReadOnly="True" SortExpression="eid" />

<asp:BoundField DataField="ename" HeaderText="ename" SortExpression="ename" />

<asp:BoundField DataField="designation" HeaderText="designation" SortExpression="designation" />

<asp:BoundField DataField="salary" HeaderText="salary" SortExpression="salary" />

</Columns>

</asp:GridView>

</div>

</form>

</body>

</html>

# Aspx.cs

using System;

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

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

using System.Web.UI.WebControls;

namespace dbDemo1

{

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

{

EmployeeDbEntities db = new EmployeeDbEntities();

public void clearTextBox()

{

TbEid.Text = ""; TbEname.Text = ""; TbDesignation.Text = ""; TbSalary.Text = "";

}

public void showData()

{

try

{

}

GvEmployee.DataSource = db.emp\_info.ToList(); GvEmployee.DataBind();

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{ showData(); }

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

{

try

{

emp\_info empObjAdd = new emp\_info(); empObjAdd.eid = Convert.ToInt32(TbEid.Text);

empObjAdd.ename = TbEname.Text; empObjAdd.designation = TbDesignation.Text; empObjAdd.salary = Convert.ToInt32(TbSalary.Text); db.emp\_info.Add(empObjAdd);

db.SaveChanges();

//Response.Write("<script type=\"text/javascript\">alert('Record added successfully!);</ script > ");

showData(); clearTextBox();

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{

try

{

emp\_info empObjDelete = db.emp\_info.Find(Convert.ToInt32(TbEid.Text)); if (empObjDelete != null)

{

db.emp\_info.Remove(empObjDelete); db.SaveChanges();

//Response.Write("<script type=\"text/javascript\">alert('Record deleted successfully!);</ script > ");

showData(); clearTextBox();

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not

found!);</ script > ");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{

try

{

emp\_info empObjSearch = db.emp\_info.Find(Convert.ToInt32(TbEid.Text)); if (empObjSearch != null)

{

}

else

{

TbEid.Text = empObjSearch.eid.ToString(); TbEname.Text = empObjSearch.ename; TbDesignation.Text = empObjSearch.designation; TbSalary.Text = empObjSearch.salary.ToString();

Response.Write("<script type=\"text/javascript\">alert('Record not found!);</ script > ");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

}

}

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

{

try

{

emp\_info empObjUpdate = db.emp\_info.Find(Convert.ToInt32(TbEid.Text)); if (empObjUpdate != null)

{

empObjUpdate.ename = TbEname.Text; empObjUpdate.designation = TbDesignation.Text; empObjUpdate.salary = Convert.ToInt32(TbSalary.Text); db.SaveChanges();

// Response.Write("<script type=\"text/javascript\">alert('Record updated successfully!);</ script > ");

showData(); clearTextBox();

}

else

{

Response.Write("<script type=\"text/javascript\">alert('Record not

found!);</ script > ");

}

}

catch (Exception ex)

{

Response.Write("<script type=\"text/javascript\">alert('Exception!'+ ex.Message);</ script > ");

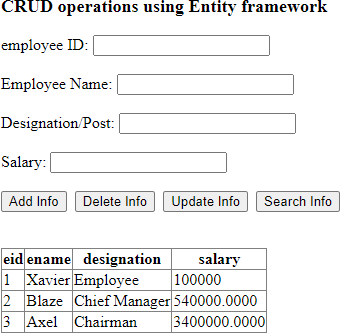
}

}

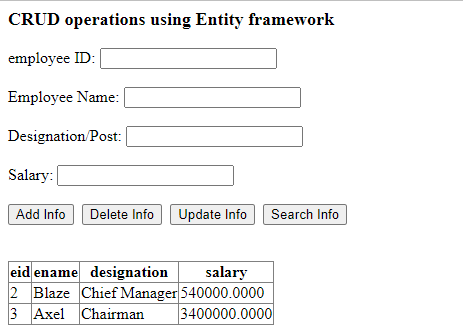
}

}

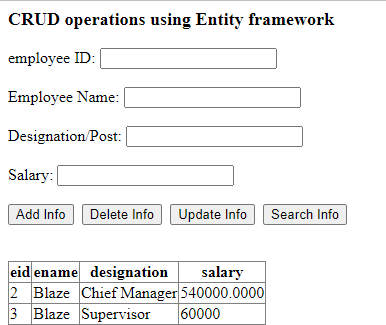
# Output: Insert:



**Delete:**



# Update:



**Search:**

