

Session 2

+1 Database Connectivity

Add new item → SQL Server Database
Database: Faculty.mdf

In server explores,
Faculty of.

Key:	Id	int	CREATE TABLE []
	Name	varchar	[FacultyInfo]
	Department	varchar	
	City	varchar	

click on \uparrow Update \rightarrow update database

& then add data as in the table

→ add new web form,

Add GridView Control

Add SQLDataSource

Click Δ \rightarrow select configure data source

Select Faculty.mdf

Do you want to save..

Yes as faculty.info.

Select query,

Test query

Gridview \rightarrow choose Data source &
SQL Data source.

Properties :-

Page size \Rightarrow 2, 5

Default key Name \rightarrow Id.

Auto Generate Edit button - True.

For configuring ^{add} update & delete query.

Configure Data source \rightarrow Advanced

Generate insert, ...

...

Refresh schema by the help of gridview.
Then run app.

Auto Generate Delete button - True.

Detail View Control

Add it.

It will display one record at a time

\rightarrow choose Data source

Enable paging

\Rightarrow You can't Insert record in gridview
it is possible by programming.

Auto Generate Edit \rightarrow True

Delete \rightarrow True

Insert \rightarrow True

Only 1 d.
1 "data"

Q In Gridview → select a record. ↗ "data"
which is displayed in detailView

For that 2 datasources are required.

Add other 'SqlDataSource' → configure it.

Select * from ' ' where Id = Control
Control ID → GridView1.

back → Advanced

Test Query

OK

→ DetailView → Datasource → 2

FormView Control

VB/SOA

```
using System.Data;  
using System.Data.Common;  
using System.Data.SqlClient;
```

```
protected void Button1_Click(object sender,  
EventArgs e)
```

{ try

```
    SqlConnection con = new SqlConnection();  
    con.ConnectionString = @"..."; ← copy & paste  
    "Data Source (LocalDB)\<name>"
```

```
    SqlCommand cmd = new SqlCommand();  
    cmd.CommandText = "Select * from  
    student";
```

```
    cmd.CommandType = CommandType.Text;
```

From
Webconfig

Options

cmd. Connection = con;

SqlDataAdapter da = new SqlDataAdapter();

~~da.Fill(ds)~~

da.SelectCommand = cmd;

DataTable dt = new DataTable();

|| DataSet ds = new DataSet();

con. Open()

da.Fill(dt); || da.Fill(ds);

con. Close();

catch (Exception e) { }

gridview1.DataSource = dt; ||= ds.Tables[0];

gridview1.DataBind();

→ Add Button, gridview.

→ DataSet is collection of DataTable

→ If DataSet is used then set of tables can be added, but here only one table we have, so, DataSet & DataTable works same.

→ When you do two times fill then in two different tables will be pointed at two different places.

→ Using System.Configuration; ← add this.

con. ConnectionString → Configuration Manager.

Connection strings ["FacultyHost" . ToString ()] ;

configuration file mathl name valucopy (name).

19/8/

student Info

allow nulls

Id

Name

Department

Password

Now update the table

- add web form

Add sql Data Source also

Id

Name

Dept

Pass

Submit

using System.Data;

using System.Data.Common;

using System.Data.SqlClient;

→ For Button Click :

In class

{

SqlConnection con;

SqlCommand cmd;

SqlDataAdapter da;

For button1 Click event

```
int id = TextBox1.Text;
String Name = TextBox2.Text;
con = new SqlConnection();
con.ConnectionString = @"data source = .\sqlexpress; initial catalog = studentinfo; user id = sa; password = 1234";
```

```
cmd = new SqlCommand();
cmd.CommandText = "Insert into studentInfo (Id, Name, Department, Password) values ('" + id + "','" + name + "','" + department + "','" + password + "')";
cmd.Connection = con;
con.Open();
cmd.ExecuteNonQuery();
con.Close();
Response.Write("No. of rows affected " + i);
```

For update query

```
update studentInfo set Name = '" + name +
" " + department
where id =
}
catch (Exception ex)
{
    Response.Write(ex.StackTrace().ToString());
}
```

For delete query ..

```
con.Delete from studentInfo where id =
```

→ Add dropdownlist & button.

[Value] []

[Retname]

In button click event.

query will be.

Select * from "Student Info";

cmd.Connection = con;

da = new SqlDataAdapter();

da.SelectCommand = cmd;

Datatable dt = new Datatable();

con.Open();

da.Fill(dt);

con.Close();

int i = dt.Rows.Count;

dropdownlist1.DataSource = dt;

dropdownlist1.DataBind();

dt.Columns[1].ToString();

dropdownlist1.DataTextField

dt.Columns[0].ToString();

dropdownlist1.DataBind();

this will
respect to
our table
result table

→ If we write select name, department
then it will show IT, IT, IT in
dropdown. Because result table will be

[Name] [Dept]

3 We can't add rows directly like at housing file stuff,

→ Create new row from (first username & pass)

Username []

pass []

I submit, }

class

{

// Create a connection,

and, commandText = "select * from
studentInfo";

string name = FB.Text;

string pass = FB.Text;
bool flag = false;

conn.Close();

for (int i = 0; i < dt.Rows.Count; i++)

{

if (uname.Equals(dt.Rows[i].

ItemArray[1].ToString()) &

6

pass.Equals(dt.Rows[i].

ItemArray[2].ToString()))

{

use Response.Write ("Logged In");

use Response.Write ("Entered Wrong Username
or Password");

```
flag = true;  
break;
```

}

```
if (flag)
```

```
{ R.W. ("logged in");
```

}

```
else
```

```
{ R.W. ("Enter valid username & password");
```

}

→ for efficient code
query:

```
Select * from _____ where name = " + B.I. Text + "
```

```
bool flag = false;
```

```
if (dt.Rows.Count > 0)
```

```
{ if (pwd.Equals(dt.Rows[0].ItemArray[3].ToString()))
```

```
    R.W. ("Logged in");
```

else {
 R.W. ("Incorrect password");
}

else {
 R.W. ("Please Enter valid info");
}

Parameterized Query (Prepared statement in java)

Con =

cmd =

```
{ SqlParameters id = new SqlParameter();
  id. parametersName = "@id";
  id. parametersValue = Convert.ToInt32
    (TB1. Text); }
```

Ctrl + V
for
name
all.

```
SqlParameters name = new SqlParameter();
name. parametersName = "@name";
name. value = TB2. Text;
cmd. Parameters. Add(id); // for all to same
query = insert into values (@id, @name,
@department, @password);
```

Instead of
SqlDataReader ds;

```
con. Open();
ds = cmd. ExecuteReader();
con. Close();
while (dr. NextResult) { }
r. w. (dr[0]. ToString() + "<br>");
```

con. Close();

→ Create a table in MS Access
Add in own explored

Using `System.Data.OleDb`;

Do same to whatever we do before
with OleDb instead of SQL.

`Con = new OleDbConnection();`

`Con.ConnectionString =`

$\downarrow i$

from

`SQLDatasource`

Query

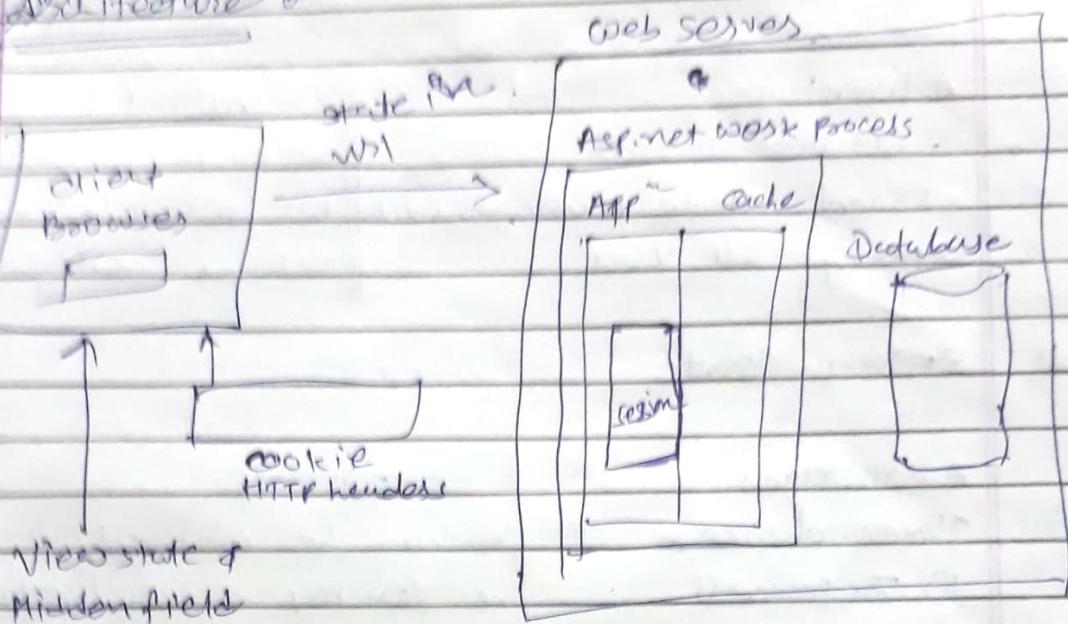
Insert into Students ([ID],
[UserName], [Password]) values
(@Id, @Username, @__);

21/8/19

State Session Management

- It is the process to store user specific app state of app
- We can store state either on client side or server side
- Each of these implementation has its own advantages & disadvantages

Architecture



Server side

App state
Cache object
Session state
Database ← most important

Client side

Cookie
Hidden field
View state
Control state
Query string (URL)

II life cycle of a request & state management process :-

A web browser makes an HTTP GET req. for page on your server, `http://.../`. The client web has never visited your site before. It's a your ASP.NET app responds by returning HTML rendered by `mypage.aspx`. Additionally...

Appⁿ state

Advantages

- fast
- shared state management among all users.

Disad

→ state is stored once per server in multiple servers configuration

session state

Advantages

Three choices :-

- I) In process
- II) Out
- (3) SQL DataBase

Dis.

- can be abused
- you pay a serialization cost when object leaves the process

DB &

→ can be accessed by any servers in multi servers configuration

→ need to pay serialization cost bcz object leaves appⁿ every time

Cache object (Appⁿ Scope)

→ like appⁿ state but includes expiration via dependencies

state stores once per server in multi servers configuration

Cookie

simple

Can be rejected by
browsers. Not
appropriate for large
amt of data. In
appropriate for
sensitive data.

Hidden field

simple for
page scoped
data

Not appropriate for
large amt of data.
Inappropriate for
sensitive data

Nice state

→ Create a web form

<@ page Trace = "True" ...
It gives all the info abt the request
headers etc.

Button - Click

HttpCookie c = new HttpCookie("name");
c.value = "Hello";
Response.Cookies.Add(c);
R.W. (" " + Request.LogonUserIdentity.
Name.ToString())

28/8/19

Add SQL Server Database .

Table = Students

Id int.

Name varchar(50)

Dept "

(updated↑)

Insert data in the table

→ Add web form

In that add button Insert Page

Add class file → student.cs

[Serializable]

```
public class Student
```

```
{ public
```

```
private int id;
```

```
private String name;
```

```
{ "dept";
```

```
public Student() { }
```

```
public Student(int ID, String NAME,
```

```
String DEPT)
```

```
{ this.id = ID;
```

```
this.name = NAME;
```

```
this.dept = DEPT;
```

```
}
```

// getter & setter methods

```
}
```

In button click event

```
{
```

```
Student s1 = new Student();
```

```
s1.id = 1;
```

```
s1.name = "Roshni";
```

```
s1.dept = "IT";
```

```
} session["Stdobj"] = s1;
```

```
Response.Redirect("Default2.aspx");
```

```
or ?Name = " + s1.name);
```

⇒ Add New web form

In page load event,

```
{
```

```
Student s1 = (Student)Session["Stdobj"];
```

```
Rw. (" " + s1.name);
```

In web config

<system.web>

<sessionState mode="StateServer" />

</system.web>

⇒ Control Panel → Administrative tools → services
→ ASP.NET state service will be running

⇒ Sessions are created per browser.

⇒ Add new web form.

page-load

if (e != null)

Request.Ob

R.W. (" + Request.QueryString["Name"]);

;

In 1st web form.

using System.Data.SqlClient;

Common;

System.Data;

Button click

SqlConnection con = new SqlConnection()
con.ConnectionString = " ";

SqlCommand cmd = new SqlCommand();
cmd.CommandText = "Select * from Student";
cmd.CommandType = CommandType.Text;
cmd.Connection = con;
DataSet ds = new DataSet();
SqlDataAdapter da = new SqlDataAdapter();
da.SelectCommand = cmd;
con.Open();
da.Fill(ds);
con.Close();
Student st = new Student();
st.Name = ds.Tables[0].Rows[0][0].ToString();
st.Session("data") = ds; Session
Response.Redirect("Default2.aspx");

In Default3.aspx:

Td Tb1 | 4 Read only,

Name

Dept.

In Page Load,

using System;

page load
student set

DataTable dt = 80Bills (DataTable)
Session["dt"]
TB1.Text = dt.Columns[0].ToString();
TB2.Text = dt.Columns[1].ToString();
TB3.Text = dt.Columns[2].ToString();
Rows[0][0]
C1
C2
Tables[0]

change the query also note

g