#### Post Graduate Program in Full Stack Software Engineering - Java Specialization – Project 1

#### Developed by: -- TANMAY SINGH

#### Registration No: R211106800003

#### Project Title: - OFFICE REGISTRATION SYSTEM

#### Name: Tanmay Singh Student ID: S211106800003 Centre Name and Location: vashi branch Course name: PGJQP Batch Code: S210050

#### Date of submission: - 22nd July, 2020

**Index**

|  |  |  |
| --- | --- | --- |
| SR.NO | CONTENT | PAGE NO. |
| 1 | **AIM** | **4** |
| 2 | **OBJECTIVE** | **4** |
| 3 | **PURPOSE** | **5** |
| 4 | **USER INTERFACE** | **5** |
| 5 | **SDLC** | **6** |
| 6 | **CONFIGURATIONS** | **7** |
| 7 | **ENTRY PROFILE** | **7** |
| 8 | **EXIT PROFILE** | **7** |
| 9 | **PROPOSED SYSTEM** | **8** |
| 10 | **WEEKLY STATUS REPORT** | **9** |
| 11 | **PROJECT REQUIREMENT** | **10** |
| 12 | **ECLIPSE JAVA CODE** | **13** |
| 13 | **SQL SERVER CODE** | **31** |
| 14 | **BIBLIOGRAPHY** | **34** |

**AIM:--**

**Synopsis of Office Registration System**

**• Manage the information of Register**

#### • Integration of all records of Employee.

**Objective of the Project:--**

**After completing this course, the students will be able to design and develop a software project**

#### Purpose of the project:--

#### This software design specification is made with the purpose of outlining the software architecture and design of the Office Registration System in detail. The document will provide developers insight in meeting client’s needs efficiently and effectively. Moreover the document facilitates communication and understanding of the system by providing several views of the system design.

**User Interface Design**

**User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs.**

**Phases in Project Execution (SDLC)**

**The project will be carried out in the following phases:**

**■ Requirements Specification: In the requirements specification phase, an in-depth study of the requirements is conducted to depict the functionality of the system.**

**■ Analysis: In the analysis phase, the processes in the system and functioning of each process are identified. The project teams will analyse the case study before moving on to the design phase.**

**■ Design: In the design phase, the architecture of the system is established. The architecture defines the components of the system, and their interfaces and behaviours. The design document describes a plan to implement the requirements.**

**■ Implementation: In the implementation phase, the project is developed based on the specifications.**

**■ Testing: In the testing phase, the developed code is tested to ensure that it works correctly. This also involves making changes to the code, if it is not working correctly.**

**■ Deployment: In the deployment phase, the developed application is packaged and deployed at the user's end.**

**Configuration**

**Hardware: -- HP envy laptop, IOS mobile phone.**

**Operating system: - Windows 10 pro, ios13.4.4**

**Software: -- Microsoft SQL Server Management Studio, Eclipse IDE, Notepad, Google Chrome and safari.**

**Entry Profile**

**Students should have completed Full Stack Software Engineering Term 1.**

**Exit Profile**

**After completing this course, the students will be able to: ■ Develop real live projects and implement various strategies and techniques of software engineering and project management concepts.**

**The proposed system has the following requirements:**

**• System needs store information about new entry of Username.**

**• System need to maintain quantity record.**

**• System need to keep the record of Login.**

**• System need to update and delete the record.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Weekly Status Report** | | | |
| **No of weeks** | **progress** | **Completion in percentage** | **Completion date** |
| **1** | **Created the database names PSDBC and added five tables** | **12** | **1ST July 2020** |
| **2** | **Input the data in the tables** | **30** | **7th July 2020** |
| **3** | **Downloaded all the required software’s (eclipse, jdk 1.8 etc.)** | **40** | **15th July 2020** |
| **4** | **Generated code in eclipse and performed** Java Database Connectivity (**JDBC**) | **80** | **19th July 2020** |
| **5** | **Made the documentation** | **100** | **22nd July 2020** |

**Project requirement:--**

1. Open SQL Server Management Studio

Create database anmed PCSDB

Create following 5 tables:

-- Employee(empId[identity(1,1), primary key], firstName, lastName, userId, password, gender, role, active)

-- Skill(skillId[identity(1,1), primary key], skillName, skillDesc, active)

-- Job(jobId[identity(1,1), primary key], jobTitle, jobDescription, companyName, Location, keySkill,salary,active)

-- EMPSkill(ESId[identity(1,1), primary key], empId, skillId, expYear)

-- EmpJob(EJId[identity(1,1), primary key],empId,jobId,recruited)

2.create project in eclipse

a. Craete following packages:

model

dao

daoImpl

controller

view

test

config

b. Add following classes and interfaces as given:

i. within config package

-- JDBCConnection.java class

ii. within model package -- add classes

-- Employee.java

-- Skill.java

-- Job.java

iii. within dao package -- add interfaces

-- IEmployeeDao.java

-- ISkillDao.java

-- IJobDao.java

iv. within daoImpl package -- add classes

-- EmployeeDaoImpl.java

-- SkillDaoImpl.java

-- JobDaoImpl.java

v. within controller package -- add classes

-- EmployeeController.java

-- SkillController.java

-- JobController.java

vi. within view package -- add frame classes

-- LoginFrame.java

-- RegistrationFrame.java

-- EmpWindow.java (following are the services by Employee)

-- View Profile

-- Update Profile

-- Update Skill

-- Apply Job

-- Logout

-- PMWindow.java (following are the services by Project Manager)

-- View Skill-wise Employee List

-- View All Skills

-- Add Job

-- View All Job

-- View Skill-wise Job

-- View Employee list who applied for Job

-- Deactivate Job

-- Logout

-- HRWindow.java (following are the services by HR)

-- Active Employee

-- Deactive Employee

-- View All Employee

-- View Selective Employee

-- Add Skill

-- View All Skills

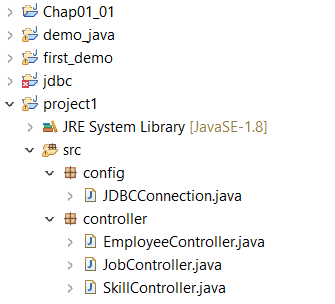
-- Deactive Skill

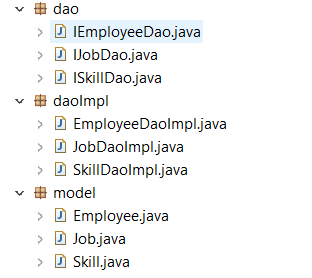
-- Logout

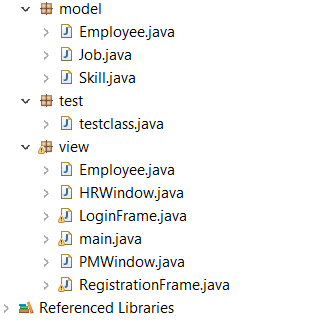
vii. within test package create test classes

**ECLIPSE IDE JAVA INPUT & OUTPUT**

**ALL PACKAGES AND CLASSES**







**\\**JDBCConnection.java

**package** config;

**import** java.sql.\*;

**public** **class** JDBCConnection {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

String url="jdbc:sqlserver:DESKTOP-CJ512VJ\\INSTANCE01:1433;databaseName=PCSDB;user=sa1;password=tanmay";

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

Connection conn=DriverManager.*getConnection*(url);

**if**(conn!=**null**)

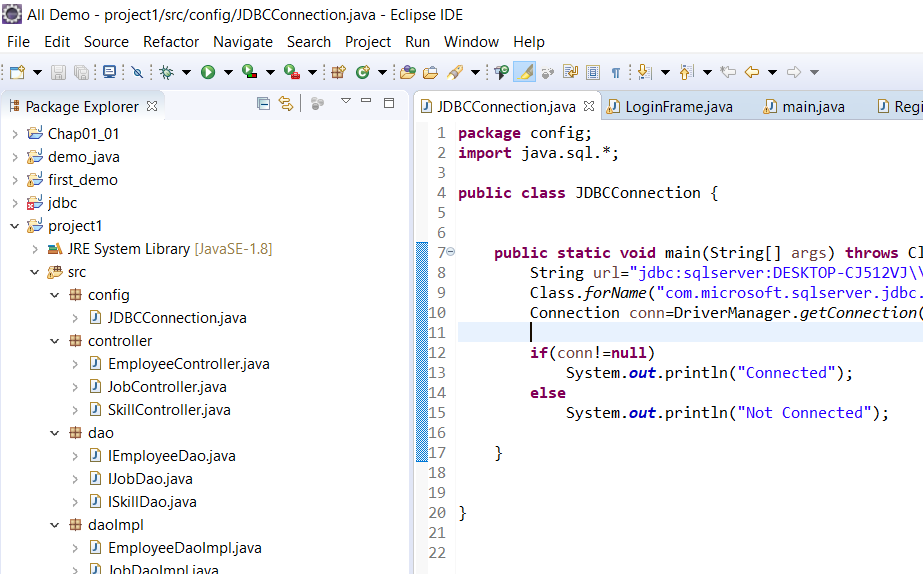
System.***out***.println("Connected");

**else**

System.***out***.println("Not Connected");

}

}



**\\LoginFrame.java**

**package** view;

**import** view.RegistrationFrame;

**import** view.Employee;

**import** view.HRWindow;

**import** config.JDBCConnection;

**import** java.awt.FlowLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

**public** **class** LoginFrame {

**public** **static** **void** main(String[] args) {

log obj=**new** log();

}

}

**class** log **extends** JFrame **implements** ActionListener

{

JLabel lTitle, lUser,lPassword,lMessage,lrole;

JTextField userid,passwords;

JPasswordField tPassword;

JButton bLogin,bregister;

JComboBox roleCombo;

**static** JDialog *dialog*;

/\*\*

\*

\*/

**public** log()

{

String[] role= {"HRA","EMP","PME"};

lTitle=**new** JLabel("Login Here");

lTitle.setBounds(180, 20, 150, 25);

lUser=**new** JLabel("User Id");

lUser.setBounds(50, 80, 150, 25);

lPassword=**new** JLabel("Password");

lPassword.setBounds(50, 160, 150, 25);

userid=**new** JTextField();

userid.setBounds(200, 80, 150, 25);

passwords=**new** JPasswordField();

passwords.setBounds(200, 160, 150, 25);

((JPasswordField) passwords).setEchoChar('\*');

lrole=**new** JLabel("Role");

lrole.setBounds(50, 200, 150, 25);

roleCombo=**new** JComboBox(role);

roleCombo.setBounds(200,200,90,20);

bLogin=**new** JButton("Login");

bLogin.setBounds(100, 250,150, 25);

bLogin.addActionListener(**this**);

bregister=**new** JButton("Register");

bregister.setBounds(250, 250,150, 25);

lMessage=**new** JLabel();

lMessage.setBounds(120, 320, 320, 25);

*dialog*=**new** JDialog(**this**,"Login",**true**);

add(lTitle);

add(lUser);

add(lPassword);

add(lrole);

add(roleCombo);

add(userid);

add(passwords);

add(bLogin);

add(bregister);

add(lMessage);

*dialog*.add(lMessage);

*dialog*.setSize(350, 200);

setSize(500,500);

setLayout(**null**);

setVisible(**true**);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

bregister.addActionListener(**new** ActionListener()

{

**public** **void** actionPerformed(ActionEvent e)

{

**new** RegistrationFrame();

}

});

}

@Override

**public** **void** actionPerformed(ActionEvent e)

{

**if**(e.getSource()==bLogin) {

**try** {

//PreparedStatement ps;

ResultSet rs;

String query = "SELECT \* FROM Employees WHERE Userid =? AND Pass =? AND Role=? And Active='TRUE'";

String url="jdbc:sqlserver:DESKTOP-CJ512VJ\\INSTANCE01:1433;databaseName=PCSDB;user=sa1;password=tanmay";

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

Connection conn=DriverManager.*getConnection*(url);

PreparedStatement ps1=conn.prepareStatement(query);

ps1.setString(1, userid.getText());

ps1.setString(2, passwords.getText());

ps1.setString(3, roleCombo.getSelectedItem().toString());

rs = ps1.executeQuery();

**if**(rs.next())

{

**if**(roleCombo.getSelectedItem().equals("HRA"))

{

HRWindow hra = **new** HRWindow();

**this**.dispose();

}

**if**(roleCombo.getSelectedItem().equals("PME"))

{

PMWindow pme = **new** PMWindow();

**this**.dispose();

}

**if**(roleCombo.getSelectedItem().equals("EMP"))

{

Employee emp = **new** Employee();

**this**.dispose();

}

}

**else**

{

JOptionPane.*showMessageDialog*(**null**, "Incorrect Username Or Password", "Login Failed", 2);

}

}

**catch** (SQLException ex) {

Logger.*getLogger*(LoginFrame.**class**.getName()).log(Level.***SEVERE***, **null**, ex);

} **catch** (ClassNotFoundException e1) {

// **TODO** Auto-generated catch block

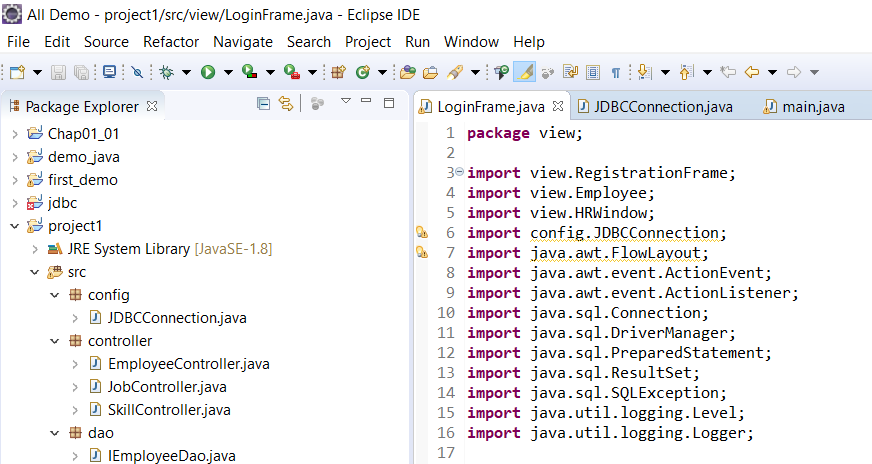
e1.printStackTrace();

}

}

}

}



**\\main.java**

**package** view;

**public** **class** main{

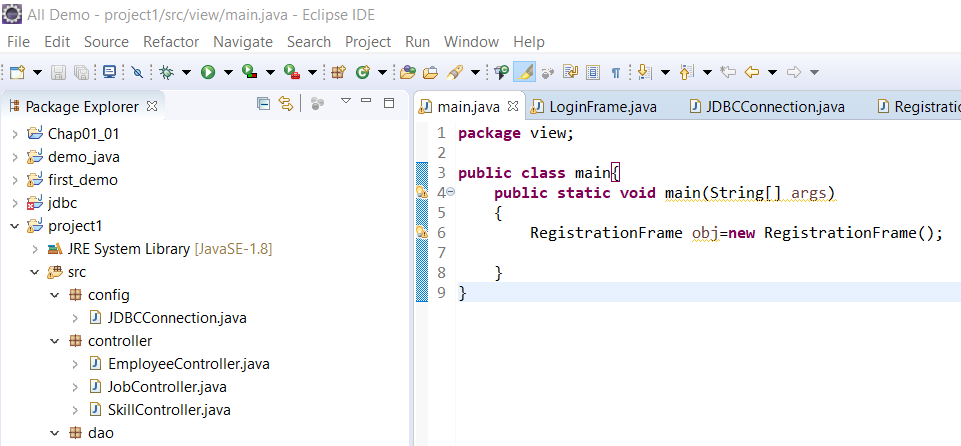
**public** **static** **void** main(String[] args)

{

RegistrationFrame obj=**new** RegistrationFrame();

}

}



\\RegistrationFrame.java

package view;

import javax.swing.\*;

import java.awt.event.ActionListener;

import java.awt.event.\*;

import java.awt.\*;

import java.sql.\*;

public class RegistrationFrame implements ActionListener {

JFrame frame;

String[] gender={"M","F"};

String[] role= {"HRA","EMP","PME"};

JLabel EmailId=new JLabel("Email Id");

JLabel FirstName=new JLabel("First Name");

JLabel LastName=new JLabel("Last Name");

JLabel Gender=new JLabel("Gender");

JLabel UserId=new JLabel("User Id");

JLabel Password=new JLabel("Password");

JLabel Role=new JLabel("Role");

JLabel Mobileno=new JLabel("Mobile No");

JTextField Emailid=new JTextField();

JTextField firstname=new JTextField();

JComboBox genderCombo=new JComboBox(gender);

JTextField lastname=new JTextField();

JTextField mobileno=new JTextField();

JTextField userid=new JTextField();

JPasswordField passwords=new JPasswordField();

JComboBox roleCombo=new JComboBox(role);

JButton registerButton=new JButton("REGISTER");

JButton resetButton=new JButton("RESET");

public RegistrationFrame()

{

createWindow();

setLocationAndSize();

addComponentsToFrame();

actionEvent();

}

public void createWindow()

{

frame=new JFrame();

frame.setTitle("Registration Form");

frame.setBounds(40,40,380,600);

frame.getContentPane().setBackground(Color.white);

frame.getContentPane().setLayout(null);

frame.setVisible(true);

//frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

}

public void setLocationAndSize()

{

EmailId.setBounds(20,20,60,70);

FirstName.setBounds(20,70,100,70);

LastName.setBounds(20,120,100,70);

UserId.setBounds(20,170,100,70);

Password.setBounds(20,220,140,70);

Role.setBounds(20,270,100,70);

Gender.setBounds(20,320,100,70);

Mobileno.setBounds(20,370,100,70);

Emailid.setBounds(180,43,165,23);

firstname.setBounds(180,93,165,23);

lastname.setBounds(180,143,165,23);

userid.setBounds(180,193,165,23);

passwords.setBounds(180,243,165,23);

roleCombo.setBounds(180,293,165,23);

genderCombo.setBounds(180,343,165,23);

mobileno.setBounds(180,393,165,23);

registerButton.setBounds(70,450,100,35);

resetButton.setBounds(220,450,100,35);

}

public void addComponentsToFrame()

{

frame.add(EmailId);

frame.add(FirstName);

frame.add(LastName);

frame.add(UserId);

frame.add(Password);

frame.add(Role);

frame.add(Gender);

frame.add(Mobileno);

frame.add(Emailid);

frame.add(firstname);

frame.add(lastname);

frame.add(userid);

frame.add(passwords);

frame.add(roleCombo);

frame.add(genderCombo);

frame.add(mobileno);

frame.add(registerButton);

frame.add(resetButton);

}

public void actionEvent()

{

registerButton.addActionListener(this);

resetButton.addActionListener(this);

}

/\*\*

\*

\*/

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==registerButton)

{

try {

Connection conn=DriverManager.getConnection("jdbc:sqlserver:DESKTOP-CJ512VJ\\INSTANCE01;databaseName=pcsdb","sa1","tanmay");

//Preapared Statement

PreparedStatement Pstatement=conn.prepareStatement("insert into Employees values(?,?,?,?,?,?,?,?,?)");

//Specifying the values of it's parameter

Pstatement.setString(1,Emailid.getText());

Pstatement.setString(2,firstname.getText());

Pstatement.setString(3,lastname.getText());

Pstatement.setString(4,userid.getText());

Pstatement.setString(5,passwords.getText());

Pstatement.setString(6,roleCombo.getSelectedItem().toString());

Pstatement.setString(7,genderCombo.getSelectedItem().toString());

Pstatement.setString(8,"FALSE");

Pstatement.setString(9,mobileno.getText());

Pstatement.executeUpdate();

Statement stmt = conn.createStatement();

String sql = "UPDATE Employees SET Active='TRUE' WHERE Role='HRA'";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(null,"Data Registered Successfully");

}

catch (SQLException e1) {

e1.printStackTrace();

}

}

if(e.getSource()==resetButton)

{

//Clearing Fields

Emailid.setText("");

firstname.setText("");

lastname.setText("");

userid.setText("");

passwords.setText("");

roleCombo.setSelectedItem("");

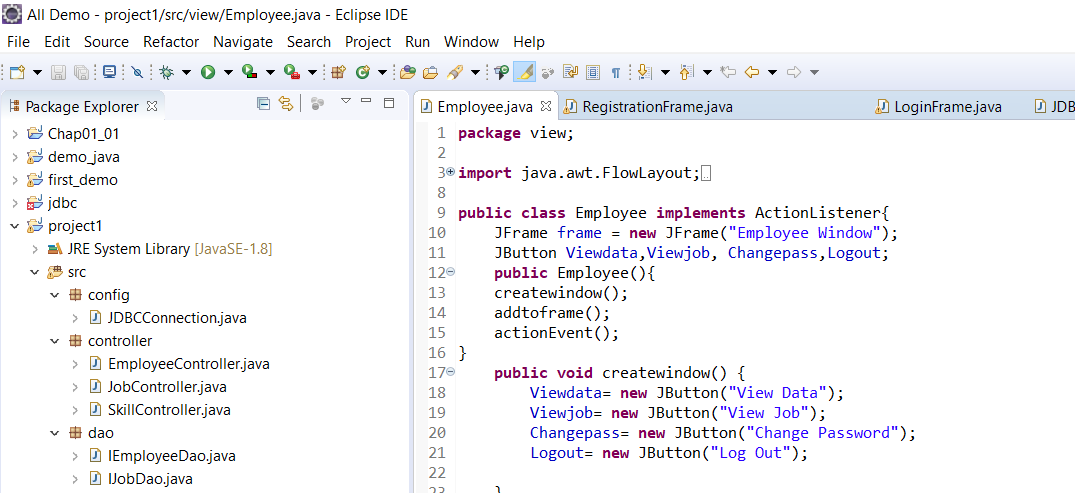
genderCombo.setSelectedItem("");

mobileno.setText("");

}

}

}



**\\Emplyee.java**

package view;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class Employee implements ActionListener{

JFrame frame = new JFrame("Employee Window");

JButton Viewdata,Viewjob, Changepass,Logout;

public Employee(){

createwindow();

addtoframe();

actionEvent();

}

public void createwindow() {

Viewdata= new JButton("View Data");

Viewjob= new JButton("View Job");

Changepass= new JButton("Change Password");

Logout= new JButton("Log Out");

}

public void addtoframe() {

frame.add(Viewdata);

frame.add(Viewjob);

frame.add(Changepass);

frame.add(Logout);

frame.setLayout(new FlowLayout());

frame.setSize(300,300);

frame.setVisible(true);

}

public void actionEvent()

{

Viewdata.addActionListener(this);

Viewjob.addActionListener(this);

Changepass.addActionListener(this);

Logout.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==Viewdata)

{

}

if(e.getSource()==Viewjob)

{

}

if(e.getSource()==Changepass)

{

}

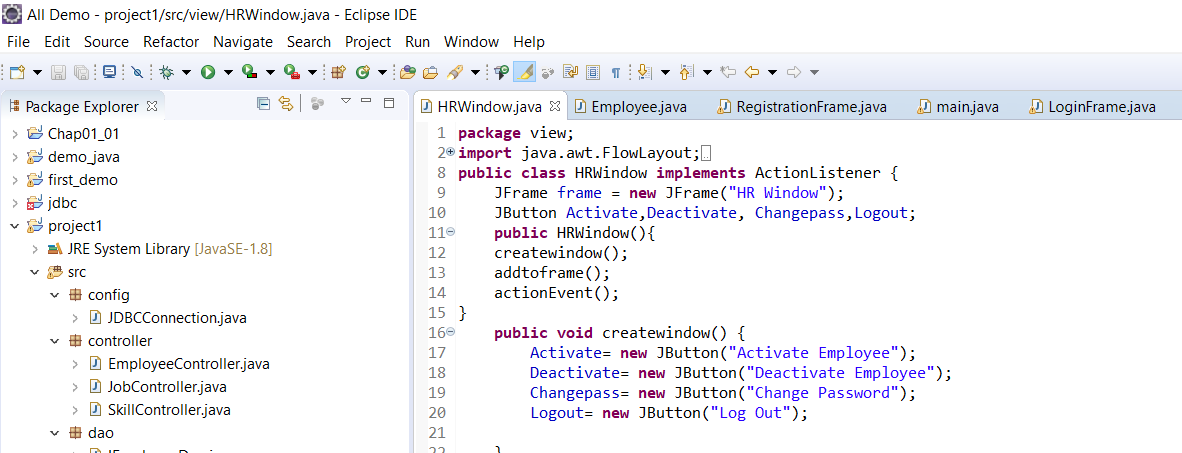
if(e.getSource()==Logout)

{

}

}

}



**\\HRWindow.java**

package view;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

public class HRWindow implements ActionListener {

JFrame frame = new JFrame("HR Window");

JButton Activate,Deactivate, Changepass,Logout;

public HRWindow(){

createwindow();

addtoframe();

actionEvent();

}

public void createwindow() {

Activate= new JButton("Activate Employee");

Deactivate= new JButton("Deactivate Employee");

Changepass= new JButton("Change Password");

Logout= new JButton("Log Out");

}

public void addtoframe() {

frame.add(Activate);

frame.add(Deactivate);

frame.add(Changepass);

frame.add(Logout);

frame.setLayout(new FlowLayout());

frame.setSize(300,300);

frame.setVisible(true);

}

public void actionEvent()

{

Activate.addActionListener(this);

Deactivate.addActionListener(this);

Changepass.addActionListener(this);

Logout.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==Activate)

{

}

if(e.getSource()==Deactivate)

{

}

if(e.getSource()==Changepass)

{

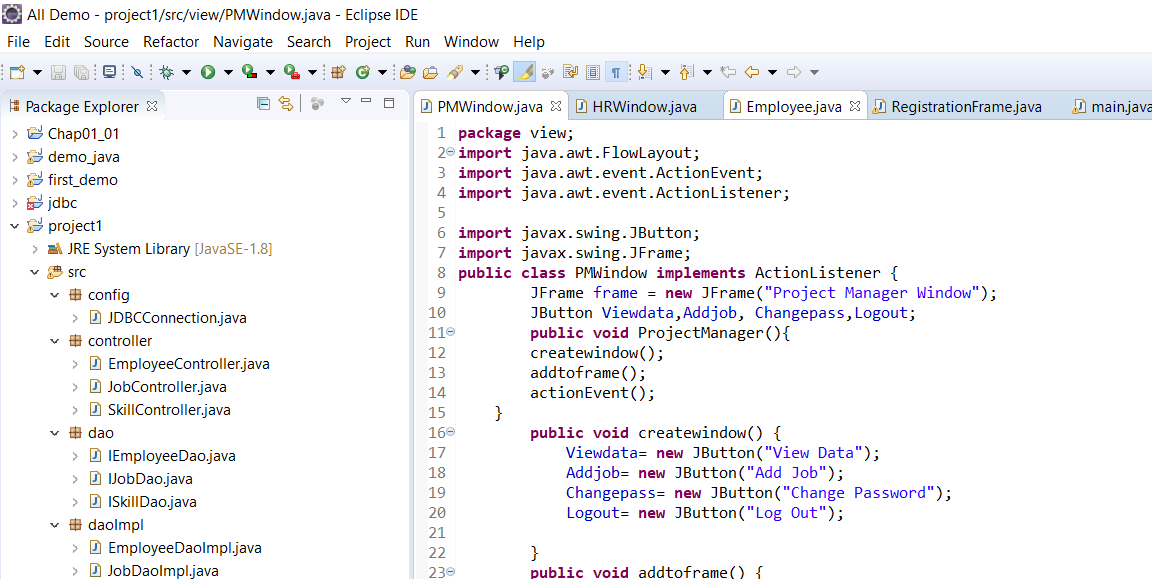
}

if(e.getSource()==Logout)

{

}

}



\\PMWindow.java (project manager):--

**package** view;

**import** java.awt.FlowLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**public** **class** PMWindow **implements** ActionListener {

JFrame frame = **new** JFrame("Project Manager Window");

JButton Viewdata,Addjob, Changepass,Logout;

**public** **void** ProjectManager(){

createwindow();

addtoframe();

actionEvent();

}

**public** **void** createwindow() {

Viewdata= **new** JButton("View Data");

Addjob= **new** JButton("Add Job");

Changepass= **new** JButton("Change Password");

Logout= **new** JButton("Log Out");

}

**public** **void** addtoframe() {

frame.add(Viewdata);

frame.add(Addjob);

frame.add(Changepass);

frame.add(Logout);

frame.setLayout(**new** FlowLayout());

frame.setSize(300,300);

frame.setVisible(**true**);

}

**public** **void** actionEvent()

{

Viewdata.addActionListener(**this**);

Addjob.addActionListener(**this**);

Changepass.addActionListener(**this**);

Logout.addActionListener(**this**);

}

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getSource()==Viewdata)

{

}

**if**(e.getSource()==Addjob)

{

}

**if**(e.getSource()==Changepass)

{

}

**if**(e.getSource()==Logout)

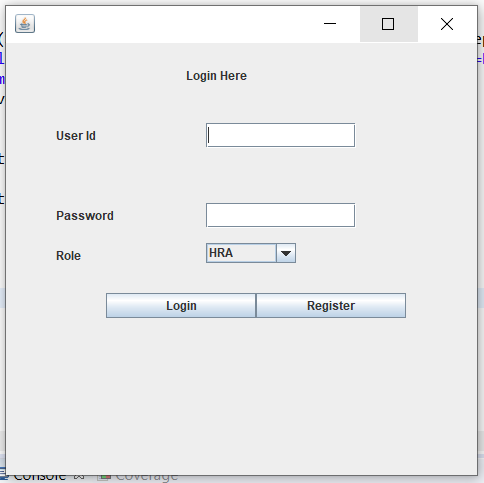
{

}

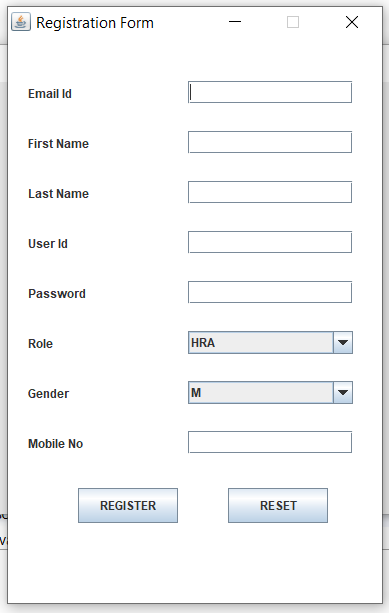
}

}

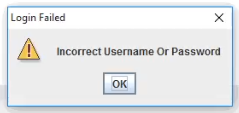
**LOGIN PAGE**



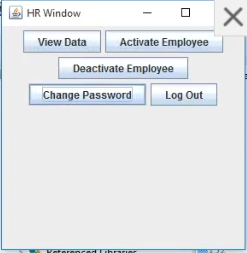
REGISTRATION PAGE



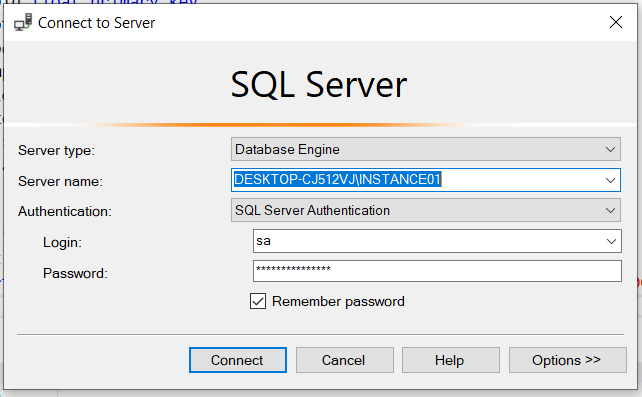
LOGIN failed page



**HR window**



**Microsoft SQL Server Management Studio INPUT & OUTPUT**



**Requirement analysis:--**

**Create database named PCSDB**

**Create following 5 tables:**

**-- Employee(empId[identity(1,1), primary key], firstName, lastName, userId, password, gender, role, active)**

**-- Skill(skillId[identity(1,1), primary key], skillName, skillDesc, active)**

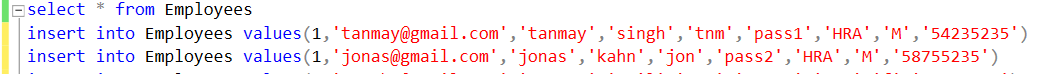
**-- Job(jobId[identity(1,1), primary key], jobTitle, jobDescription, companyName, Location, keySkill,salary,active)**

**-- EMPSkill(ESId[identity(1,1), primary key], empId, skillId, expYear)**

**-- EmpJob(EJId[identity(1,1), primary key],empId,jobId,recruited)**

create database PSCDB

use PCSDB



create table Employees(

empid int primary key identity(1,1),

Emailid varchar(30),

Firstname varchar(30),

Lastname varchar(30),

Userid varchar(30),

Pass varchar(30),

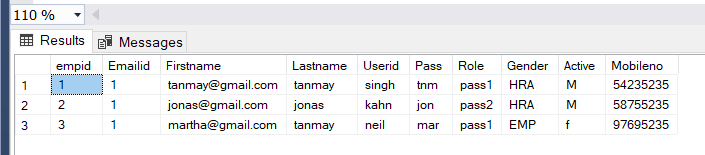
Role varchar(30),

Gender varchar(30),

Active varchar(30),

Mobileno varchar(30));

select \*from Employees



CREATE TABLE job (

jobid float primary key,

jobtitle varchar(30),

jobdesc varchar(30),

companyName varchar(30),

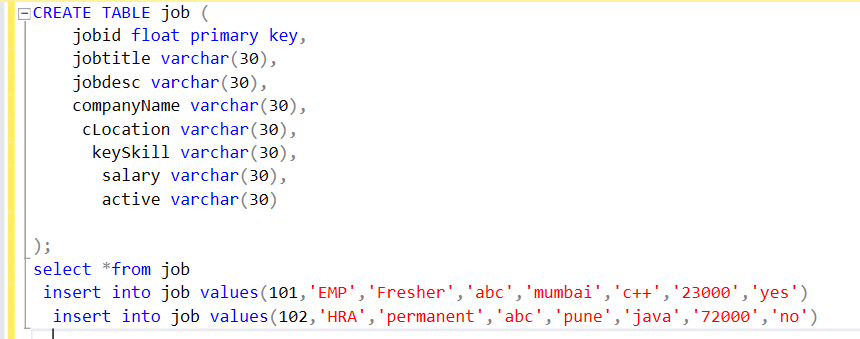
cLocation varchar(30),

keySkill varchar(30),

salary varchar(30),

active varchar(30)

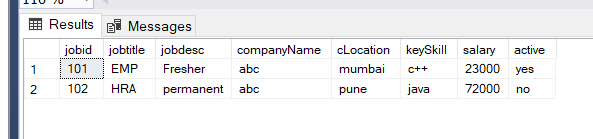
);



select \*from job

insert into job values(101,'EMP','Fresher','abc','mumbai','c++','23000','yes')

insert into job values(102,'HRA','permanent','abc','pune','java','72000','no')



CREATE TABLE skill (

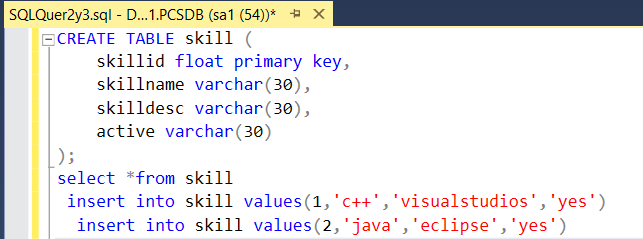
skillid float primary key,

skillname varchar(30),

skilldesc varchar(30),

active varchar(30)

);

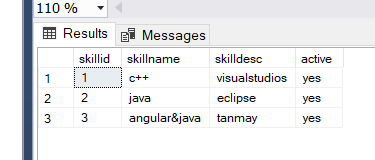


select \*from skill

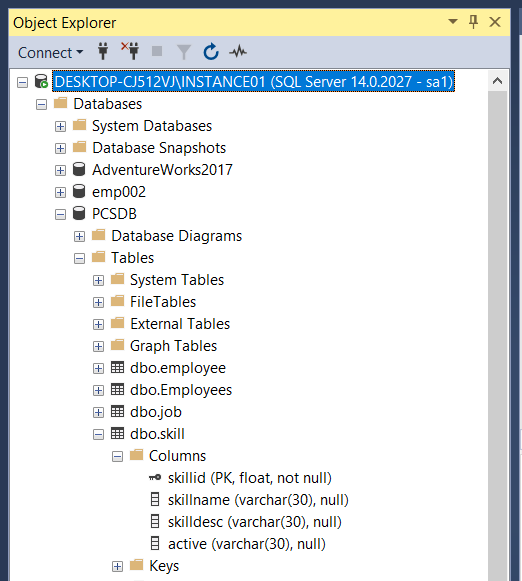
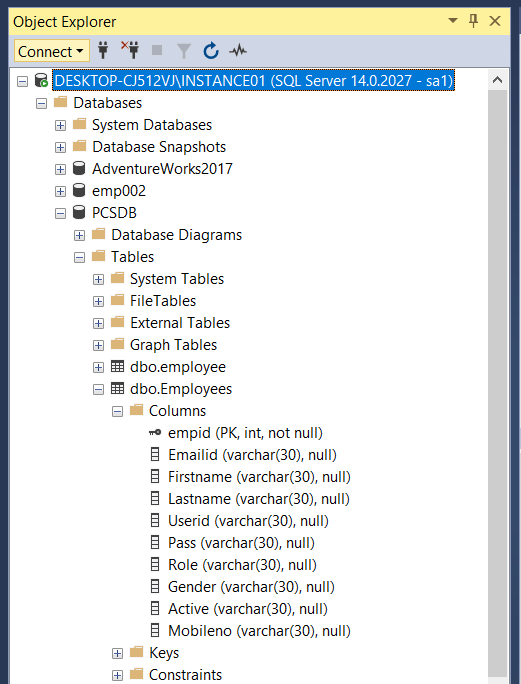
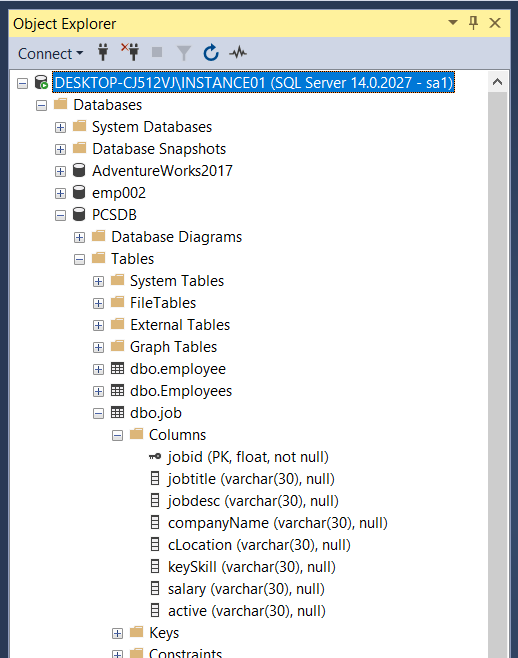
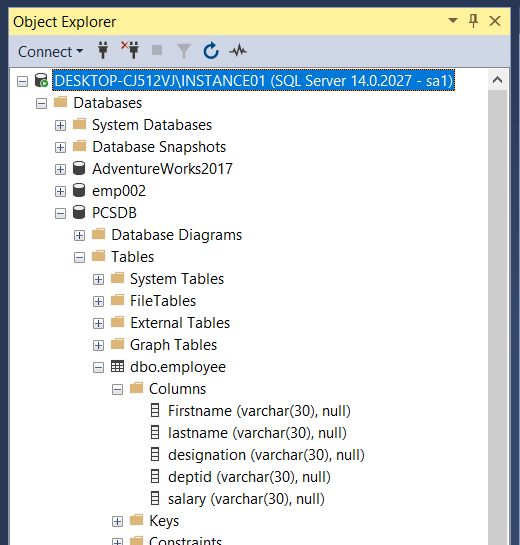
insert into skill values(1,'c++','visualstudios','yes')

insert into skill values(2,'java','eclipse','yes')

insert into skill values(3,'angular&java','tanmay','yes')



**OBJECT EXPLORER:--**

**  **

**References and Bibliography:**

**• Google for problem solving**

**•** [**http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html**](http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html)

**• Database Programming with JDBC and Java by O'Reilly**

**• Head First Java 2nd Edition**

**•** [**http://www.jdbc-tutorial.com/**](http://www.jdbc-tutorial.com/)

**• Java and Software Design Concepts by Apress**

**•https://www.tutorialspoint.com/java/ • http://www.javatpoint.com/java-tutorial**

**•** [**https://docs.oracle.com/javase/tutorial/**](https://docs.oracle.com/javase/tutorial/)

**• http://www.wampserver.com/en/ • http://www.JSP.net/ •** [**http://www.tutorialspoint.com/mysql/**](http://www.tutorialspoint.com/mysql/)

**• httpd.apache.org/docs/2.0/misc/tutorials.html**

**THANK YOU!!!**

--TANMAY SINGH

-- Student ID: S211106800003

#### -- Registration No: R211106800003