EMPLOYEE MANAGEMENT SYSTEM

SANKAR M (RA2211031010024)

ANAND BALAJI S N (RA2211031010012)

GIRISAI S (RA2211031010009)

INTRODUCTION

- An Employee Management System (EMS) is a comprehensive and integrated software solution designed to streamline and optimize the various aspects of managing an organization's workforce. This system plays a crucial role in enhancing efficiency, transparency, and overall productivity within a company by automating and centralizing employee-related processes.

- The primary objective of an Employee Management System is to simplify the human resources (HR) functions, enabling organizations to manage their workforce more effectively. This system encompasses a wide range of functionalities, including but not limited to the database for storing and managing essential employee information, such as personal details, contact information, job history, and qualifications.

OBJECTIVE

- The objective of this work is to give a complete approach to personnel information management. This will be accomplished by developing and deploying an HR management system that will result in a significant shift in the way employee data is managed.

This system's objectives include the following:

- Design of an HR management system to meet needs such as adding and deleting employees, viewing and printing employee data, and updating employee information.

- Employee data is stored in a well-designed database.

- An easy-to-use interface that will let user interact with the system.

- Availability – All the employee details stored permanently in the database admin can see the data, whatever needed.

- Maintain Cost – Reduce the cost of maintenance. It is standalone application so no required of cost for maintain

- **Enable Real-time Updates:** Implement features that allow real-time updates of stocks availability and member activities, ensuring the information is always current and reliable.

- **Ensure Security:** Implement robust security measures to protect member information, ensuring data privacy and confidentiality.

ALGORITHM

- First we need to have the proper jdk and ide for the purpose of coding and eclipse is used for the same. Mysql database is used for the maintenance of data that is being stored in the library for example the books the students and so on.

- Now, we need to connect java and the database which is Mysql for the purpose of showing the data in a proper format and also for accessing the data. For this we use an appropriate jdbc connector and in this case it is mysql-connector-j-8.1.0.jar.

- Now we create the application using java swing modules and for this purpose the window builder library is installed.

- Now, the consequent classes and the connection between the data in mysql and the main program is being done.

- **Initialize Employee Database:**
  - **Create a database to store employee information.**
  - **Define data fields for each employee, such as ID, name, contact details, job title, department, etc.**

- **Add Employee:Collect**
  - **employee details (ID, name, contact information, etc.).**
  - **Verify that the employee does not already exist in the system.**
  - **Add the new employee to the database.**

- **Update Employee Information:**
  - **Identify the employee to be updated.**
  - **Display current information and provide options for modification (name, contact details, etc.).**
  - **Update the database with the modified information.**

- **Remove Employee:**
  - **Identify the employee to be removed.**
  - **Confirm the deletion with appropriate permissions.**
  - **Remove the employee from the database.**

LITERATURE REVIEW

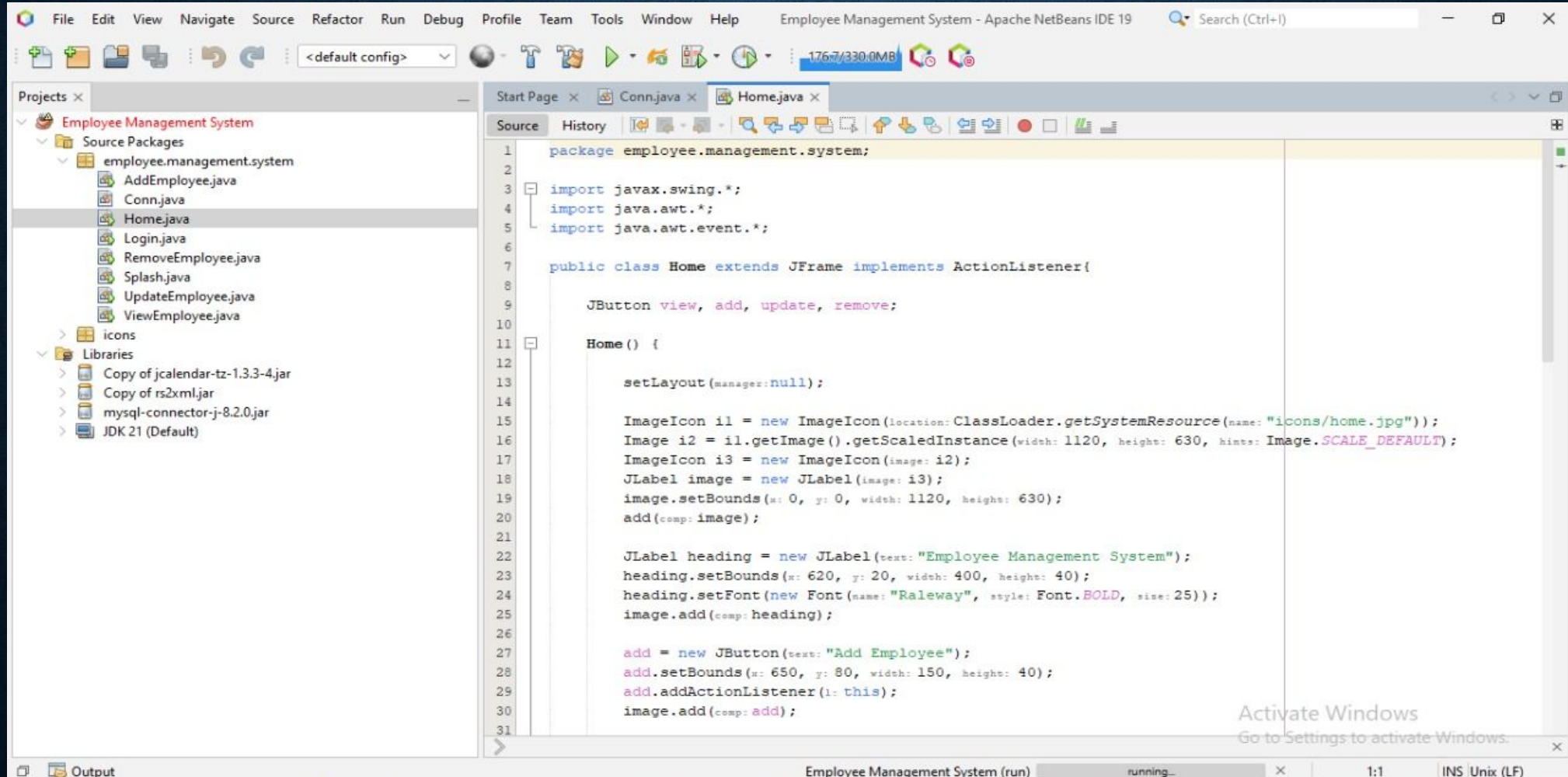| S.No	Author	Title	Inference	Drawback
1.	Ferdianto; Lius Steven Sanjaya; Titan	**Design of Employee Management Application for Small Medium Enterprise**	The research seeks to develop an application for SMEs, employing SDLC, literature review, and interviews, to streamline employee management tasks and enhance SME owners' access to data and attendance monitoring.	One potential drawback is that the implementation of the information technology system may face resistance or challenges in terms of employee adaptation and initial integration into existing company processes.
2.	Mayur Dange, Ashwini Bahadure, Priti Rathod, Rani Kanhekar, Rohini Guldagad, Prof.P.R.Patil	**Advance Employee Management System**	The project aims to streamline and centralize human resources management by developing an advanced employee management system with features catering to both managers and employees, enhancing efficiency and accessibility in handling various HR-related tasks.	One potential drawback is the initial cost and time investment required for implementing and training personnel on the advanced employee management system, which could pose challenges for smaller companies with limited resources.
3.	Ipshita Pal; Rohan Basak; Aprataikya Bandyopadayay; Titlii Ghosh; Mili Sarkar	**Design of a employee management system**	It aim for the collection of data in sequence order,user friendly	The draw is that it has many bugs ,which can be easily hacked and it doesn't print the employee informarion

ACTIVITY DIAGRAM
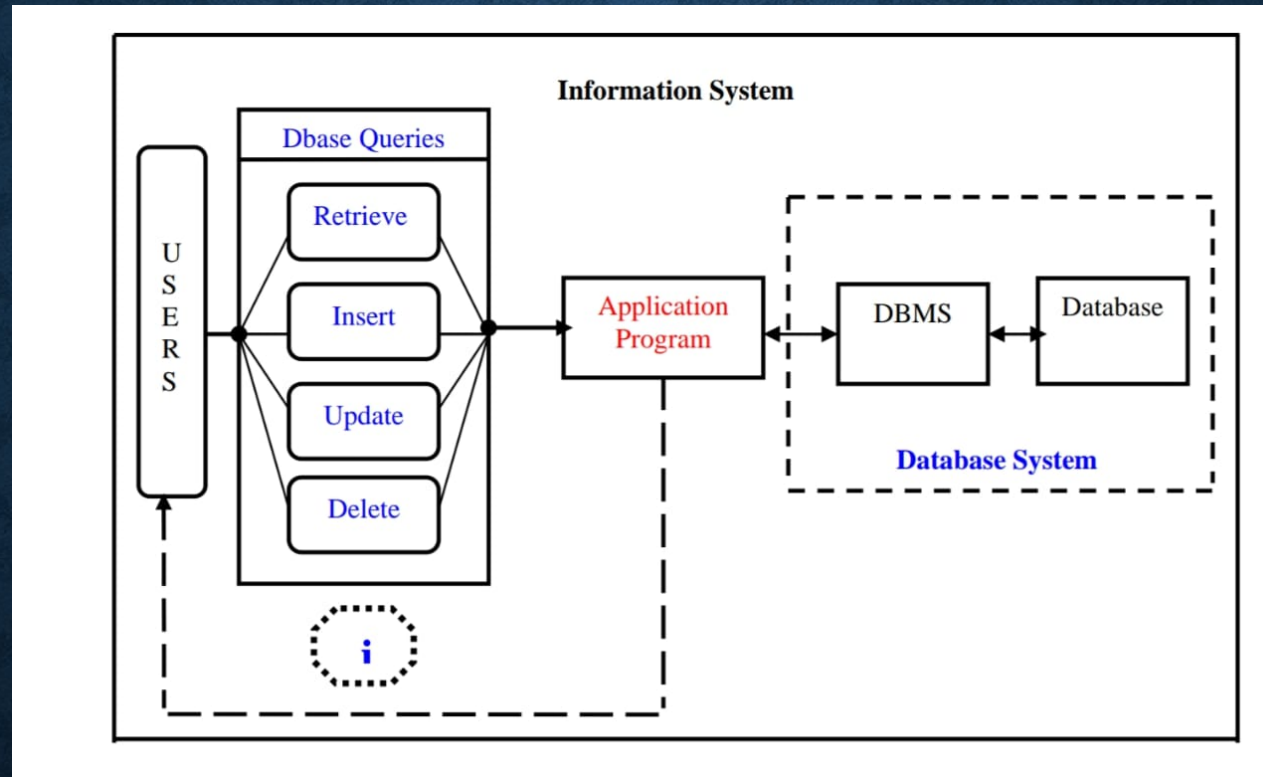
MODULES DESCRIPTION

- Login frame – This is the login frame of this system where user have to enter the required credentials to have access for the main dashboard.

- Main Dashboard – After login in, user is directed to the main dashboard of this system where user can perform various operations like adding an employee, deleting an employee

- Add employee – Here user have to enter all the required credentials to add a new employee to the system.

- Remove employee – User has to enter the employee id in order to delete his information from the system.

- View and update employee – In order to view and update employee information, the user have to enter employee ID.

IMPLEMENTATION

```java
31
32          view = new JButton(text: "View Employees");
33          view.setBounds(x: 820, y: 80, width: 150, height: 40);
            view.addActionListener(l: this);
35          image.add(comp: view);
36
37          update = new JButton(text: "Update Employee");
38          update.setBounds(x: 650, y: 140, width: 150, height: 40);
            update.addActionListener(l: this);
40          image.add(comp: update);
41
42          remove = new JButton(text: "Remove Employee");
43          remove.setBounds(x: 820, y: 140, width: 150, height: 40);
            remove.addActionListener(l: this);
45          image.add(comp: remove);
46
47          setSize(width: 1120, height: 630);
48          setLocation(x: 250, y: 100);
49          setVisible(b: true);
50      }
51
52      public void actionPerformed(ActionEvent ae) {
53          if (ae.getSource() == add) {
54              setVisible(b: false);
                new AddEmployee();
56          } else if (ae.getSource() == view) {
57              setVisible(b: false);
                new ViewEmployee();
59          } else if (ae.getSource() == update) {
60              setVisible(b: false);
                new ViewEmployee();
```

287.0/372.0MB

Employee Management System
  Source Packages
    employee.management.system
      AddEmployee.java
      Conn.java
      Home.java
      Login.java
      RemoveEmployee.java
      Splash.java
      UpdateEmployee.java
      ViewEmployee.java
    icons
  Libraries
    Copy of jcalendar-tz-1.3.3-4.jar
    Copy of rs2xml.jar
    mysql-connector-j-8.2.0.jar
    JDK 21 (Default)

Source   History

```java
 8
 9    public class AddEmployee extends JFrame implements ActionListener{
10
11        Random ran = new Random();
12        int number = ran.nextInt(bound: 999999);
13
14        JTextField tfname, tffname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
15        JDateChooser dcdob;
16        JComboBox cbeducation;
17        JLabel lblempId;
18        JButton add, back;
19
20        AddEmployee() {
21            getContentPane().setBackground(c: Color.WHITE);
22            setLayout(manager: null);
23
24            JLabel heading = new JLabel(text: "Add Employee Detail");
25            heading.setBounds(x: 320, y: 30, width: 500, height: 50);
26            heading.setFont(new Font(name: "SAN_SERIF", style: Font.BOLD, size: 25));
27            add(comp: heading);
28
29            JLabel labelname = new JLabel(text: "Name");
30            labelname.setBounds(x: 50, y: 150, width: 150, height: 30);
31            labelname.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
32            add(comp: labelname);
33
34            tfname = new JTextField();
35            tfname.setBounds(x: 200, y: 150, width: 150, height: 30);
36            add(comp: tfname);
37
38            JLabel labelfname = new JLabel(text: "Father's Name");
```

Activate Windows
Go to Settings to activate Windows.

```java
42          remove = new JButton(text:"Remove Employee");
43          remove.setBounds(x: 820, y: 140, width: 150, height: 40);
44          remove.addActionListener(l: this);
45          image.add(comp: remove);
46
47          setSize(width: 1120, height: 630);
48          setLocation(x: 250, y: 100);
49          setVisible(b: true);
50      }
51
52      public void actionPerformed(ActionEvent ae) {
53          if (ae.getSource() == add) {
54              setVisible(b: false);
55              new AddEmployee();
56          } else if (ae.getSource() == view) {
57              setVisible(b: false);
58              new ViewEmployee();
59          } else if (ae.getSource() == update) {
60              setVisible(b: false);
61              new ViewEmployee();
62          } else {
63              setVisible(b: false);
64              new RemoveEmployee();
65          }
66      }
67
68      public static void main(String[] args) {
69          new Home();
70      }
71  }
72
```

Projects ×

Start Page ×   Conn.java ×   Home.java ×   Login.java ×

Source   History

▼ Employee Management System
  ▼ Source Packages
    ▼ employee.management.system
        AddEmployee.java
        Conn.java
        Home.java
        Login.java
        RemoveEmployee.java
        Splash.java
        UpdateEmployee.java
        ViewEmployee.java
    ▶ icons
  ▼ Libraries
    ▶ Copy of jcalendar-tz-1.3.3-4.jar
    ▶ Copy of rs2xml.jar
    ▶ mysql-connector-j-8.2.0.jar
    ▶ JDK 21 (Default)

```java
1   package employee.management.system;
2
3   import javax.swing.*;
4   import java.awt.*;
5   import java.awt.event.*;
6   import java.sql.*;
7
8   public class Login extends JFrame implements ActionListener{
9
10      JTextField tfusername, tfpassword;
11
12      Login() {
13
14          getContentPane().setBackground(c: Color.WHITE);
15          setLayout(manager: null);
16
17          JLabel lblusername = new JLabel(text: "Username");
18          lblusername.setBounds(x: 40, y: 20, width: 100, height: 30);
19          add(comp: lblusername);
20
21          tfusername = new JTextField();
22          tfusername.setBounds(x: 150, y: 20, width: 150, height: 30);
23          add(comp: tfusername);
24
25          JLabel lblpassword = new JLabel(text: "Password");
26          lblpassword.setBounds(x: 40, y: 70, width: 100, height: 30);
27          add(comp: lblpassword);
28
29          tfpassword = new JTextField();
30          tfpassword.setBounds(x: 150, y: 70, width: 150, height: 30);
31          add(comp: tfpassword);
```

employee.management.system.Login >   Login >

Output

Employee Management System (run)   running...   12:14   INS

Projects ×

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×

Employee Management System
  Source Packages
    employee.management.system
      AddEmployee.java
      Conn.java
      Home.java
      Login.java
      RemoveEmployee.java
      Splash.java
      UpdateEmployee.java
      ViewEmployee.java
    icons
  Libraries
    Copy of jcalendar-tz-1.3.3-4.jar
    Copy of rs2xml.jar
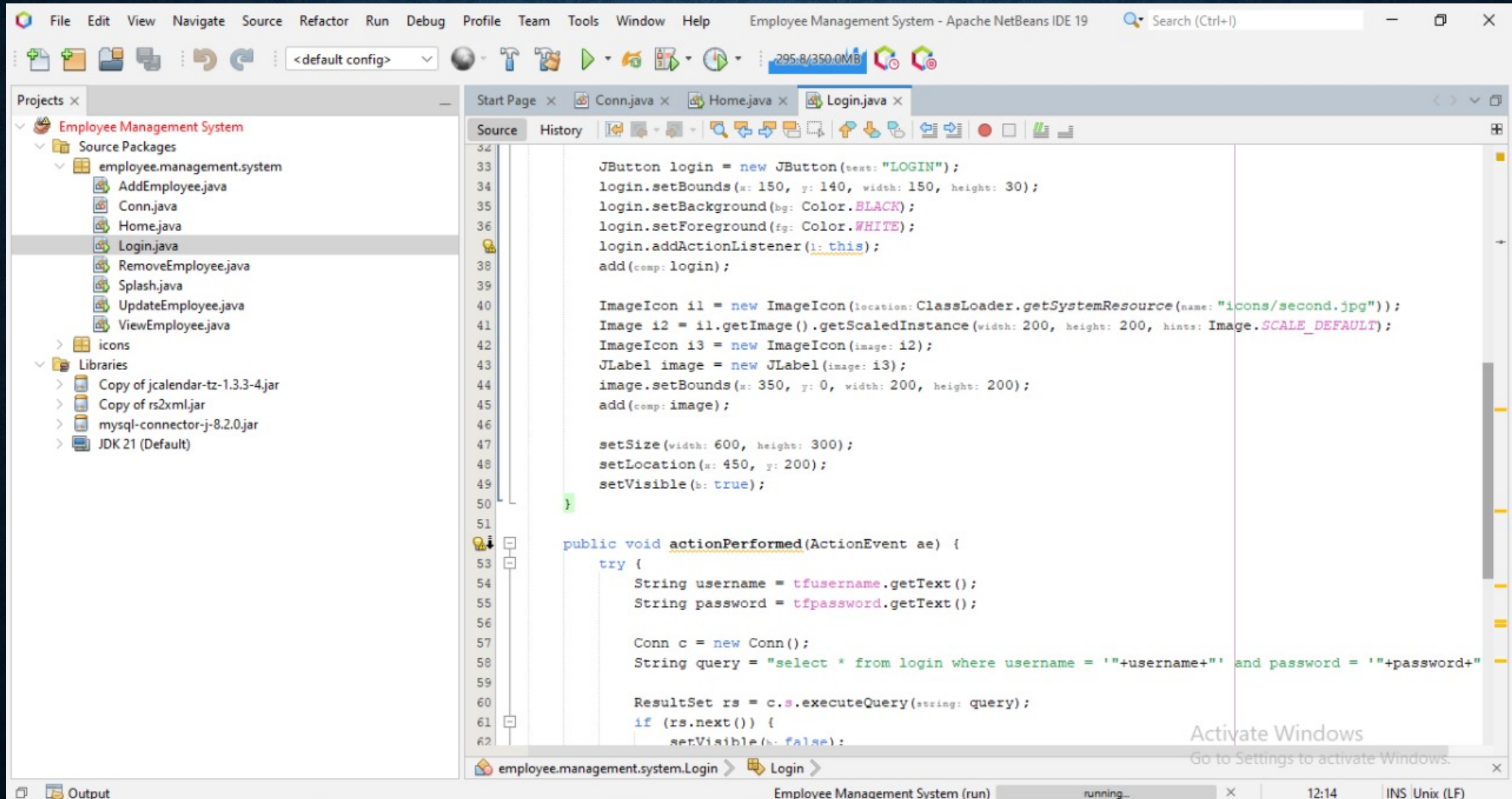   mysql-connector-j-8.2.0.jar
    JDK 21 (Default)

Source   History

```java
48        labeldob.setBounds(x: 50, y: 200, width: 150, height: 30);
49        labeldob.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
50        add(comp: labeldob);
51
52        dcdob = new JDateChooser();
53        dcdob.setBounds(x: 200, y: 200, width: 150, height: 30);
54        add(comp: dcdob);
55
56        JLabel labelsalary = new JLabel(text: "Salary");
57        labelsalary.setBounds(x: 400, y: 200, width: 150, height: 30);
58        labelsalary.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
59        add(comp: labelsalary);
60
61        tfsalary = new JTextField();
62        tfsalary.setBounds(x: 600, y: 200, width: 150, height: 30);
63        add(comp: tfsalary);
64
65        JLabel labeladdress = new JLabel(text: "Address");
66        labeladdress.setBounds(x: 50, y: 250, width: 150, height: 30);
67        labeladdress.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
68        add(comp: labeladdress);
69
70        tfaddress = new JTextField();
71        tfaddress.setBounds(x: 200, y: 250, width: 150, height: 30);
72        add(comp: tfaddress);
73
74        JLabel labelphone = new JLabel(text: "Phone");
75        labelphone.setBounds(x: 400, y: 250, width: 150, height: 30);
76        labelphone.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
77        add(comp: labelphone);
78
```

Output   Employee Management System (run)   running...   1:1   INS   Unix (LF)

Source   History

```
32
33          JButton login = new JButton(text: "LOGIN");
34          login.setBounds(x: 150, y: 140, width: 150, height: 30);
35          login.setBackground(bg: Color.BLACK);
36          login.setForeground(fg: Color.WHITE);
            login.addActionListener(l: this);
38          add(comp: login);
39
40          ImageIcon i1 = new ImageIcon(location: ClassLoader.getSystemResource(name: "icons/second.jpg"));
41          Image i2 = i1.getImage().getScaledInstance(width: 200, height: 200, hints: Image.SCALE_DEFAULT);
42          ImageIcon i3 = new ImageIcon(image: i2);
43          JLabel image = new JLabel(image: i3);
44          image.setBounds(x: 350, y: 0, width: 200, height: 200);
45          add(comp: image);
46
47          setSize(width: 600, height: 300);
48          setLocation(x: 450, y: 200);
49          setVisible(b: true);
50      }
51
52      public void actionPerformed(ActionEvent ae) {
53          try {
54              String username = tfusername.getText();
55              String password = tfpassword.getText();
56
57              Conn c = new Conn();
58              String query = "select * from login where username = '"+username+"' and password = '"+password+"
59
60              ResultSet rs = c.s.executeQuery(string: query);
61              if (rs.next()) {
62                  setVisible(b: false);
```

<default config>

250.4/384.0MB

Start Page × 📄 Conn.java × 📄 Home.java × 📄 Login.java × 📄 AddEmployee.java × 📄 ViewEmployee.java ×

Source History

- 🍰 Employee Management System
  - 📁 Source Packages
    - 📦 employee.management.system
      - 📄 AddEmployee.java
      - 📄 Conn.java
      - 📄 Home.java
      - 📄 Login.java
      - 📄 RemoveEmployee.java
      - 📄 Splash.java
      - 📄 UpdateEmployee.java
      - 📄 ViewEmployee.java
    - 📁 icons
  - 📁 Libraries
    - 📄 Copy of jcalendar-tz-1.3.3-4.jar
    - 📄 Copy of rs2xml.jar
    - 📄 mysql-connector-j-8.2.0.jar
    - 📄 JDK 21 (Default)

```java
  8
  9     public class ViewEmployee extends JFrame implements ActionListener{
 10
 11         JTable table;
 12         Choice cemployeeId;
 13         JButton search, print, update, back;
 14
 15         ViewEmployee() {
 16
 17             getContentPane().setBackground(c: Color.WHITE);
 18             setLayout(manager: null);
 19
 20             JLabel searchlbl = new JLabel(text: "Search by Employee Id");
 21             searchlbl.setBounds(x: 20, y: 20, width: 150, height: 20);
 22             add(comp: searchlbl);
 23
 24             cemployeeId = new Choice();
 25             cemployeeId.setBounds(x: 180, y: 20, width: 150, height: 20);
 26             add(comp: cemployeeId);
 27
 28             try {
 29                 Conn c = new Conn();
 30                 ResultSet rs = c.s.executeQuery(string: "select * from employee");
 31                 while(rs.next()) {
 32                     cemployeeId.add(item: rs.getString(string: "empId"));
 33                 }
 34             } catch (Exception e) {
 35                 e.printStackTrace();
 36             }
 37
 38             table = new JTable();
```

Projects ×

- Employee Management System
  - Source Packages
    - employee.management.system
      - AddEmployee.java
      - Conn.java
      - Home.java
      - Login.java
      - RemoveEmployee.java
      - Splash.java
      - UpdateEmployee.java
      - ViewEmployee.java
    - icons
  - Libraries
    - Copy of jcalendar-tz-1.3.3-4.jar
    - Copy of rs2xml.jar
    - mysql-connector-j-8.2.0.jar
    - JDK 21 (Default)

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×

Source   History

```
134            add.setBackground(bg: Color.BLACK);
135            add.setForeground(fg: Color.WHITE);
136            add(comp: add);
137
138            back = new JButton(text: "Back");
139            back.setBounds(x: 450, y: 550, width: 150, height: 40);
               back.addActionListener(l: this);
141            back.setBackground(bg: Color.BLACK);
142            back.setForeground(fg: Color.WHITE);
143            add(comp: back);
144
145            setSize(width: 900, height: 700);
146            setLocation(x: 300, y: 50);
147            setVisible(b: true);
148        }
149
150        public void actionPerformed(ActionEvent ae) {
151            if (ae.getSource() == add) {
152                String name = tfname.getText();
153                String fname = tffname.getText();
154                String dob = ((JTextField) dcdob.getDateEditor().getUiComponent()).getText();
155                String salary = tfsalary.getText();
156                String address = tfaddress.getText();
157                String phone = tfphone.getText();
158                String email = tfemail.getText();
159                String education = (String) cbeducation.getSelectedItem();
160                String designation = tfdesignation.getText();
161                String aadhar = tfaadhar.getText();
162                String empId = lblempId.getText();
163
```

Output   Employee Management System (run)   running...   1:1   INS Unix (LF)

File   Edit   View   Navigate   Source   Refactor   Run   Debug   Profile   Team   Tools   Window   Help

Search (Ctrl+I)

<default config>

321.3/384.0MB

Projects ×

- Employee Management System
  - Source Packages
    - employee.management.system
      - AddEmployee.java
      - Conn.java
      - Home.java
      - Login.java
      - RemoveEmployee.java
      - Splash.java
      - UpdateEmployee.java
      - ViewEmployee.java
    - icons
  - Libraries
    - Copy of jcalendar-tz-1.3.3-4.jar
    - Copy of rs2xml.jar
    - mysql-connector-j-8.2.0.jar
    - JDK 21 (Default)

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×   ViewEmployee.java ×

Source   History

```java
        } catch (Exception e) {
            e.printStackTrace();
36      }
37
38      table = new JTable();
39
40      try {
41          Conn c = new Conn();
42          ResultSet rs = c.s.executeQuery(string: "select * from employee");
43          table.setModel(dataModel: DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            e.printStackTrace();
46      }
47
48      JScrollPane jsp = new JScrollPane(view: table);
49      jsp.setBounds(x: 0, y: 100, width: 900, height: 600);
50      add(comp: jsp);
51
52      search = new JButton(text: "Search");
53      search.setBounds(x: 20, y: 70, width: 80, height: 20);
        search.addActionListener(l: this);
55      add(comp: search);
56
57      print = new JButton(text: "Print");
58      print.setBounds(x: 120, y: 70, width: 80, height: 20);
        print.addActionListener(l: this);
60      add(comp: print);
61
62      update = new JButton(text: "Update");
63      update.setBounds(x: 220, y: 70, width: 80, height: 20);
```

Output

Employee Management System (run)   running...

1:1   INS   Unix (LF)

Activate Windows
Go to Settings to activate Windows.

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×   ViewEmployee.java ×

Source   History

```java
55          search.addActionListener(l: this);
            add(comp: search);

56
57          print = new JButton(text: "Print");
58          print.setBounds(x: 120, y: 70, width: 80, height: 20);
            print.addActionListener(l: this);
60          add(comp: print);

61
62          update = new JButton(text: "Update");
63          update.setBounds(x: 220, y: 70, width: 80, height: 20);
            update.addActionListener(l: this);
65          add(comp: update);

66
67          back = new JButton(text: "Back");
68          back.setBounds(x: 320, y: 70, width: 80, height: 20);
            back.addActionListener(l: this);
70          add(comp: back);

71
72          setSize(width: 900, height: 700);
73          setLocation(x: 300, y: 100);
74          setVisible(b: true);
75      }

76
77      public void actionPerformed(ActionEvent ae) {
78          if (ae.getSource() == search) {
79              String query = "select * from employee where empId = '"+cemployeeId.getSelectedItem()+"'";
80              try {
81                  Conn c = new Conn();
82                  ResultSet rs = c.s.executeQuery(string: query);
83                  table.setModel(dataModel: DbUtils.resultSetToTableModel(rs));
                } catch (Exception e) {
```

employee.management.system.ViewEmployee   ViewEmployee

273.4/372.0MB

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×

Source   History

Employee Management System
  Source Packages
    employee.management.system
      AddEmployee.java
      Conn.java
      Home.java
      Login.java
      RemoveEmployee.java
      Splash.java
      UpdateEmployee.java
      ViewEmployee.java
    icons
  Libraries
    Copy of jcalendar-tz-1.3.3-4.jar
    Copy of rs2xml.jar
    mysql-connector-j-8.2.0.jar
    JDK 21 (Default)

```
103        JLabel labeldesignation = new JLabel(text: "Designation");
104        labeldesignation.setBounds(x: 50, y: 350, width: 150, height: 30);
105        labeldesignation.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
106        add(comp: labeldesignation);
107
108        tfdesignation = new JTextField();
109        tfdesignation.setBounds(x: 200, y: 350, width: 150, height: 30);
110        add(comp: tfdesignation);
111
112        JLabel labelaadhar = new JLabel(text: "Aadhar Number");
113        labelaadhar.setBounds(x: 400, y: 350, width: 150, height: 30);
114        labelaadhar.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
115        add(comp: labelaadhar);
116
117        tfaadhar = new JTextField();
118        tfaadhar.setBounds(x: 600, y: 350, width: 150, height: 30);
119        add(comp: tfaadhar);
120
121        JLabel labelempId = new JLabel(text: "Employee id");
122        labelempId.setBounds(x: 50, y: 400, width: 150, height: 30);
123        labelempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
124        add(comp: labelempId);
125
126        lblempId = new JLabel("" + number);
127        lblempId.setBounds(x: 200, y: 400, width: 150, height: 30);
128        lblempId.setFont(new Font(name: "serif", style: Font.PLAIN, size: 20));
129        add(comp: lblempId);
130
131        add = new JButton(text: "Add Details");
132        add.setBounds(x: 250, y: 550, width: 150, height: 40);
```

Projects ×

- Employee Management System
  - Source Packages
    - employee.management.system
      - AddEmployee.java
      - Conn.java
      - Home.java
      - Login.java
      - RemoveEmployee.java
      - Splash.java
      - UpdateEmployee.java
      - ViewEmployee.java
    - icons
  - Libraries
    - Copy of jcalendar-tz-1.3.3-4.jar
    - Copy of rs2xml.jar
    - mysql-connector-j-8.2.0.jar
    - JDK 21 (Default)

Start Page ×   Conn.java ×   Home.java ×   Login.java ×   AddEmployee.java ×   ViewEmployee.java ×

Source   History

```java
73              setLocation(x: 300, y: 100);
74              setVisible(b: true);
75          }
76
77      public void actionPerformed(ActionEvent ae) {
78          if (ae.getSource() == search) {
79              String query = "select * from employee where empId = '"+cemployeeId.getSelectedItem()+"'";
80              try {
81                  Conn c = new Conn();
82                  ResultSet rs = c.s.executeQuery(string: query);
83                  table.setModel(dataModel: DbUtils.resultSetToTableModel(rs));
84              } catch (Exception e) {
85                  e.printStackTrace();
86              }
87          } else if (ae.getSource() == print) {
88              try {
89                  table.print();
90              } catch (Exception e) {
91                  e.printStackTrace();
92              }
93          } else if (ae.getSource() == update) {
94              setVisible(b: false);
95              new UpdateEmployee(empId: cemployeeId.getSelectedItem());
96          } else {
97              setVisible(b: false);
98              new Home();
99          }
100         }
101
102     public static void main(String[] args) {
            new ViewEmployee();
```

employee.management.system.ViewEmployee >   ViewEmployee >

Output   Employee Management System (run)   running...   47:9/4:52   INS   Unix (LF)

RESULT

Username admin

Password 12345

LOGIN

Employee Management System

Add Employee   View Employees

Update Employee   Remove Employee

Activate Windows
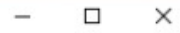
Add Employee Detail

Name

Father's Name

Date of Birth

Salary

Address

Phone

Email

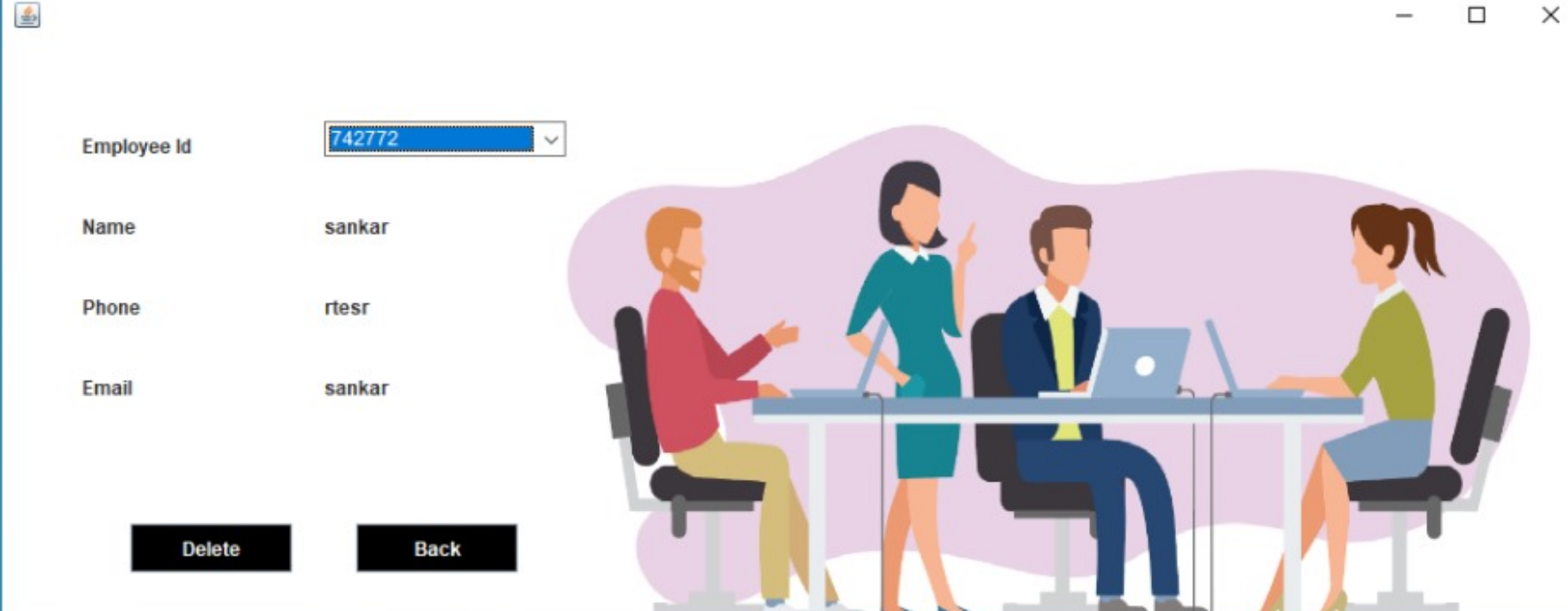Higest Education   BBA

Designation

Aadhar Number

Employee id   882375

Add Details

Back

Search by Employee Id   742772

| Search | Print | Update | Back |

name	fname	dob	salary	address	phone	email	education	designation	aadhar	empId
sankar	eresr	Feb 19, 2005	etresr	chennai	rtesr	sankar	BTech	jtdtrhhd	aresrtgesr	742772

Update Employee Detail

Name	**sankar**	Father's Name	eresr
Date of Birth	**Feb 19, 2005**	Salary	etresr
Address	chennai	Phone	rtesr
Email	sankar	Higest Education	BTech
Designation	jtdtrhhd	Aadhar Number	**aresrtgesr**
Employee id	**742772**		

Update Details   **Back**

THANK YOU