
EMPLOYEE MANAGEMENT SYSTEM

Created

By

MUKESH PRAJAPATI

Acknowledgement

It is high privilege for me to express my deep sense of gratitude to those entire faculty Members who helped me in the completion of the project, specially my internal guide Mr. VIVEK KRISHNA who was always there at hour of need.

My special thanks to all other faculty members, Batch mate & friends for helping me in the completion of project work and its report submission.

Secondly, I would also like to thank my friends who helped me a lot in finalizing this project within the limited time frame.

TABLE OF FIGURES

Figure No.	Caption of the Figure	Page No.
1	Login frame	5
2	Main Dashboard	6
3	Add employee	7
4	Remove employee	8
5	View and Update employee	9

I. Introduction

Everything has been digitised in our age of ever-increasing technology. The human workforce has grown as a result of the abundance of job options. As a result, a system that can handle the data of such a vast number of people in a company is required. Because of its user-friendly design, this project makes the process of keeping records easier. The "EMPLOYEE MANAGEMENT SYSTEM" was created to address the issues that plagued the previous manual system. This programme is designed to eliminate, and in some cases, decrease, the problems that the current system has.

To eliminate data entry mistakes, the software is kept as simple as possible. When inputting incorrect data, it also displays an error notice. The user doesn't require any formal expertise to operate this system. The admin will be able to add new employees to this project. Employee data may also be seen and printed by the administrator. Admins can also remove an employee and change their details.

III. Methodology

The methodology to complete this project is as follows:

1. I explored net beans, concepts of swings and applets.
2. For further and a deeper understanding, I even referred to some articles, books, journals, websites and news articles.

Below are the important concepts on which the work has been done and with the support of these I was able to work on my project.

NET BEANS- NetBeans is a Java-based integrated development environment (IDE). Eclipse enables the creation of applications using a set of modular software components known as modules. Eclipse is compatible with Windows, Mac OS X, Linux, and Solaris. It also allows other programming languages to be extended. In addition to Java programming, Third-party developers can expand Eclipse-based applications, including the Eclipse IDE.

JAVA- High-level, Object-Oriented programming language which help programmers to run their applications efficiently. JAVA is the programming language which comes into our minds when we talk about android application. By using JAVA as a programming language, programmer can develop any type of android application easily. JAVA also provides many

libraries which also helps in making efficient android application. Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC), which provides an API for creating graphical-user- interfaces for Java programmes.

SWING- Swing is a Java GUI widget toolkit. It's part of Oracle's Java Foundation Classes (JFC), which provides an API for creating-graphical-user-interfaces for Java programmes. Swing was created to give a more advanced collection of graphical user interface components than the previous Abstract Window Toolkit (AWT). Swing offers a pluggable look and feel that allows applications to have alook &feel that is unconnected to the underlying platform, as well as a look & feel that emulates the look & feel of numerous platforms.

SQL- SQL (Structured Query Language) is a computer language that is used to manage data in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS). It's especially beneficial for dealing with structured data, or data that has relationships between entities and variables

Chapter-5: Software/ Hardware Requirements and specifications

The Software and hardware requirements for an employee management system (ECM) in java will vary depending on the specific features and functionality of the system. However , some general requirements include:

Hardware Requirements:

- Processor: Intel Pentium 4 or higher
- Hard Disk: 1GB or more
- Ram: 512 MB or more

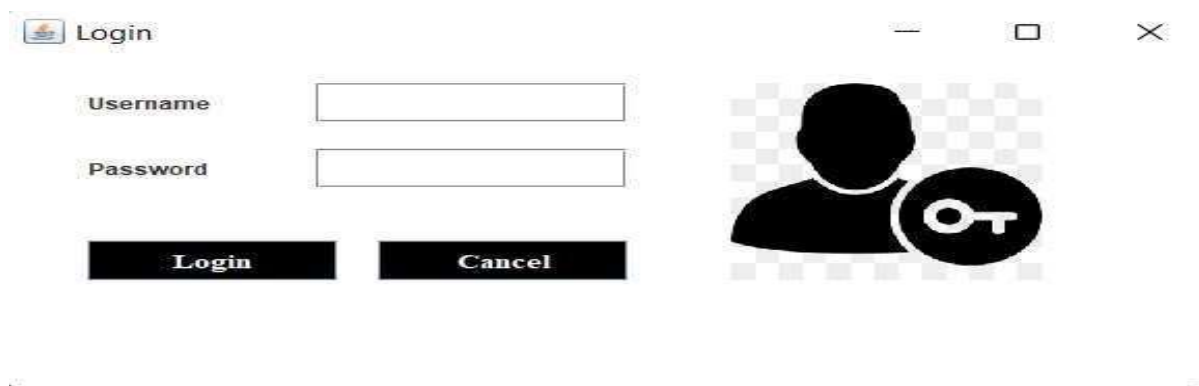
Software Requirements:

- Java Development Kit (JDK)
- Java Swing and AWT for GUI development
- SQL database knowledge and SQL query skills
- Understanding of JAR files in Java Projects
- A Java IDE, such as NET-BEANS

8. IMPLEMENTATION AND RESULT

Following are the screens of the Employee Management System where you can see all the features of this system in use and you can also see the GUI of the system:

1. **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.



Figr 1.

2. 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



Fig. 2

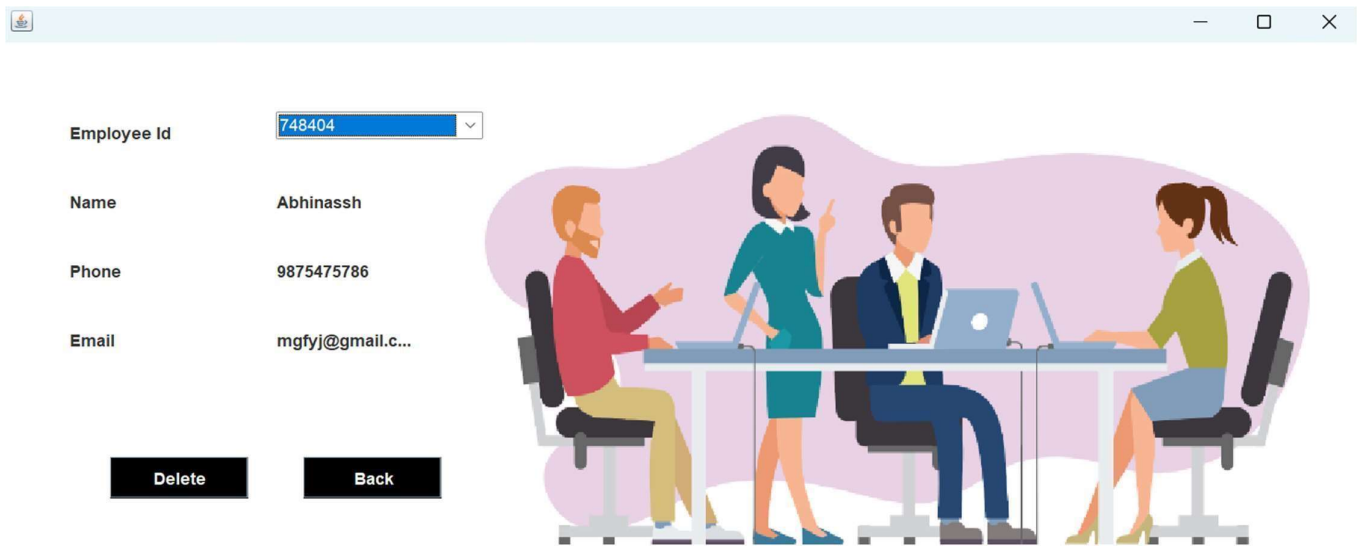
3. Add employee – Here user have to enter all the required credentials to add a new employee to the system.

Add Employee Detail

Name	<input type="text"/>	Father's Name	<input type="text"/>
Date of Birth	<input type="text"/>	Salary	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Email	<input type="text"/>	Highest Education	<input type="text" value="BBA"/>
Designation	<input type="text"/>	Aadhar Number	<input type="text"/>
Employee id	421483		

Figr 3.

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



Employee Id: 748404

Name: Abhinassh

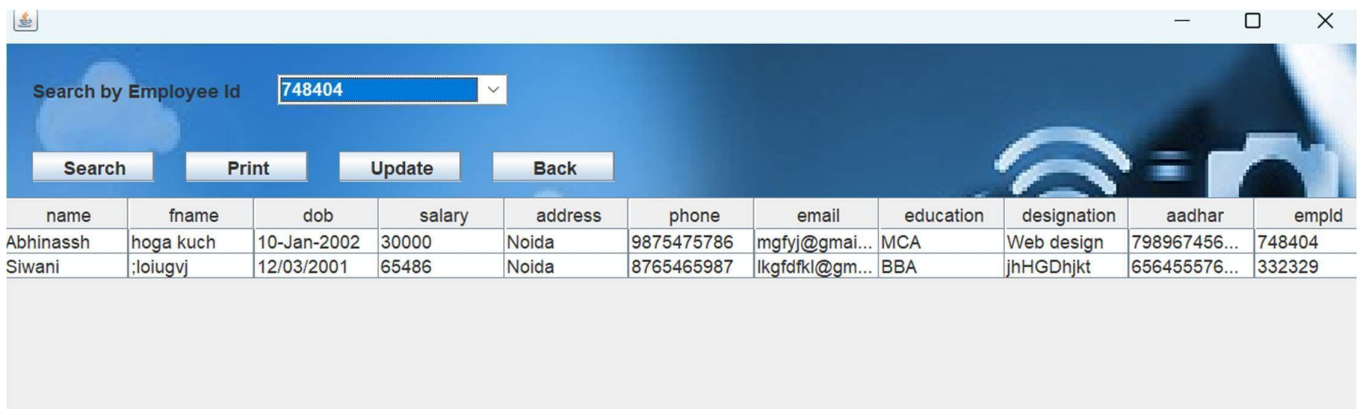
Phone: 9875475786

Email: mgfyj@gmail.c...

Delete **Back**

Figr 4.

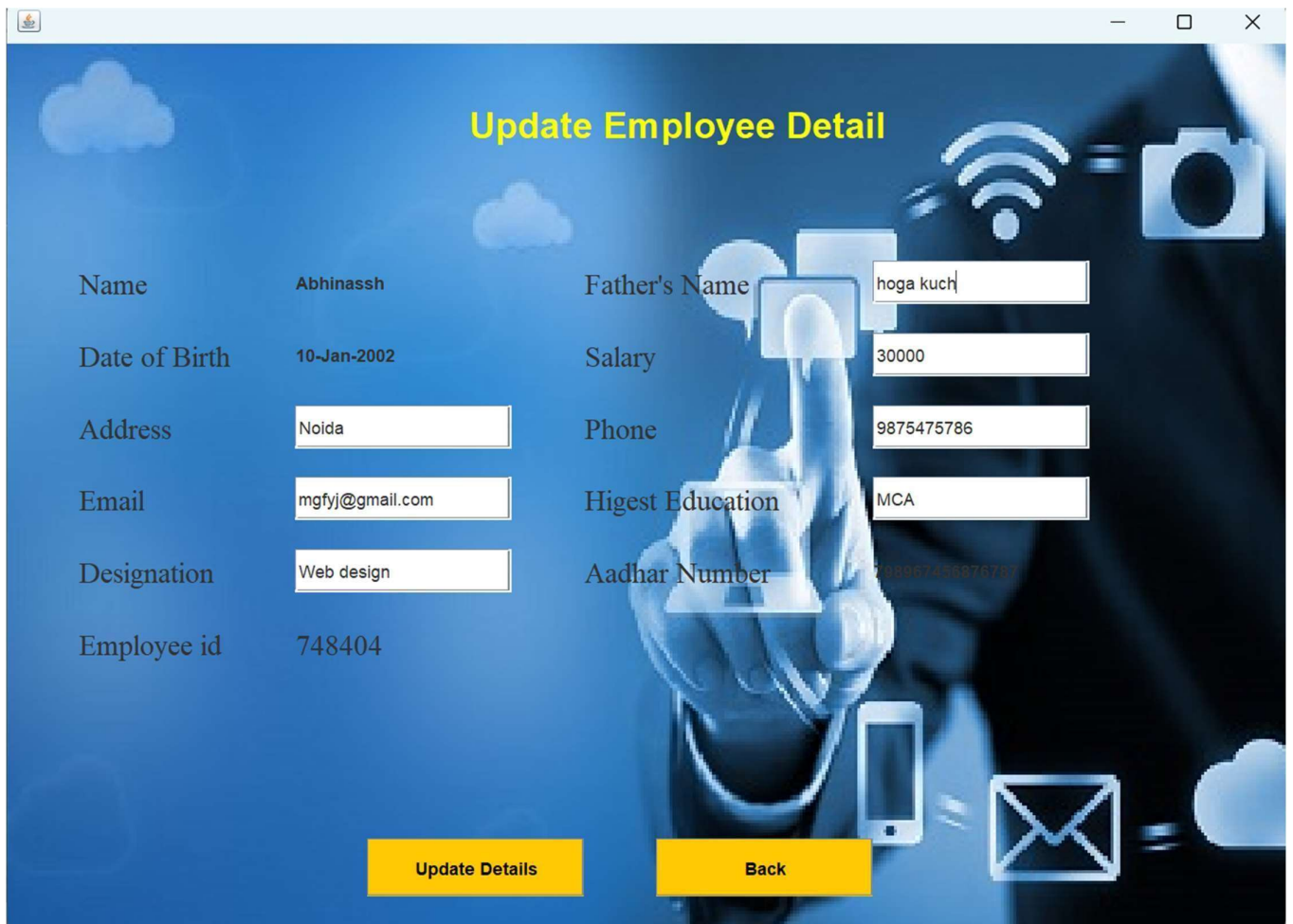
5.View and update employee – In order to view and update employee information, the user have to enter employee ID.



Search by Employee Id: 748404

Search **Print** **Update** **Back**

name	fname	dob	salary	address	phone	email	education	designation	aadhar	empld
Abhinassh	hoga kuch	10-Jan-2002	30000	Noida	9875475786	mgfyj@gmail...	MCA	Web design	798967456...	748404
Siwani	:loiugvj	12/03/2001	65486	Noida	8765465987	lkghdfkl@gm...	BBA	jhHGDhjkt	656455576...	332329



The image shows a web application window titled "Update Employee Detail". The background is a blue gradient with icons of clouds, a Wi-Fi signal, a camera, a hand pointing at a screen, a smartphone, and an envelope. The form contains two columns of input fields. The left column has fields for Name, Date of Birth, Address, Email, Designation, and Employee id. The right column has fields for Father's Name, Salary, Phone, Highest Education, and Aadhar Number. At the bottom, there are two yellow buttons: "Update Details" and "Back".

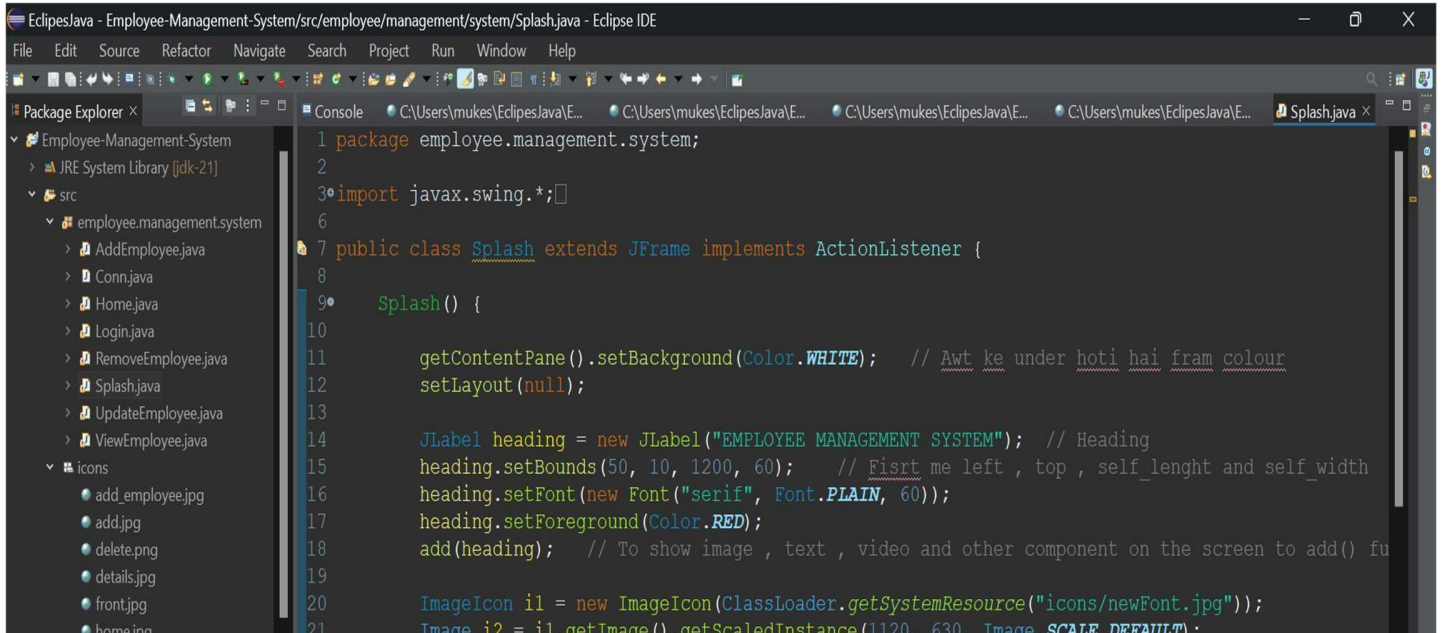
Field	Value
Name	Abhinassh
Date of Birth	10-Jan-2002
Address	Noida
Email	mgfyj@gmail.com
Designation	Web design
Employee id	748404
Father's Name	hoga kuch
Salary	30000
Phone	9875475786
Highest Education	MCA
Aadhar Number	748967448876787

Update Details **Back**

Fig. 5

9. Source code:-

1. File Name in Eclipse



2. Main Java File (Splash.java)

```
package employee.management.system;
```

```
import javax.swing.*; import
```

```
java.awt.*;
```

```
import java.awt.event.*;
```

```
public class Splash extends JFrame implements ActionListener {
```

```
    Splash() {
```

```
        getContentPane().setBackground(Color.WHITE); // Awt ke under hoti hai fram colour
        setLayout(null);
```

```
        JLabel heading = new JLabel("EMPLOYEE MANAGEMENT SYSTEM"); // Heading
        heading.setBounds(50, 10, 1200, 60); // Fisrt me left , top , self_lenght and self_width
        heading.setFont(new Font("serif", Font.PLAIN, 60));
        heading.setForeground(Color.RED);
```

```
        add(heading); // To show image , text , video and other component on the screen to add()
        function use
```

```
        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/newFont.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
```

10.

```
image.setBounds(0, 0, 1120, 630);  
add(image);
```

```
    JButton clickhere = new JButton("CLICK HERE TO CONTINUE");  
clickhere.setBounds(400, 400, 300, 70);  
clickhere.setBackground(Color.BLACK);  
clickhere.setForeground(Color.WHITE);  
clickhere.addActionListener(this); // To click button then action perform  
image.add(clickhere);
```

```
setSize(1100, 650); // Frame length and width  
setLocation(100, 30); // Frame open while location on the screen To left and Top locations  
setVisible(true); // any frame to visibilty bydefault false hoti hai
```

```
    while(true) {  
        heading.setVisible(false);  
try {  
            Thread.sleep(500);  
        } catch (Exception e){  
  
        }  

```

```
        heading.setVisible(true);  
try {  
            Thread.sleep(500);  
        } catch (Exception e){  
  
        }  
    }  
}
```

```
    public void actionPerformed(ActionEvent ae) {  
setVisible(false); // Current frame to close  
new Login();  
    }
```

```
    public static void main(String args[]) {  
new Splash();  
    }  
}
```

3. Login page(login.java)

```
package employee.management.system;

import javax.swing.*;
import java.awt.*; import
java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener{

    JTextField tfusername, tfpassword;

    Login() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel lblusername = new JLabel("Username");
        lblusername.setBounds(40, 20, 100, 30); // Left=40 ,
        top=20    add(lblusername);

        tfusername = new JTextField();
        tfusername.setBounds(150, 20, 150, 30);
        add(tfusername);

        JLabel lblpassword = new JLabel("Password");
        lblpassword.setBounds(40, 70, 100, 30);    add(lblpassword);

        tfpassword = new JTextField();
        tfpassword.setBounds(150, 70, 150, 30);
        add(tfpassword);

        JButton login = new JButton("LOGIN");
        login.setBounds(150, 140, 150, 30);
        login.setBackground(Color.BLACK);
        login.setForeground(Color.WHITE);
        login.addActionListener(this);
        add(login);

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/second.jpg"));
        Image i2 = i1.getImage().getScaledInstance(200, 200,
        Image.SCALE_DEFAULT);    ImageIcon i3 = new ImageIcon(i2);    JLabel
        image = new JLabel(i3);
```

```

        image.setBounds(350, 0, 200, 200);
        add(image);

        setSize(600, 300);
        setLocation(450, 200);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
    try {
        String username = tfusername.getText();
        String password = tfpassword.getText();

        Conn c = new Conn();
        String query = "select * from login where username = '"+username+"' and password = '"+password+"'";

        ResultSet rs =
c.s.executeQuery(query);        if (rs.next())
{            setVisible(false);        new
Home();
        } else {
            JOptionPane.showMessageDialog(null, "Invalid username or password");
setVisible(false);
        }
    } catch (Exception e) {
e.printStackTrace();
    }
}

    public static void main(String[] args) {
new Login();
    }
}

```

4. Home Page(Home.java)

```

package employee.management.system;

```

```

import javax.swing.*;
import java.awt.*; import
java.awt.event.*;

public class Home extends JFrame implements ActionListener{

    JButton view, add, update, remove;

    Home() {

        setLayout(null);

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/home.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(0, 0, 1120, 630);
        add(image);

        JLabel heading = new JLabel("Employee Management System");
        heading.setBounds(620, 20, 400, 40);
        heading.setFont(new Font("Raleway", Font.BOLD, 25));
        image.add(heading);

        add = new JButton("Add Employee");
        add.setBounds(650, 80, 150, 40);
        add.addActionListener(this);
        image.add(add);

        view = new JButton("View Employees");
        view.setBounds(820, 80, 150, 40);
        view.addActionListener(this);
        image.add(view);

        update = new JButton("Update Employee");
        update.setBounds(650, 140, 150, 40);
        update.addActionListener(this);
        image.add(update);

        remove = new JButton("Remove Employee");
        remove.setBounds(820, 140, 150, 40);
        remove.addActionListener(this);
        image.add(remove);
    }
}

```

```
        setSize(1120, 630);  
setLocation(100, 30);  
setVisible(true);  
}
```

```

        public void actionPerformed(ActionEvent ae) {
            if (ae.getSource() == add) {
                setVisible(false);
                new AddEmployee();
            } else if (ae.getSource() == view) {
                setVisible(false);
                new ViewEmployee();
            } else if (ae.getSource() == update) {
                setVisible(false);
                new ViewEmployee();
            } else {
                setVisible(false);
                new RemoveEmployee();
            }
        }

        public static void main(String[] args) {
            new Home();
        }
    }

```

5.AddEmployee.java

```

package employee.management.system;

import java.awt.*; import
javax.swing.*;
import com.toedter.calendar.JDateChooser;
import java.util.*; import java.awt.event.*; public class AddEmployee
extends JFrame implements ActionListener{

    Random ran = new Random();
    int number = ran.nextInt(999999);

    JTextField tfname, tfname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
    JDateChooser dcdob;
    JComboBox cbeducation;
    JLabel lblempId;
    JButton add, back;

    AddEmployee() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
    }

```



```

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/add.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(0, 0, 1120, 630);
        add(image);

```

```

        JLabel heading = new JLabel("Add Employee Detail");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD,
25));
        heading.setForeground(Color.yellow);
        image.add(heading);

```

```

        JLabel labelname = new JLabel("Name");
        labelname.setBounds(50, 150, 150, 30);
        labelname.setFont(new Font("serif", Font.PLAIN, 20));
        image.add(labelname);

```

```

        tfname = new JTextField();
        tfname.setBounds(200, 150, 150, 30);
        image.add(tfname);

```

```

        JLabel labelfname = new JLabel("Father's Name");
        labelfname.setBounds(400, 150, 150, 30);
        labelfname.setFont(new Font("serif", Font.PLAIN, 20));
        image.add(labelfname);

```

```

        tffname = new JTextField();
        tffname.setBounds(600, 150, 150, 30);
        image.add(tffname);

```

```

        JLabel labeldob = new JLabel("Date of Birth");
        labeldob.setBounds(50, 200, 150, 30);
        labeldob.setFont(new Font("serif", Font.PLAIN, 20));
        image.add(labeldob);

```

```

        dcdob = new JDateChooser();
        dcdob.setBounds(200, 200, 150, 30);
        image.add(dcdob);

```

```
JLabel labelsalary = new JLabel("Salary");
labelsalary.setBounds(400, 200, 150, 30);
labelsalary.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labelsalary);
tfsalary = new JTextField();
tfsalary.setBounds(600, 200, 150, 30);
image.add(tfsalary);
```

```
JLabel labeladdress = new JLabel("Address");
labeladdress.setBounds(50, 250, 150, 30);
labeladdress.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labeladdress);
```

```
tfaddress = new JTextField();
tfaddress.setBounds(200, 250, 150, 30);
image.add(tfaddress);
```

```
JLabel labelphone = new JLabel("Phone");
labelphone.setBounds(400, 250, 150, 30);
labelphone.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labelphone);
```

```
tfphone = new JTextField();
tfphone.setBounds(600, 250, 150, 30);
image.add(tfphone);
```

```
JLabel lalemail = new JLabel("Email");
lalemail.setBounds(50, 300, 150, 30);
lalemail.setFont(new Font("serif", Font.PLAIN, 20));
image.add(lalemail);
```

```
tfemail = new JTextField();
tfemail.setBounds(200, 300, 150, 30);
image.add(tfemail);
```

```
JLabel labeleducation = new JLabel("Higest Education");
labeleducation.setBounds(400, 300, 150, 30);
labeleducation.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labeleducation);
```

```
String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA",
"MA",
"MTech", "MSC", "PHD"};    cbeducation =
new JComboBox(courses);
```

```
cbeducation.setBackground(Color.WHITE);
cbeducation.setBounds(600, 300, 150, 30);
image.add(cbeducation);
```

```
JLabel labeldesignation = new JLabel("Designation");
labeldesignation.setBounds(50, 350, 150, 30);
labeldesignation.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labeldesignation);
```

```
tfdesignation = new JTextField();
tfdesignation.setBounds(200, 350, 150, 30);
image.add(tfdesignation);
```

```
JLabel labelaadhar = new JLabel("Aadhar Number");
labelaadhar.setBounds(400, 350, 150, 30);
labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labelaadhar);
```

```
tfaadhar = new JTextField();
tfaadhar.setBounds(600, 350, 150, 30);
image.add(tfaadhar);
```

```
JLabel labelempId = new JLabel("Employee id");
labelempId.setBounds(50, 400, 150, 30);
labelempId.setFont(new Font("serif", Font.PLAIN, 20));
image.add(labelempId);
```

```
lblempId = new JLabel("" + number);
lblempId.setBounds(200, 400, 150, 30);
lblempId.setFont(new Font("serif", Font.PLAIN, 20));
image.add(lblempId);
```

```
add = new JButton("Add Details");
add.setBounds(250, 550, 150, 40);
add.addActionListener(this);
add.setBackground(Color.BLACK);
add.setForeground(Color.WHITE);
image.add(add);
```

```
back = new JButton("Back");
back.setBounds(450, 550, 150, 40);
back.addActionListener(this);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
```

```
        image.add(back);

        setSize(900, 700);
        setLocation(300, 50);
        setVisible(true);
    }
```

```

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == add) {

        String name = tfname.getText();
        String fname = tffname.getText();
        String dob = ((JTextField) dcdob.getDateEditor().getUiComponent()).getText();
        String salary = tfsalary.getText();
        String address = tfaddress.getText();
        String phone = tfphone.getText();
        String email = tfemail.getText();
        String education = (String) cbeducation.getSelectedItem();
        String designation = tfdesignation.getText();
        String aadhar = tfaadhar.getText();
        String empId = lblempId.getText();

        try {
            Conn conn = new Conn();
            String query = "insert into employee values('"+name+"', '"+fname+"', '"+dob+"',
            '"+salary+"', '"+address+"', '"+phone+"', '"+email+"', '"+education+"', '"+designation+"', '"+aadhar+"',
            '"+empId+"')";
            conn.s.executeUpdate(query);
            JOptionPane.showMessageDialog(null, "Details added
            successfully");          setVisible(false);          new Home();          }
            catch (Exception e) {          e.printStackTrace();
            }
            } else {
                setVisible(false);
                new Home();
            }
        }

        public static void main(String[] args) {
            new AddEmployee();
        }
    }
}

```

6. UpdateEmployee.java

```
package employee.management.system;

import java.awt.*; import
javax.swing.*; import
java.awt.event.*; import
java.sql.*;
public class UpdateEmployee extends JFrame implements ActionListener{

    JTextField tfeducation, tffname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
    JLabel lblempId;
    JButton add, back;
    String empId;

    UpdateEmployee(String empId) {
        this.empId = empId;
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/view.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(0, 0, 1120, 630);
        add(image);

        JLabel heading = new JLabel("Update Employee Detail");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        heading.setForeground(Color.yellow);      image.add(heading);

        JLabel labelname = new JLabel("Name");
        labelname.setBounds(50, 150, 150, 30);
        labelname.setFont(new Font("serif", Font.PLAIN, 20));
        image.add(labelname);

        JLabel lblname = new JLabel();
        lblname.setBounds(200, 150, 150, 30);
        image.add(lblname);

        JLabel labelfname = new JLabel("Father's Name");
        labelfname.setBounds(400, 150, 150, 30);

        labelfname.setFont(new Font("serif", Font.PLAIN, 20));
        image.add(labelfname);
```

```
tffname = new JTextField();  
tffname.setBounds(600, 150, 150, 30);  
image.add(tffname);
```

```
JLabel labeldob = new JLabel("Date of Birth");  
labeldob.setBounds(50, 200, 150, 30);  
labeldob.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labeldob);
```

```
JLabel lbldob = new JLabel();  
lbldob.setBounds(200, 200, 150, 30);  
image.add(lbldob);
```

```
JLabel labelsalary = new JLabel("Salary");  
labelsalary.setBounds(400, 200, 150, 30);  
labelsalary.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labelsalary);
```

```
tfsalary = new JTextField();  
tfsalary.setBounds(600, 200, 150, 30);  
image.add(tfsalary);
```

```
JLabel labeladdress = new JLabel("Address");  
labeladdress.setBounds(50, 250, 150, 30);  
labeladdress.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labeladdress);
```

```
tfaddress = new JTextField();  
tfaddress.setBounds(200, 250, 150, 30);  
image.add(tfaddress);
```

```
JLabel labelphone = new JLabel("Phone");  
labelphone.setBounds(400, 250, 150, 30);  
labelphone.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labelphone);
```

```
tfphone = new JTextField();  
tfphone.setBounds(600, 250, 150, 30);  
image.add(tfphone);
```

```
JLabel lalemail = new JLabel("Email");  
lalemail.setBounds(50, 300, 150, 30);
```

```
lalemail.setFont(new Font("serif", Font.PLAIN, 20));
```

```
image.add(labelemail);
```

```
tfemail = new JTextField();  
tfemail.setBounds(200, 300, 150, 30);  
image.add(tfemail);
```

```
JLabel labeleducation = new JLabel("Higest Education");  
labeleducation.setBounds(400, 300, 150, 30);  
labeleducation.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labeleducation);
```

```
tfeducation = new JTextField();  
tfeducation.setBounds(600, 300, 150, 30);  
image.add(tfeducation);
```

```
JLabel labeldesignation = new JLabel("Designation");  
labeldesignation.setBounds(50, 350, 150, 30);  
labeldesignation.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labeldesignation);
```

```
tfdesignation = new JTextField();  
tfdesignation.setBounds(200, 350, 150, 30);  
image.add(tfdesignation);
```

```
JLabel labelaadhar = new JLabel("Aadhar Number");  
labelaadhar.setBounds(400, 350, 150, 30);  
labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labelaadhar);
```

```
JLabel lblaadhar = new JLabel();  
lblaadhar.setBounds(600, 350, 150, 30);  
image.add(lblaadhar);
```

```
JLabel labelempId = new JLabel("Employee id");  
labelempId.setBounds(50, 400, 150, 30);  
labelempId.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(labelempId);
```

```
lblempId = new JLabel();  
lblempId.setBounds(200, 400, 150, 30);  
lblempId.setFont(new Font("serif", Font.PLAIN, 20));  
image.add(lblempId);
```



```

try {
    Conn c = new Conn();
    String query = "select * from employee where empId = '"+empId+"'";
    ResultSet rs = c.s.executeQuery(query);
    while(rs.next()) {
        lblname.setText(rs.getString("name"));
        tffname.setText(rs.getString("fname"));
        lbldob.setText(rs.getString("dob"));
        tfaddress.setText(rs.getString("address"));
        tfsalary.setText(rs.getString("salary"));
        tfphone.setText(rs.getString("phone"));
        tfemail.setText(rs.getString("email"));
        tfeducation.setText(rs.getString("education"));
        lblaadhar.setText(rs.getString("aadhar"));
        lblempId.setText(rs.getString("empId"));
        tfdesignation.setText(rs.getString("designation"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

add = new JButton("Update Details");
add.setBounds(250, 550, 150, 40);
add.addActionListener(this);
add.setBackground(Color.orange);
add.setForeground(Color.black);
image.add(add);

back = new JButton("Back");
back.setBounds(450, 550, 150, 40);
back.addActionListener(this);
back.setBackground(Color.orange);
back.setForeground(Color.black);
image.add(back);

setSize(900, 650);
setLocation(220, 30);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == add) {
        String fname = tffname.getText();

```

```

        String salary = tfsalary.getText();
String address = tfaddress.getText();
        String phone = tfphone.getText();
        String email = tfemail.getText();
        String education = tfeducation.getText();
        String designation = tfdesignation.getText();

        try {
            Conn conn = new Conn();
            String query = "update employee set fname = '"+fname+"', salary = '"+salary+"', address =
            '"+address+"', phone = '"+phone+"', email = '"+email+"', education = '"+education+"', designation =
            '"+designation+"' where empId = '"+empId+"'";
            conn.s.executeUpdate(query);
            JOptionPane.showMessageDialog(null, "Details updated
            successfully");          setVisible(false);          new Home();          } catch
            (Exception e) {          e.printStackTrace();
            }
        } else {
            setVisible(false);
            new Home();
        }
    }

    public static void main(String[] args) {
        new UpdateEmployee("");
    }
}

```

7. ViewEmployee.java

```

package employee.management.system;

import javax.swing.*;
import java.awt.*; import
java.sql.*;
import net.proteanit.sql.DbUtils; import
java.awt.event.*;

public class ViewEmployee extends JFrame implements ActionListener{

    JTable table;
    Choice cemployeeId;
    JButton search, print, update, back;

```

31.

```

ViewEmployee() {
    getContentPane().setBackground(Color.WHITE);    setLayout(null);

    ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/view.jpg"));
    Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
    ImageIcon i3 = new ImageIcon(i2);
    JLabel image = new JLabel(i3);
    image.setBounds(0, 0, 1120, 630);
    add(image);

    JLabel searchlbl = new JLabel("Search by Employee Id");
    searchlbl.setBounds(20, 20, 150, 20);
    image.add(searchlbl);

    cemployeeId = new Choice();
    cemployeeId.setBounds(180, 20, 150, 20);
    image.add(cemployeeId);

    try {
        Conn c = new Conn();
        ResultSet rs = c.s.executeQuery("select * from
employee");    while(rs.next()) {
cemployeeId.add(rs.getString("empId"));
        }
    } catch (Exception e) {
e.printStackTrace();
    }

    table = new JTable();

    try {
        Conn c = new Conn();
        ResultSet rs = c.s.executeQuery("select * from
employee");
        table.setModel(DbUtils.resultSetToTableModel(rs));    } catch
(Exception e) {    e.printStackTrace();
    }

    JScrollPane jsp = new JScrollPane(table);
    jsp.setBounds(0, 100, 900, 600);    image.add(jsp);

    search = new JButton("Search");    search.setBounds(20, 70, 80, 20);
    search.addActionListener(this);
    image.add(search);

```

32.

```

        print = new JButton("Print");
print.setBounds(120, 70, 80, 20);
print.addActionListener(this);    image.add(print);

        update = new JButton("Update");
update.setBounds(220, 70, 80, 20);
update.addActionListener(this);    image.add(update);

        back = new JButton("Back");
back.setBounds(320, 70, 80, 20);
back.addActionListener(this);    image.add(back);

        setSize(900, 600);
        setLocation(220, 30);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
if (ae.getSource() == search) {
    String query = "select * from employee where empId = '"+employeeId.getSelectedItemAt()+"'";
    try {
        Conn c = new Conn();
        ResultSet rs = c.s.executeQuery(query);

        table.setModel(DbUtils.resultSetToTableModel(rs));    }
        catch (Exception e) {    e.printStackTrace();
        }
    } else if (ae.getSource() == print) {
    try {
        table.print();
    } catch (Exception e) {
        e.printStackTrace();
    }
    } else if (ae.getSource() == update) {
        setVisible(false);    new
        UpdateEmployee(cemployeeId.getSelectedItemAt());
    } else {
        setVisible(false);
        new Home();
    }
}

    public static void main(String[] args) {
        new ViewEmployee();
    }
}

```

8. RemoveEmployee.java

```
package employee.management.system;

import javax.swing.*.*;
import java.awt.*.*; import
java.sql.*;
import java.awt.event.*;

public class RemoveEmployee extends JFrame implements ActionListener {

    Choice cEmpId;
    JButton delete, back;

    RemoveEmployee() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel labelempId = new JLabel("Employee Id");
        labelempId.setBounds(50, 50, 100, 30);    add(labelempId);

        cEmpId = new Choice();
        cEmpId.setBounds(200, 50, 150, 30);
        add(cEmpId);

        try {
            Conn c = new Conn();
            String query = "select * from employee";
            ResultSet rs = c.s.executeQuery(query);
            while(rs.next()) {
                cEmpId.add(rs.getString("empId"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        JLabel labelname = new JLabel("Name");

        labelname.setBounds(50, 100, 100, 30);
        add(labelname);

        JLabel lblname = new JLabel();
        lblname.setBounds(200, 100, 100, 30);    add(lblname);
```

```
JLabel labelphone = new JLabel("Phone");
labelphone.setBounds(50, 150, 100, 30);    add(labelphone);
```

```
JLabel lblphone = new JLabel();
lblphone.setBounds(200, 150, 100, 30);
add(lblphone);
```

```
JLabel labelemail = new JLabel("Email");
labelemail.setBounds(50, 200, 100, 30);    add(labelemail);
```

```
JLabel lblemail = new JLabel();
lblemail.setBounds(200, 200, 100, 30);    add(lblemail);
```

```
try {
    Conn c = new Conn();
    String query = "select * from employee where empId = '"+cEmpId.getSelectedItem()+"'";
    ResultSet rs = c.s.executeQuery(query);
    while(rs.next()) {
        lblname.setText(rs.getString("name"));
        lblphone.setText(rs.getString("phone"));
        lblemail.setText(rs.getString("email"));
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

```
cEmpId.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ie) {
try {
    Conn c = new Conn();
    String query = "select * from employee where empId = '"+cEmpId.getSelectedItem()+"'";
    ResultSet rs = c.s.executeQuery(query);
    while(rs.next()) {
        lblname.setText(rs.getString("name"));
        lblphone.setText(rs.getString("phone"));
        lblemail.setText(rs.getString("email"));
    }
} catch (Exception e) {
    e.printStackTrace();
}
    }
});
```

```

        delete = new JButton("Delete");
delete.setBounds(80, 300, 100,30);
delete.setBackground(Color.BLACK);
delete.setForeground(Color.WHITE);
delete.addActionListener(this);    add(delete);

        back = new JButton("Back");
back.setBounds(220, 300, 100,30);
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.addActionListener(this);    add(back);

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icons/delete.png"));
        Image i2 = i1.getImage().getScaledInstance(600, 400, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(350, 0, 600, 400);
add(image);

        setSize(1000, 500);
setLocation(220, 100);
setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == delete) {
            try {
                Conn c = new Conn();
                String query = "delete from employee where empId = '"+cEmpId.getSelectedItemAt()+"'";
c.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null, "Employee Information Deleted
Sucessfully");
                setVisible(false);
                new Home();
            } catch (Exception e) {
e.printStackTrace();
            }
        } else {
            setVisible(false);
            new Home();
        }
    }

    public static void main(String[] args) {
        new RemoveEmployee();
    }
}

```

9. Database Connectivity (Conn.java)

```
package employee.management.system;

import java.sql.*;

public class Conn {

    Connection    c;
    Statement     s;
    public Conn () {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/employeemanagementsystem", "root",
            "mukesh@123");
            s = c.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Create Database

```
create database employeemanagementsystem;
```

```
show databases;
```

```
use employeemanagementsystem;
```

```
create table login(username varchar(20), password varchar(20));
```

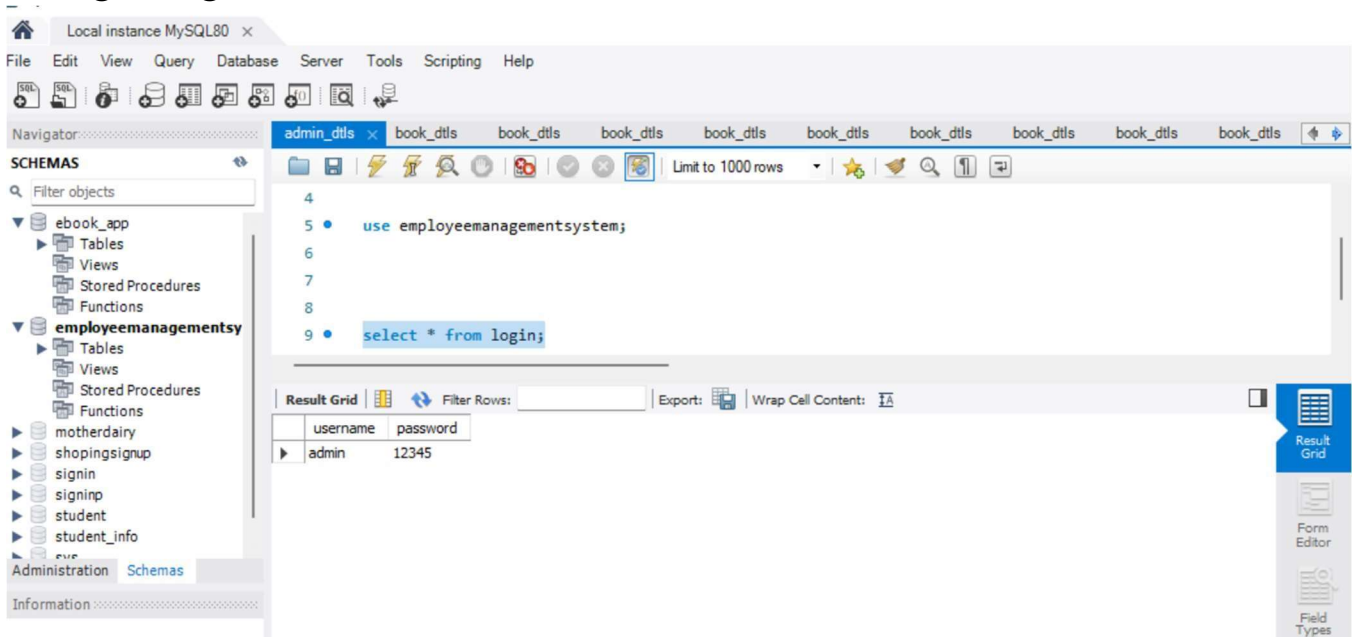
```
insert into login values('admin', '12345');
```

```
select * from login;
```

```
create table employee(name varchar(20),fname varchar(20),  
dob varchar(30),salary varchar(20),address varchar(100)  
,phone varchar(12) ,email varchar(40) ,  
education varchar(20),designation varchar(30),  
aadhar varchar(25),empId varchar(20) );
```

```
select * from employee;
```

1.Login Page With Database



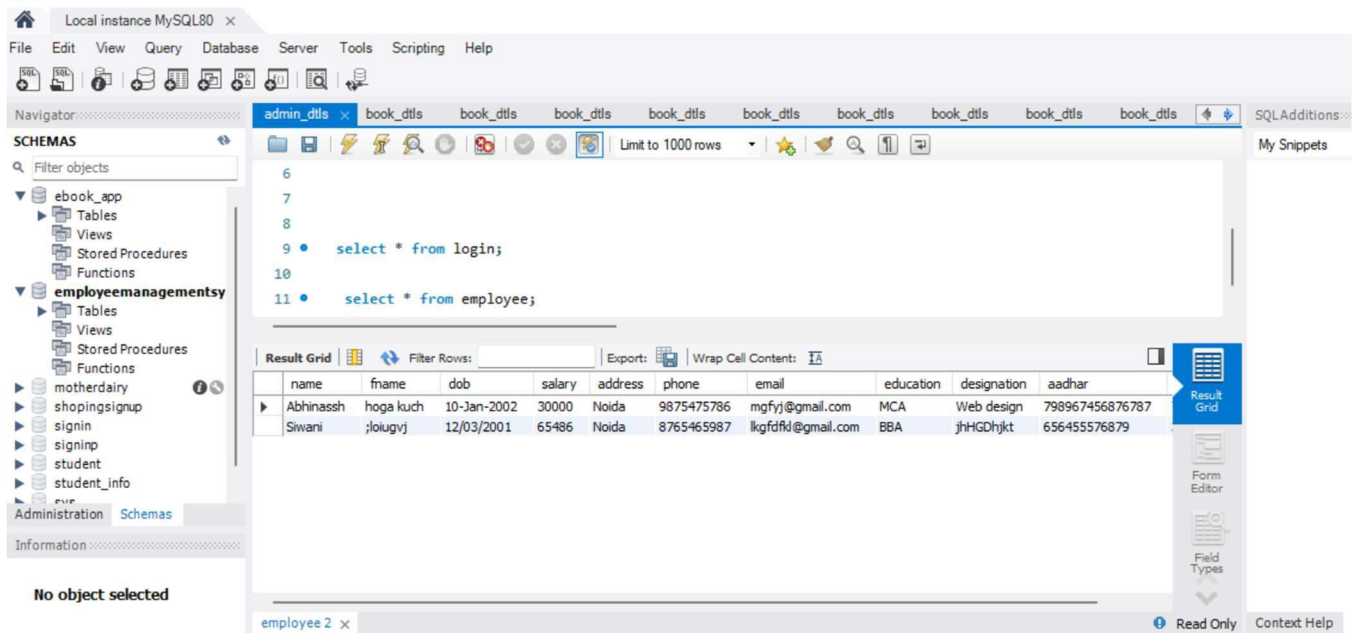
The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Schemas' pane on the left shows the 'employeemanagementsystem' database selected. The main query editor contains the following SQL code:

```
4  
5 • use employeemanagementsystem;  
6  
7  
8  
9 • select * from login;
```

The 'Result Grid' at the bottom shows the output of the query, displaying a single row with the username 'admin' and password '12345'.

username	password
admin	12345

2.Employee Details in Database



X.Conclusion

The goal of the initiative is to digitise personnel databases in businesses and provide administrators access to computers. Employees and administrators use software as an information system. The user can store his or her database safe and secure for an indefinite amount of time here. Adding, deleting, accessing, and changing employee information is simple and easy using the Employee Management System.

XI. Future Scope

The GUI and the features added to this system are the basic ones. In future, there will be a better Graphical User Interface and there will be more features added to this system. If Graphical User Interface is improved then this system will be more user friendly and more features added will make this system a lot better and HR will be able to perform more operations.

