JOB POSTING-RECRUITMENT PORTAL

DECEMBER 12

Developed by: Kaniyan Pandiarasan K

Reg No: R200002100292

Batch Code: 210167

Name of the Cordinator: Ms. Lopamudra Bera

Date of Submission: 12/12/2020

NIIT

CERTIFICATE

This is to certify that this report, titled Job Posting – Recruitment Portal embodies the original work done by Kaniyan Pandiarasan K, in partial fulfillment of his course requirement at NIIT.

Coordinator: Ms. Lopamudra Bera

ACKNOWLEDGEMENT

It gives me great pleasure in expressing my gratitude to all those people who have supported us and had their contribution in making this project work possible. First and foremost, I must acknowledge and thank the Almighty for blessing, protecting and guiding me throughout this period.

I express my profound sense of reverence to my Coordinator Ms. Lopamudra Bera, for her constant guidance, support, motivation and untiring help during the course of my project work. She has given us enough freedom during our project work, and she has always been nice to us. I am thankful to the Almighty for giving us a teacher like her.

ABSTRACT

This Project will mainly use for professional networking, allowing registered employees to post jobs and their information is organized and easily accessible to all registered users such as job seekers, employers and HR consultant.

Users will register themselves in the application and fill in their professional details along with their core competencies. And Employers can be able to search relevant profiles for their openings using the keywords that matches with the profiles. Based on the match, relevant profiles should be shown to the employers with their contact details and International Business Unit name.

CONFIGURATION

- ***Hardware***
    - ***Core i3 Processor 2.10 GHz x64 bit***
    - ***8 Gigabytes of RAM***

- ***Operating System***
    - ***Windows 10 Home***

- ***Software***
  - ***Eclipse IDE***
    - ***JAVA***
    - ***MYSQL***

TABLE OF CONTENTS

OBJECTIVE

This Project will mainly use for professional networking, allowing registered employees to post jobs and their information is organized and easily accessible to all registered users such as job seekers, employers and HR consultant.

Users will register themselves in the application and fill in their professional details along with their core competencies. And Employers can be able to search relevant profiles for their openings using the keywords that matches with the profiles. Based on the match, relevant profiles should be shown to the employers with their contact details and International Business Unit name.

CASE STUDY

Professionet Consultancy Services (PCS) is a business consultancy that has established itself as a renowned service provider of a wide range of business services to its clients.

PCS offers an offline platform for their employees to share their profiles to initiate internal job posting process with the expert Human Resources consultants.

There are over 22,000 new and 50,000 experienced PCS professionals providing their services to 150 clients aligned with the consultancy. The consultancy needs to maintain the information of every PCS employee focusing on their industry vertical. All PCS employees are registered with PCS and are given a unique identification number. Profile validation is done by the HR experts and requirements are full filled by mapping skills and requirements manually.

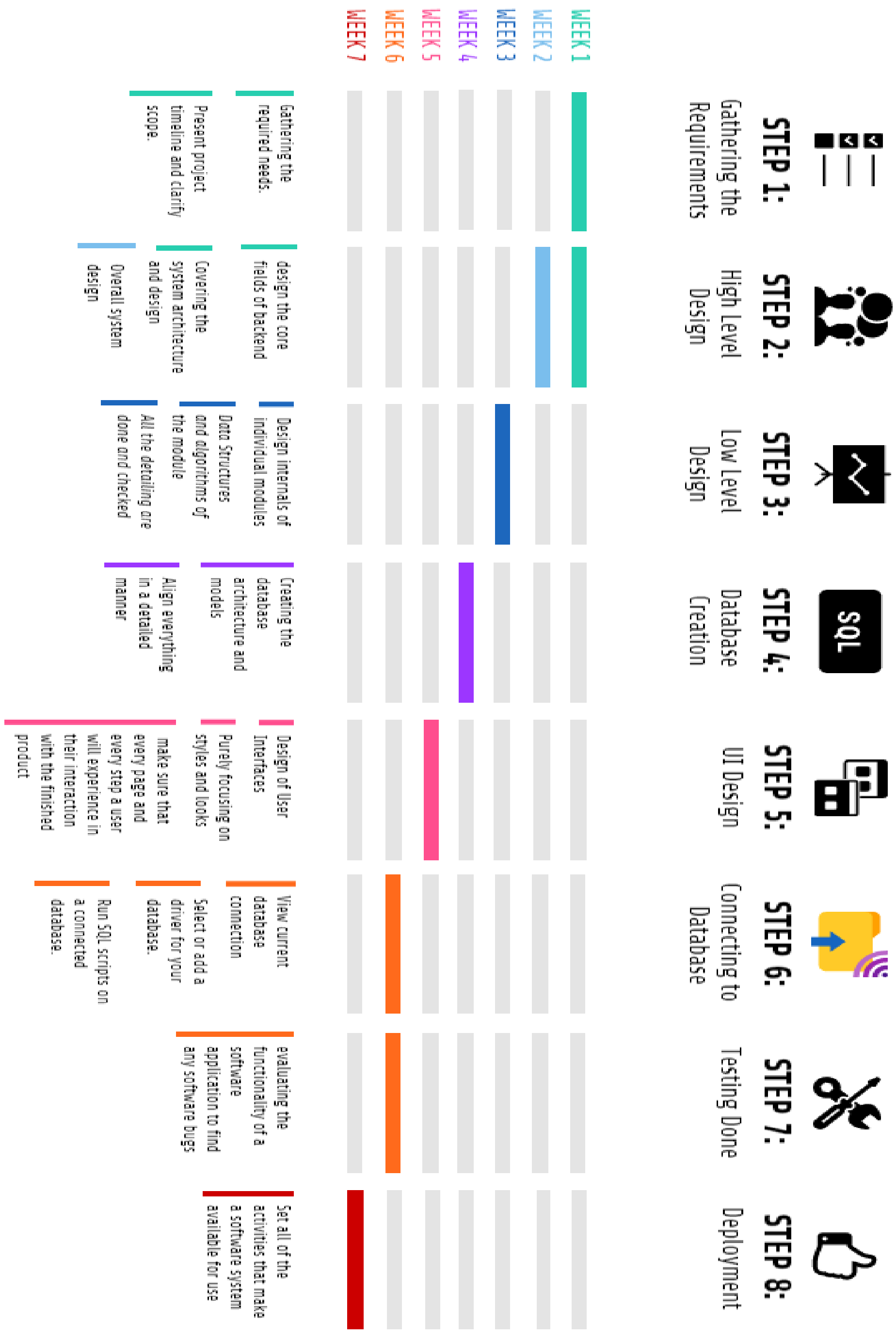
The Consultancy wants to introduce automation in their Internal Job Posting (IJP) selection and recruitment process so that the potential PCS employees and HR department have an online platform based on a skill map engine to connect directly with each other and aid their job search within the company. HR and PM can also post their jobs and refine their search by using keywords that matches the profiles.

STATEMENT OF REQUIREMENTS

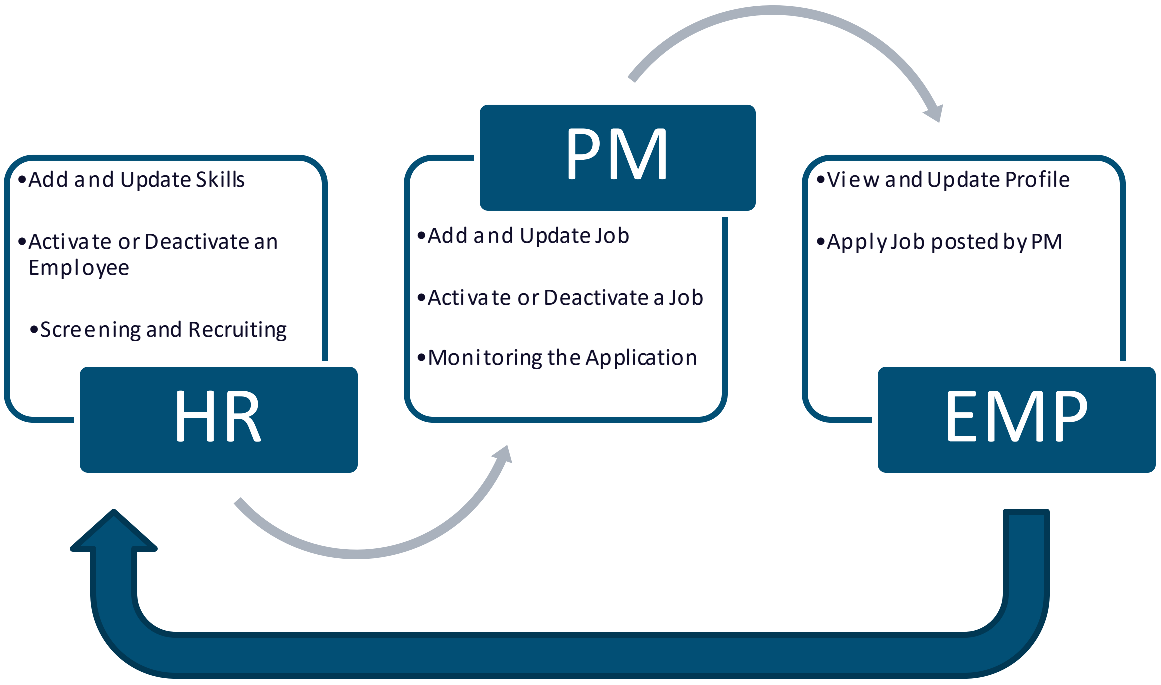
Key Features of the application:

- ✓ **Skill mapping** feature is the most prominent feature of this software that will reduce time and effort spent on profile screening and mapping the employees' profiles with the current business requirement.
- ✓ Also this software holds the other features such as having a separate window for Employees, Project Managers and HR's to access their portal.
- ✓ It has the registration window to register with the company by themselves.
- ✓ Application has the same login window for everyone and it detects by itself which designation you are in and give access accordingly.
- ✓ HR can able to activate and deactivate an employee's profile and recruit an employee to their organization.
- ✓ HR can able to view all the employee's status and can add, delete, activate or deactivate a skill in the portal.
- ✓ Project Manager can add a job on the portal and deactivate or activate a job while it is needed.
- ✓ PM can also monitor the employees who has applied for the posted job on the portal and also sort out the jobs using particular skill name.
- ✓ Employee will have access to view and update their profile anytime and can see what the available jobs on their required verticals are.
- ✓ They can apply for the jobs by mentioning the job id that will immediately reflect on the HR's portal.

PROJECT PLAN



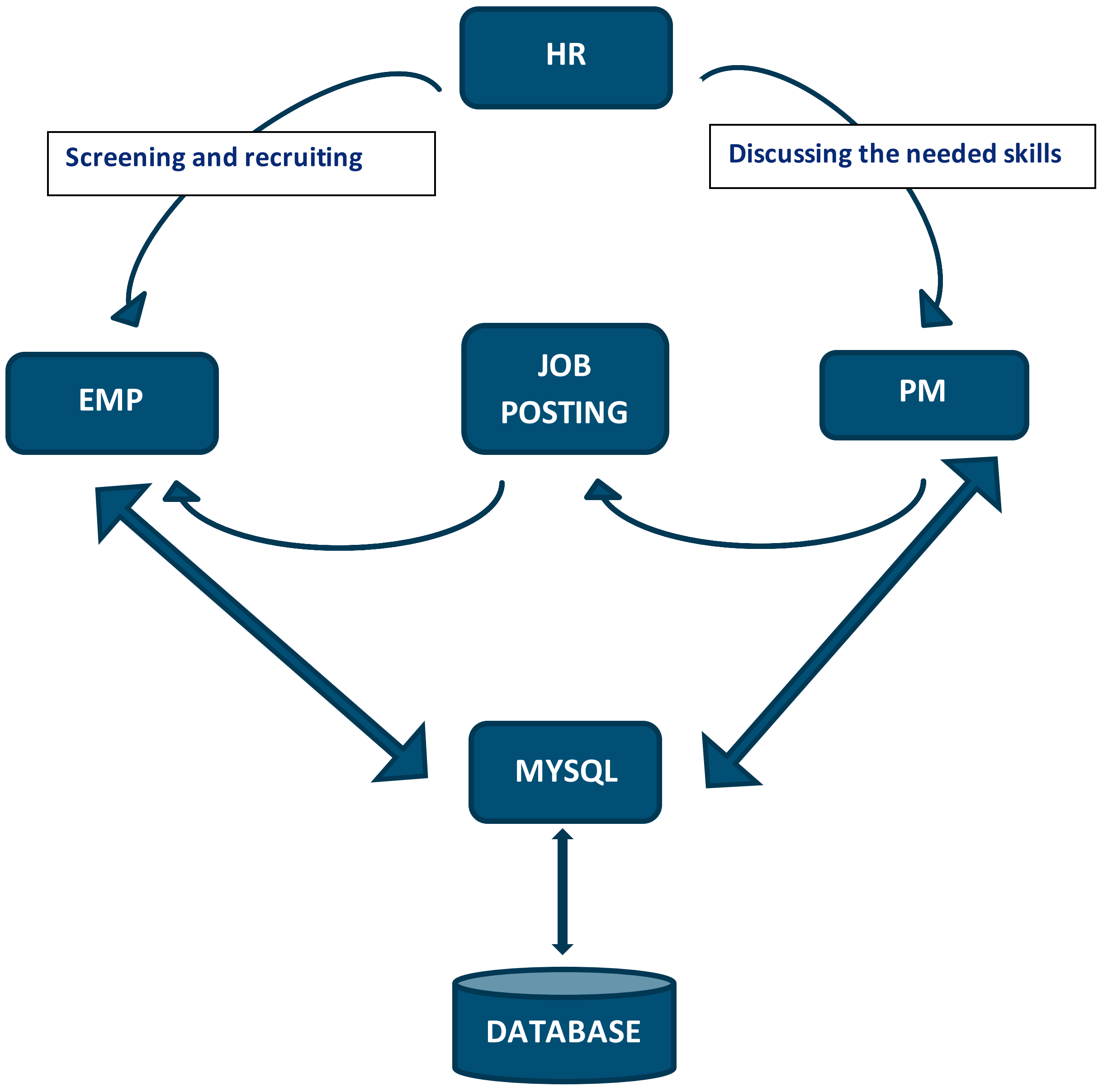
BUSINESS MODEL



•Add and Update Skills
•Activate or Deactivate an Employee
•Screening and Recruiting
HR
PM
•Add and Update Job
•Activate or Deactivate a Job
•Monitoring the Application
•View and Update Profile
•Apply Job posted by PM
EMP

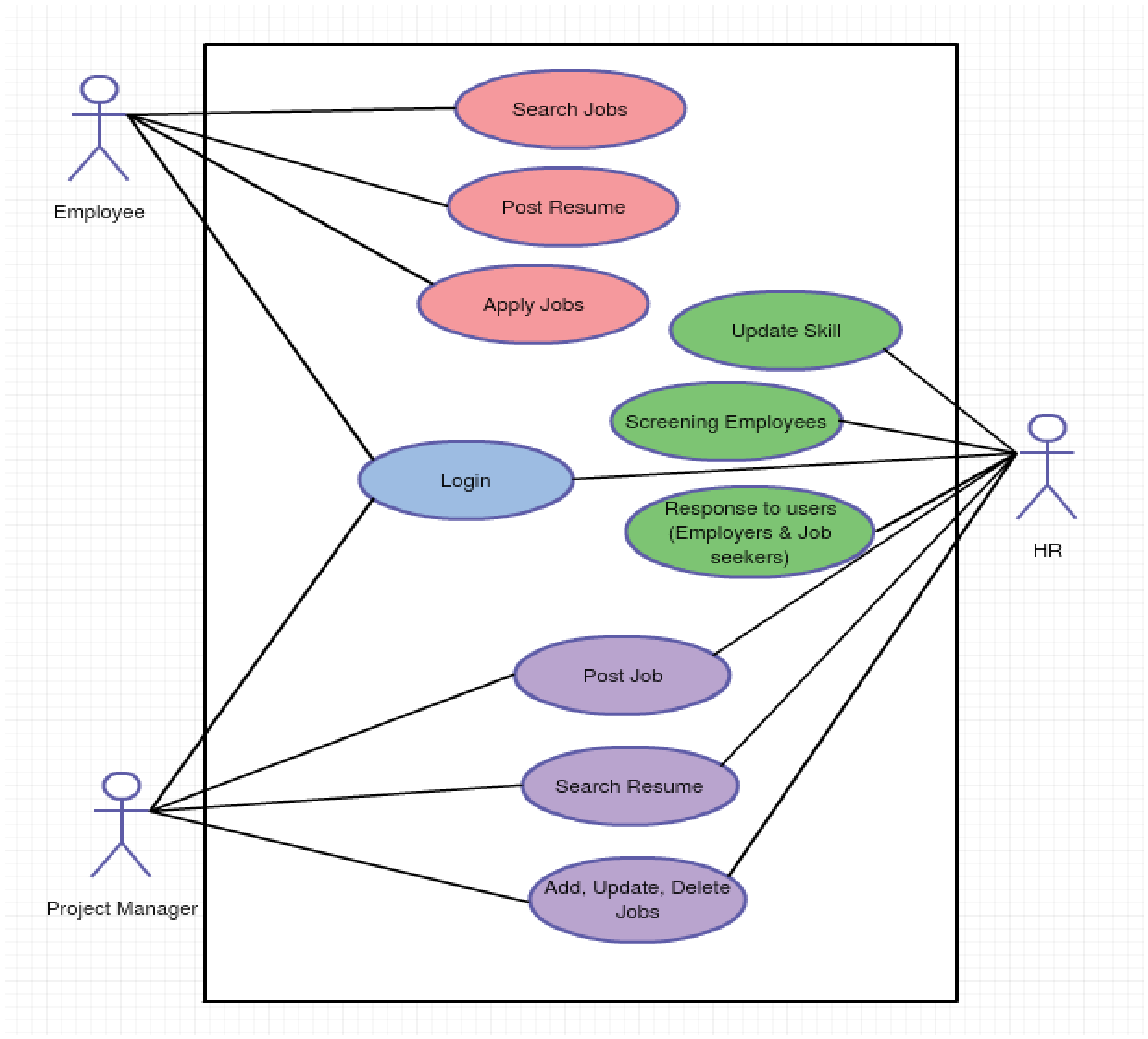
In the above mentioned model is that the overall functionality model of the application that we have already discussed in the previous chapter.

HIGH LEVEL CASE DIAGRAM



A High-level design document adds the necessary details to the current project description to represent a suitable model for coding.

LOW LEVEL CASE DIAGRAM



Here is the Low level case diagram which represents everything a user can do. An Employee can search for a job, update their profile and apply for a job.

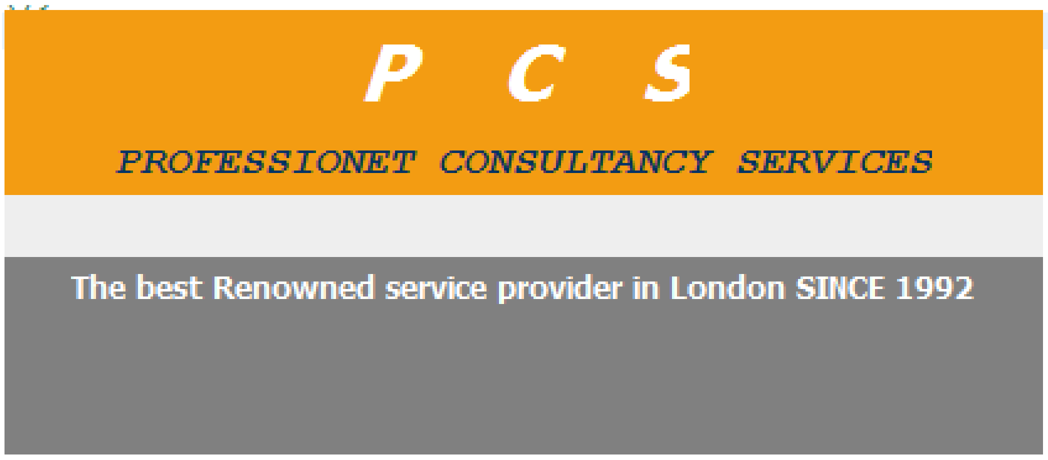
Project Manager can post the job necessary for the organization and monitor the cycle. He can able to activate and deactivate a job according to the situation of the company.

Similarly, HR can monitor the profiles who are applied for the particular jobs and screening them by matching the skill sets to pick a suitable employee to fit the project.

USER INTERFACE DESIGN

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

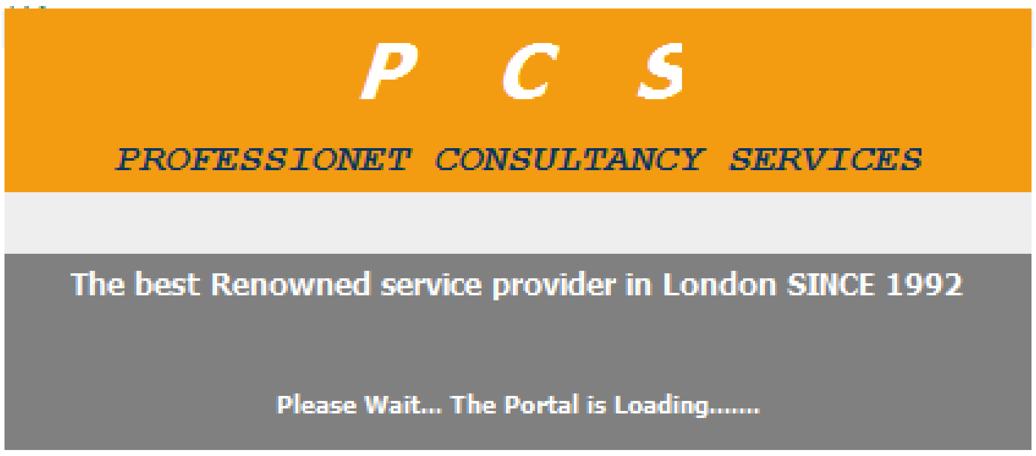
While stating up the software it displays a frameless window until the database is loading to the software.



P C S

PROFESSIONET CONSULTANCY SERVICES

The best Renowned service provider in London SINCE 1992

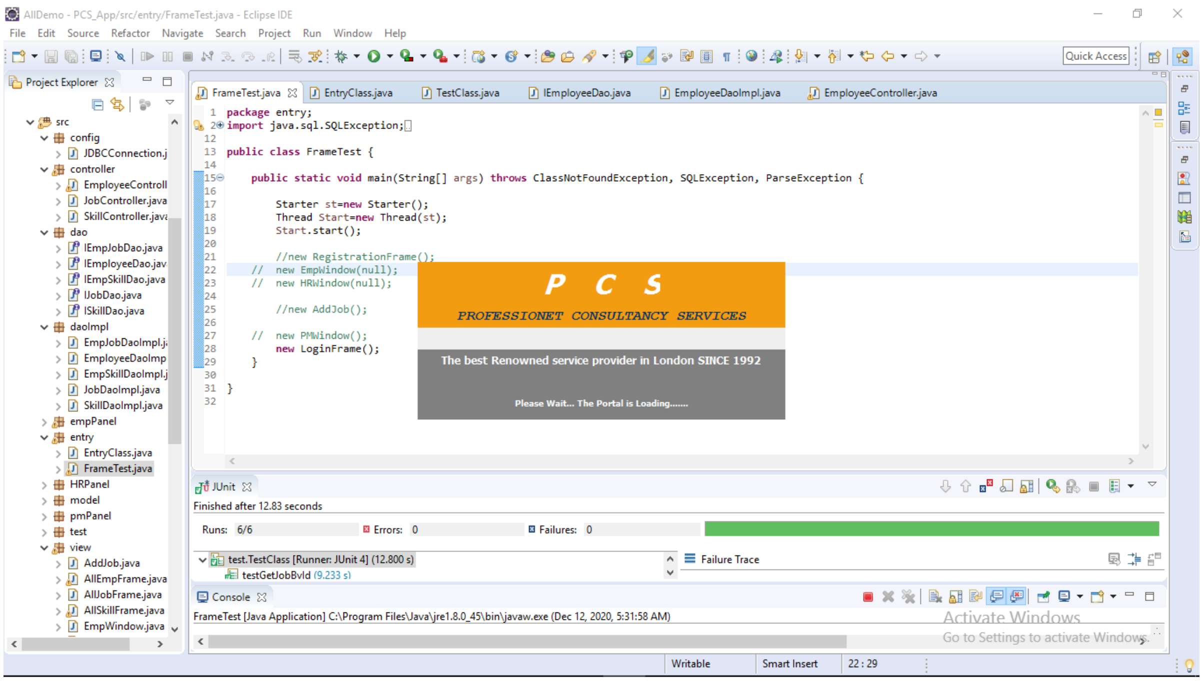


P C S

PROFESSIONET CONSULTANCY SERVICES

The best Renowned service provider in London SINCE 1992

Please Wait... The Portal is Loading.......



```java
package entry;
import java.sql.SQLException;

public class FrameTest {

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

		Starter st=new Starter();
		Thread Start=new Thread(st);
		Start.start();

		//new RegistrationFrame();
	//	new EmpWindow(null);
	//	new HRWindow(null);

		//new AddJob();

	//	new PMWindow();
		new LoginFrame();
	}

}
```

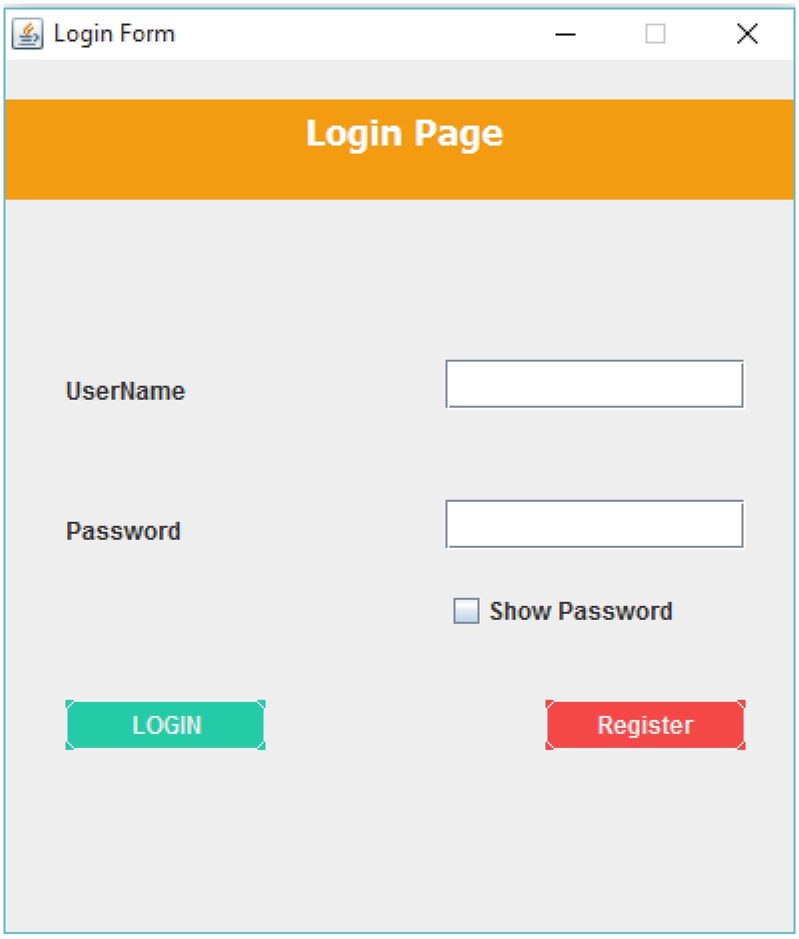
P C S
PROFESSIONET CONSULTANCY SERVICES
The best Renowned service provider in London SINCE 1992
Please Wait... The Portal is Loading.......

```
Finished after 12.83 seconds
Runs: 6/6    Errors: 0    Failures: 0
test.TestClass [Runner: JUnit 4] (12.800 s)
  testGetJobById (9.233 s)
```

```
FrameTest [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Dec 12, 2020, 5:31:58 AM)
```

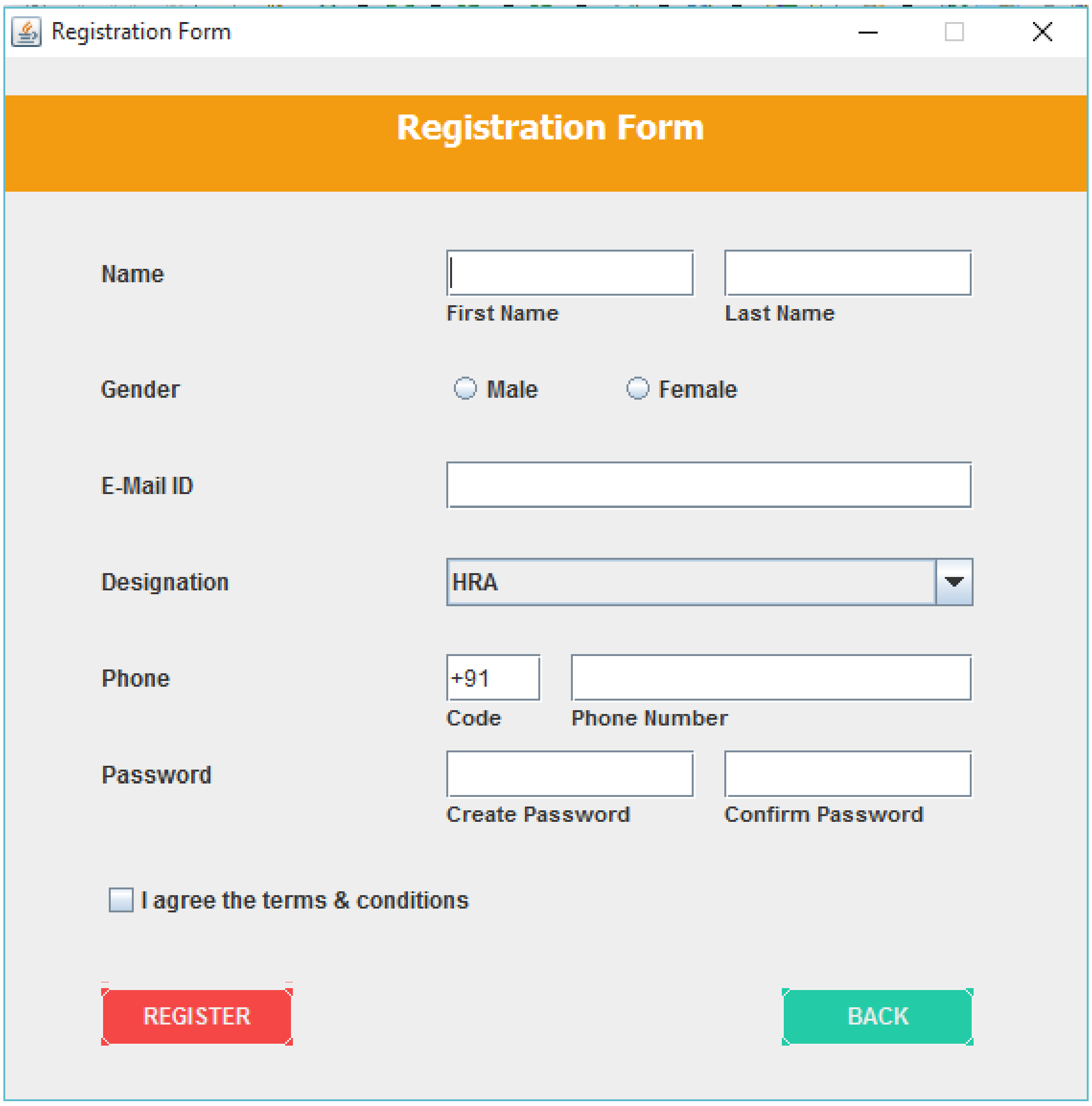
This is how the actual window looks like on the desktop. It will take 6 to 8 seconds to connect to the database. In that mean time, this frameless window will be displayed and once the database loaded, it will automatically close by itself and leads you to the login frame. That will be shown in the next image.

The common user interface for the entire system that includes an Employee, Project Manager and HR.



The user has to be registered as PCS Employee before login. This frame will connect to the database and checks the user's authentication and also decides which designation they belong to (EMP, PM, HR) and will give access to the user accordingly.

This is the Registration form that the user has to fill their details to get registered with the PCS Community.



Registration Form

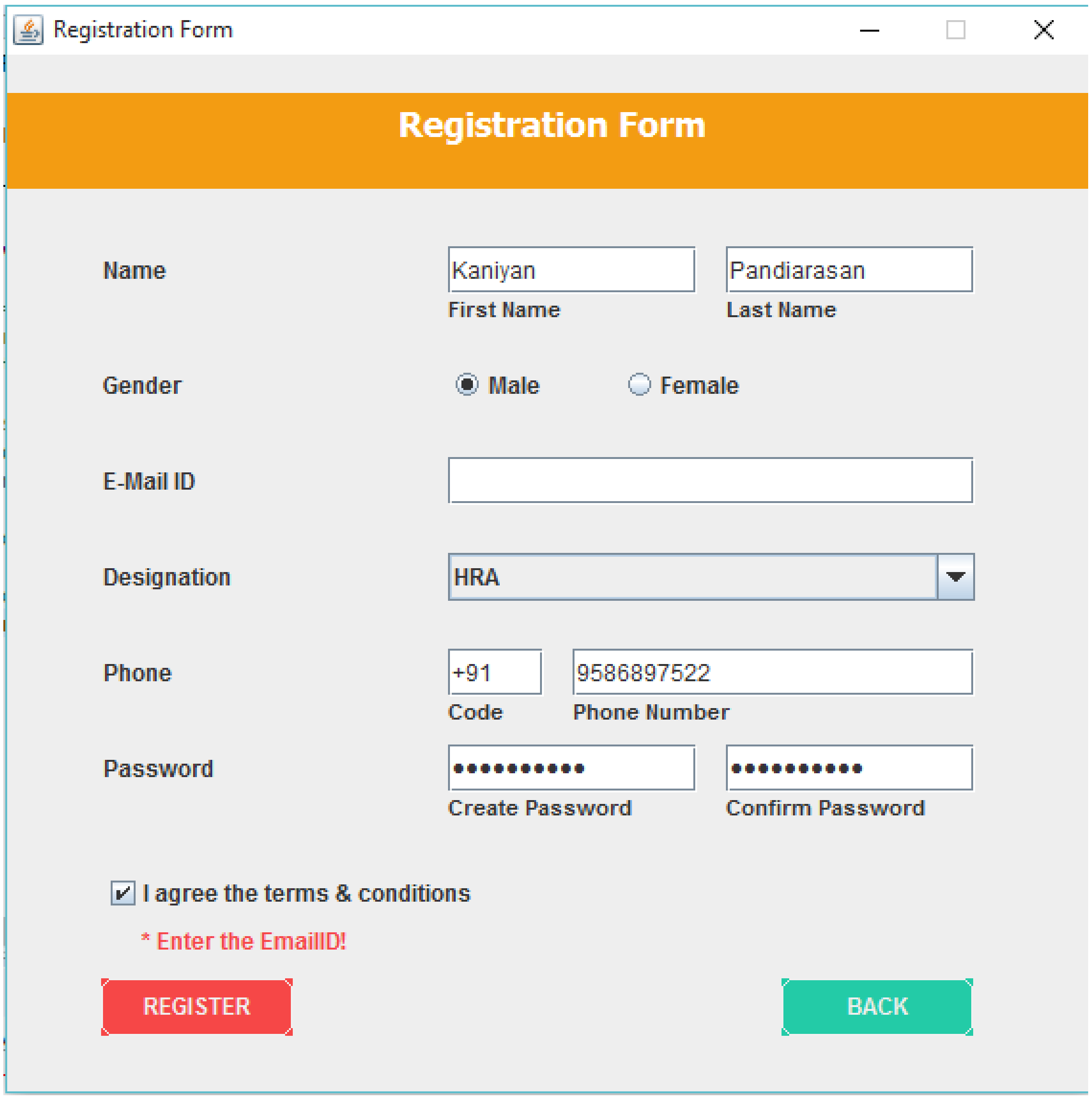
Registration Form

Name	First Name	Last Name
Gender	○ Male	○ Female
E-Mail ID		
Designation	HRA	
Phone	Code: +91	Phone Number
Password	Create Password	Confirm Password

☐ I agree the terms & conditions

The System must not miss out any kind of information while dealing with the registration form. So, the authentication is very much important for the organization. The registration form would check each and every column for null value and alert the user to fill that missing field and also it makes sure that the entered E-mail id, phone number, password is a valid one and meet the system requirements.

Here are some snaps with explanation.


Registration Form

Registration Form

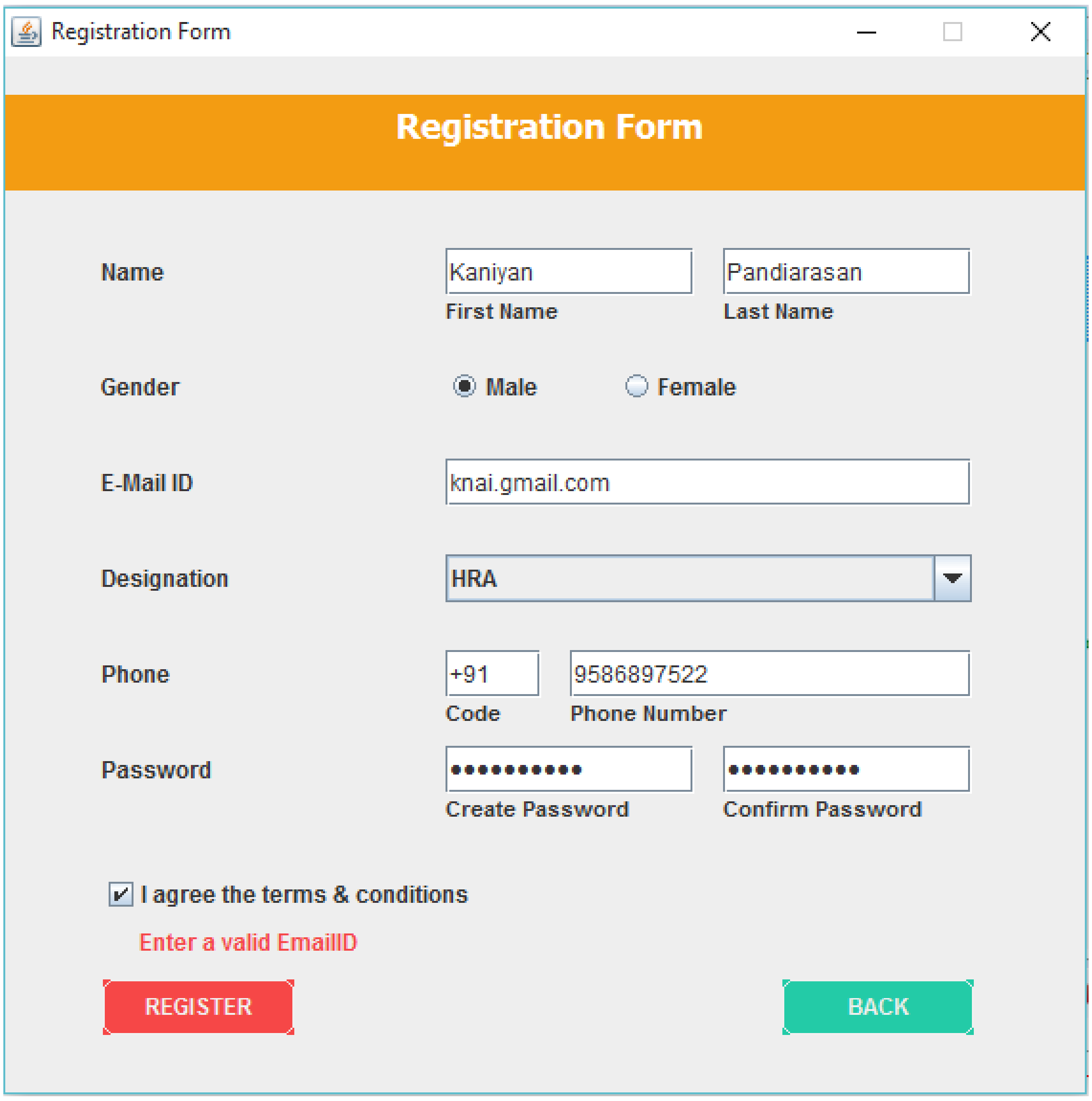
Name	Kaniyan First Name	Pandiarasan Last Name
Gender	[✓ Male]	Female
E-Mail ID		
Designation	HRA	
Phone	+91 Code	9586897522 Phone Number
Password	●●●●●●●●●●● Create Password	●●●●●●●●●●● Confirm Password

[✓ I agree the terms & conditions]

* Enter the EmailID!

In above form, E-mail id field is missed to fill. And if you click the register button, it will throw an indication stating that to fill the required field to complete registration.

If the user enters the invalid mail address then a message will be displayed stating that to enter the valid email ID. Following is the example:



Registration Form

Registration Form

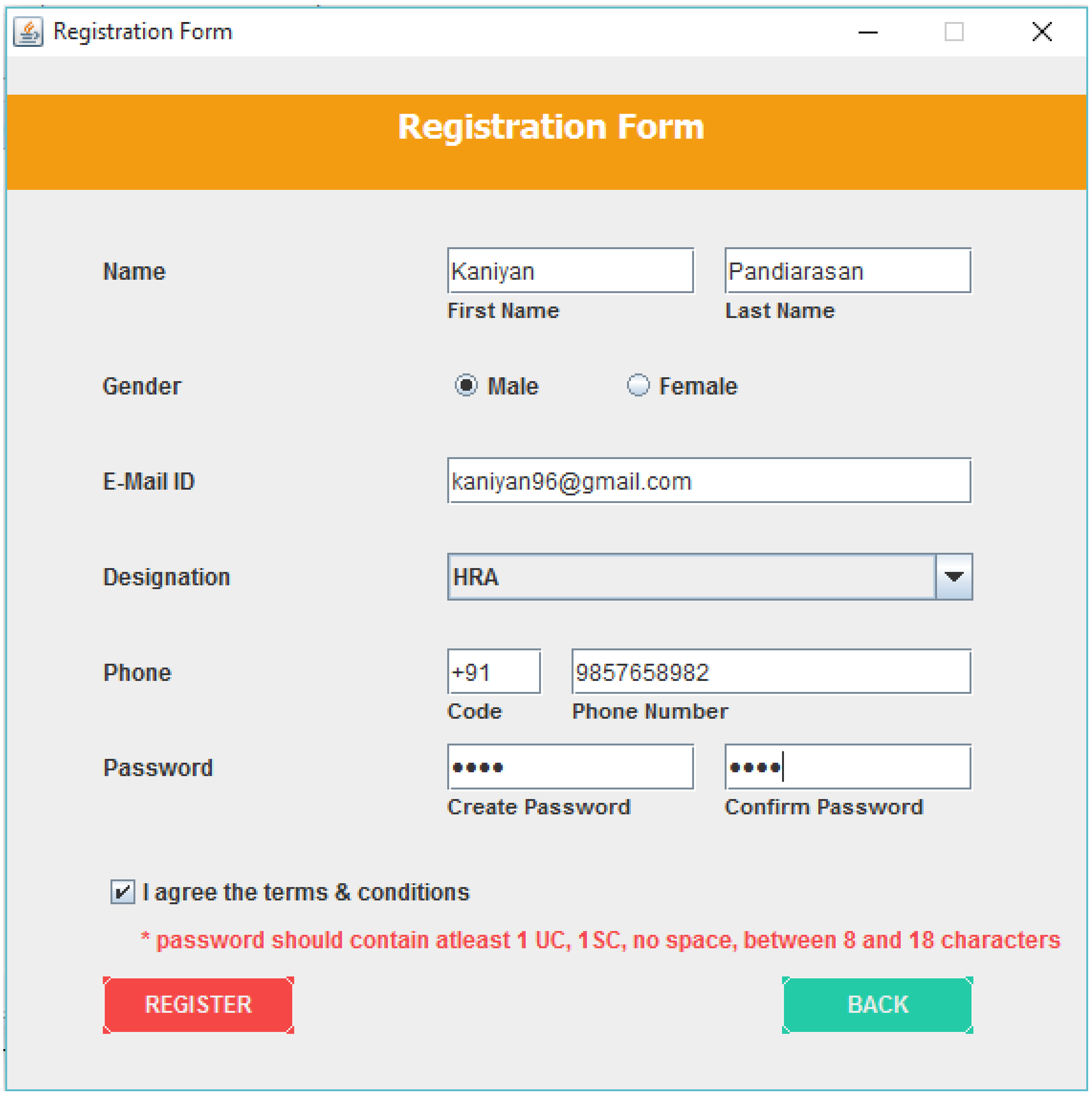
Name	Kaniyan First Name	Pandiarasan Last Name
Gender	[✓ Male]	Female
E-Mail ID	knai.gmail.com	
Designation	HRA	
Phone	+91 Code	9586897522 Phone Number
Password	●●●●●●●●●●● Create Password	●●●●●●●●●●● Confirm Password

[✓] I agree the terms & conditions

Enter a valid EmailID

The same will be checked for validating the phone number. The phone number column has to have 10 digits, doesn't accept the letters and should start with 7, 8, and 9.

Here comes the password validation. The password has to have 8-18 characters with no space and contains at least one uppercase, one lowercase, one special character and one number. If it entered wrong, a message will be displayed on the screen to enter a valid password.


Registration Form

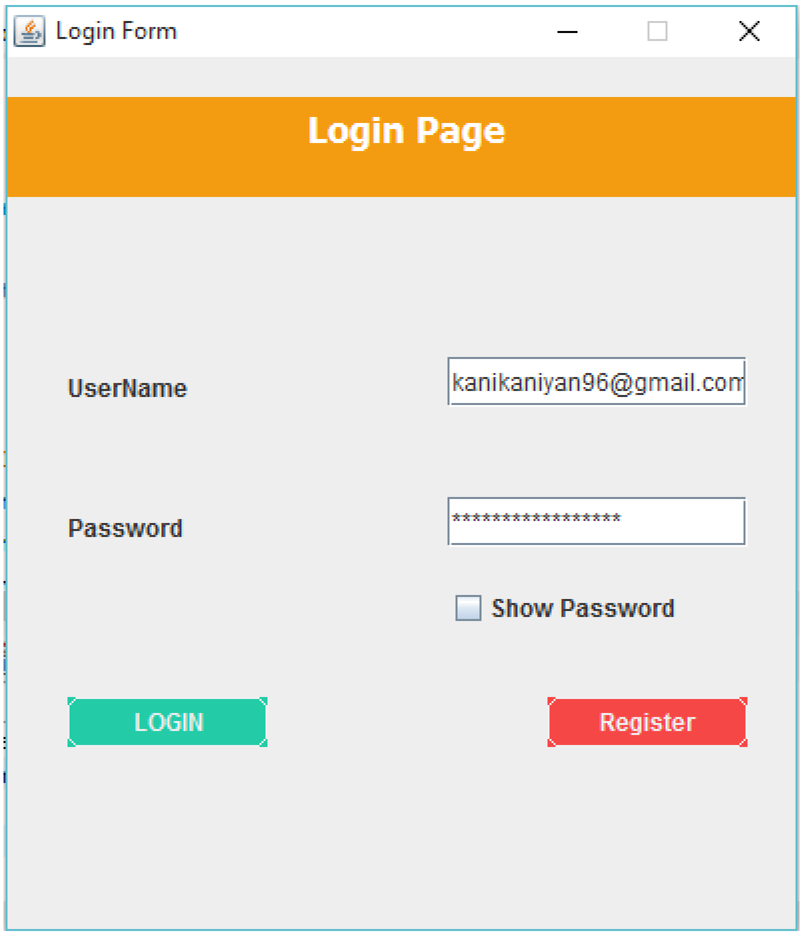
Registration Form

Name	Kaniyan First Name	Pandiarasan Last Name
Gender	[● Male]	○ Female
E-Mail ID	kaniyan96@gmail.com	
Designation	HRA	
Phone	+91 Code	9857658982 Phone Number
Password	•••• Create Password	•••• Confirm Password

[✓] I agree the terms & conditions

* password should contain atleast 1 UC, 1 SC, no space, between 8 and 18 characters

If the password matches with the required criteria, then it will cross check with the confirm password to make sure that the user is entering the valid known password. In case, password doesn't match with the confirm password, then a message would be displayed as password doesn't match.



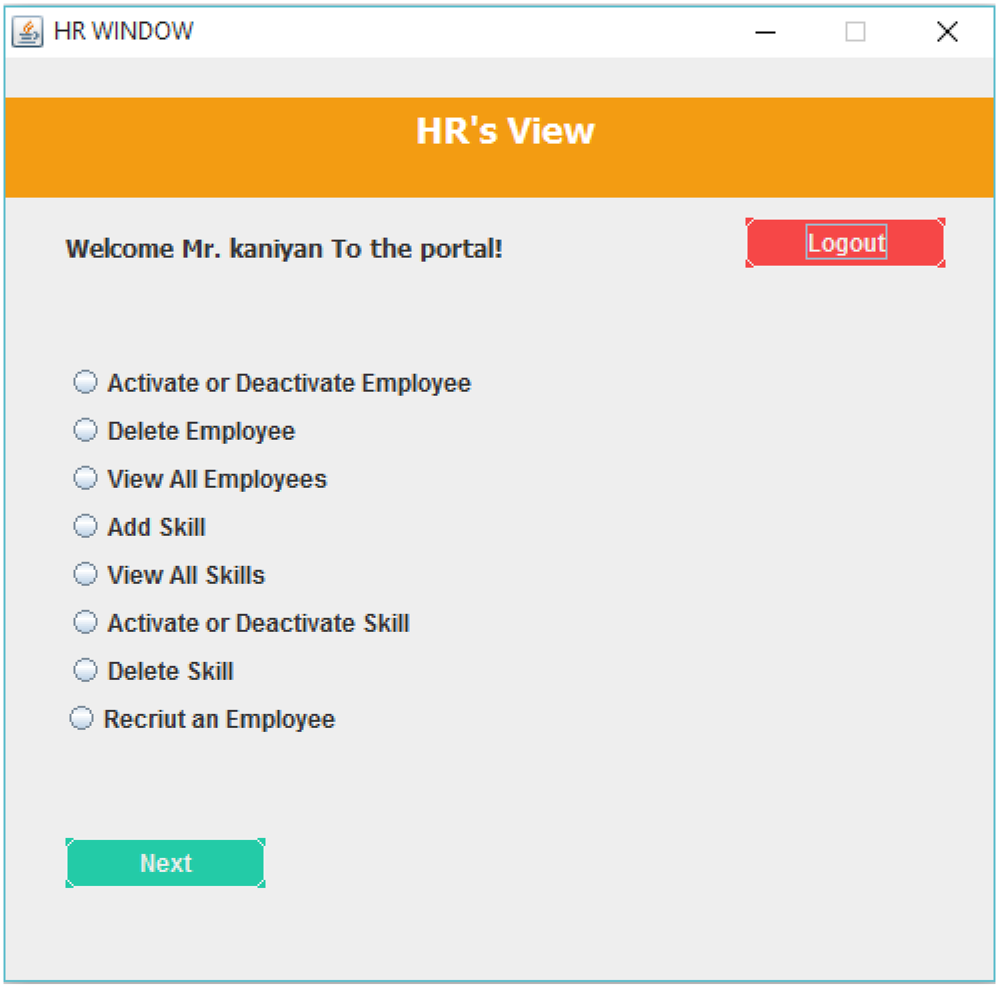
UserName kanikaniyan96@gmail.com

Password *****************

Show Password

This is how a user can login to their portal. This is a HR login credentials. This navigates us to HR portal. While registering with the portal, one needs to choose that the user is an employee / Project Manager / HR. It will walk us through which portal we are having the access to.

HR Window



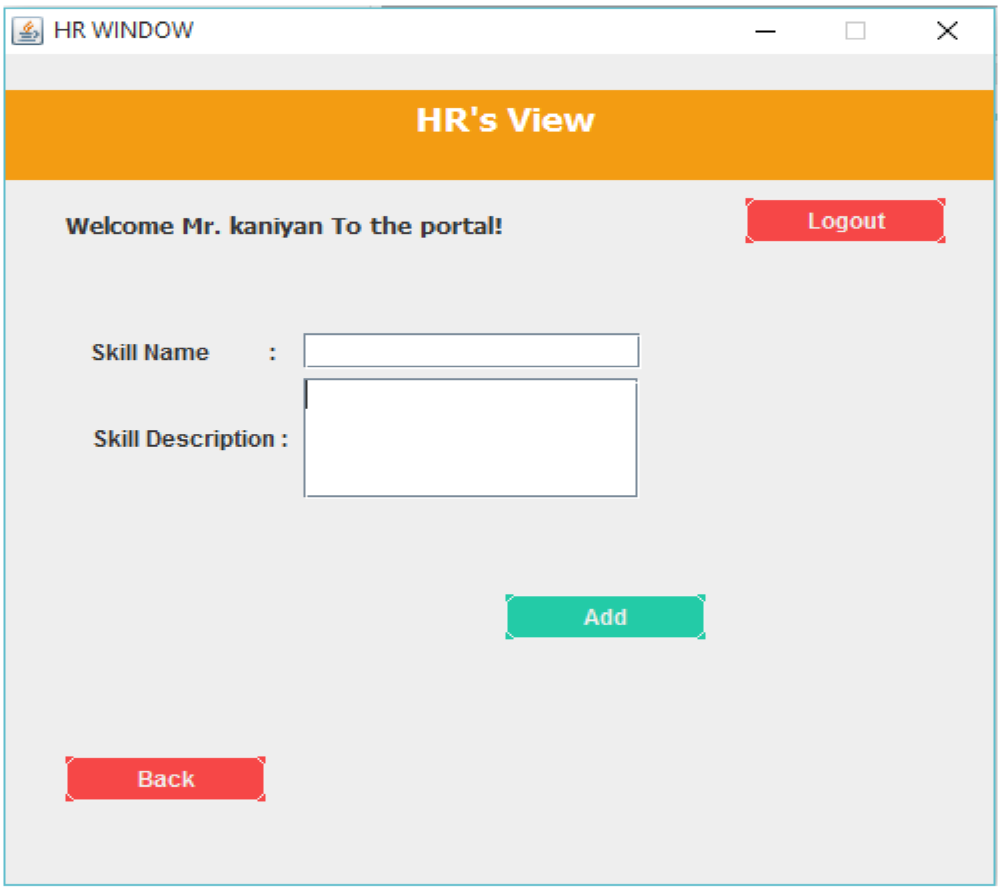
Welcome Mr. kaniyan To the portal!

- Activate or Deactivate Employee
- Delete Employee
- View All Employees
- Add Skill
- View All Skills
- Activate or Deactivate Skill
- Delete Skill
- Recriut an Employee

This is how the HR Window looks like. Nicely designed user interface for the HR’s View. Here, can see that what HR can have the access to do. HR can

- Activate or deactivate employee
- Delete an employee
- View all employees
- Add a necessary skill
- View all available skills in the database
- Activate, deactivate or delete a skill.

HR can able to add a skill.



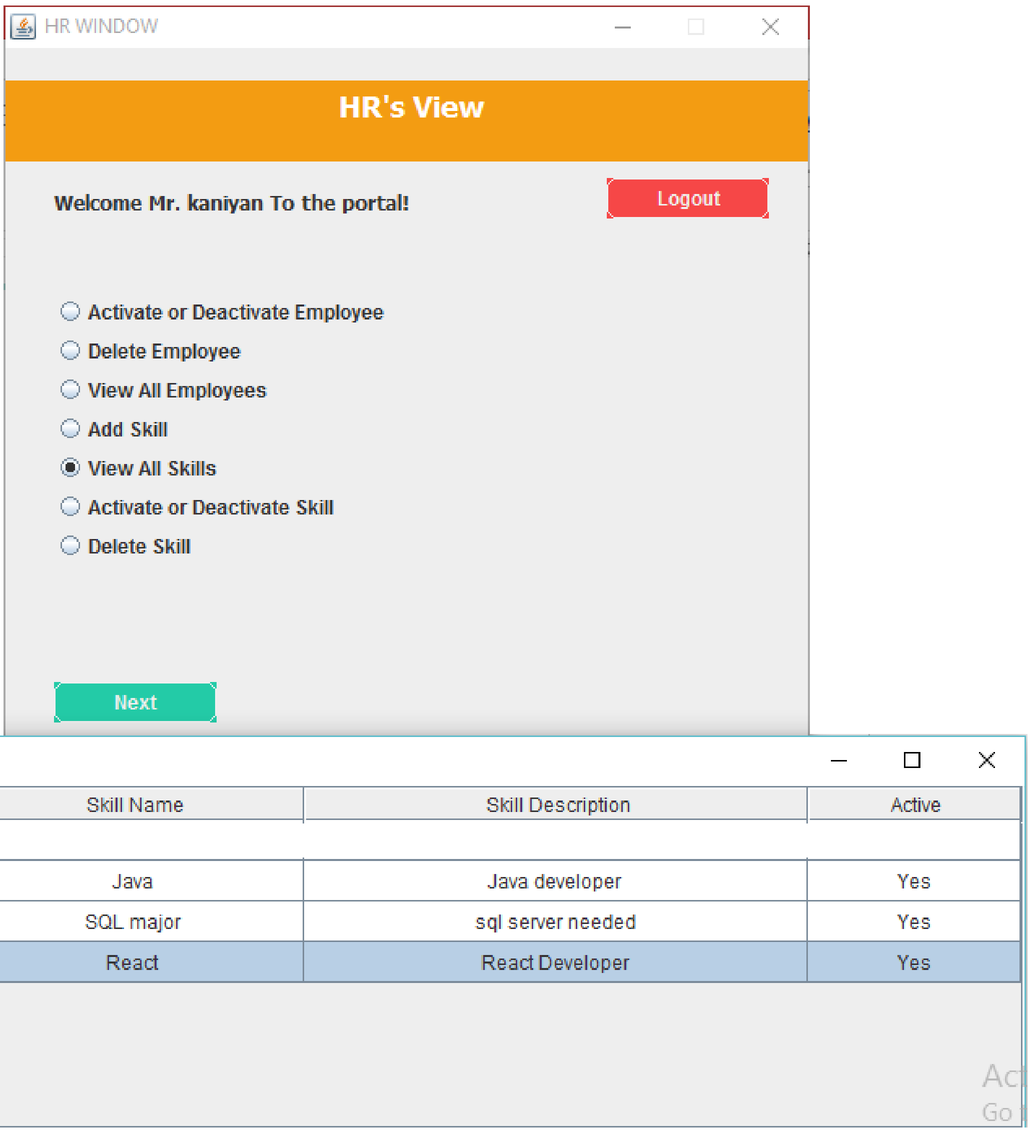
Welcome Mr. kaniyan To the portal!

Skill Name :

Skill Description :

HR can able to add the skill by entering the skill name and the skill description. It will be added to the database and immediately it will reflects on the application.

HR can see all the skills that includes deactivated skills to check the data.



HR WINDOW

HR's View

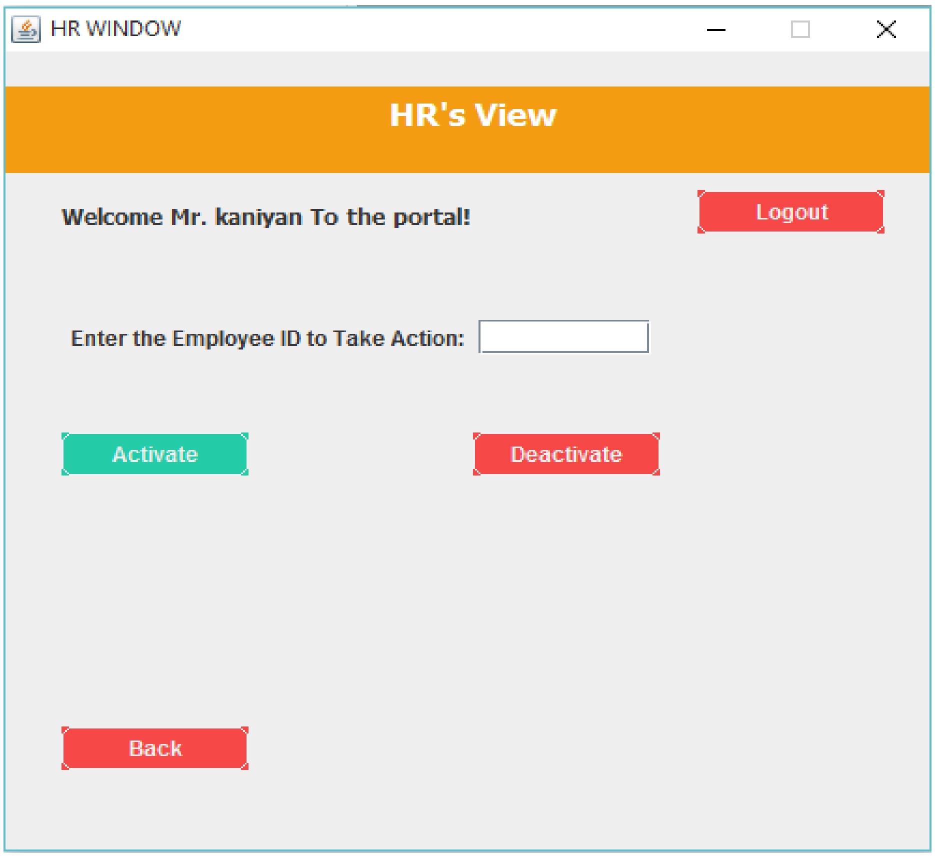
Welcome Mr. kaniyan To the portal!

- ◯ Activate or Deactivate Employee
- ◯ Delete Employee
- ◯ View All Employees
- ◯ Add Skill
- ● View All Skills
- ◯ Activate or Deactivate Skill
- ◯ Delete Skill

table

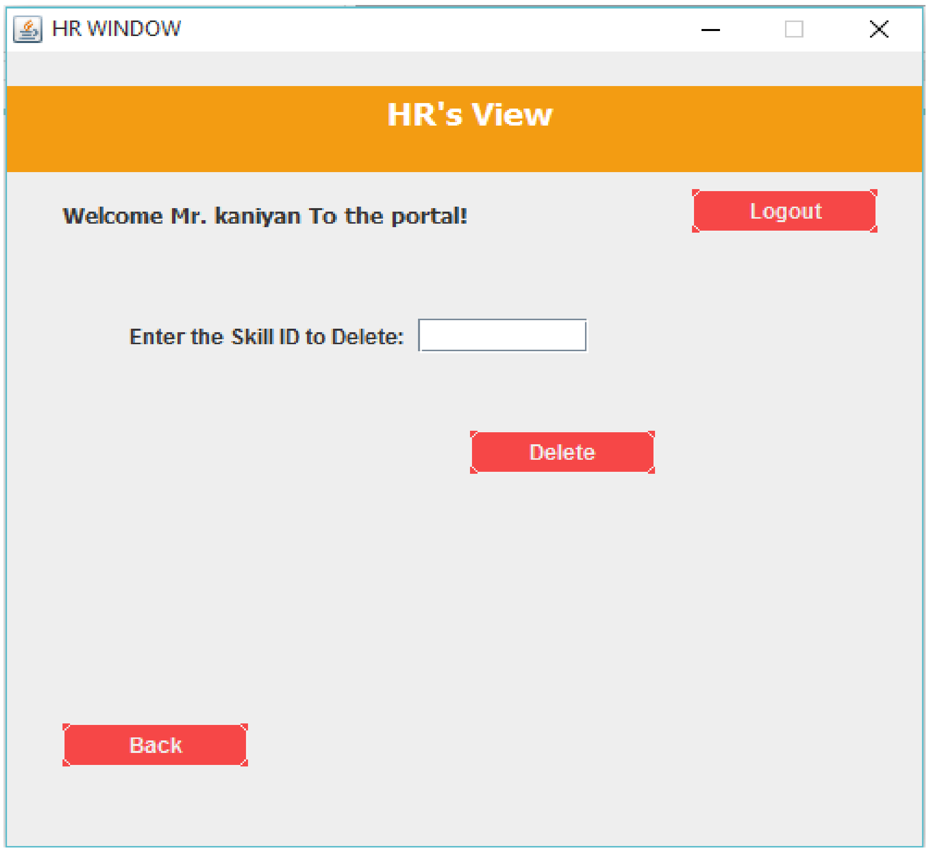
Skill ID	Skill Name	Skill Description	Active
209	Java	Java developer	Yes
210	SQL major	sql server needed	Yes
211	React	React Developer	Yes

Activate, deactivate and delete Employee/skill with employee ID/skill ID.



Welcome Mr. kaniyan To the portal!

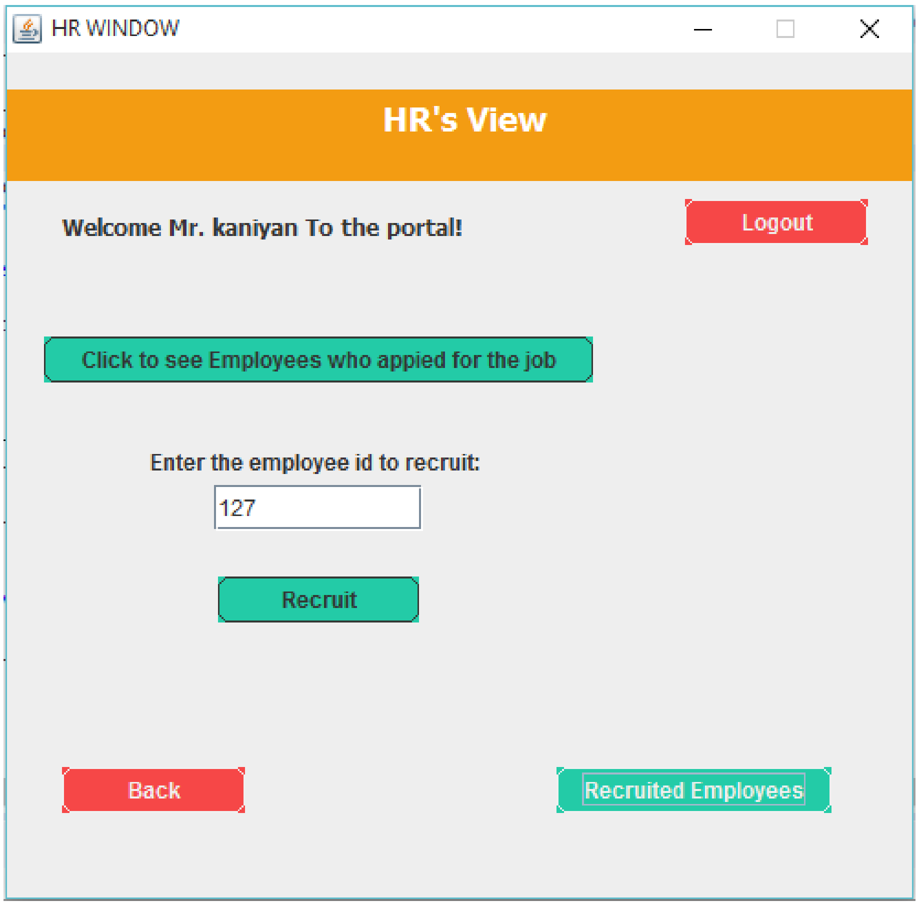
Enter the Employee ID to Take Action:

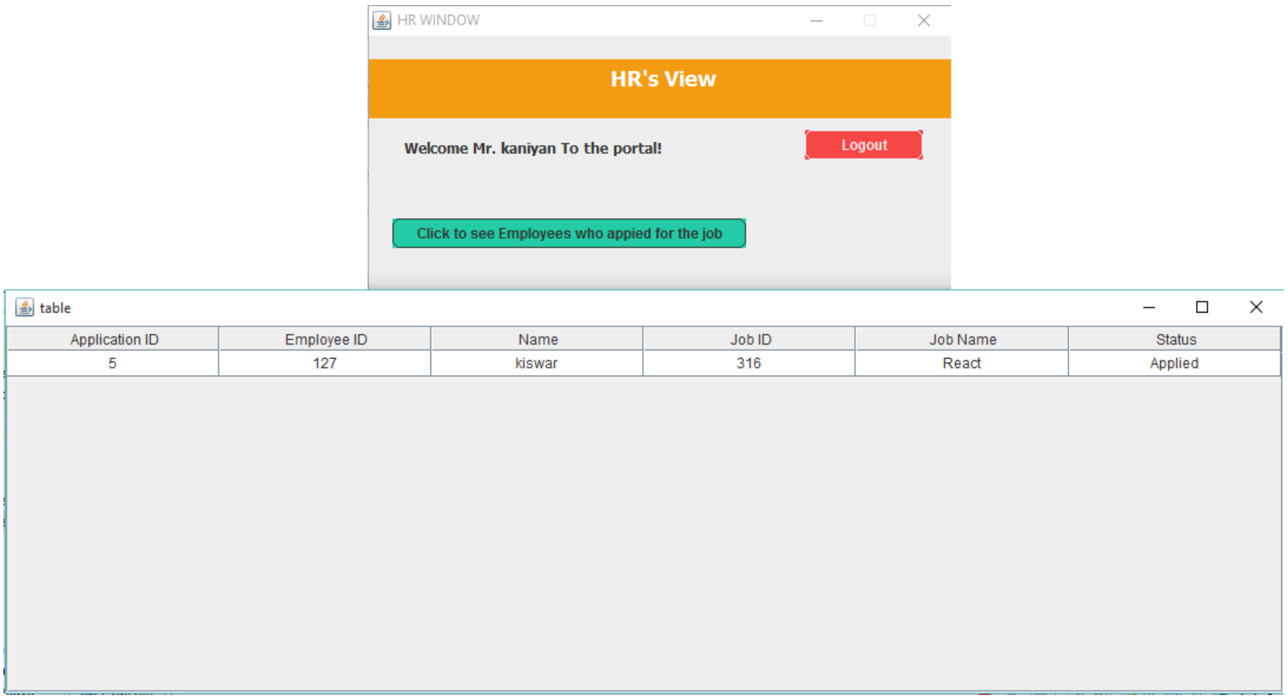


Welcome Mr. kaniyan To the portal!

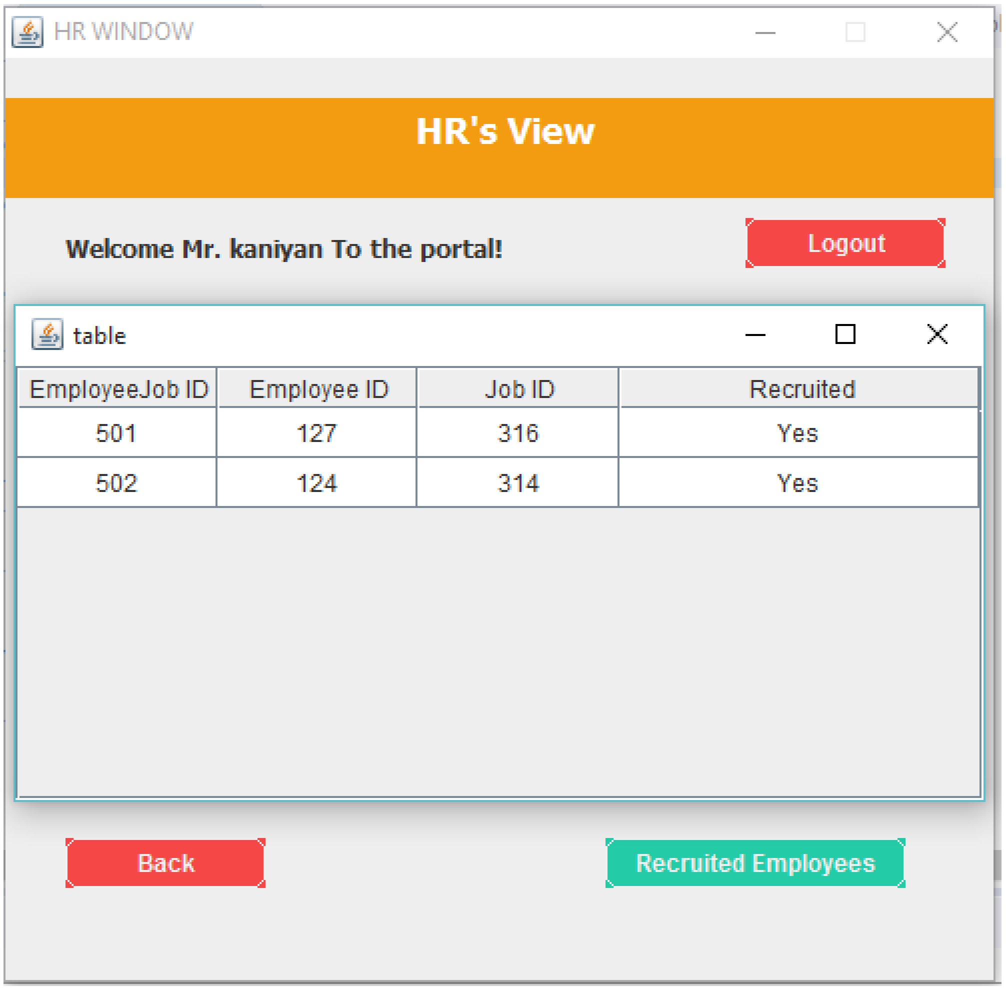
Enter the Skill ID to Delete:

HR’s Recruitment view


HR WINDOW
HR's View
Welcome Mr. kaniyan To the portal!
Logout
Click to see Employees who appied for the job
Enter the employee id to recruit:
127
Recruit
Back
Recruited Employees


HR WINDOW
HR's View
Welcome Mr. kaniyan To the portal!
Logout
Click to see Employees who appied for the job
table

Application ID	Employee ID	Name	Job ID	Job Name	Status
5	127	kiswar	316	React	Applied

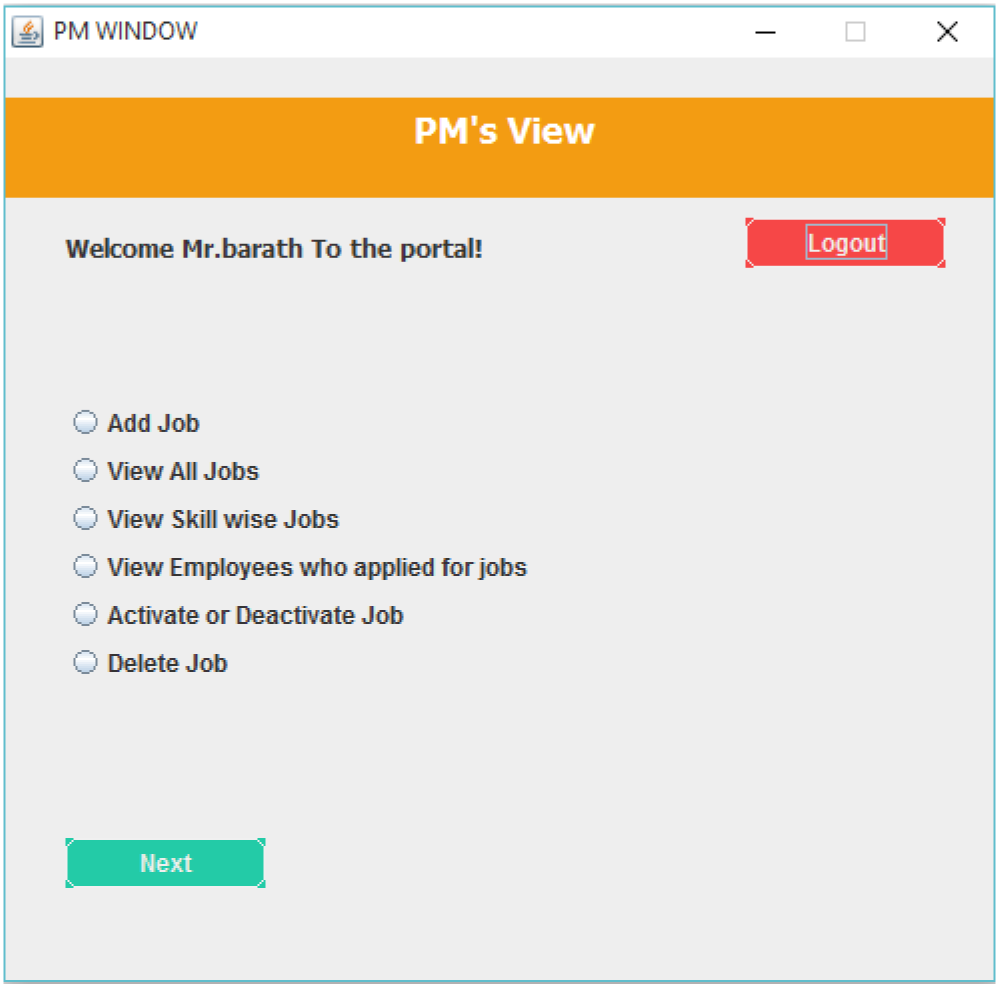


Welcome Mr. kaniyan To the portal!

EmployeeJob ID	Employee ID	Job ID	Recruited
501	127	316	Yes
502	124	314	Yes

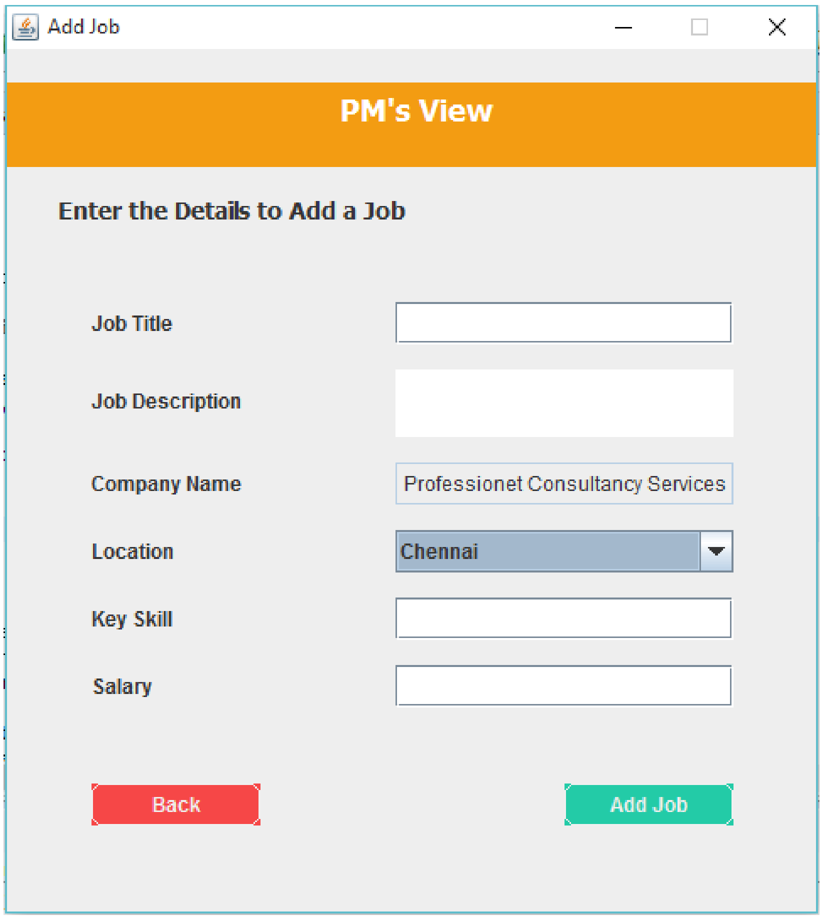
This the recruited employees by the HR by entering the employee ID on the TextField.

Project Manager's View


PM WINDOW
PM's View
Welcome Mr.barath To the portal!
Logout
Add Job
View All Jobs
View Skill wise Jobs
View Employees who applied for jobs
Activate or Deactivate Job
Delete Job
Next

Nice and sleekly designed user interface that allows the managers to take navigate through the process quickly and effectively. The above pictures shows that what a PM capable of doing in that organization. He is responsible for the entire process of monitoring employees who belongs to that firm.

PM can add a Job by entering the necessary details.



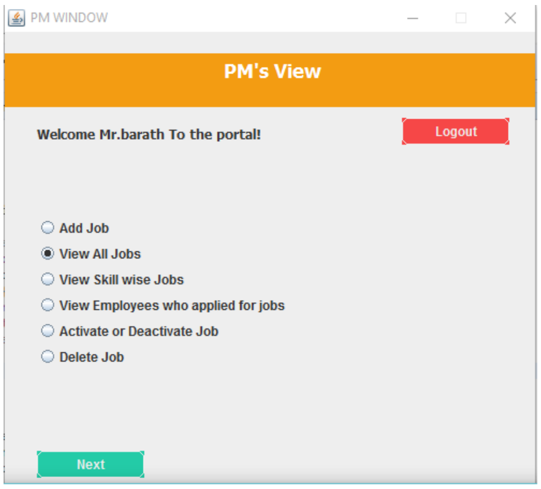
Add Job

PM's View

Enter the Details to Add a Job

Job Title	
Job Description	
Company Name	Professionet Consultancy Services
Location	Chennai
Key Skill	
Salary	

PM can able to view all the jobs that he has added to maintain the circle.

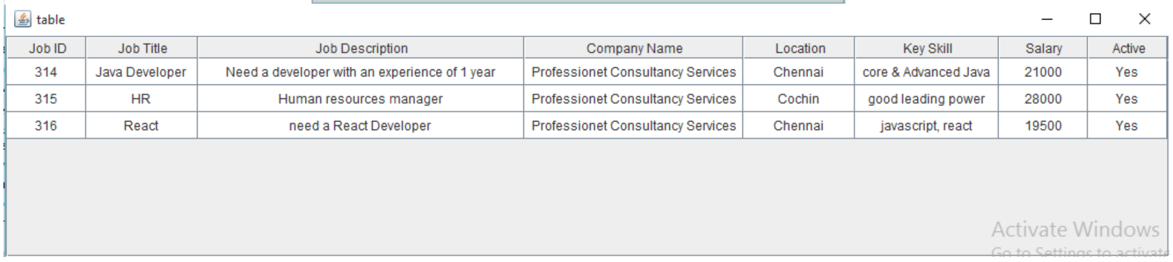


PM WINDOW

PM's View

Welcome Mr.barath To the portal!

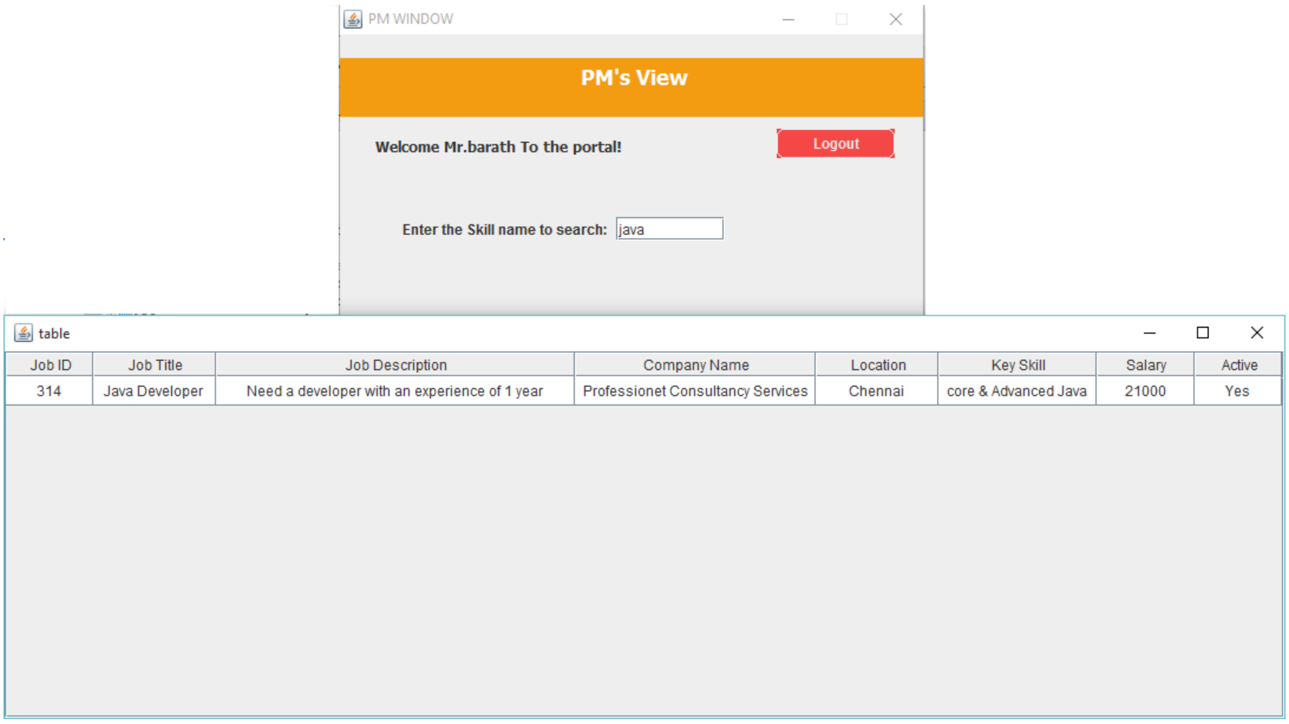
- Add Job
- [✓ View All Jobs]
- View Skill wise Jobs
- View Employees who applied for jobs
- Activate or Deactivate Job
- Delete Job



table

Job ID	Job Title	Job Description	Company Name	Location	Key Skill	Salary	Active
314	Java Developer	Need a developer with an experience of 1 year	Professionet Consultancy Services	Chennai	core & Advanced Java	21000	Yes
315	HR	Human resources manager	Professionet Consultancy Services	Cochin	good leading power	28000	Yes
316	React	need a React Developer	Professionet Consultancy Services	Chennai	javascript, react	19500	Yes

He can filter the jobs using the keywords.

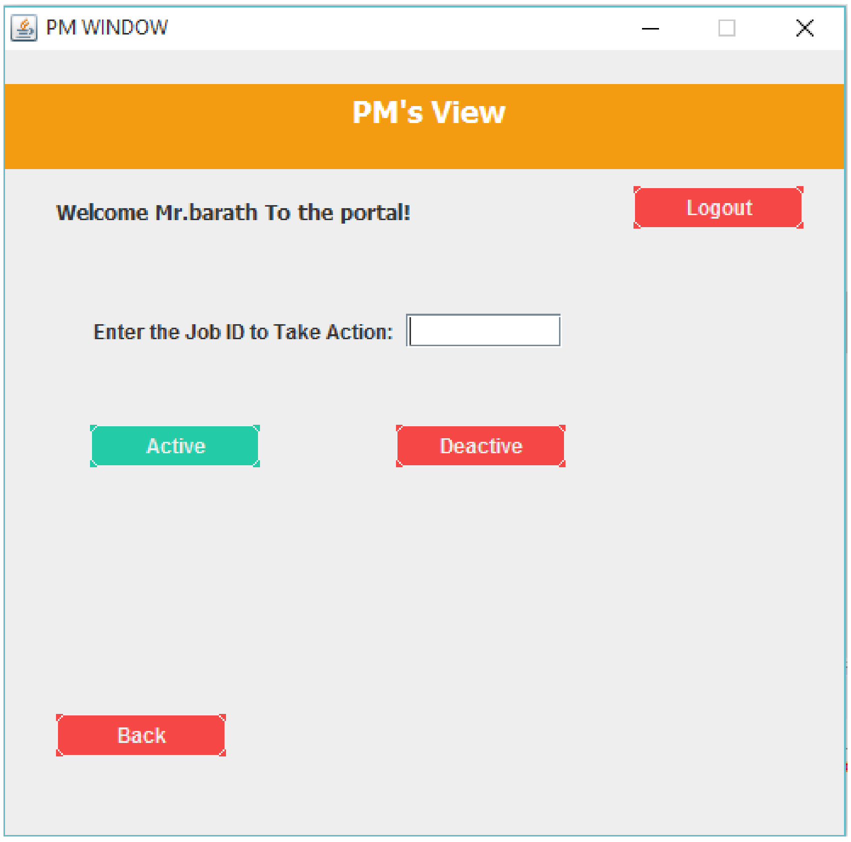


Welcome Mr.barath To the portal!

Enter the Skill name to search: java

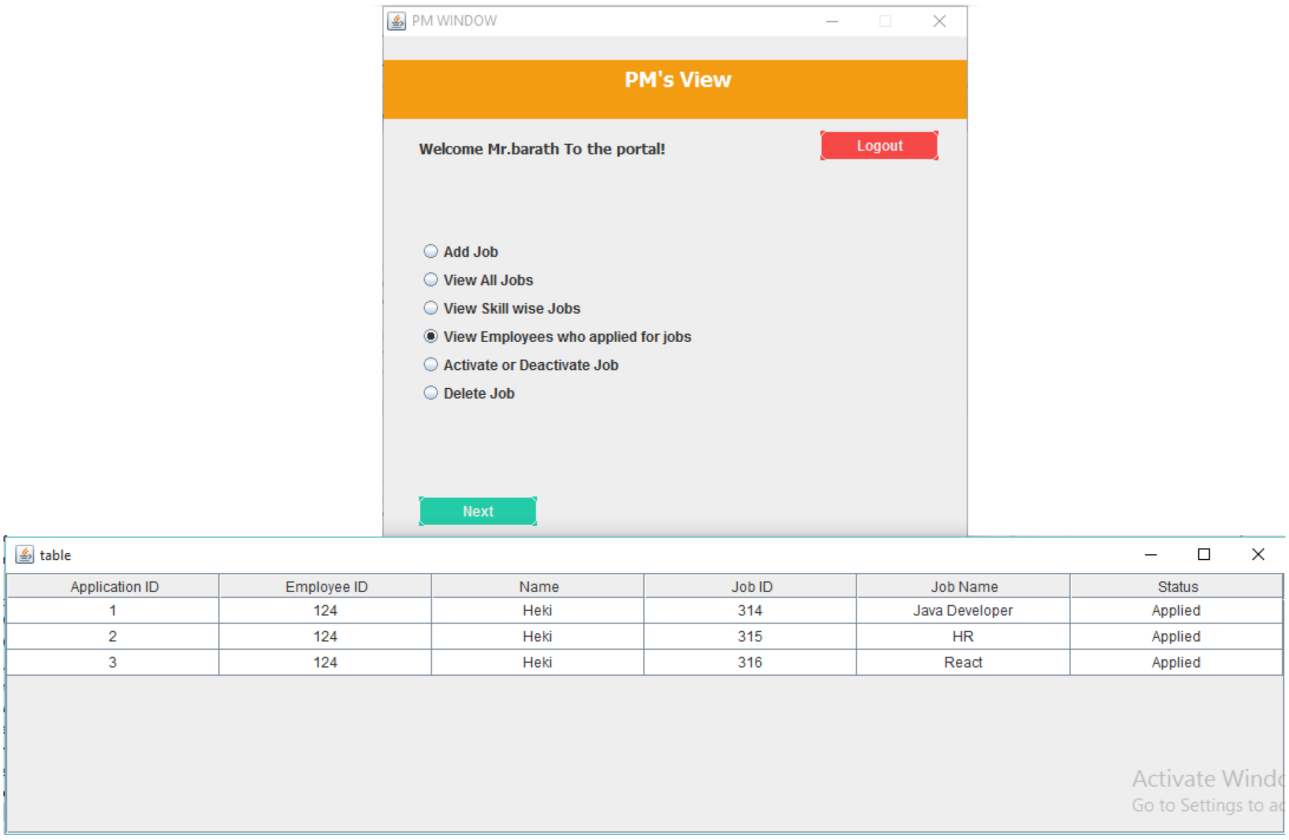
Job ID	Job Title	Job Description	Company Name	Location	Key Skill	Salary	Active
314	Java Developer	Need a developer with an experience of 1 year	Professionet Consultancy Services	Chennai	core & Advanced Java	21000	Yes

PM can activate and deactivate the posted job in the portal.



Welcome Mr.barath To the portal!

Enter the Job ID to Take Action:


PM WINDOW

PM's View

Welcome Mr.barath To the portal!

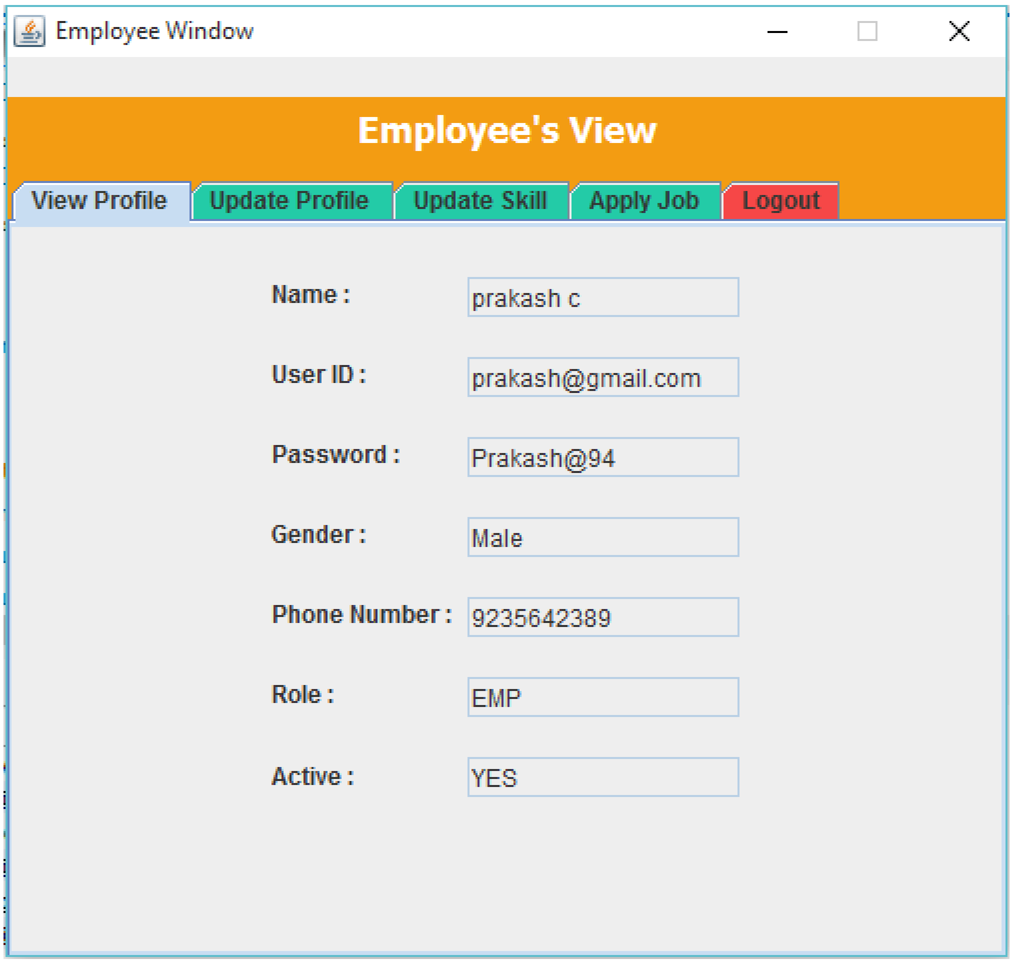
- Add Job
- View All Jobs
- View Skill wise Jobs
- [✓ View Employees who applied for jobs]
- Activate or Deactivate Job
- Delete Job

table

Application ID	Employee ID	Name	Job ID	Job Name	Status
1	124	Heki	314	Java Developer	Applied
2	124	Heki	315	HR	Applied
3	124	Heki	316	React	Applied

Manager can monitor the employees who have applied for the job and manage the job flow and response to that particular project.

Employee’s View

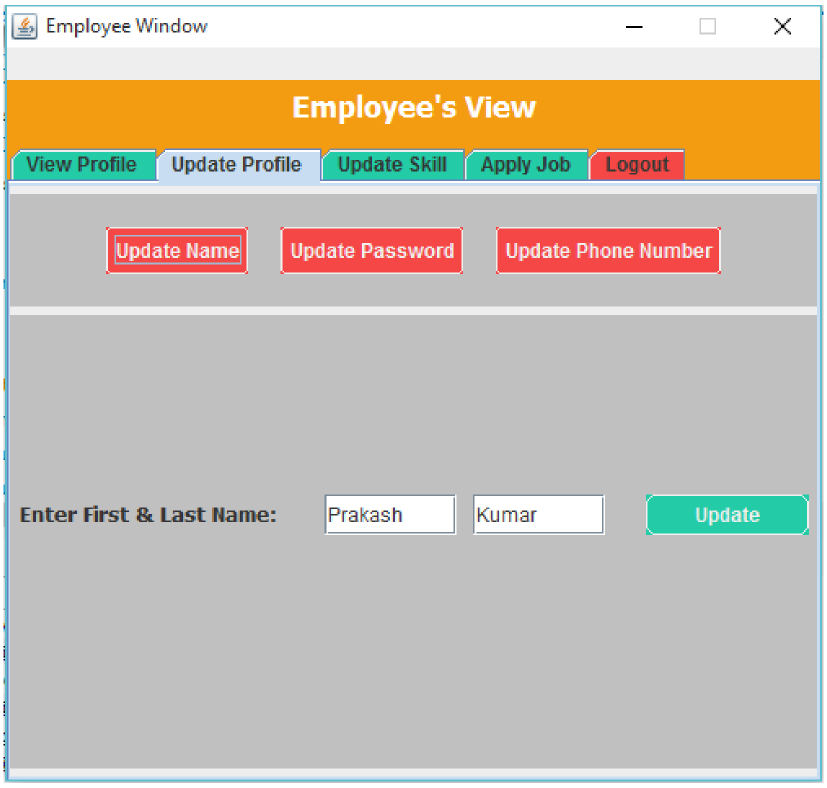


Name :	prakash c
User ID :	prakash@gmail.com
Password :	Prakash@94
Gender :	Male
Phone Number :	9235642389
Role :	EMP
Active :	YES

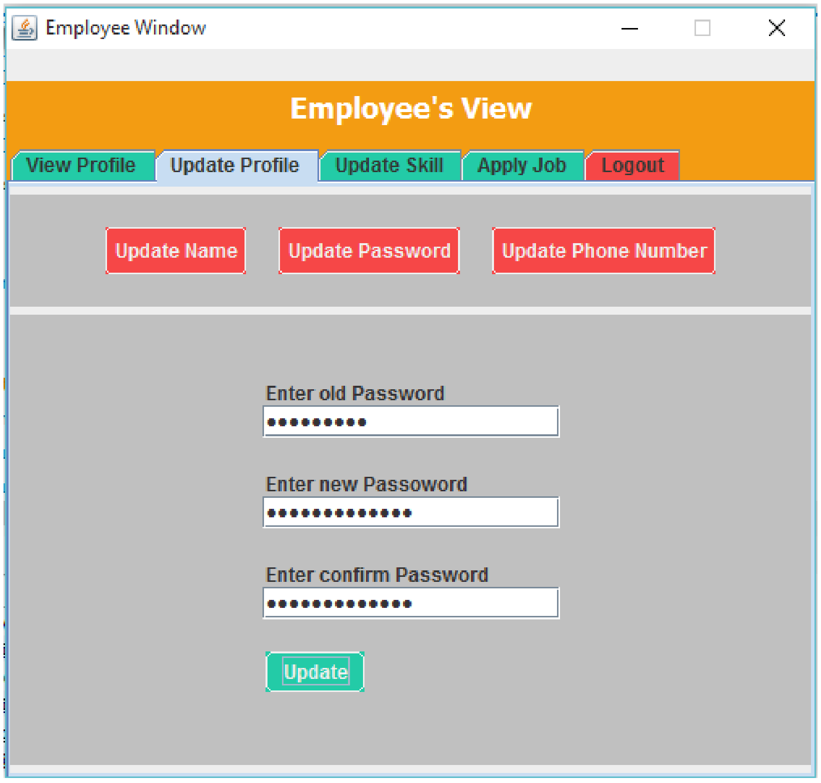
Once you login as an employee, you will be automatically directed to your profile view.

Employee’s view can give access to the employees to view their profile. Update their profile and skill and also apply the available jobs.

Update profile:



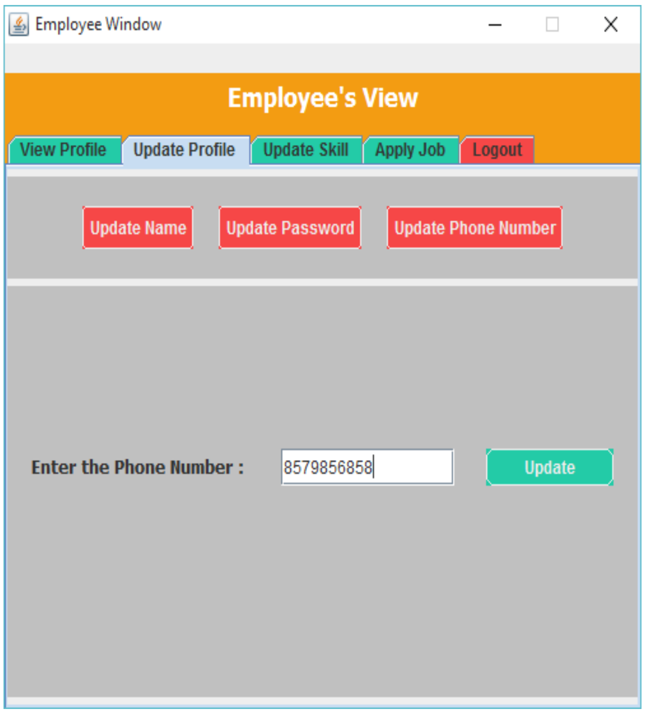
Enter First & Last Name: Prakash Kumar



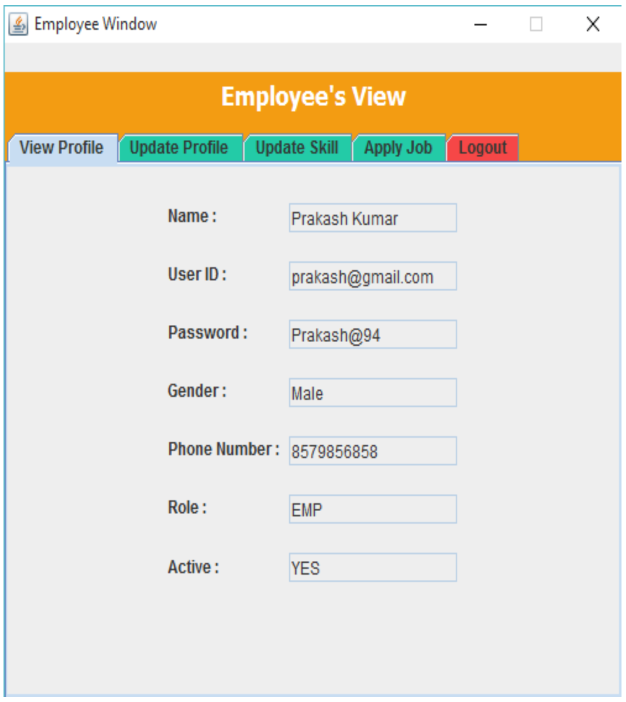
Enter old Password ●●●●●●●●●

Enter new Passoword ●●●●●●●●●●●●●

Enter confirm Password ●●●●●●●●●●●●●



Enter the Phone Number : 8579856858



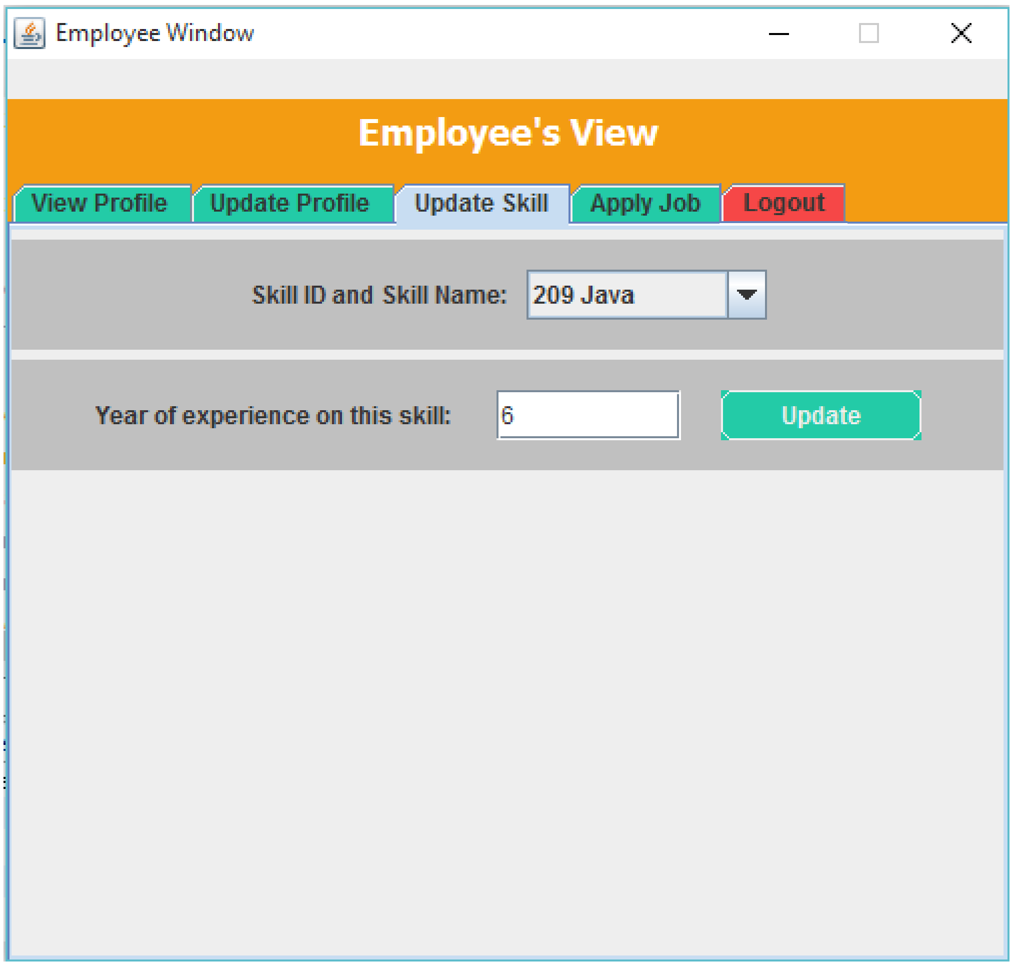
Name : Prakash Kumar
User ID : prakash@gmail.com
Password : Prakash@94
Gender : Male
Phone Number : 8579856858
Role : EMP
Active : YES

The above four images belongs to update profile in which the user can able to update their name, password and phone number.

When comes to the password, authentication is really important that some else in the room can able to change your password and access your data. So, we makes sure that the user has to enter the old password to change to a new one. The old password entered by the user will be crossed checked with the database. Once the authentication is approved by the system, the user can change their password. And similarly, for the phone number, the user has to enter 10 digit number and that has to starts with 7, 8 and 9 respectively.

As you see on the updating images, **every update will reflect immediately on their portal** and they need not logout and login again.

Skill Updating:

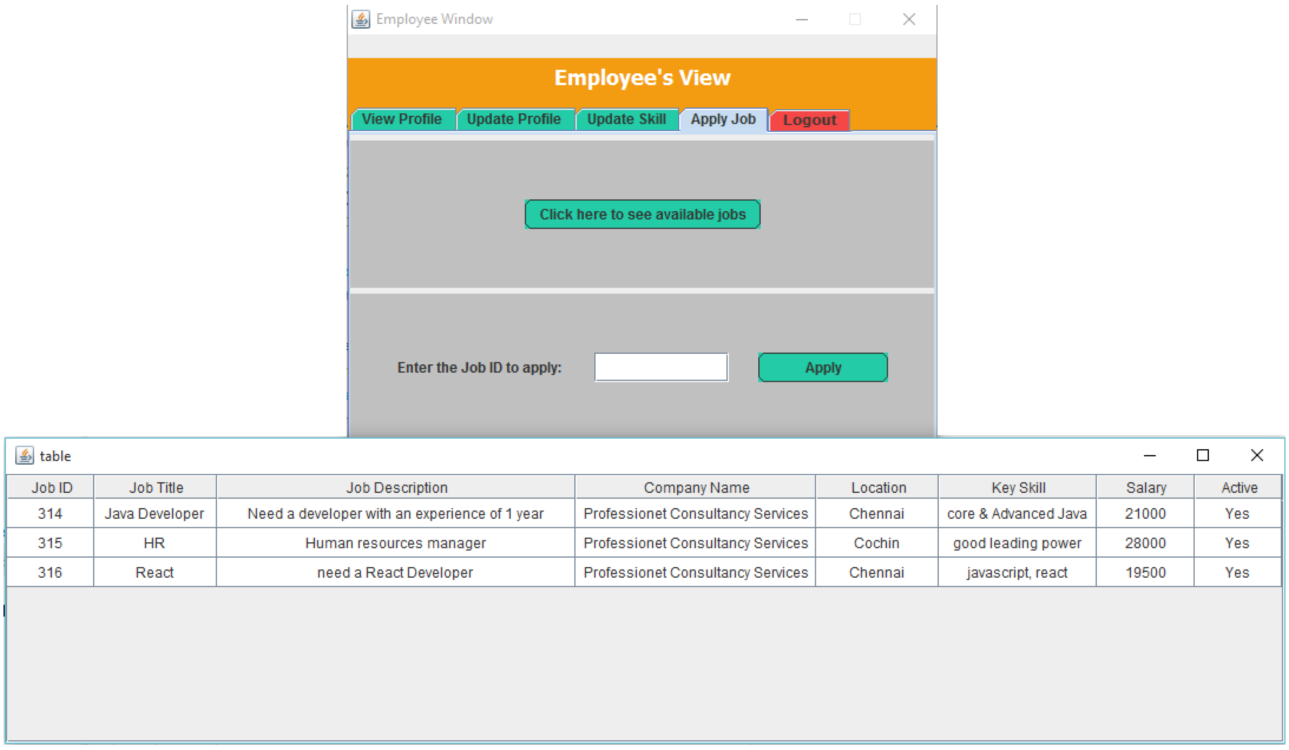


Skill ID and Skill Name: 209 Java

Year of experience on this skill: 6

Employee can able to update their skill which are made available and required by the HR and PM. In this panel, the user required to select the skill id and name of the skill along with the number of professional experience the user having. And the user can update the skill.

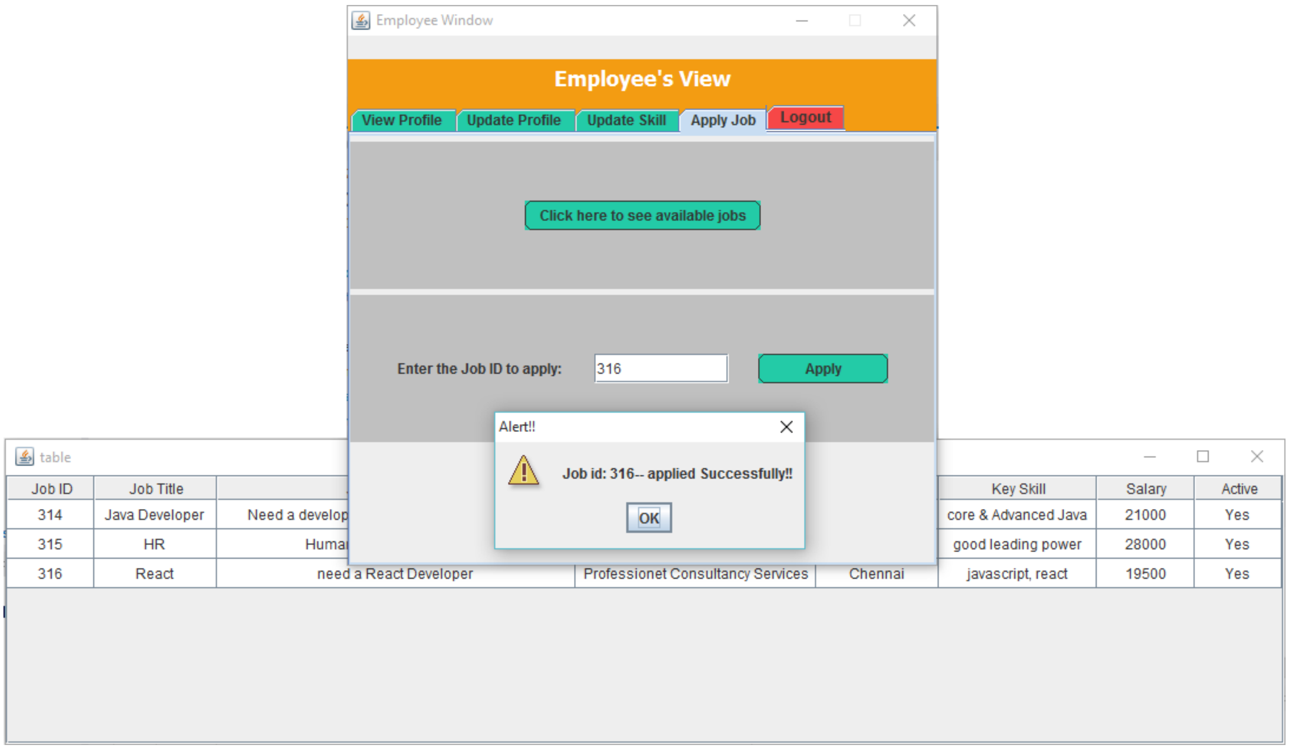
Apply Job:



Enter the Job ID to apply:

Job ID	Job Title	Job Description	Company Name	Location	Key Skill	Salary	Active
314	Java Developer	Need a developer with an experience of 1 year	Professionet Consultancy Services	Chennai	core & Advanced Java	21000	Yes
315	HR	Human resources manager	Professionet Consultancy Services	Cochin	good leading power	28000	Yes
316	React	need a React Developer	Professionet Consultancy Services	Chennai	javascript, react	19500	Yes

In the apply job panel, the user can have the access to view the available jobs on the portal and accordingly the one can apply for the job by entering the job id with ease. Below is the successful application.

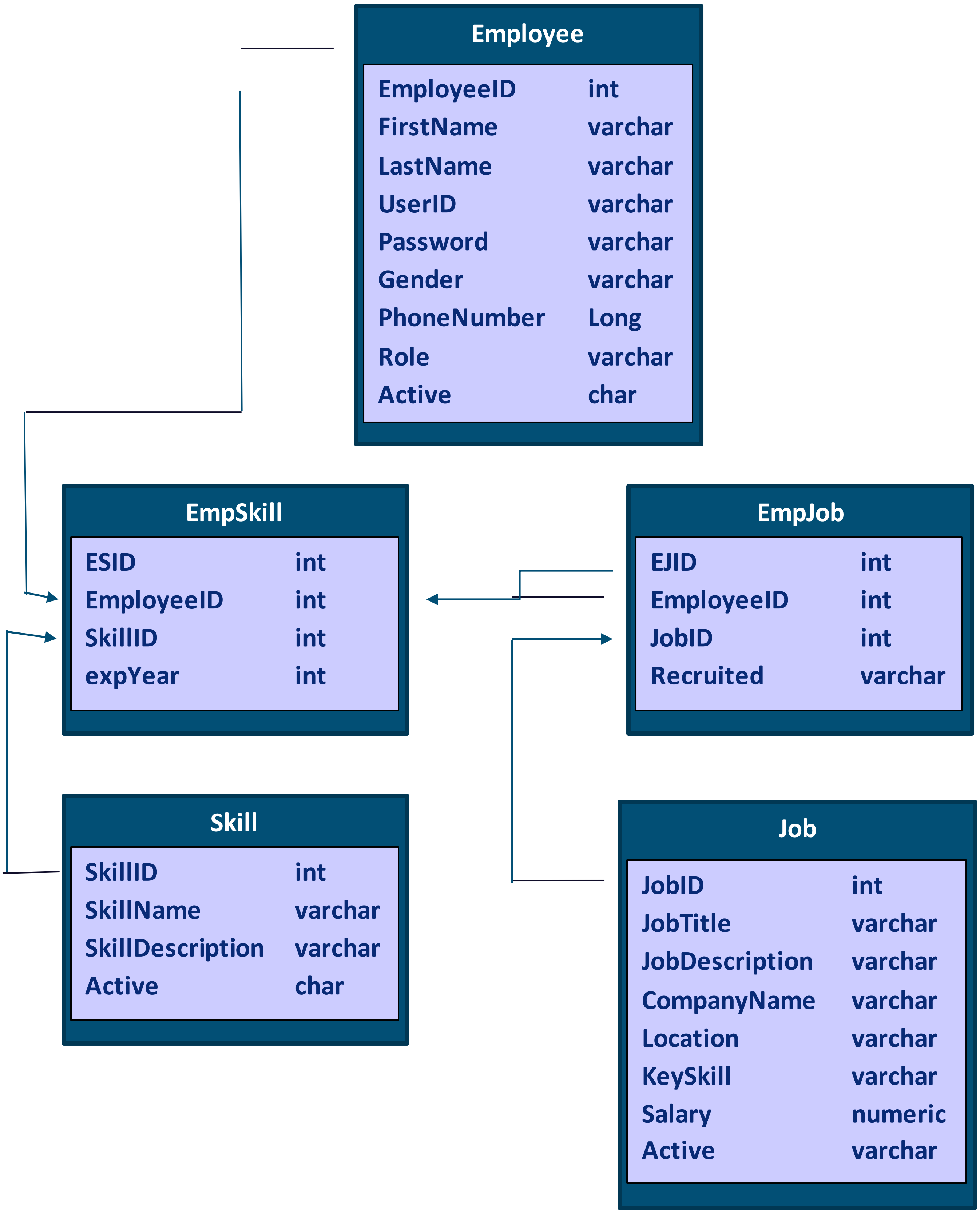


Enter the Job ID to apply: 316

Job id: 316-- applied Successfully!!

Job ID	Job Title				Key Skill	Salary	Active
314	Java Developer	Need a develop			core & Advanced Java	21000	Yes
315	HR	Huma			good leading power	28000	Yes
316	React	need a React Developer	Professionet Consultancy Services	Chennai	javascript, react	19500	Yes

DATA STRUCTURE MODEL



Employee

EmployeeID	int
FirstName	varchar
LastName	varchar
UserID	varchar
Password	varchar
Gender	varchar
PhoneNumber	Long
Role	varchar
Active	char

EmpSkill

ESID	int
EmployeeID	int
SkillID	int
expYear	int

EmpJob

EJID	int
EmployeeID	int
JobID	int
Recruited	varchar

Skill

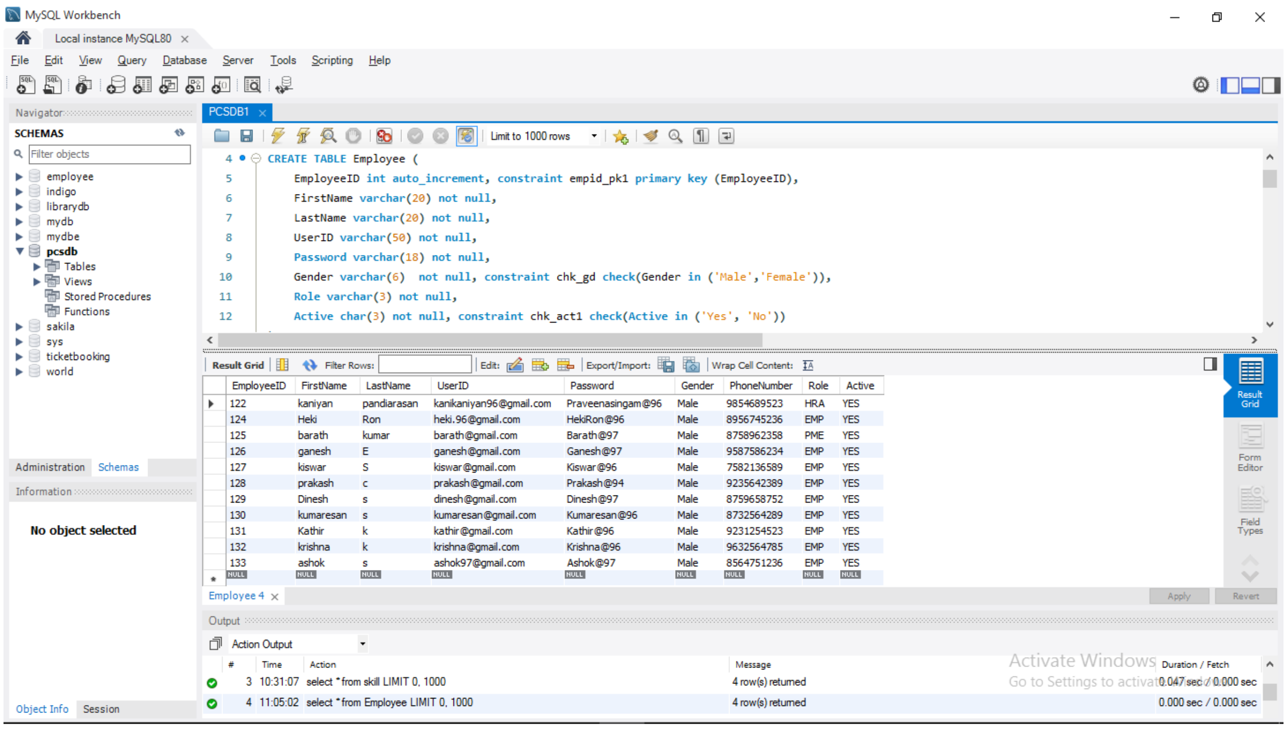
SkillID	int
SkillName	varchar
SkillDescription	varchar
Active	char

Job

JobID	int
JobTitle	varchar
JobDescription	varchar
CompanyName	varchar
Location	varchar
KeySkill	varchar
Salary	numeric
Active	varchar

DATA MODEL

EMPLOYEE TABLE:



```
CREATE TABLE Employee (
    EmployeeID int auto_increment, constraint empid_pk1 primary key (EmployeeID),
    FirstName varchar(20) not null,
    LastName varchar(20) not null,
    UserID varchar(50) not null,
    Password varchar(18) not null,
    Gender varchar(6)  not null, constraint chk_gd check(Gender in ('Male','Female')),
    Role varchar(3) not null,
    Active char(3) not null, constraint chk_act1 check(Active in ('Yes', 'No'))
```

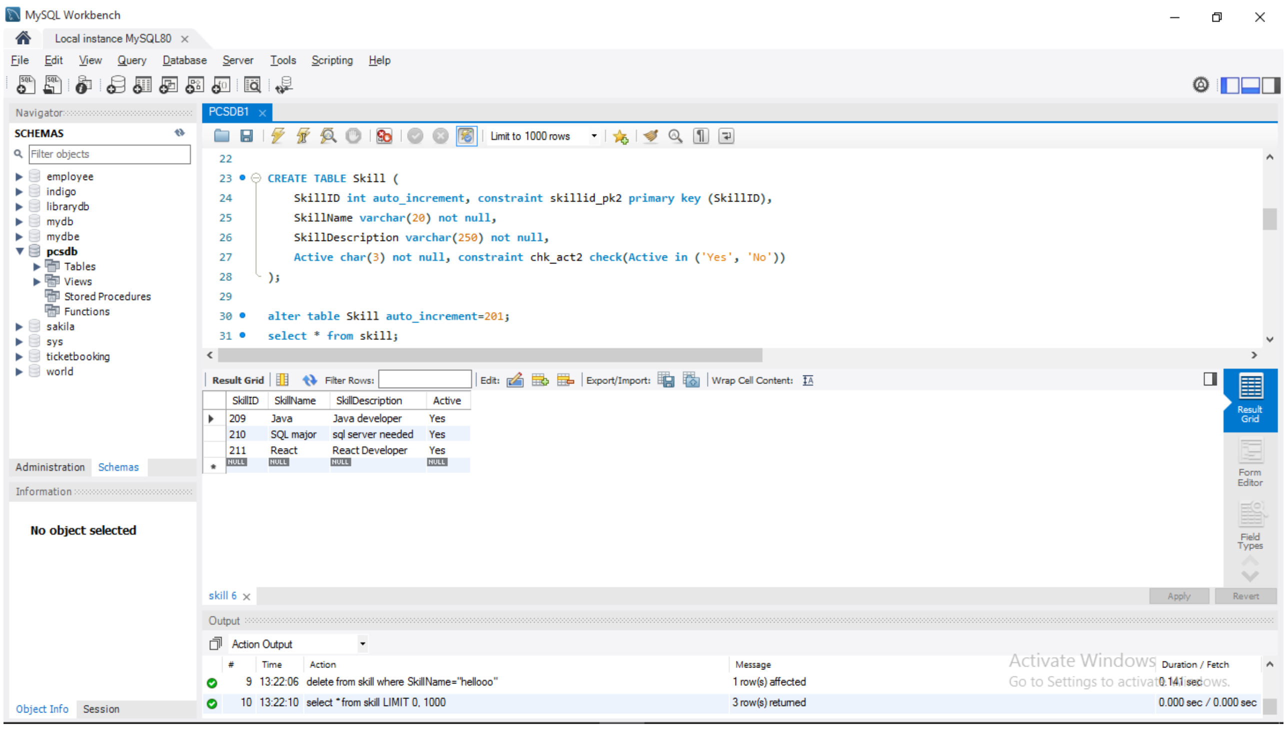
EmployeeID	FirstName	LastName	UserID	Password	Gender	PhoneNumber	Role	Active
122	kaniyan	pandiarasan	kanikaniyan96@gmail.com	Praveenasingam@96	Male	9854689523	HRA	YES
124	Heki	Ron	heki.96@gmail.com	HekiRon@96	Male	8956745236	EMP	YES
125	barath	kumar	barath@gmail.com	Barath@97	Male	8758962358	PME	YES
126	ganesh	E	ganesh@gmail.com	Ganesh@97	Male	9587586234	EMP	YES
127	kiswar	S	kiswar@gmail.com	Kiswar@96	Male	7582136589	EMP	YES
128	prakash	c	prakash@gmail.com	Prakash@94	Male	9235642389	EMP	YES
129	Dinesh	s	dinesh@gmail.com	Dinesh@97	Male	8759658752	EMP	YES
130	kumaresan	s	kumaresan@gmail.com	Kumaresan@96	Male	8732564289	EMP	YES
131	Kathir	k	kathir@gmail.com	Kathir@96	Male	9231254523	EMP	YES
132	krishna	k	krishna@gmail.com	Krishna@96	Male	9632564785	EMP	YES
133	ashok	s	ashok97@gmail.com	Ashok@97	Male	8564751236	EMP	YES
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                  Message            Duration / Fetch
3  10:31:07  select * from skill LIMIT 0, 1000       4 row(s) returned  0.047 sec / 0.000 sec
4  11:05:02  select * from Employee LIMIT 0, 1000    4 row(s) returned  0.000 sec / 0.000 sec
```

Create database PCSDB;
Use PCSDB;

CREATE TABLE Employee (
    EmployeeID int auto_increment, constraint empid_pk1 primary key (EmployeeID),
    FirstName varchar (20) not null,
    LastName varchar (20) not null,
    UserID varchar (50) not null,
    Password varchar (18) not null,
    Gender varchar (6) not null, constraint chk_gd check (Gender in ('Male’, ‘Female')),
    PhoneNumber long not null,
    Role varchar (3) not null,
    Active char (3) not null, constraint chk_act1 check (Active in ('Yes', 'No'))
);
Select * from Employee;

SKILL TABLE:



```
CREATE TABLE Skill (
    SkillID int auto_increment, constraint skillid_pk2 primary key (SkillID),
    SkillName varchar(20) not null,
    SkillDescription varchar(250) not null,
    Active char(3) not null, constraint chk_act2 check(Active in ('Yes', 'No'))
);

alter table Skill auto_increment=201;
select * from skill;
```

SkillID	SkillName	SkillDescription	Active
209	Java	Java developer	Yes
210	SQL major	sql server needed	Yes
211	React	React Developer	Yes
NULL	NULL	NULL	NULL

```
#   Time      Action                                     Message             Duration / Fetch
9   13:22:06  delete from skill where SkillName="hellooo"  1 row(s) affected   0.141 sec
10  13:22:10  select * from skill LIMIT 0, 1000           3 row(s) returned   0.000 sec / 0.000 sec
```

CREATE TABLE Skill (

SkillID int auto_increment, constraint skillid_pk2 primary key (SkillID),

SkillName varchar (20) not null,

SkillDescription varchar (250) not null,

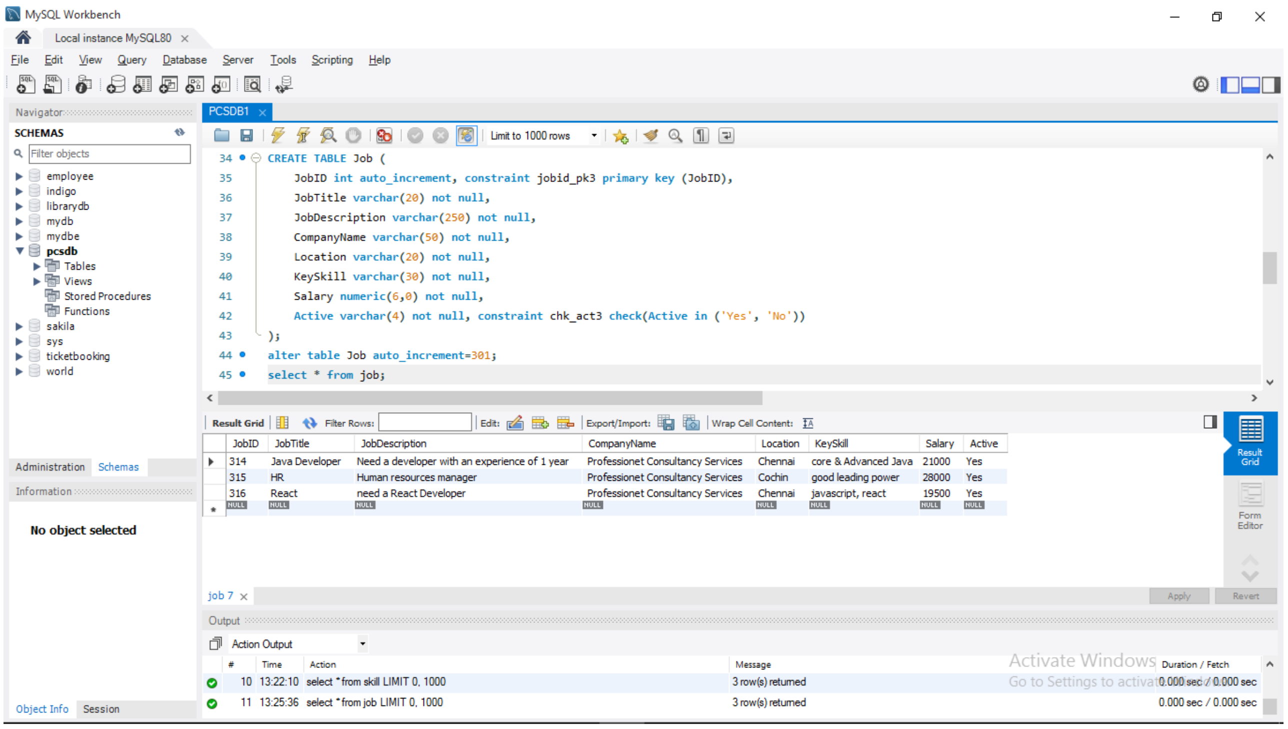
Active char (3) not null, constraint chk_act2 check (Active in ('Yes', 'No'))

);

Alter table Skill auto_increment=201;

Select * from skill;

JOB TABLE:



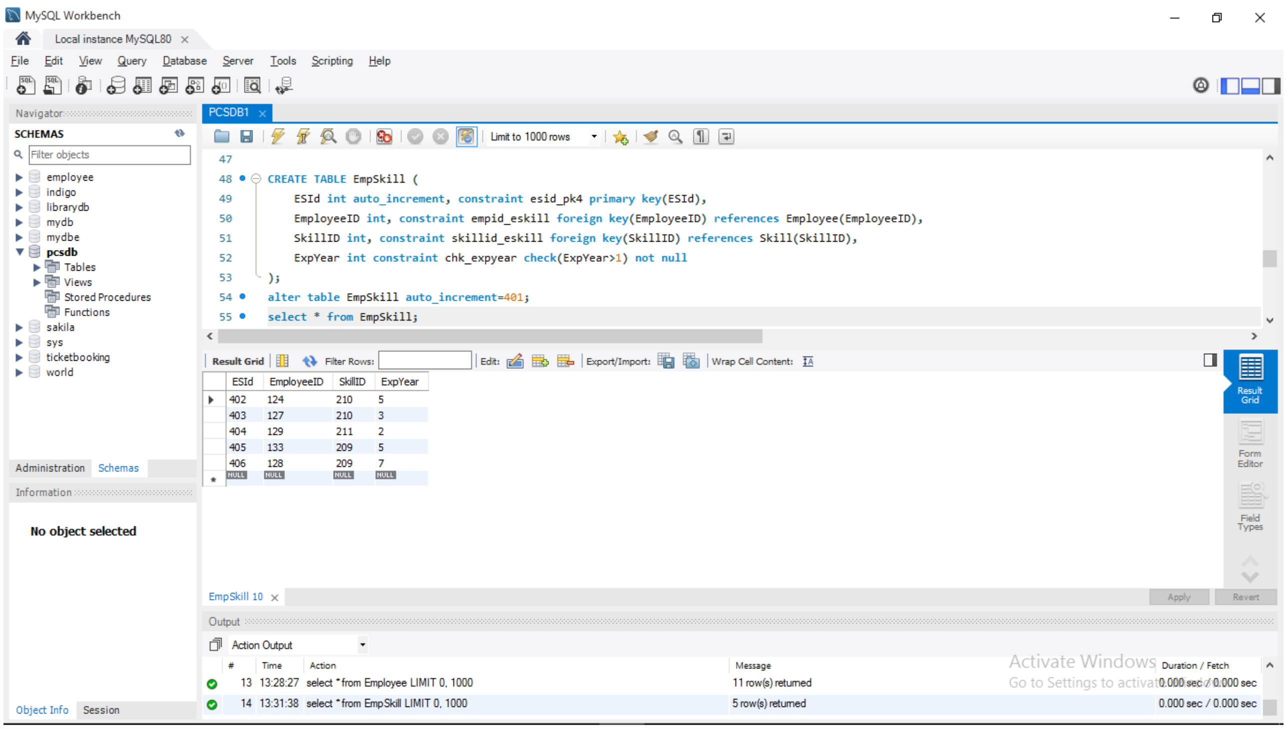
```
CREATE TABLE Job (
    JobID int auto_increment, constraint jobid_pk3 primary key (JobID),
    JobTitle varchar(20) not null,
    JobDescription varchar(250) not null,
    CompanyName varchar(50) not null,
    Location varchar(20) not null,
    KeySkill varchar(30) not null,
    Salary numeric(6,0) not null,
    Active varchar(4) not null, constraint chk_act3 check(Active in ('Yes', 'No'))
);
alter table Job auto_increment=301;
select * from job;
```

JobID	JobTitle	JobDescription	CompanyName	Location	KeySkill	Salary	Active
314	Java Developer	Need a developer with an experience of 1 year	Professionet Consultancy Services	Chennai	core & Advanced Java	21000	Yes
315	HR	Human resources manager	Professionet Consultancy Services	Cochin	good leading power	28000	Yes
316	React	need a React Developer	Professionet Consultancy Services	Chennai	javascript, react	19500	Yes
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                                Message              Duration / Fetch
10  13:22:10  select * from skill LIMIT 0, 1000     3 row(s) returned    0.000 sec / 0.000 sec
11  13:25:36  select * from job LIMIT 0, 1000       3 row(s) returned    0.000 sec / 0.000 sec
```

```
CREATE TABLE Job (
        JobID int auto_increment, constraint jobid_pk3 primary key (JobID),
        JobTitle varchar (20) not null,
        JobDescription varchar (250) not null,
        CompanyName varchar (50) not null,
        Location varchar (20) not null,
        KeySkill varchar (30) not null,
        Salary numeric (6,0) not null,
        Active varchar (4) not null, constraint chk_act3 check(Active in ('Yes', 'No'))
);
Alter table Job auto_increment=301;
Select * from job;
```

EMPSKILL TABLE:



```
CREATE TABLE EmpSkill (
    ESId int auto_increment, constraint esid_pk4 primary key(ESId),
    EmployeeID int, constraint empid_eskill foreign key(EmployeeID) references Employee(EmployeeID),
    SkillID int, constraint skillid_eskill foreign key(SkillID) references Skill(SkillID),
    ExpYear int constraint chk_expyear check(ExpYear>1) not null
);
alter table EmpSkill auto_increment=401;
select * from EmpSkill;
```

ESId	EmployeeID	SkillID	ExpYear
402	124	210	5
403	127	210	3
404	129	211	2
405	133	209	5
406	128	209	7
NULL	NULL	NULL	NULL

```
#   Time      Action                                  Message            Duration / Fetch
13  13:28:27  select * from Employee LIMIT 0, 1000    11 row(s) returned 0.000 sec / 0.000 sec
14  13:31:38  select * from EmpSkill LIMIT 0, 1000    5 row(s) returned  0.000 sec / 0.000 sec
```

CREATE TABLE EmpSkill (

ESId int auto_increment, constraint esid_pk4 primary key (ESId),

EmployeeID int, constraint empid_eskill foreign key (EmployeeID) references Employee (EmployeeID),

SkillID int, constraint skillid_eskill foreign key (SkillID) references Skill (SkillID),

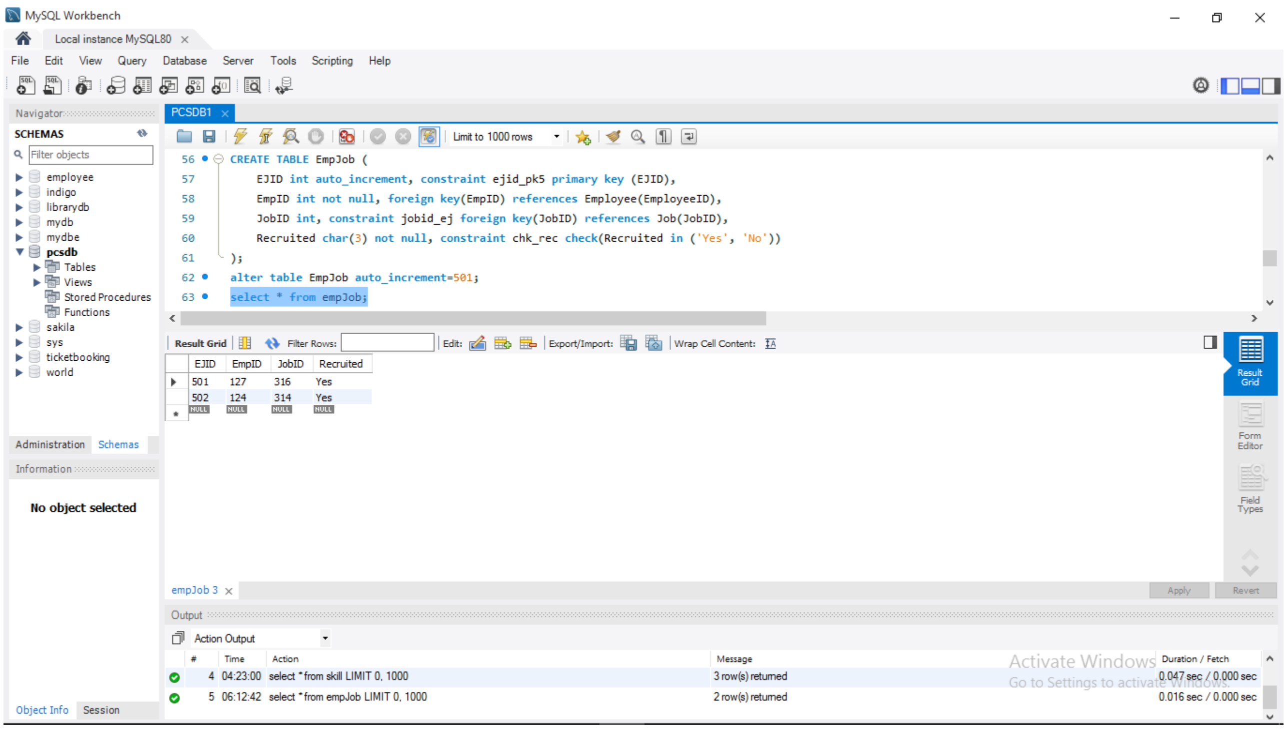
ExpYear int constraint chk_expyear check (ExpYear>1) not null

);

Alter table EmpSkill auto_increment=401;

Select * from EmpSkill;

EMPJOB TABLE:



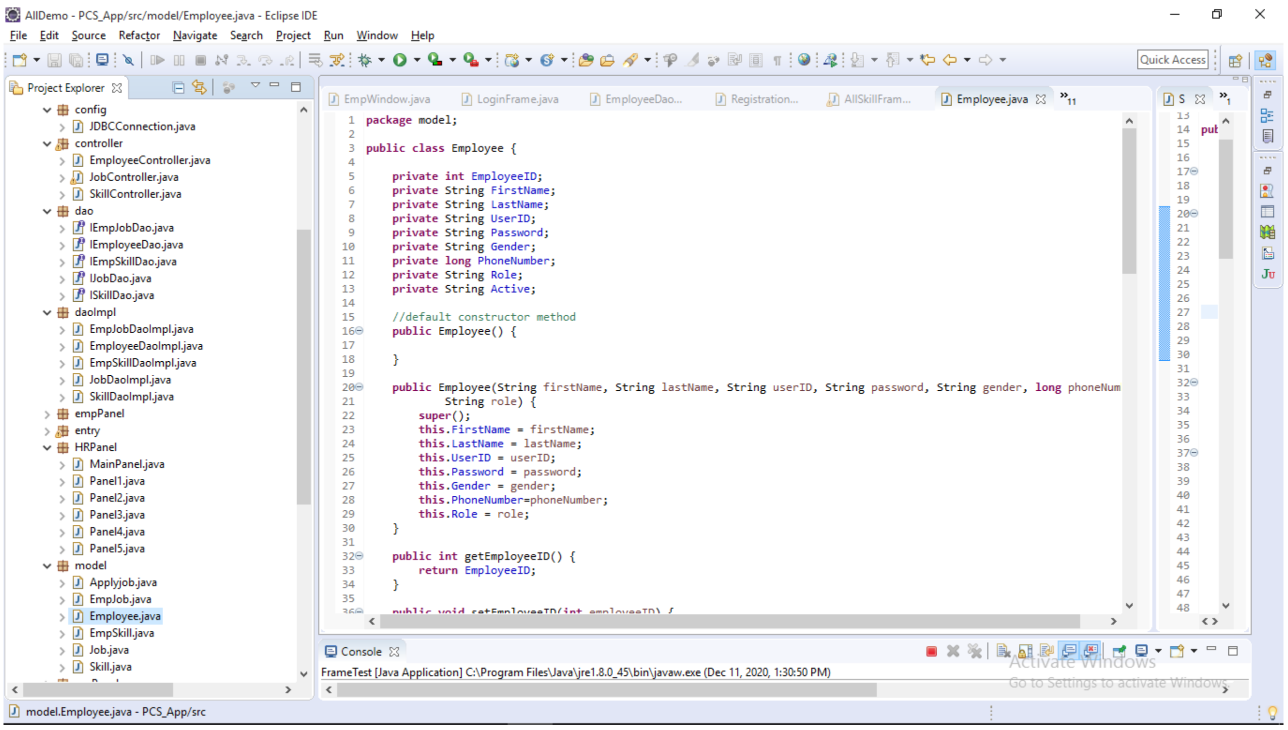
```
CREATE TABLE EmpJob (
    EJID int auto_increment, constraint ejid_pk5 primary key (EJID),
    EmpID int not null, foreign key(EmpID) references Employee(EmployeeID),
    JobID int, constraint jobid_ej foreign key(JobID) references Job(JobID),
    Recruited char(3) not null, constraint chk_rec check(Recruited in ('Yes', 'No'))
);
alter table EmpJob auto_increment=501;
select * from empJob;
```

EJID	EmpID	JobID	Recruited
501	127	316	Yes
502	124	314	Yes
NULL	NULL	NULL	NULL

```
#  Time      Action                               Message             Duration / Fetch
4  04:23:00  select * from skill LIMIT 0, 1000    3 row(s) returned   0.047 sec / 0.000 sec
5  06:12:42  select * from empJob LIMIT 0, 1000   2 row(s) returned   0.016 sec / 0.000 sec
```

```
CREATE TABLE EmpJob (
        EJID int auto_increment, constraint ejid_pk5 primary key (EJID),
        JobID int, constraint jobid_ej foreign key (JobID) references Job (JobID),
        Recruited char (3) not null, constraint chk_rec check (Recruited in ('Yes', 'No'))
);
Alter table EmpJob auto_increment=501;
Select * from empJob;
```

PROJECT MODEL


AllDemo - PCS_App/src/model/Employee.java - Eclipse IDE

```
config
  JDBCConnection.java
controller
  EmployeeController.java
  JobController.java
  SkillController.java
dao
  IEmpJobDao.java
  IEmployeeDao.java
  IEmpSkillDao.java
  IJobDao.java
  ISkillDao.java
daoImpl
  EmpJobDaoImpl.java
  EmployeeDaoImpl.java
  EmpSkillDaoImpl.java
  JobDaoImpl.java
  SkillDaoImpl.java
empPanel
entry
HRPanel
  MainPanel.java
  Panel1.java
  Panel2.java
  Panel3.java
  Panel4.java
  Panel5.java
model
  Applyjob.java
  EmpJob.java
  Employee.java
  EmpSkill.java
  Job.java
  Skill.java
```

```java
package model;

public class Employee {

    private int EmployeeID;
    private String FirstName;
    private String LastName;
    private String UserID;
    private String Password;
    private String Gender;
    private long PhoneNumber;
    private String Role;
    private String Active;

    //default constructor method
    public Employee() {

    }

    public Employee(String firstName, String lastName, String userID, String password, String gender, long phoneNum
            String role) {
        super();
        this.FirstName = firstName;
        this.LastName = lastName;
        this.UserID = userID;
        this.Password = password;
        this.Gender = gender;
        this.PhoneNumber=phoneNumber;
        this.Role = role;
    }

    public int getEmployeeID() {
        return EmployeeID;
    }
```

```
FrameTest [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Dec 11, 2020, 1:30:50 PM)
```

model.Employee.java - PCS_App/src

This is the project model that I have created on Eclipse IDE. The structure of the project is present on the left most side of the window under Project Explorer. There you will get to see different kind of packages like config, controller, dao, daoimpl, entry, model etc., which holds different classes that are categorized in terms of different packages. These are the classes which can handle the entire backend process of the project structure.

Some code snippets are given for your reference.

Config package provides the configuration and it loads the JDBC driver to connect to the database.

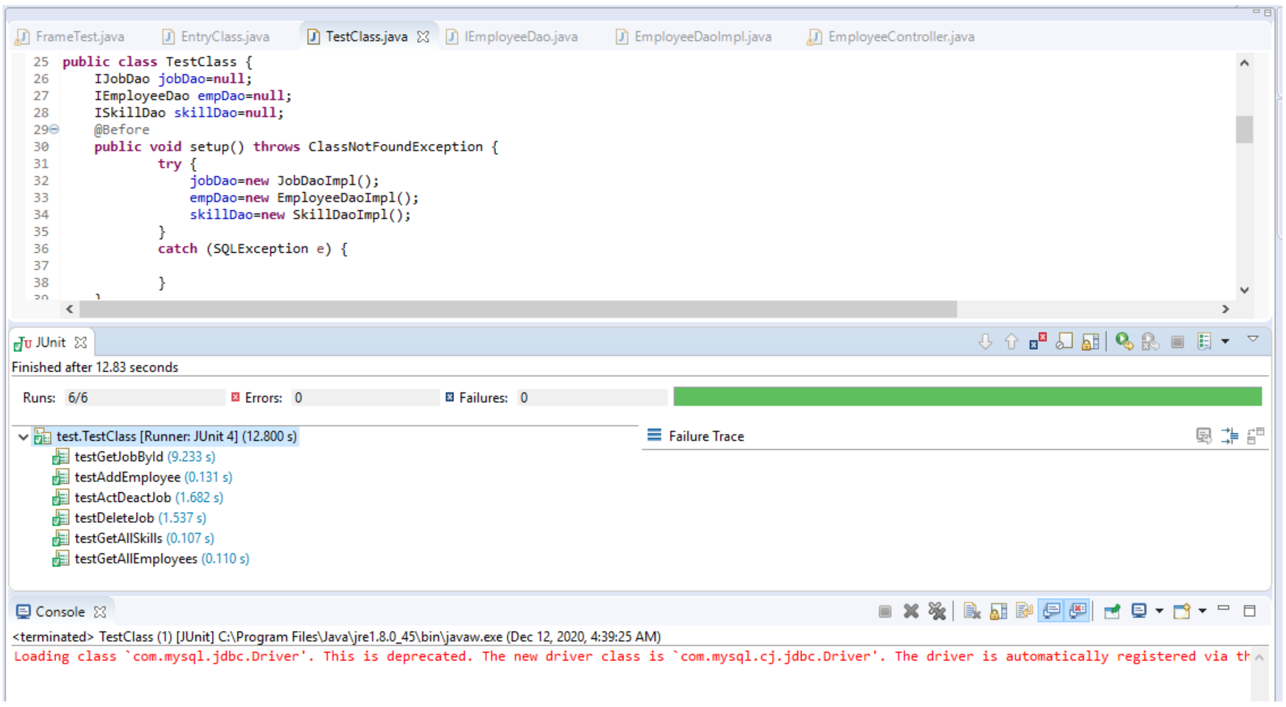
```
package config;
import java.sql.*;

public class JDBCConnection {
	public static Connection getDBConnection() throws ClassNotFoundException,
SQLException {

		String url= "jdbc:mysql://localhost:3306/PCSDB";
		String username = "root";
		String password = "***************";
		//LOADING DRIVER
		Class.forName("com.mysql.jdbc.Driver");
		Connection conn=DriverManager.getConnection(url, username, password);
		return conn;
	}
}
```

Model package provides the base for the structure.

```
package model;

public class Employee {

	private int EmployeeID;
	private String FirstName;
	private String LastName;
	private String UserID;
	private String Password;
	private String Gender;
	private long PhoneNumber;
	private String Role;
	private String Active;

	//default constructor method
	public Employee () {

	}

	public Employee (String firstName, String lastName, String userID, String 
password, String gender, long phoneNumber,
			String role) {
		super ();
		this.FirstName = firstName;
		this.LastName = lastName;
		this.UserID = userID;
		this.Password = password;
		this.Gender = gender;
		this.PhoneNumber=phoneNumber;
		this.Role = role;
	}

	public int getEmployeeID() {
		return EmployeeID;
	}

	public void setEmployeeID (int employeeID) {
		this.EmployeeID = employeeID;
	}

	public String getFirstName() {
		return FirstName;
	}

	public void setFirstName(String firstName) {
		this.FirstName = firstName;
	}

	public String getLastName() {
		return LastName;
	}

	public void setLastName (String lastName) {
		this.LastName = lastName;
	}

	public String getUserID() {
		return UserID;
	}
	public void setUserID(String userID) {
		this.UserID = userID;
	}

	public String getPassword() {
		return Password;
	}

	public void setPassword(String password) {
		this.Password = password;
	}

	public String getGender() {
		return Gender;
	}

	public void setGender(String gender) {
		this.Gender = gender;
	}

	public long getPhoneNumber() {
		return PhoneNumber;
	}

	public void setPhoneNumber(long phoneNumber) {
		this.PhoneNumber = phoneNumber;
	}

	public String getRole() {
		return Role;
	}

	public void setRole(String role) {
		this.Role = role;
	}

	public String getActive() {
		return Active;
	}

	public void setActive(String active) {
		this.Active = active;
	}

	@Override
	public String toString() {
		return "Employee [EmployeeID=" + EmployeeID + ", FirstName=" + FirstName +
", LastName=" + LastName
				+ ", UserID=" + UserID + ", Password=" + Password + ",
Gender=" + Gender + ", PhoneNumber="
				+ PhoneNumber + ", Role=" + Role + ", Active=" + Active +
"]";
	}



}
```

TEST CASES

In Software Engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program oath or to verify compliance with a specific requirement.



```
public class TestClass {
    IJobDao jobDao=null;
    IEmployeeDao empDao=null;
    ISkillDao skillDao=null;
    @Before
    public void setup() throws ClassNotFoundException {
            try {
                jobDao=new JobDaoImpl();
                empDao=new EmployeeDaoImpl();
                skillDao=new SkillDaoImpl();
            }
            catch (SQLException e) {

            }
```

```
JUnit
Finished after 12.83 seconds
Runs: 6/6    Errors: 0    Failures: 0
test.TestClass [Runner: JUnit 4] (12.800 s)
    testGetJobById (9.233 s)
    testAddEmployee (0.131 s)
    testActDeactJob (1.682 s)
    testDeleteJob (1.537 s)
    testGetAllSkills (0.107 s)
    testGetAllEmployees (0.110 s)
Failure Trace
```

```
Console
<terminated> TestClass (1) [JUnit] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Dec 12, 2020, 4:39:25 AM)
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via th
```

The above picture shows that the test cases that I have tested with JUNIT and all the test cases has been progressed successfully.

Here are some test cases:

```
@Test
	public void testGetJobById() {
		Job jobActual=new Job();
		Job jobExpected=new Job(314, "Java Developer", "Need a developer with an experience of 1 year", " Professionet Consultancy Services", "Chennai", "core & Advanced Java", 21000, "Yes");
		jobActual=jobDao.getJobById(314);
		assertEquals(jobExpected.getJobTitle(), jobActual.getJobTitle());
	}
```

The above test case checks whether the expected output matches with the actual output by using assertEquals statement.

```
@Test
	public void testActDeactJob() {
		Job jobActual=new Job();
		Job JobExpected=new Job(314, "Java Developer", "Need a developer with an 
experience of 1 year", " Professionet Consultancy Services", "Chennai", "core & 
Advanced Java", 21000, "NO");
		Job job=jobDao.getJobById(314);
		job.setActive("NO");
		jobActual=jobDao.deactivateJob(job, "Deactivated");
		assertEquals(JobExpected.getActive(), jobActual.getActive());
	}
```

The above test case monitors whether the job has been deactivated or not. If its deactivated, it will be set to “NO” and that will be compared with the assert equals statement.

```
@Test
	public void testDeleteJob() {
		Job jobActual=new Job();
		new Job(315, "HR", "Human resources manager", " Professionet Consultancy 
Services", "Cochin", "good leading power", 28000, "Yes");
		jobActual=jobDao.deleteJob(315);
		assertNull(jobActual);
	}
```

This test case implies that the actual output will be null. Because, here we are deleting the employee. Once it is deleted, it will be no longer available in the database. So, it will be checked using the assertNull statement.

```
@Test
      public void testGetAllEmployees() {
            List<Employee> allEmpList=empDao.getAllEmployees();

            assertNotNull(allEmpList);

            Employee emp =allEmpList.get (3);
            AssertEquals (126, emp.getEmployeeID ());

            Employee emp1 =allEmpList.get (8);
            assertEquals ("Kathir", emp1.getFirstName());

            assertTrue ("Is Active ?", emp1.getActive().equals("YES"));

            assertThat (allEmpList.size(), is (11));
      }
```

Here, in the above test case, the details of all the employees are retrieved to the user interface to display to the user. It has to be checked whether it's collecting all the available data from the database. So, the entire details will be collected in the List and first, we are checking for null values using *assertNotNull*. Then, using get method we are randomly checking the data using the index value. And also here *AssertThat* is used to check size of the array that is equals to the actual size of the List.

PERSONAL OBSERVATION

- ✓ Spending enough time interacting to get the needed data as much as possible.
- ✓ Reducing the problem of reactivity. Gaining intuitive understanding of the meaning of the structure.
- ✓ Addressing problems that are received while compiling the code snippets and sort out the possible solutions.
- ✓ Moreover, I personally enjoyed while working in this project and gained more knowledge, boosts my confidence and brings me to successfully submit the project on time.

Software project developers are always open to something new, but they are not oracles and cannot foresee all the issues that may arise on the way. Moreover, many problems are not an obstruction, but a chance to see the entire project from a different point view. Apply all our responsibility, attention, skills, and mental flexibility, embrace problem-solving and we will always come up with the best possible solution for any problem.