

## UCS1412 – Database Lab Assignment – 10

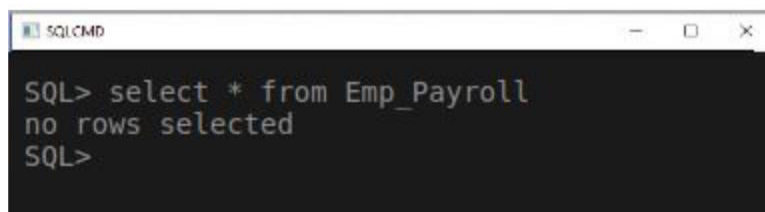
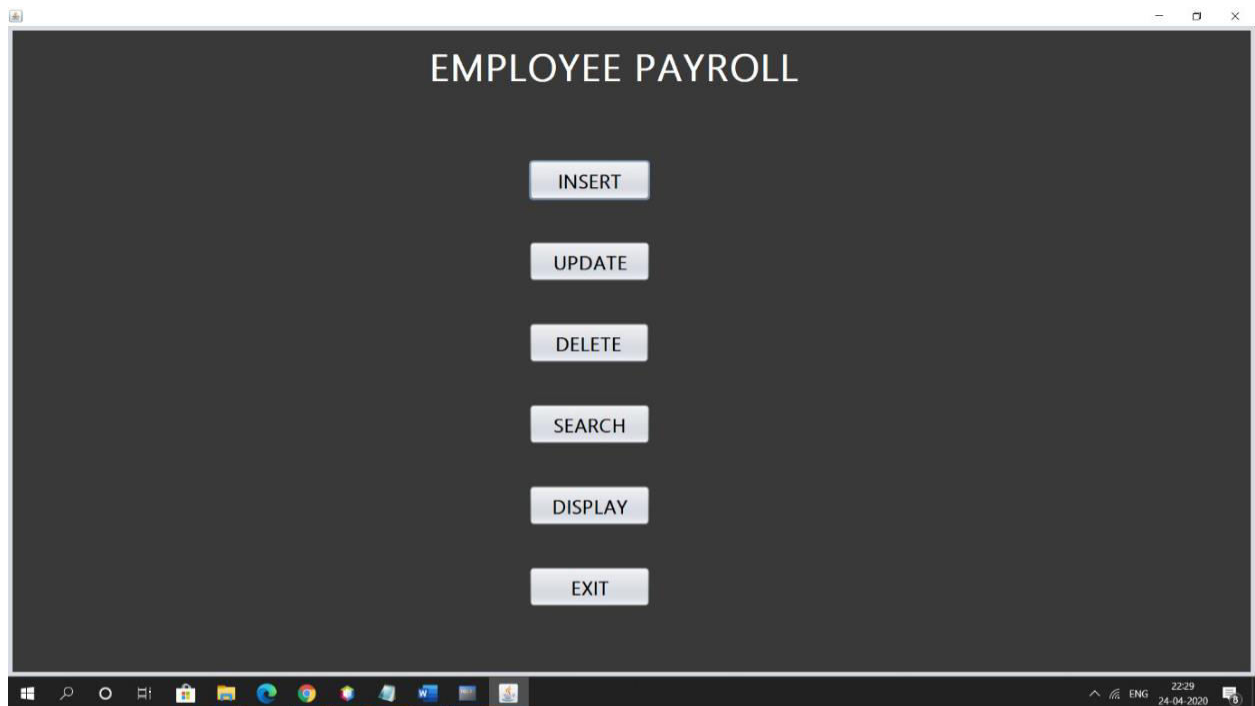
**Likhitha Verma A-185001084**

### **Title: Database Application Programming using JDBC**

This application maintains the employee payroll and allows insert, update, delete, search and display the details of the employee.

### **MODULES IN THE APPLICATION->**

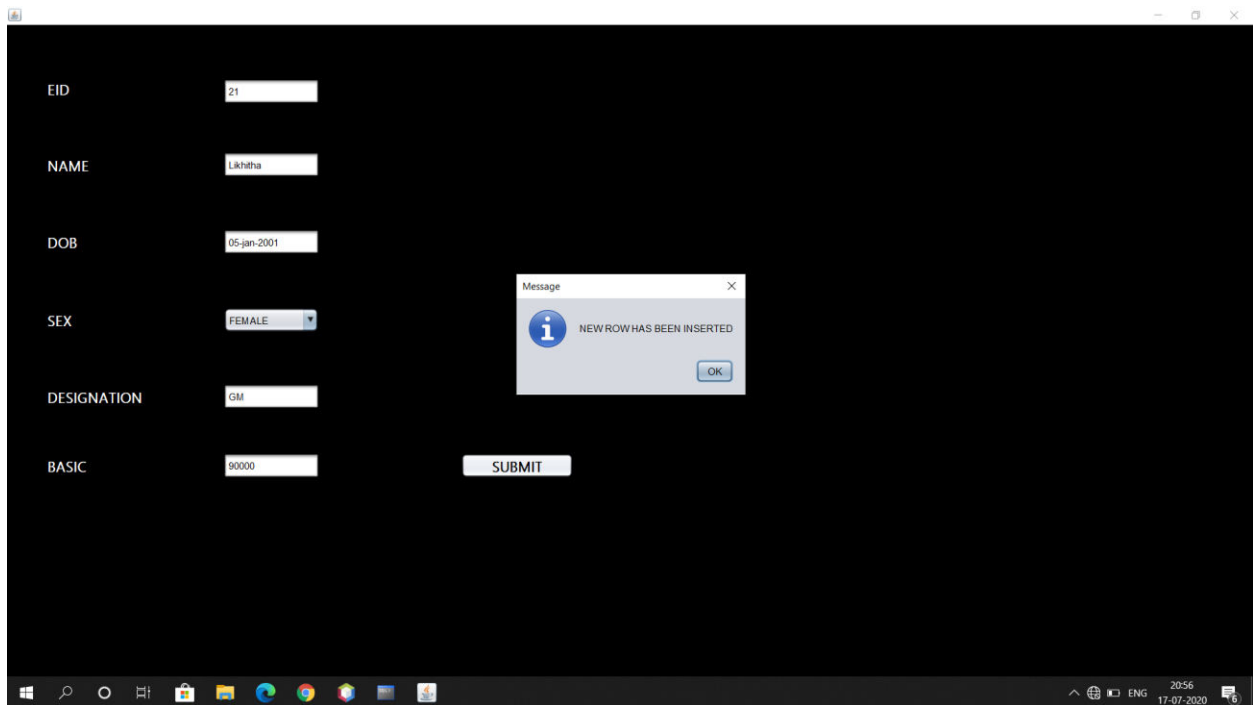
Basic Front End



## Insert Module

Allows insertion of an employee details into the database table Emp\_Payroll and calculates the net pay with basic using a procedure .

It also shows an error message if we try to insert another employee with the same id since it is the primary key of the table. After insert it will be redirected to the first page.



The screenshot shows a web application interface for inserting employee details. The form is set against a black background with white text and input fields. The fields are labeled EID, NAME, DOB, SEX, DESIGNATION, and BASIC. The values entered are 21, Lakshya, 05-jan-2001, FEMALE, GM, and 90000 respectively. A SUBMIT button is located at the bottom right of the form. A modal message box is displayed in the center, stating 'NEW ROW HAS BEEN INSERTED' with an 'OK' button. The Windows taskbar is visible at the bottom, showing the time as 20:56 on 17-07-2020.

Field	Value
EID	21
NAME	Lakshya
DOB	05-jan-2001
SEX	FEMALE
DESIGNATION	GM
BASIC	90000

Message: NEW ROW HAS BEEN INSERTED

SUBMIT

```
Run SQL Command Line
SQL> select * from Emp_Payroll;
```

EID	ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	TOT_DEDUC	NETPAY
21	Likhiitha	05-jan-2001	FEMALE	GM	90000	54000	9900	3600	2700	153900	6300	147600

```
SQL>
```

Insertion of 21(emp id) that is already present shows a pop up window.

EID:

NAME:


DOB:

SEX:

DESIGNATION:

BASIC:

Message

 VIOLATION OF PRIMARY KEY CONSTRAINT

EID

56

NAME

Jack

DOB

07-may-1995

SEX

MALE

DESIGNATION

Employee

BASIC

40000

SUBMIT

Message

i

NEW ROW HAS BEEN INSERTED

OK

Windows

Search

Task View

File Explorer

Google Chrome

Microsoft Edge

Internet Explorer

Visual Studio Code

Terminal

20:59

17-07-2020

ENG

EID

55

NAME

Yamini

DOB

12-sep-1988

SEX

FEMALE

DESIGNATION


Employee

BASIC

50000

SUBMIT

Message



NEW ROW HAS BEEN INSERTED

OK



ENG

21:01

17-07-2020

Run SQL Command Line

SQL> select \* from Emp\_Payroll;

EID	ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	TOT_DEDUC	NETPAY
21	Likhitha	05-jan-2001	FEMALE	GM	90000	54000	9900	3600	2700	153900	6300	147600

SQL> select \* from Emp\_Payroll;

EID	ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	TOT_DEDUC	NETPAY
21	Likhitha	05-jan-2001	FEMALE	GM	90000	54000	9900	3600	2700	153900	6300	147600
56	Jack	07-may-1995	MALE	Employee	40000	24000	4400	1600	1200	68400	2800	65600
55	Yamini	12-sep-1988	FEMALE	Employee	50000	30000	5500	2000	1500	85500	3500	82000

SQL>

21:01

17-07-2020

ENG

## Update Module

This option allows to modify the details of existing employees and the updation of basic pay will also calculate the new net pay.

After updating it will be redirected to the first page.

Updating an employee whose id is not present will be an error.

The screenshot displays the 'UPDATE' module interface and the SQL Command Line window. The 'UPDATE' form has a dark blue background with white text and input fields. It includes a label 'ENTER EID IN THE TEXT BOX BELOW TO UPDATE:' followed by a text box containing '55'. Below this are several labeled input fields: 'ENAME' (Yamini), 'DOB' (12-sep-1988), 'SEX' (FEMALE), 'DESIGNATION' (Employee), and 'BASIC' (60000). A 'Message' dialog box with an information icon and the text 'UPDATED!' is overlaid on the form. An 'UPDATE' button is at the bottom right of the form. The SQL Command Line window shows the results of a query on the 'Emp\_Payroll' table.

**UPDATE Form Fields:**

- ENTER EID IN THE TEXT BOX BELOW TO UPDATE: 55
- ENAME: Yamini
- DOB: 12-sep-1988
- SEX: FEMALE
- DESIGNATION: Employee
- BASIC: 60000
- UPDATE button

**SQL Command Line Output:**

```
SQL> select * from Emp_Payroll;
```

EID	ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	TOT_DEDUC	NETPAY
21	Likhitha	05-Jan-2001	FEMALE	GR	90000	54000	9900	3600	2700	153000	6300	147600
56	Jack	07-may-1995	MALE	Employee	40000	24000	4400	1600	1200	68400	2800	65600
55	Yamini	12-sep-1988	FEMALE	Employee	60000	36000	6600	2400	1800	102600	4200	98400

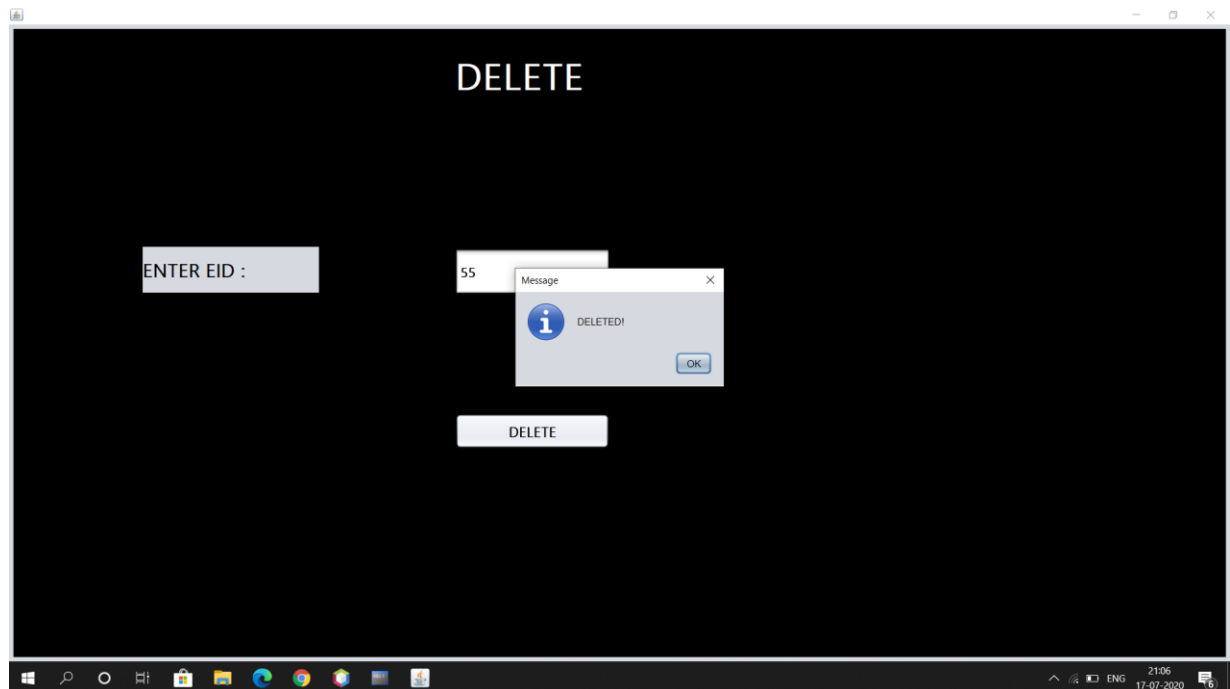
```
SQL> select * from Emp_Payroll;
```

EID	ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	TOT_DEDUC	NETPAY
21	Likhitha	05-Jan-2001	FEMALE	GR	90000	54000	9900	3600	2700	153000	6300	147600
56	Jack	07-may-1995	MALE	Employee	40000	24000	4400	1600	1200	68400	2800	65600

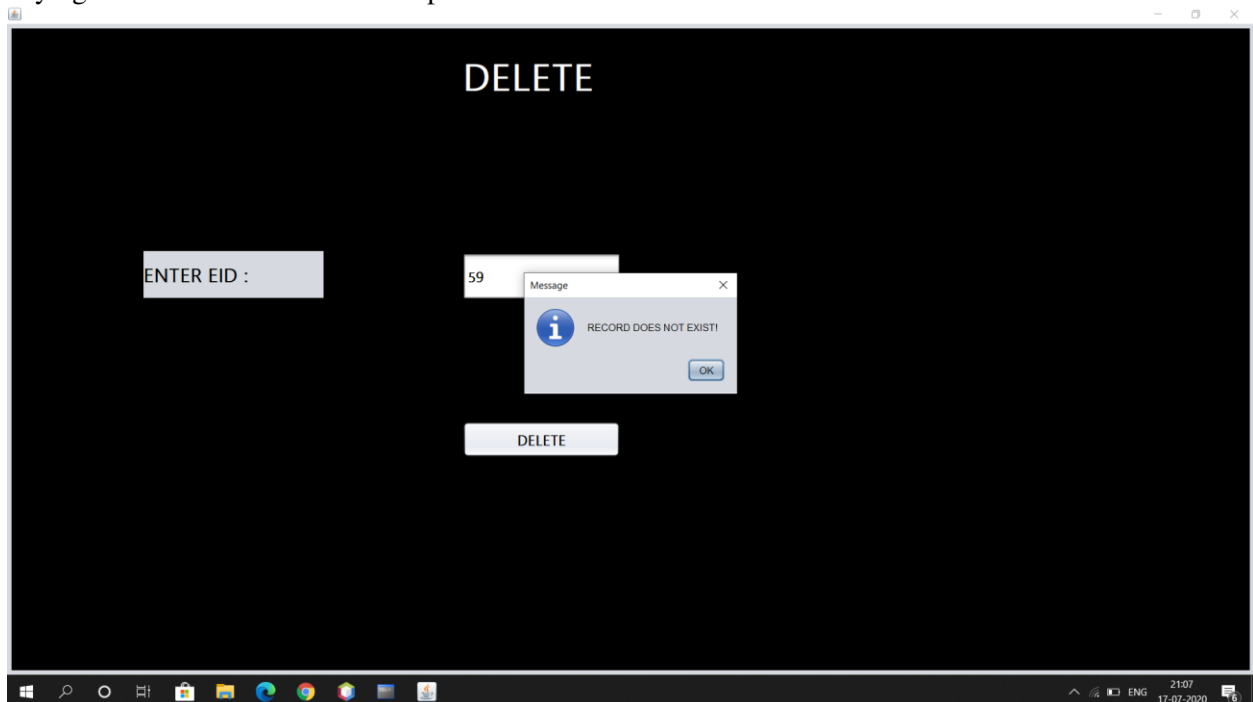
```
SQL>
```

## Delete Module

This module deletes the existing record and will show an error if the employee id is not present.



Trying to delete a record that is not present will show an error.





## Search Module

This module displays the searched record and shows a popup window if the employee id is not present. Back button will redirect to the first page.



SEARCH FOR A RECORD

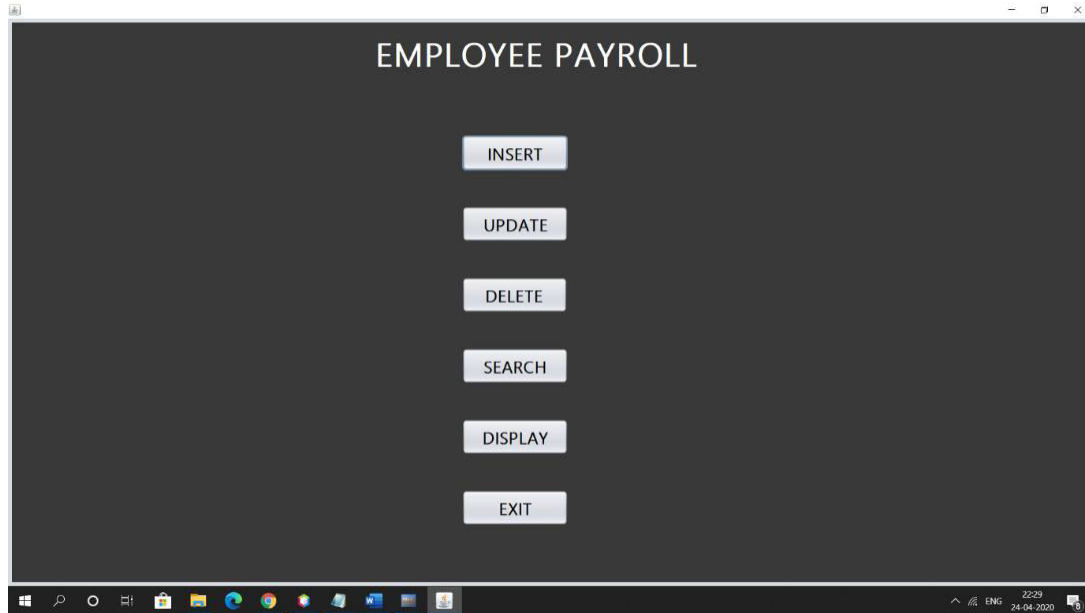
ENTER EID : 21 SEARCH

ENAME	DOB	SEX	DESIGNATION	BASIC	DA	HRA	PF	MC	GROSS	DEDUCTION	NETPAY
Likhitha	05-jan-2001	FEMALE	GM	90000.0	54000.0	9900.0	3600.0	2700.0	153900.0	6300.0	147600.0

BACK

## Exit Module

Exit option terminates the program.



**SQL CODE→**

**drop table Emp\_Payroll;**

**create table Emp\_Payroll(eid varchar2(20) primary key,ename char(20),dob  
char(20),sex char(20),designation char(20),basic number(10),**

**da number(10),hra number(10),pf number(10),mc number(10),gross  
number(10),tot\_deduc number(10),netpay number(10));**

**rem create procedure**

**create or replace procedure comp(id in Emp\_Payroll.eid % type,pay in Emp\_Payroll.basic % type) is**

**d        n**

**umber(10**

**); hr**

**number(**

**10); p\_f**

**number(**

**10); m\_c**

**number(**

**10); gr**

**number(**

**10); ded**

**number(**

**10); np**

**number(**

**10);**

**BEGIN**

**d:=(60\*pay)/100;**

**hr:=(11\*pay)/100;**

```
p_f:=(4*pay)/100;  
m_c:=(3*pay)/100;  
gr:=pay+d+hr;  
ded:=p_f+m_c;  
np:=gr-ded;  
update Emp_Payroll set basic=pay,da=d,hra=hr,pf=p_f,mc=m_c,gross=gr,tot_deduc=ded,  
netpay=np where eid=id;  
  
END;
```

## **JAVA APPLICATION 10(Extracted from netbeans 10)**

**src file contents-->Under java application 10->**

**Javaapplication10.java(CONTENTS)**

```
package javaapplication10;

import java.sql.*;

/*
 * @author
 * Likhitha*/

public class JavaApplication10
{
    public static void main(String[] args)
    {
        Connection conn=null;

        try
        {
            String driverName="oracle.jdbc.driver.OracleDriver"; Class.forName(driverName);

            conn=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-N6BHVVF:1521:XE",
"likki", "likki123");

        }

        catch (ClassNotFoundException e)

        {
            System.out.println("could not find conn db driver"+e.getMessage());
        }

        catch(SQLException e)

        {
            System.out.println("could not connect to db"+e.getMessage());
        }

    }
}
```

## IMPLEMENTATION CODE->

MODULE CODES→

### If insert button is clicked

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
insertop insert=new insertop();  
insert.setVisible(true);  
}
```

### If update button is clicked

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
updateop update=new updateop();  
update.setVisible(true);  
}
```

### If delete button is clicked

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
deleteop delete=new deleteop();  
delete.setVisible(true);  
}
```

### If search button is clicked

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
searchop search=new searchop();  
search.setVisible(true);  
}
```

### If display button is clicked

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
displayop display=new displayop();  
display.setVisible(true);  
}
```

### If exit button is clicked

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
this.dispose();  
}
```

## NAVIGATING BUTTON CODES->

### If SUBMIT button is clicked

```
String eid=jTextField1.getText();
String name=jTextField2.getText();
String dob=jTextField3.getText();
String sex=jComboBox1.getSelectedItem().toString();
String desig=jTextField4.getText();
float basic=Float.parseFloat(jTextField5.getText());
try{
String msg="select * from Emp_Payroll";
PreparedStatement s=conn.prepareStatement(msg);
ResultSet res = s.executeQuery();
while(res.next()) {
String data=res.getString(1);
if(data.equals(eid))
{
JOptionPane.showMessageDialog(null,"VIOLATION OF PRIMARY KEY CONSTRAINT");
this.dispose();
}
}
s.close();
res.close();
PreparedStatement st=conn.prepareStatement("insert into
Emp_Payroll(eid,ename,dob,sex,designation,basic) values(?,?,?,?,?,?)");
st.setString(1,eid);st.setString(2,name);st.setString(3,dob);st.setString(4,sex);st.setString(5,desig);
st.setFloat(6, basic);
int rs=st.executeUpdate();
System.out.println(rs+" status");
st.close();
PreparedStatement ss=conn.prepareStatement("call comp(?,?)");
ss.setString(1,eid);
ss.setFloat(2, basic);
int rss=ss.executeUpdate();
ss.close();
JOptionPane.showMessageDialog(null,"NEW ROW HAS BEEN INSERTED");
this.dispose();
} catch(SQLException e)
{ e.printStackTrace(); }
```

#### If UPDATE Button is clicked

```
try {
String id=jTextField1.getText();
String msg="update Emp_Payroll set ename='"+jTextField2.getText()+"
',dob='"+jTextField3.getText()+"",
sex='"+jComboBox1.getSelectedItem().toString()+"',
designation='"+jTextField4.getText()+"
where eid='"+jTextField1.getText()+"'";
PreparedStatement st=conn.prepareStatement(msg);
int res=st.executeUpdate();
System.out.println(res+" rest status");
st.close();
PreparedStatement s=conn.prepareStatement("call comp(?,?)");
s.setString(1,id);
s.setFloat(2,Float.parseFloat(jTextField5.getText()));
int rs=s.executeUpdate();
s.close();
if(res==1)
{JOptionPane.showMessageDialog(null,"UPDATED!");
this.dispose();
} else
{
JOptionPane.showMessageDialog(null,"RECORD NOT FOUND!");
this.dispose();
}
}
catch (SQLException ex)
{ ex.printStackTrace();
}
```

#### If DELETE Button is clicked

```
try {
String id=jTextField1.getText();
String msg="delete from Emp_Payroll where eid='"+id+"'";
PreparedStatement st=conn.prepareStatement(msg); int
res=st.executeUpdate();
System.out.println(res+" rest status");
st.close();
if(res==1)
JOptionPane.showMessageDialog(null,"DELETED!");
else
JOptionPane.showMessageDialog(null,"RECORD DOES NOT EXIST!");
} catch (SQLException ex)
{ ex.printStackTrace(); }
this.dispose();
```



#### If SEARCH Button is clicked

```
try {
String id=jTextField1.getText();
String msg="select * from Emp_Payroll where eid='"+id+"'";
PreparedStatement st=conn.prepareStatement(msg);
ResultSet rs = st.executeQuery();
if (rs.next())
{ Object data[]={ rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getFloat(6),rs.getFloat(7),
rs.getFloat(8) ,rs.getFloat(9),rs.getFloat(10),rs.getFloat(11),rs.getFloat(12), rs.getFloat(13)

};
DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
tb.addRow(data);
}
else
JOptionPane.showMessageDialog(null,"RECORD NOT FOUND!");
st.close();
} catch (SQLException ex)
{ ex.printStackTrace();
}
```

#### If BACK Button is clicked

```
this.dispose();
```

#### If DISPLAY Button is clicked

```
try {
String msg="select * from Emp_Payroll";
PreparedStatement st=conn.prepareStatement(msg);
ResultSet rs = st.executeQuery();
while(rs.next())
{
Object data[] = {rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getFloat(6) ,rs.getFloat(7),rs.getFloat(8),rs.getFloat(9),rs.getFloat(10),rs.getFloat(11),rs.getFloat(12),
rs.getFloat(13)};

DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
tb.addRow(data);
}
st.close();
}
catch (SQLException ex) {
ex.printStackTrace();
}
```