**Connection String**

* Binding the data in the table of data base to web form

1. Create a table in the data base
2. Create a stored procedure in sql of any functionality like a method to call it

create table people ( Id int, PName varchar(30), PAddress varchar(30))

insert into people values ( 1,'Nikhil','vizag')

insert into people values (2,'Hari','vizag')

creating stored procedure

CREATE PROCEDURE getPData

as

Begin

select Id,PName,PAddress from people

end

🡪while calling the sp name we can get the data

Create a asp.net web form project

Which contains or create

* Webform.aspx = HTML page to get output
* Webform.aspx.cs = c# page to code functionality an d connection
* Webform.config = where we write connection string

**1)WEBCONFIG**

There is a web config page where we write the below code after the </system.web>

<connectionStrings>

<add name ="connection\_people" connectionString="data source=APOLBO03-213; initial catalog=NIKHIL;Integrated security =True"

providerName="System.Data.SqlClient"></add>

</connectionStrings>

-- > Here after add name = name is the name we desired to give for the connection which we call it in c# in sqlConnection line

-- > connectionString = “data source = APOLBO03-213; is your sql server and we can give login and password if we have

-- > initial catalog = NIKHIL where it is the database name

-- > remaining integrated and provider name is necessary to write for the connection

**2)WEBFORM.ASPX**

Here we write code as html ,css, javaScript and for linking

For now we use grid code to show the data table from data base in body in a div section like below

<body>

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

<div>

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

<asp:GridView ID="grdPeople" runat="server"></asp:GridView><br />

// <asp:GridView ID="grdPeoples" runat="server"></asp:GridView>

</div>

</form>

</body>

-- >Here <form id = “form1” runat=”server”> runat is asp.net Attributes and it indicates that this is an server side form and this form is for server side processing of controls

-- > <asp:label> code line : this is an asp.net label control it has an id lbl indicating a server side reference for the control

And it is used in catch block to output exception and gridview is used to output the table

-- ><asp:gridview> code line : this is a asp.net grid view control with the id and run at server attribute makes it a server side control , grid view controls are used to display tabular data on a webpage

**3)WEBFORM.ASPX.CS**

To get data from a single table we use this datatable and for multiple tables view on webpage we use dataset

DATA TABLE

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Xml;

/\* 1)the up written libraries should be included\*/

/\* 2)the down three lines are created when we create a aspx.cs page automatically \*/

namespace DataBind.views

{

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

{

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

{

grdPeople.DataSource = getPeople();

grdPeople.DataBind();

}

/\* 4) the grdpeople is gridview id in aspx \*/

/\* 5) get people is a method we create \*/

public DataTable getPeople()

{

DataTable dt = new DataTable(); /\* 6) datatable creation \*/

try

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["connection\_people"].ConnectionString); /\* 7) sql connection with sp of connectionpeople we created in sql \*/

Con.open(0);

SqlCommand cmd = con.CreateCommand();

cmd.CommandText = "getPData";

cmd.CommandType = System.Data.CommandType.StoredProcedure;

SqlDataAdapter objDataAdapter = new SqlDataAdapter(cmd);

objDataAdapter.Fill(dt);

con.Close();

}

catch (Exception ex) /\* 8) catch exception \*/

{

lblException.Text = ex.Message;

}

return dt; /\* 9) return data \*/

}

For multiple tables view in webform we use dataset we should again create grdpeoples id in aspx

DATA SETS

1)

namespace DataBind.views

{

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

{

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

{

DataSet dsStudent = new DataSet();

dsStudent=getPeople();

grdPeople.DataSource = dsStudent.Tables[0];

grdPeople.DataBind();

grdPeoples.DataSource = dsStudent.Tables[1];

grdPeoples.DataBind();

}

public DataSet getPeople()

{

DataSet ds = new DataSet();

try

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["connection\_people"].ConnectionString);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandText = "DataNeed";

cmd.CommandType = System.Data.CommandType.StoredProcedure;

SqlDataAdapter da = new SqlDataAdapter(cmd);

da.Fill(ds);

con.Close();

}

catch (Exception ex)

{

lblException.Text = ex.Message;

}

return ds;

}

}

}

namespace DataBind.views

{

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

{

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

{

grdPeople.DataSource = getPeople();

grdPeople.DataBind();

grdPeoples.DataSource = getPeople();

grdPeoples.DataBind();

}

public DataSet getPeople()

{

DataSet ds = new DataSet();

try

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["connection\_people"].ConnectionString);

con.Open();

SqlCommand cmd = con.CreateCommand();

cmd.CommandText = "DataNeed";

cmd.CommandType = System.Data.CommandType.StoredProcedure;

SqlDataAdapter da = new SqlDataAdapter(cmd);

da.Fill(ds);

con.Close();

}

catch (Exception ex)

{

lblException.Text = ex.Message;

}

return ds;

}

**4)** grdPeople.DataSource = getEmployee();

grdPeople.DataBind();

--> here grdpeople is the id of gridview and datasource is the given to method name where the method is called and stored into id and in next line .dataBind(); function the data was binded

**5)** we created a method for getting connection and we called it in above 4) step and the method contains try and catchblocks

🡪dataTable dt = new dataTable();

Here datatable is inbuilt class and a fundamental data structure in ado.net framework it is used to create a structure for data similar to database table and we created a object dt for it

🡪in try block we created a sql connection by

SqlConnection con = new SqlConnection (ConfigurationManager.ConnectionStrings["connection\_people"].ConnectionString);

Where sqlConnection was inbuilt class provided by ado.net and it represents connection to sql server and we created con named object for it

🡪configurationmanager is a class that will be access to application configuration settings here we checks any connection\_people is name of connectionstring we given in web.config and .connectionstring specifies type of material we checking for here we are checking for connectionstring and ConfigurationManager.ConnectionString checks the connectionString in web.Config page

🡪 con.Open(); : opens con object

🡪 SqlCommand cmd = con.CreateCommand(); : sql command is a class provided by ado.net that you can execute against sql server and we create a object named cmd and stores con.createCommand where con is sql connection object and create command is a method in a sql connection class it is used to interact with database like insert and update

🡪 cmd.CommandText = "DataNeed"; here dataNeed is the stored procedure name we given in sql and command text is property of sql command class it should tell that do a specific task according to given value

🡪 cmd.CommandType = System.Data.CommandType.StoredProcedure; specifies the type of text on above line here we specify it is a stored procedure type

🡪SqlDataAdapter da = new SqlDataAdapter(cmd);

da.Fill(ds);

con.Close();

here sqladapter is a bridge between database and application and it will take cmd as parameter it is used to fill a dataset or table from database and we close the con

🡪 catch (Exception ex) {

lblException.Text = ex.Message;

}

return dt;

here we use catch exception . we also given lblexception in web.config page to output in webform and it shows the text which causes the error

and we return the dt which prints output;

PROGRAM FLOW