A PROJECT REPORT

On

EMPLOYEE MANAGEMENT SYSTEM USING JAVA

SUBMITTED TO THE DIRECTORATE OF DISTANCE &

CONTINUING EDUCATION IN PARTIAL FULLFILLMENT

OF THE

BACHELORS IN COMPUTER APPLICATIONS

Submitted by

Sagar Patel (822151453893BCA)

Aashu Patel (822151452020BCA)

Kunj Thakkar (822151452028BCA)

Under the Guidance of

Name of Internal Guide

Ms. Hiral Patel

PROJECT REPORT

On

(Employee Management System Using Java)

SUBMITTED TO

Dr. Babasaheb Ambedkar Open University

By

Name: _____

Enrollment No: _____

Study Centre Name: _____

Study Centre Code: _____

CERTIFICATE OF THE GUIDE

Guide Name: MS. HIRAL PATEL

Designation: Assistant Professor

This is to certify that the project report entitled "Employee Management System Using Java" has been prepared by Sagar Patel, Aashu Patel, Kunj Thakkar under my supervision and guidance, as a Project work (BCAR-404-PRO). Their Project work is satisfactory

Date: **Signature of Guide**

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, especially my internal guide Ms. HIRAL PATEL who was always there at hour of need. My special thanks to all other faculty members, Batch mate & Seniors of S. B. COLLEGE OF COMPUTER APPLICATION & MANAGEMENT for helping me in the completion of project work and its report submission.

DECLARATION

I do hereby declare that this project work entitled **"Employee Management System Using Java"** submitted by me for the partial fulfilment of the requirement for the second Semester BCAR-404-PRO is a record of my own work. The report embodies the finding based on my study and observation and has not been submitted earlier for the award of any degree or diploma to any Institute or University.

Name:

Sagar Patel 822151453893BCA

Aashu Patel 822151452020BCA

Date: Kunj Thakkar 822151452028BCA

STUDENT'S PROJECT REPORT EVALUATION BY INTERNAL EXAMINER

Date: Year:2024

Program: BCA Semester:4th

Enrollment No: 822151453893BCA

822151452020BCA

822151452028BCA

Study Centre Name: S. B. College of Computer Application & Management - Savli

Study Centre Code: 0791514

Student's Name: Sagar Patel, Aashu Patel, Kunj Thakkar

SR.NO	PARTICULARS	MARKS OUT OF	MARKS OBTAINED
1	Project Definition, Its Size, Complexity, and Quantum of Work:		
2	Coding Style Including (I) Generalized Parameterized, (II) Structured-Modular Coding Style, (III) Compactness & Clarity, (IV) Checkpoints for intermediate results, (V) Naming Conventions, (VI) Self-Documented:		
3	Completion and Operational		
4	Quality of Output and Testing Plan, etc.		
5	A Section in Report Containing: Analysis of Various Alternative and the Justification for the Selected Approach		
6	Overall		
TOTAL			

Guide Name: MS. HIRAL PATEL

Designation: ASSISTANT PROFESSOR

Signature:

Seal of the Study Centre

Signature of Study Centre Head _____

Date: _____

INDEX

CHAPTER – 1

➢ INTRODUCTION:

Everything has been digitized 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 program 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.

CHAPTER – 2

➢ 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:
    1. Design of an HR management system to meet needs such as adding and deleting employees, viewing and printing employee data, and updating employee information.
    2. Employee data is stored in a well-designed database.
    3. An easy-to-use interface that will let user interact with the system.

CHAPTER – 3

➢ 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). NetBeans enables the creation of applications using a set of modular software components known as modules. NetBeans 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 NetBeans-based applications, including the NetBeans 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 programs.

❖ **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 programs. 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 a look &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 – 4

➢ FEASIBILITY STUDY:

In order to do a feasibility study, we must consider the following:

1. Technical Feasibility:

The availability of hardware & Software necessary for the creation of the system, as-well-as the compatibility and maturity of the technology planned to be used, and the availability of the requisite technical staff to create the system, are all factors to consider.

2. Operational Feasibility:

Problems that may develop during operations are the focus of operation feasibility. There are two parts to this problem to consider:

- ✓ What are the chances that the solution provided will not be used or will not work?
- ✓ What is the inclination of-the management and end users towards the solution?

3. Economic Feasibility:

The concept of economic feasibility is determining whether or not the potential benefit of fixing difficulties is worthwhile. Because member needs &alternative solutions haven't been specified at this point, it is difficult to estimate the cost at this level.

CHAPTER – 5

➢ DESIGN:

The System was designed in NetBeans Software. The System Design Phase Describes the Functional Capabilities of the Proposed System. This is divided into the following Design Phases:

- ✓ System data flow diagram
- ✓ Class diagram

CHAPTER – 6

➢ DESCRIBE HOW TO RUN A PROGRAM?

- ✓ That is too simple to run a program in Apache NetBeans.
- ✓ Simply Click on run button on the task bar to run the program.
- ✓ After running the program, you can see Splash Page of Employee Management System, After Click on CLICK HERE TO CONTINUE Button Redirect to the Login Page, Enter Usar Name and Password Open Home Page.

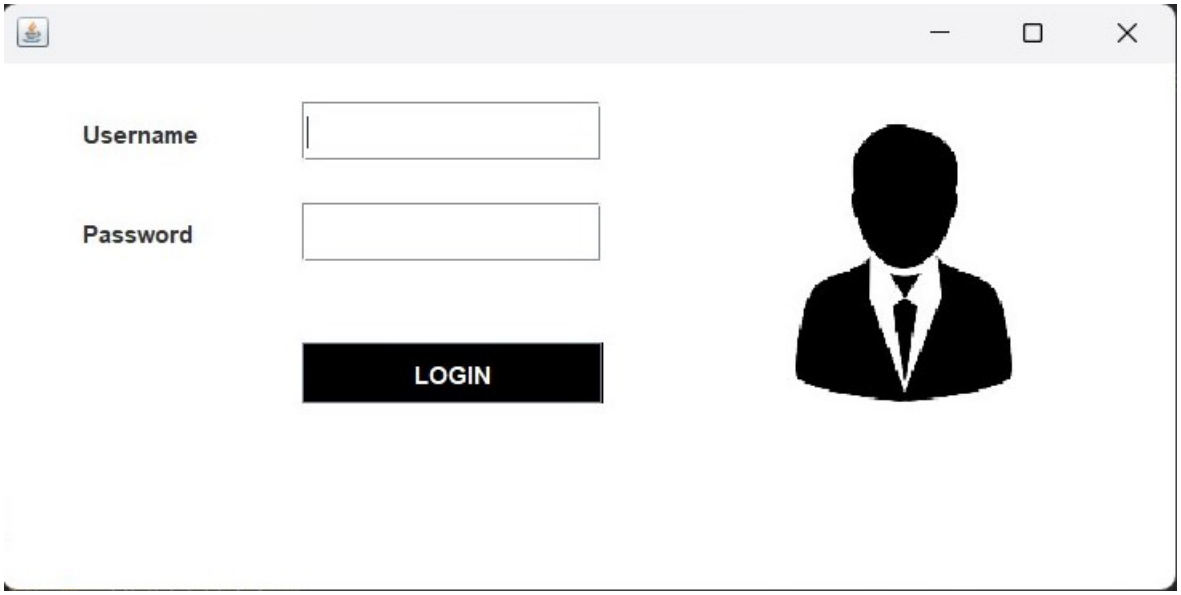
CHAPTER – 7

➢ 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.

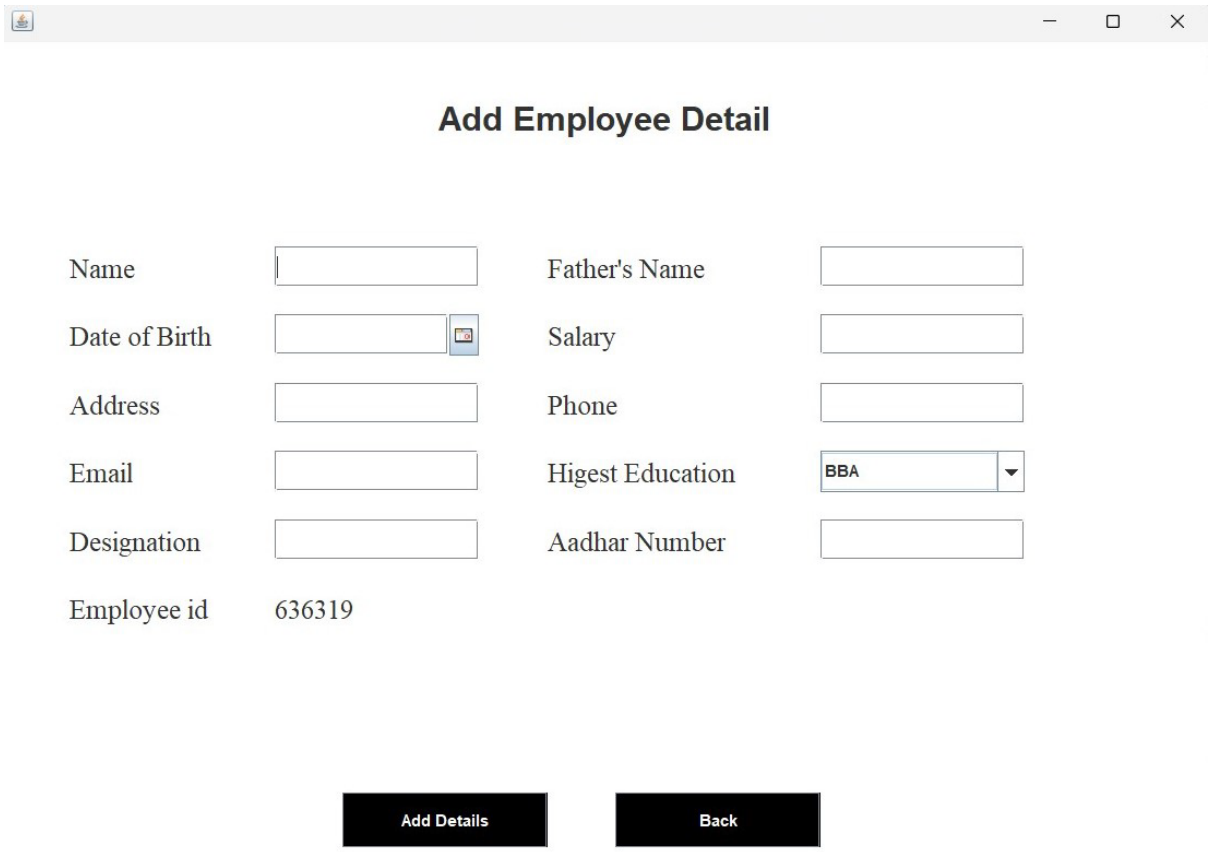


2. Main Dashboard:

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.

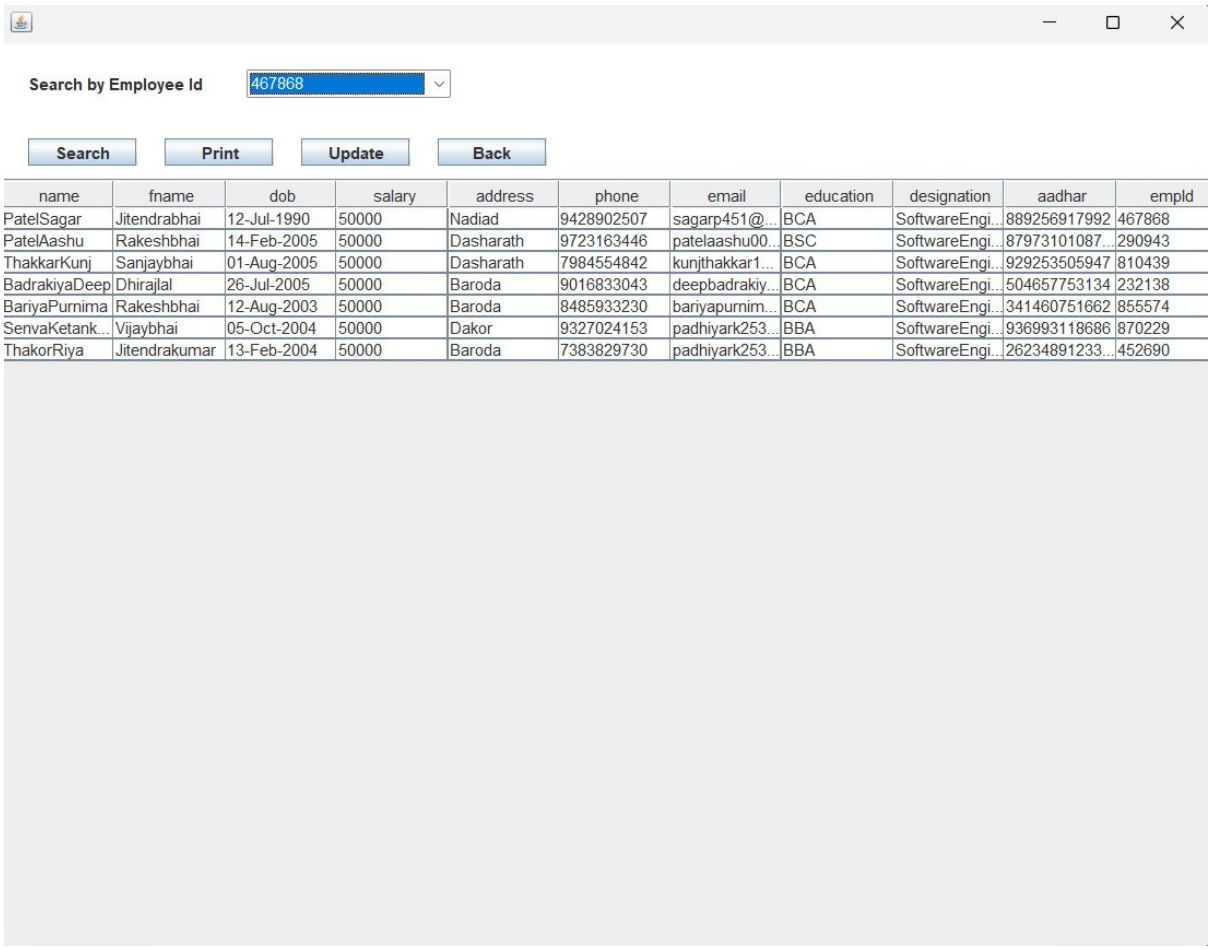
3. Add Employee:

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



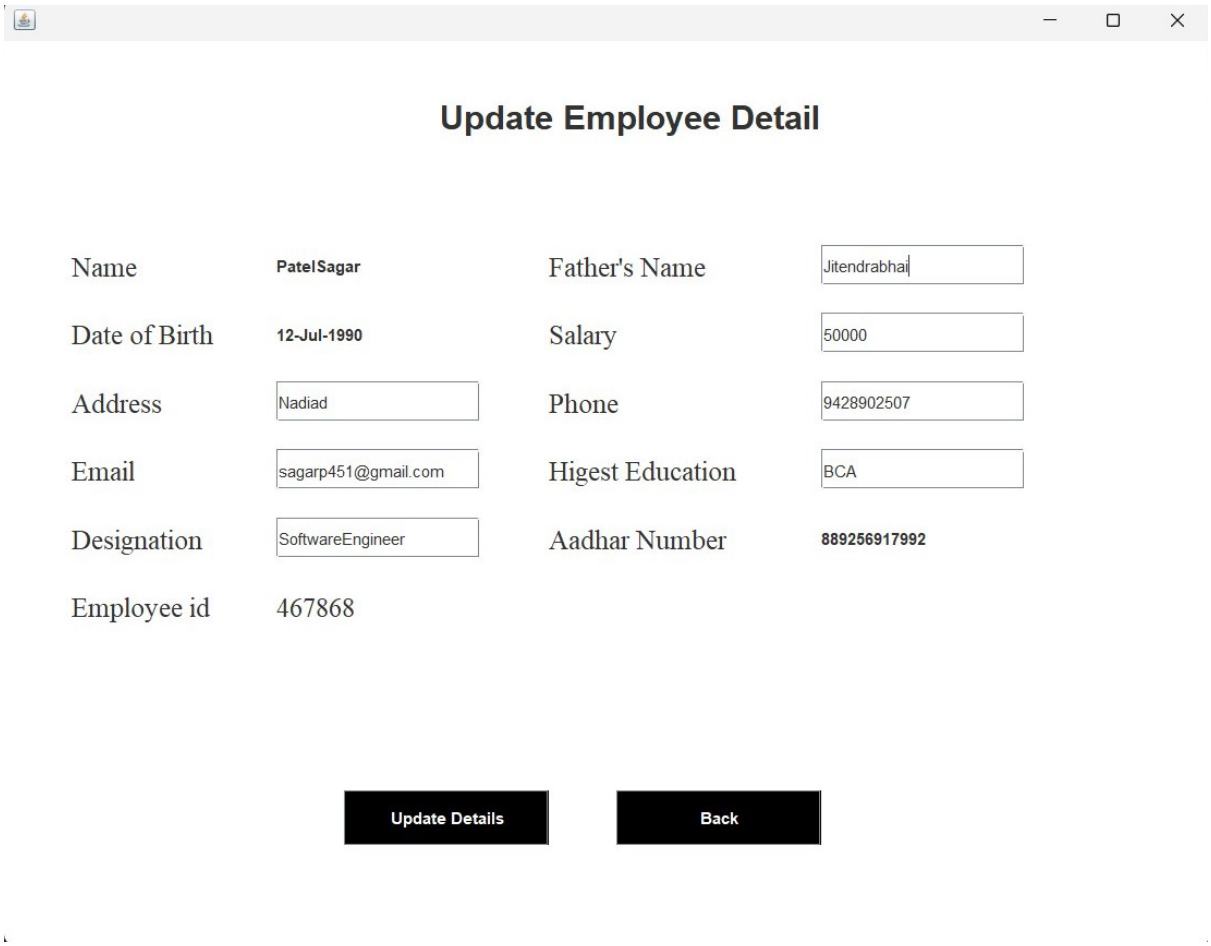
4. View Employee:

In order to view employee information, the user have to enter employee ID.



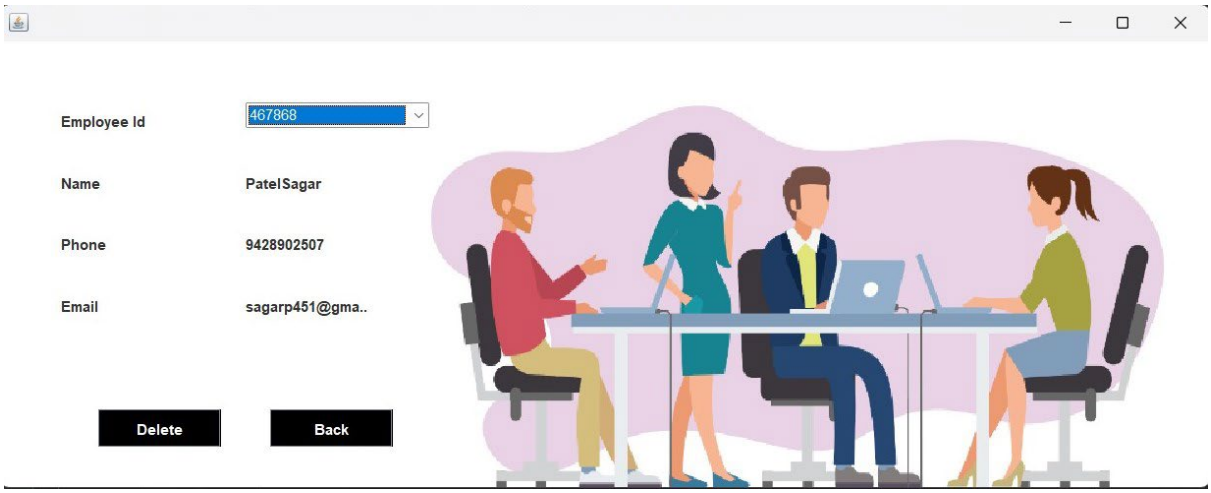
5. Update Employee:

In order to and update employee information, the user have to enter employee ID.



6. Remove employee:

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



CHAPTER – 8

➢ SOFTWARE REQUIREMENT

❖ Software Used:

- ✓ Apache NetBeans IDE 21

➢ HARDWARE REQUIREMENT:

❖ Hardware Used

- ✓ Intel(R) Core (TM) i3-8145U CPU @ 2.10GHz 2.30 GHz
- ✓ 12 GB Ram
- ✓ 256 GB SSD
- ✓ 1 TB HDD
- ✓ Personal Computer / Laptop

CHAPTER – 9

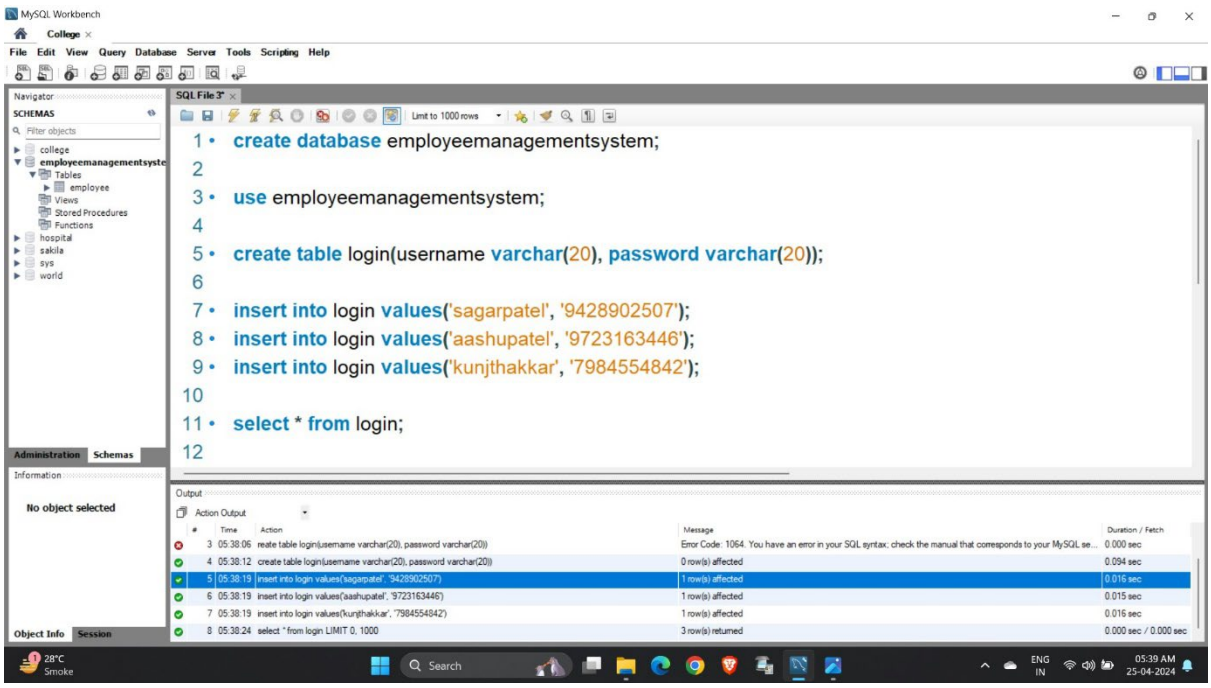
✓ TESTING:

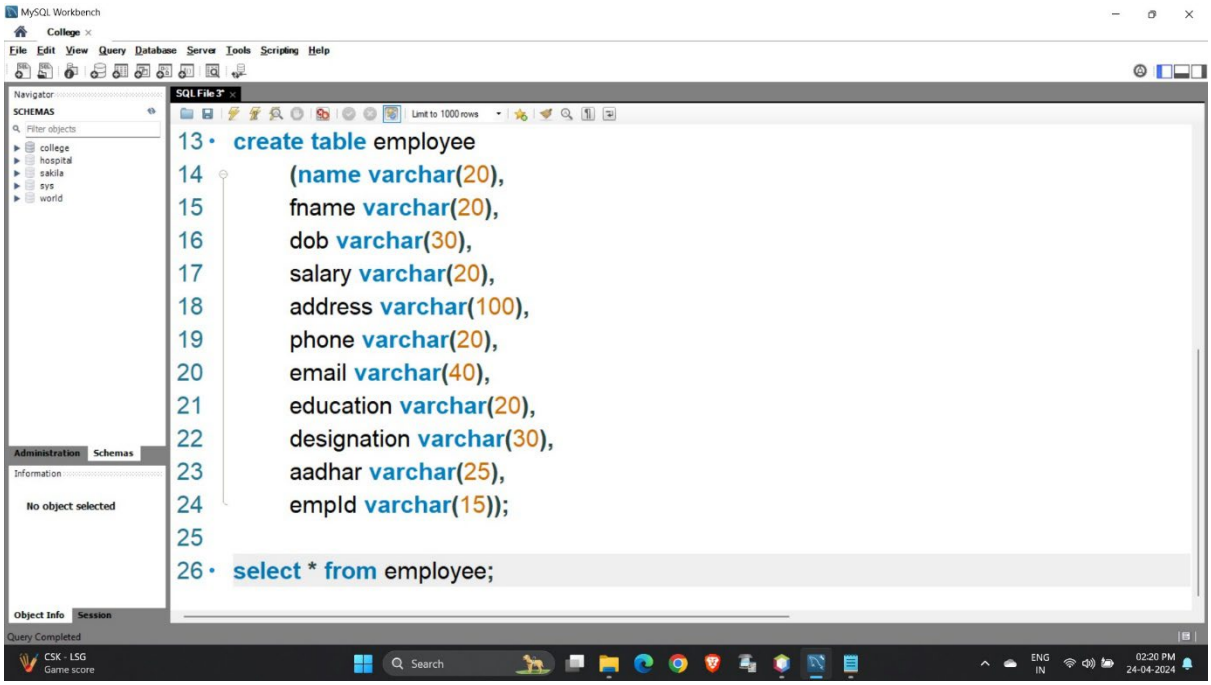
Test ID	Test-Purpose	Test-Condition	Expected-Output	Output	Remark
TC1	Check Username & Password	If user details are not correct, display error message	Grant access to main dashboard.	Access granted to main dashboard	Test Successful
TC2	To add new user to the system	If user already exists, error message should be displayed.	New user should be added.	New user added successfully	Test Successful
TC3	To view existing employee information	If employee exists, then information should be displayed, else error message should be displayed.	Employee information should be displayed.	Employee information displayed.	Test Successful
TC4	To remove an employee	If employee exists, then employee should be removed else error message should be displayed.	Employee should be removed.	Employee removed successfully.	Test Successful
TC5	Update employee information	If employee exists, then information should be updated.	Employee information should be updated.	Employee information updated successfully	Test Successful

CHAPTER –10

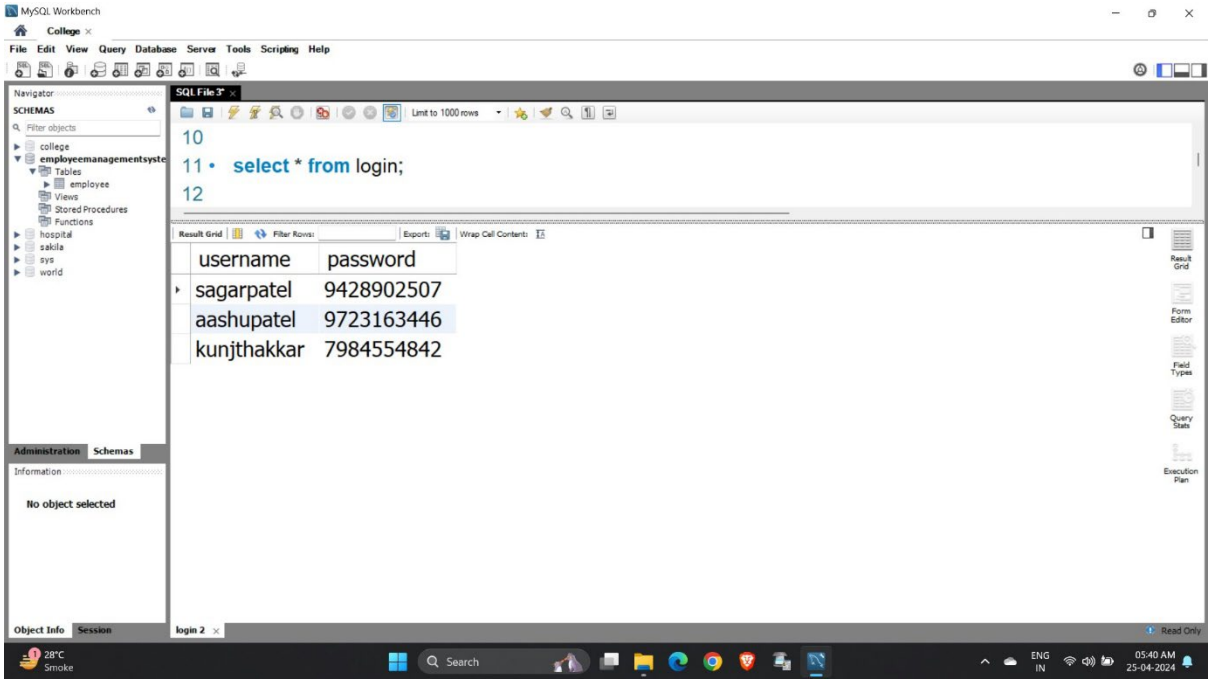
➢ CODE SCREENSHOT:

✓ My SQL Database:

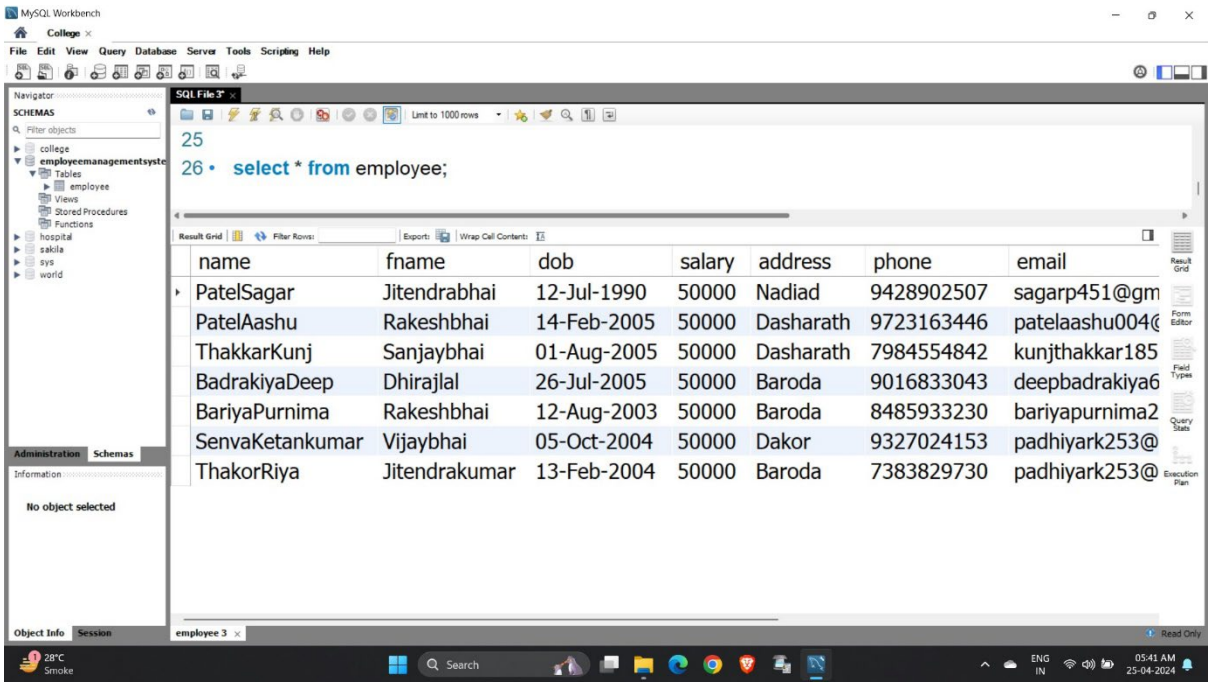




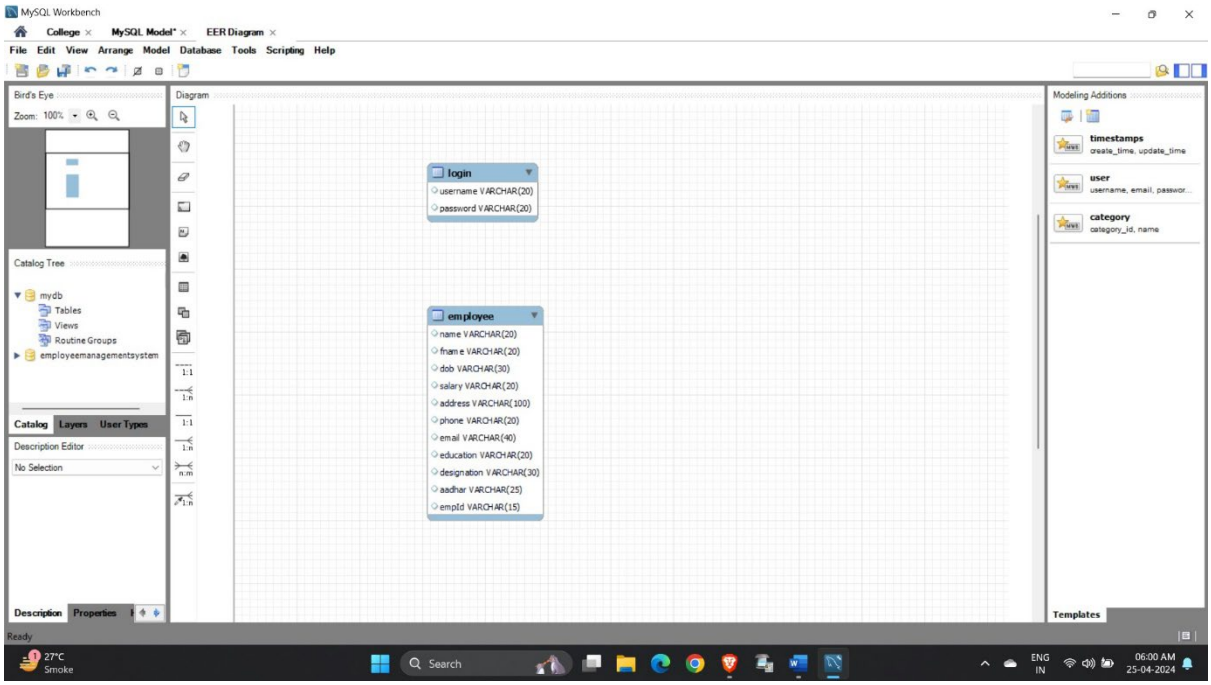
✓ Login Table:



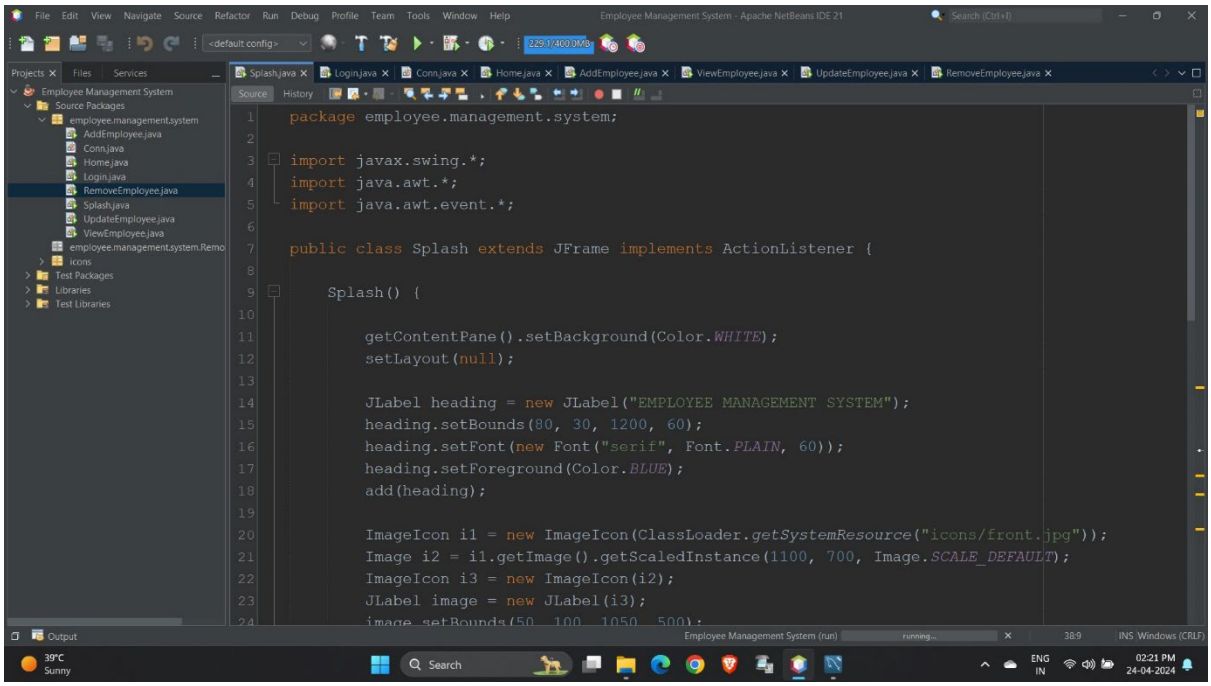
✓ Employee Table:

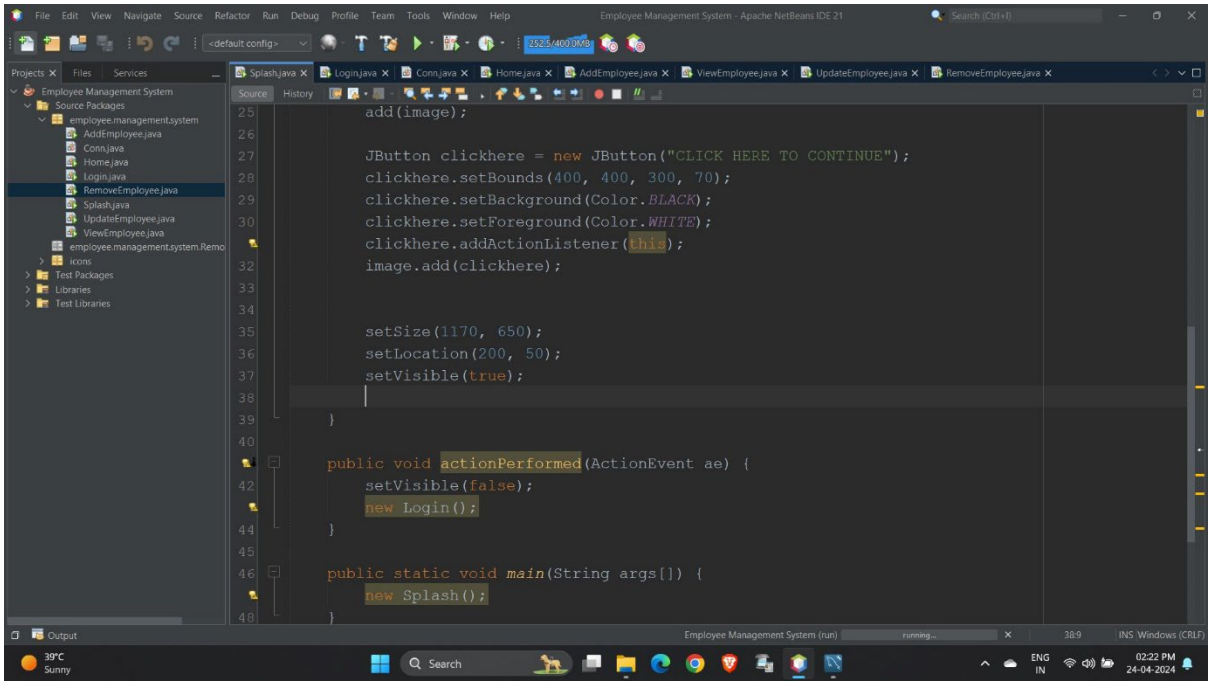


✓ ER Diagram:



✓ Splash Page:

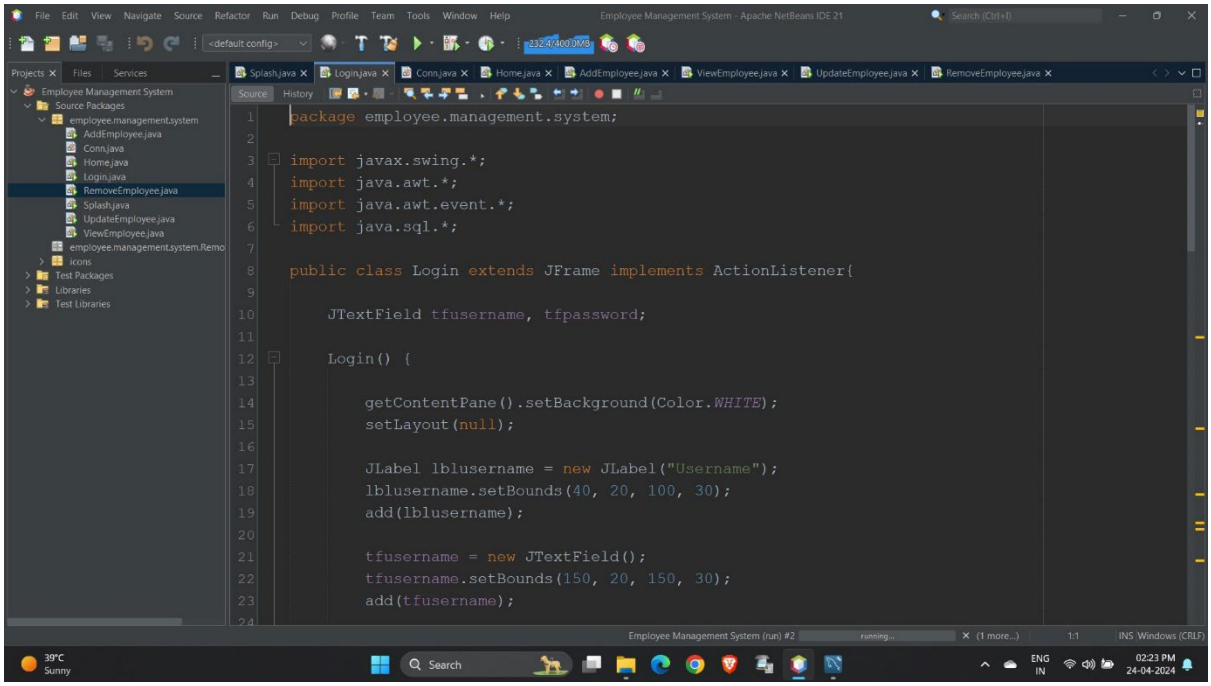




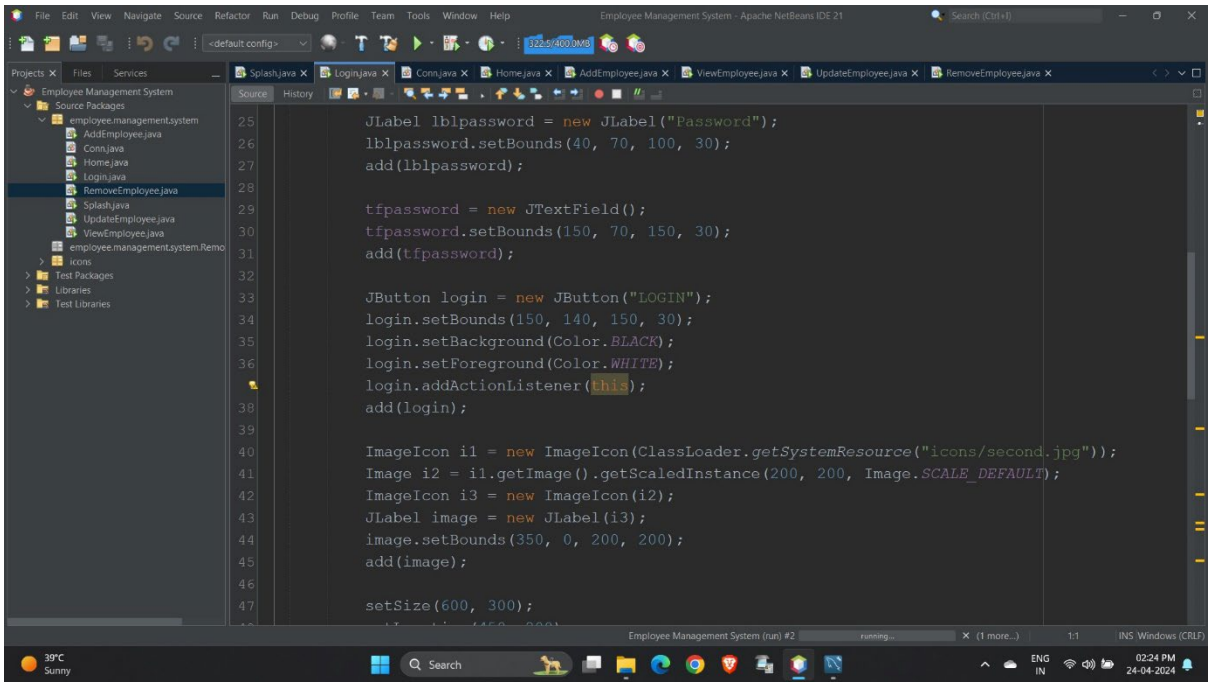
```
            add(image);

            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);
            image.add(clickhere);


            setSize(1170, 650);
            setLocation(200, 50);
            setVisible(true);

        }

        public void actionPerformed(ActionEvent ae) {
            setVisible(false);
            new Login();
        }

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

✓ Output of Splash Page:

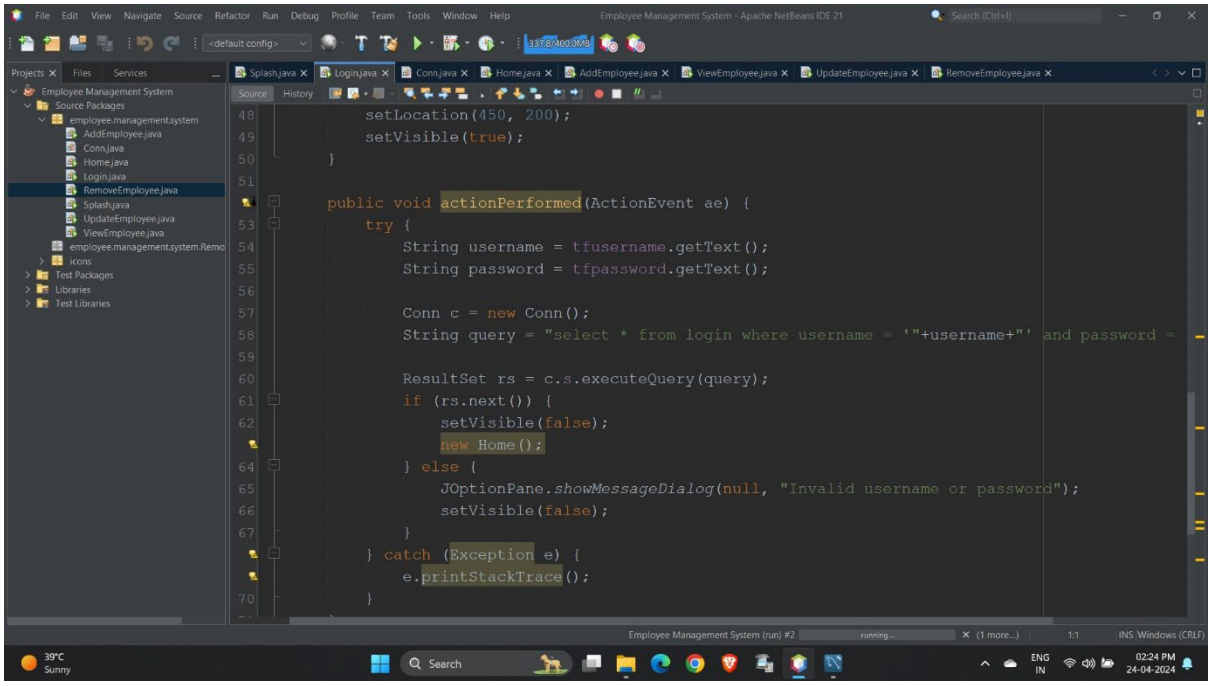
✓ Login Page:



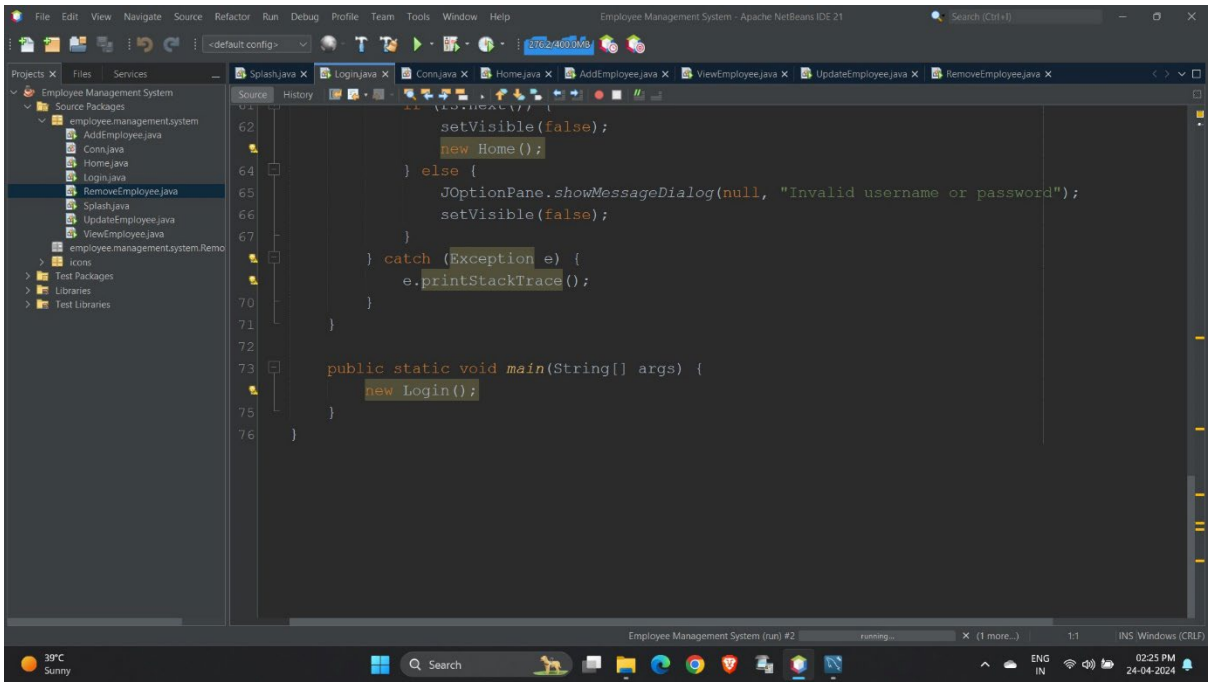
```
package employee.management.system;

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);
        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.getSystemResource("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);
```

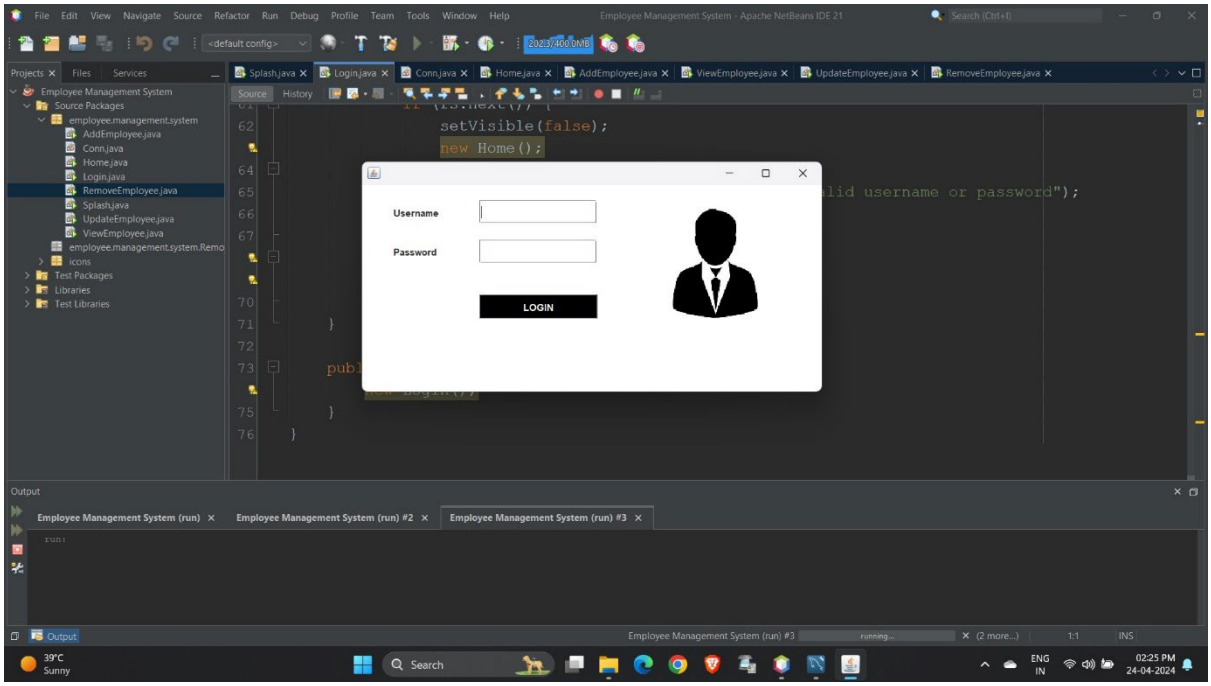


```
            setLocation(450, 200);
            setVisible(true);
        }

        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 =

                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();
            }
```

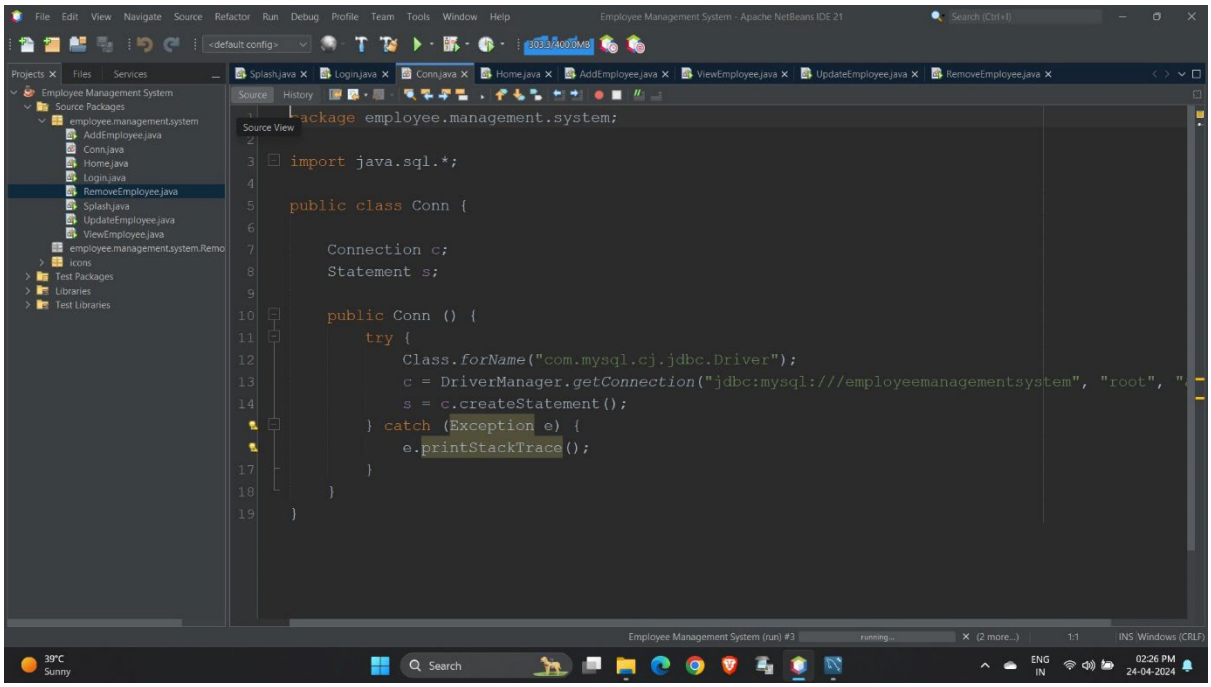


```
                    setVisible(false);
                    new Home();
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid username or password");
                    setVisible(false);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

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

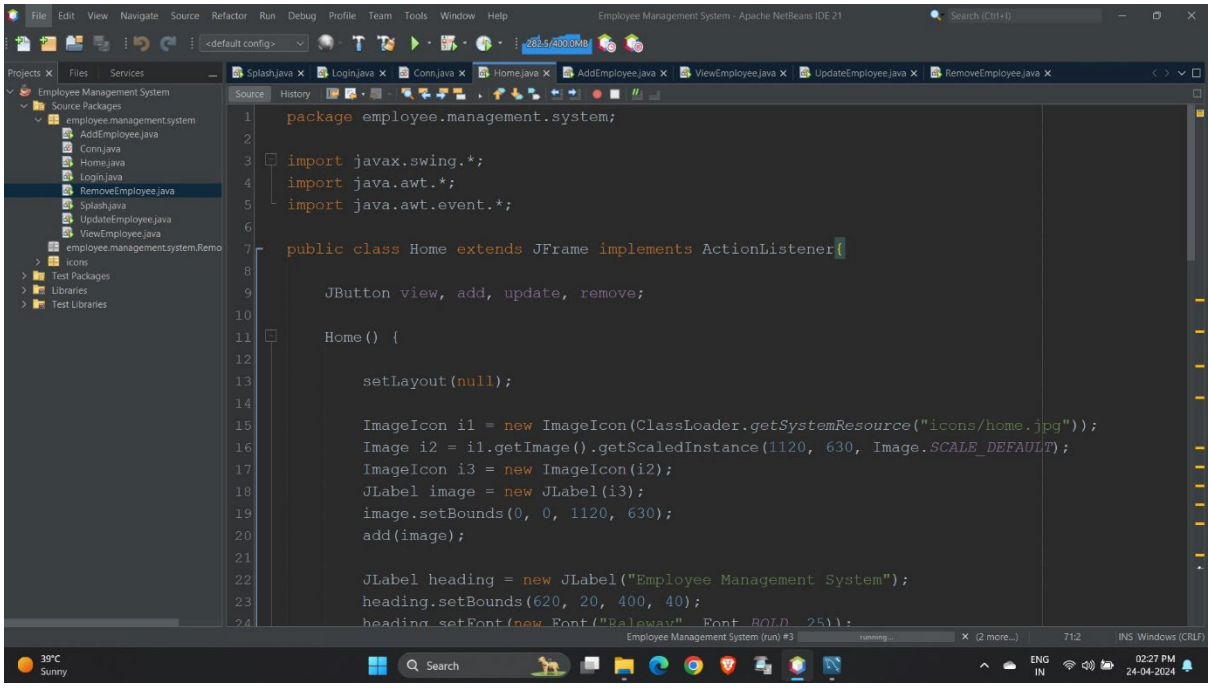
✓ Output of Login Page:



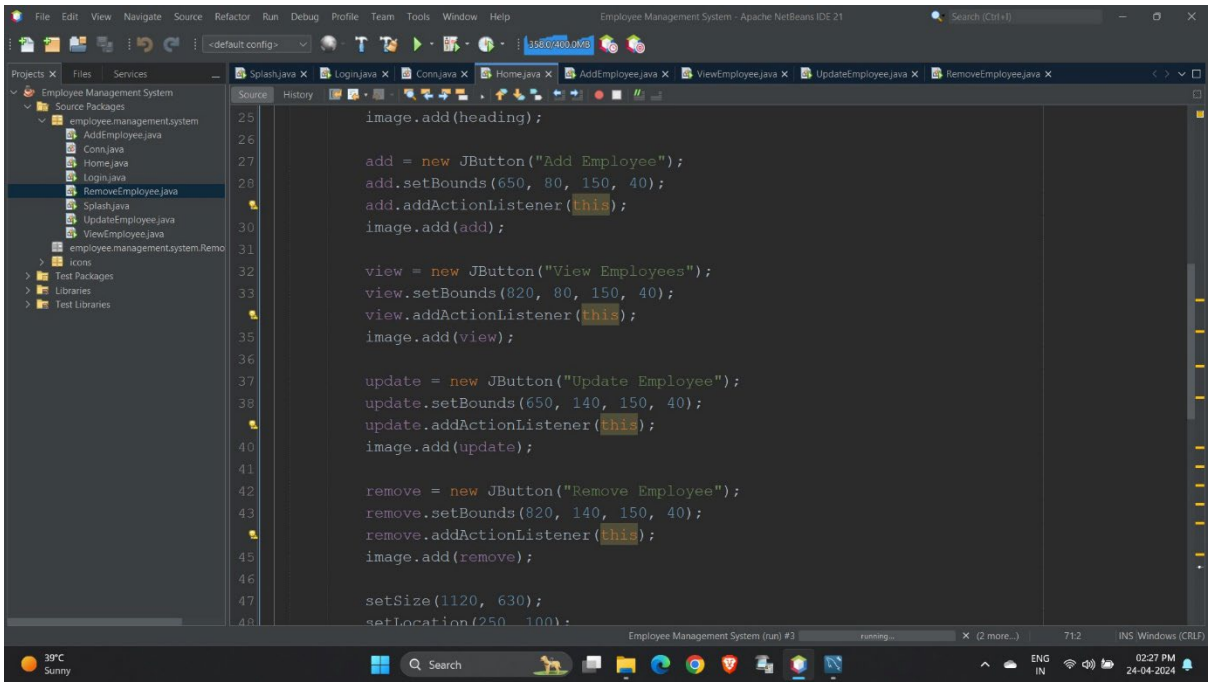
✓ Connection Page:



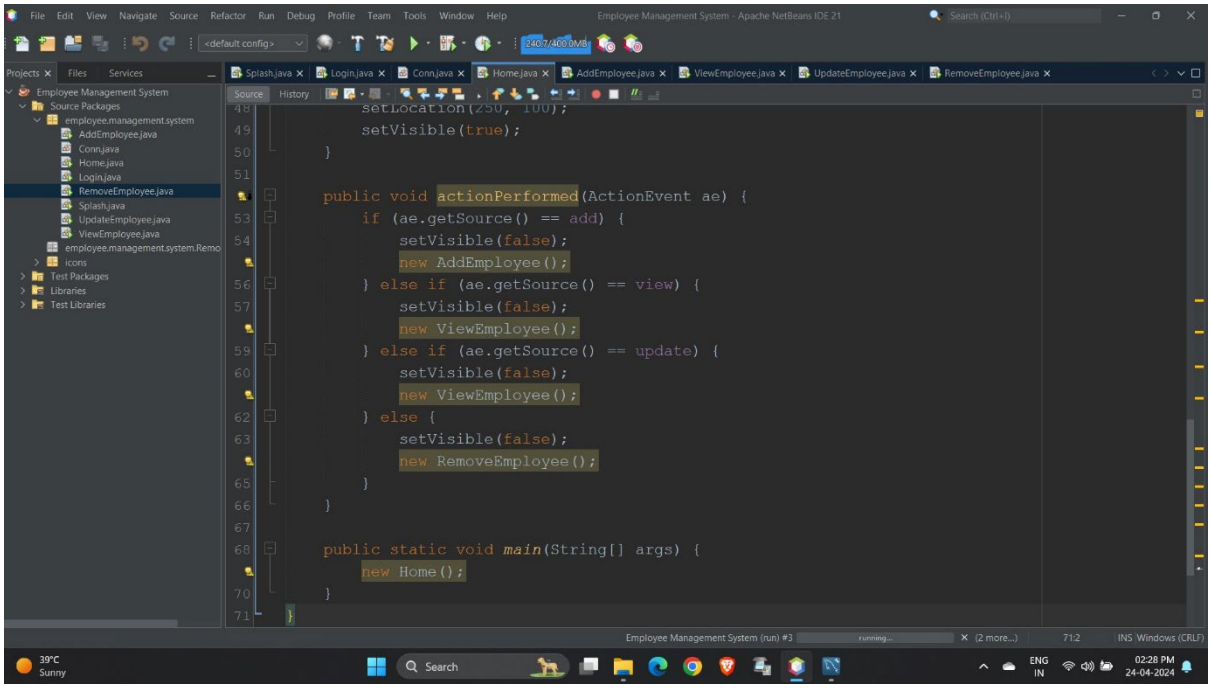
✓ Home Page:



```
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.getSystemResource("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));
```



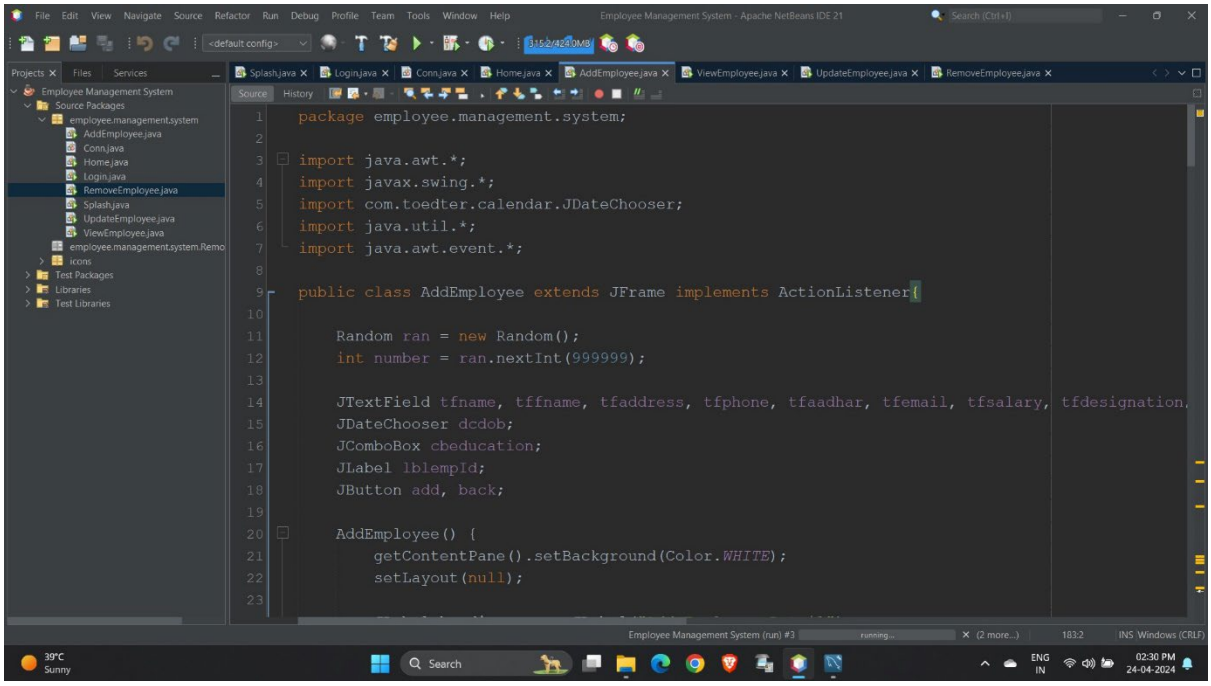
```
        image.add(heading);

        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(250, 100);
```



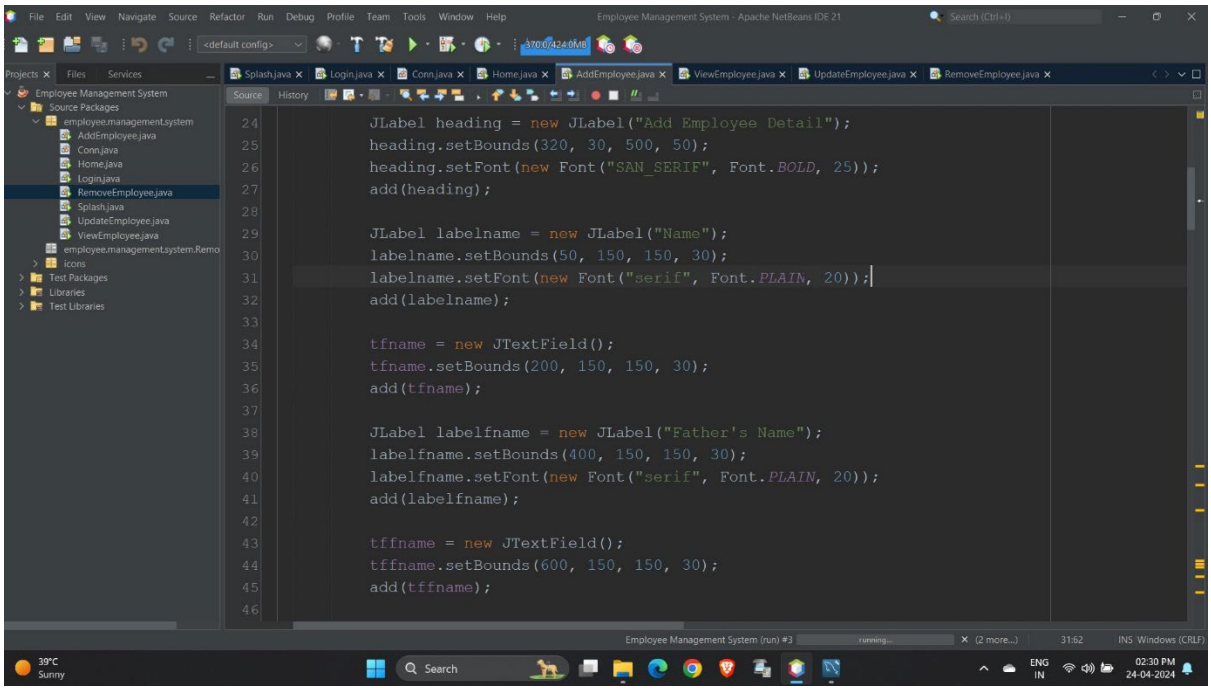
```
        setLocation(250, 100);
        setVisible(true);
    }

    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();
    }
}
```

✓ Output of Home Page:

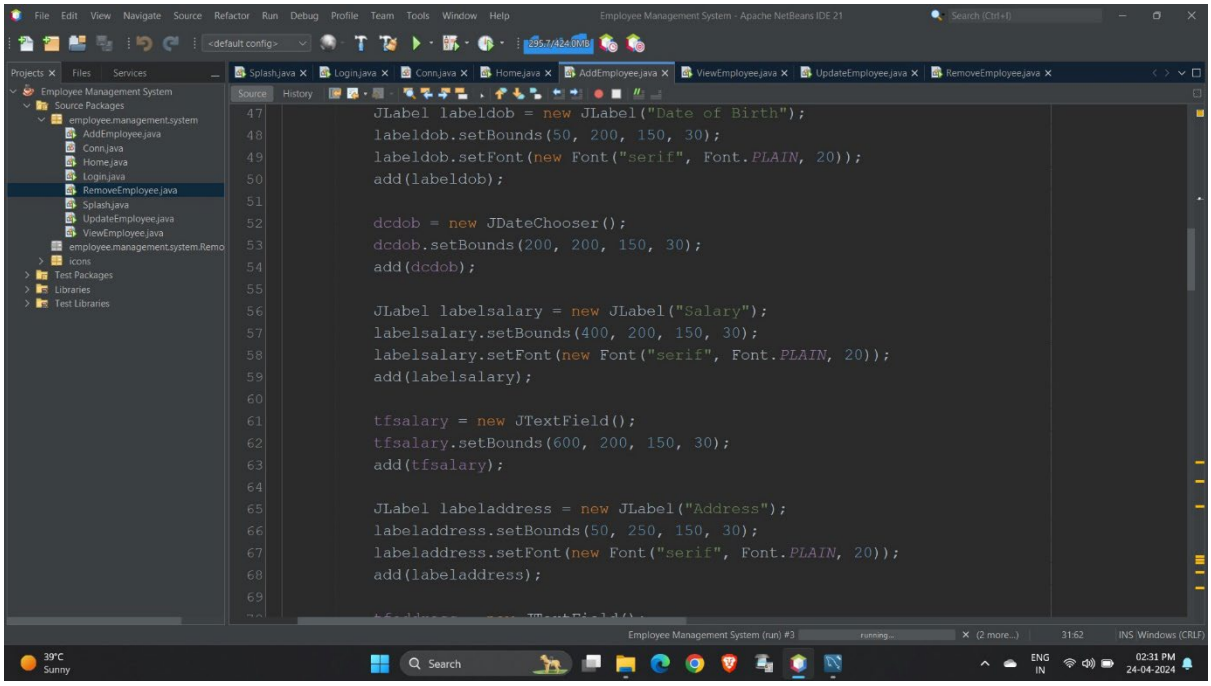
✓ Add Employee Page:



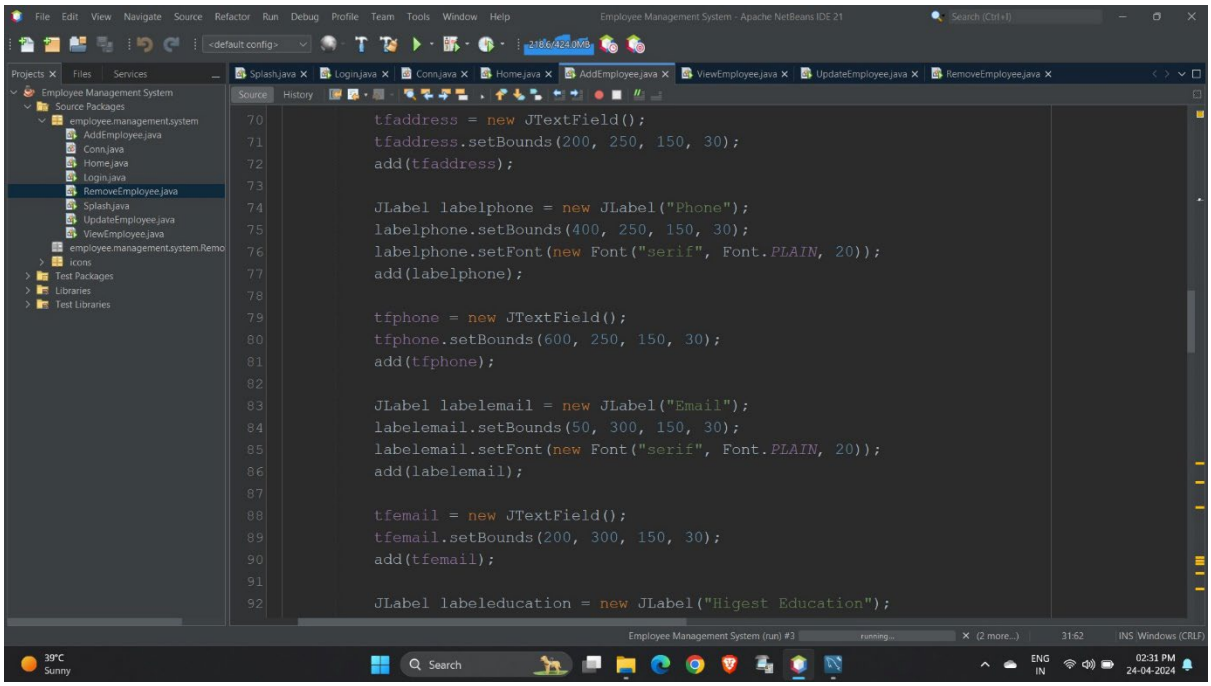
```
package employee.management.system;

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

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



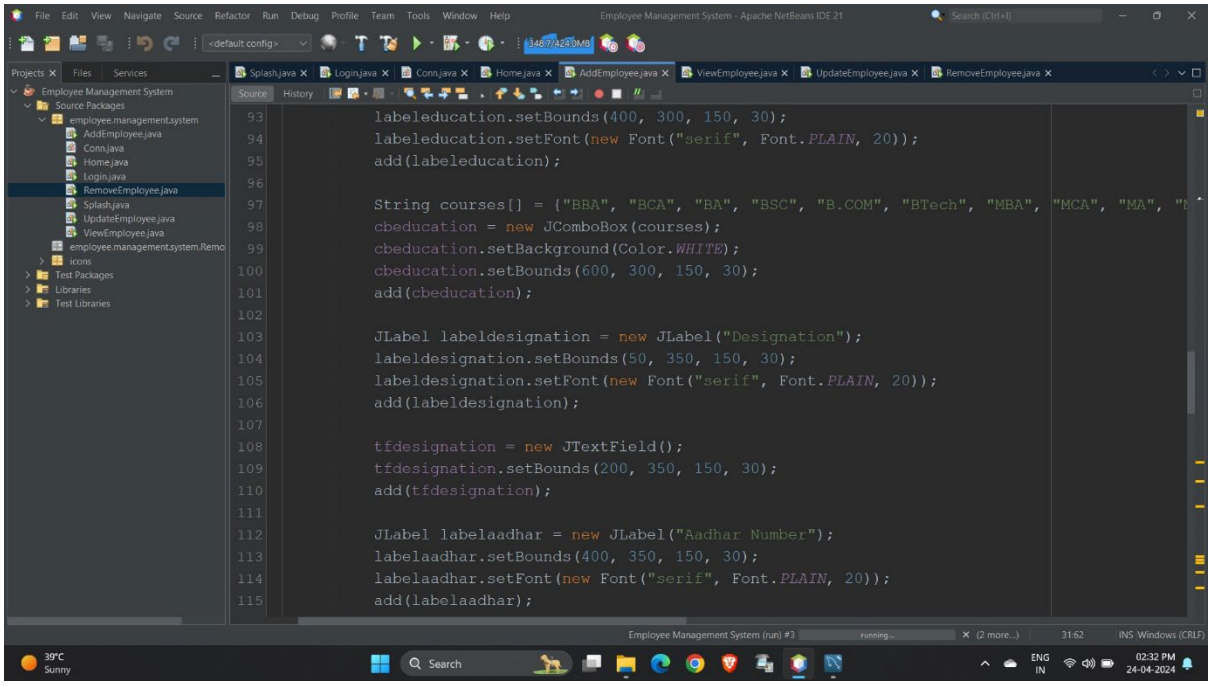
```
        JLabel heading = new JLabel("Add Employee Detail");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        add(heading);

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

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

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

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



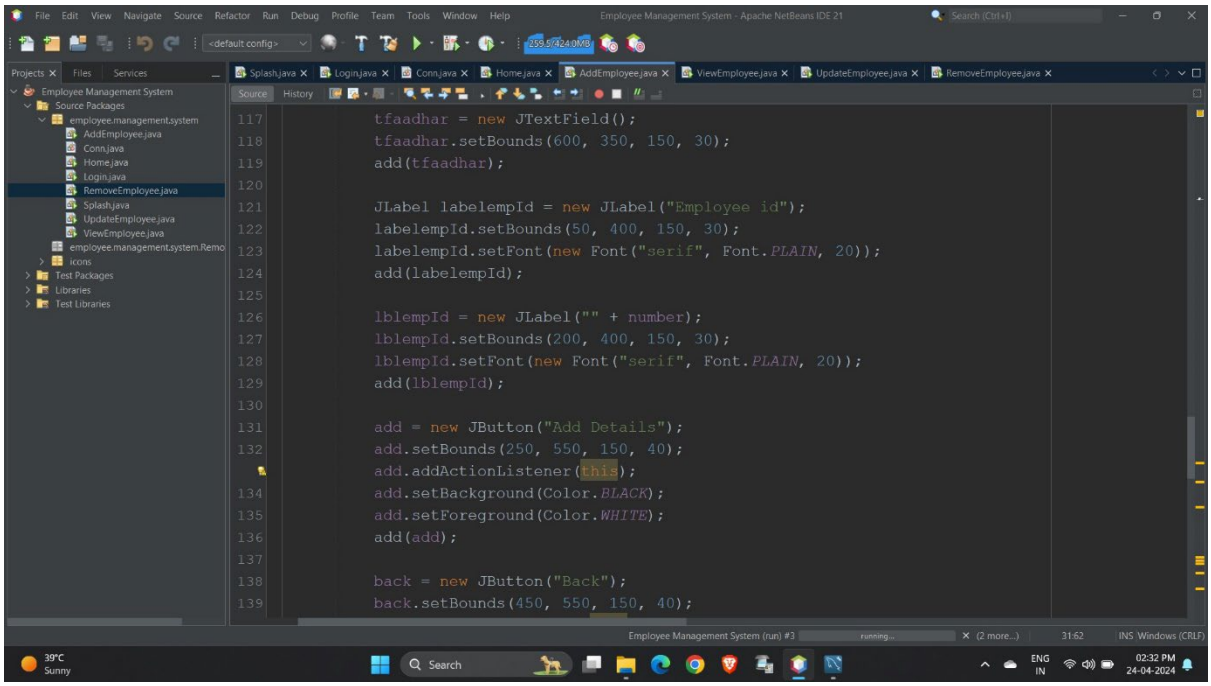
```
47          JLabel labeldob = new JLabel("Date of Birth");
48          labeldob.setBounds(50, 200, 150, 30);
49          labeldob.setFont(new Font("serif", Font.PLAIN, 20));
50          add(labeldob);
51
52          dcdob = new JDateChooser();
53          dcdob.setBounds(200, 200, 150, 30);
54          add(dcdob);
55
56          JLabel labelsalary = new JLabel("Salary");
57          labelsalary.setBounds(400, 200, 150, 30);
58          labelsalary.setFont(new Font("serif", Font.PLAIN, 20));
59          add(labelsalary);
60
61          tfsalary = new JTextField();
62          tfsalary.setBounds(600, 200, 150, 30);
63          add(tfsalary);
64
65          JLabel labeladdress = new JLabel("Address");
66          labeladdress.setBounds(50, 250, 150, 30);
67          labeladdress.setFont(new Font("serif", Font.PLAIN, 20));
68          add(labeladdress);
69
```



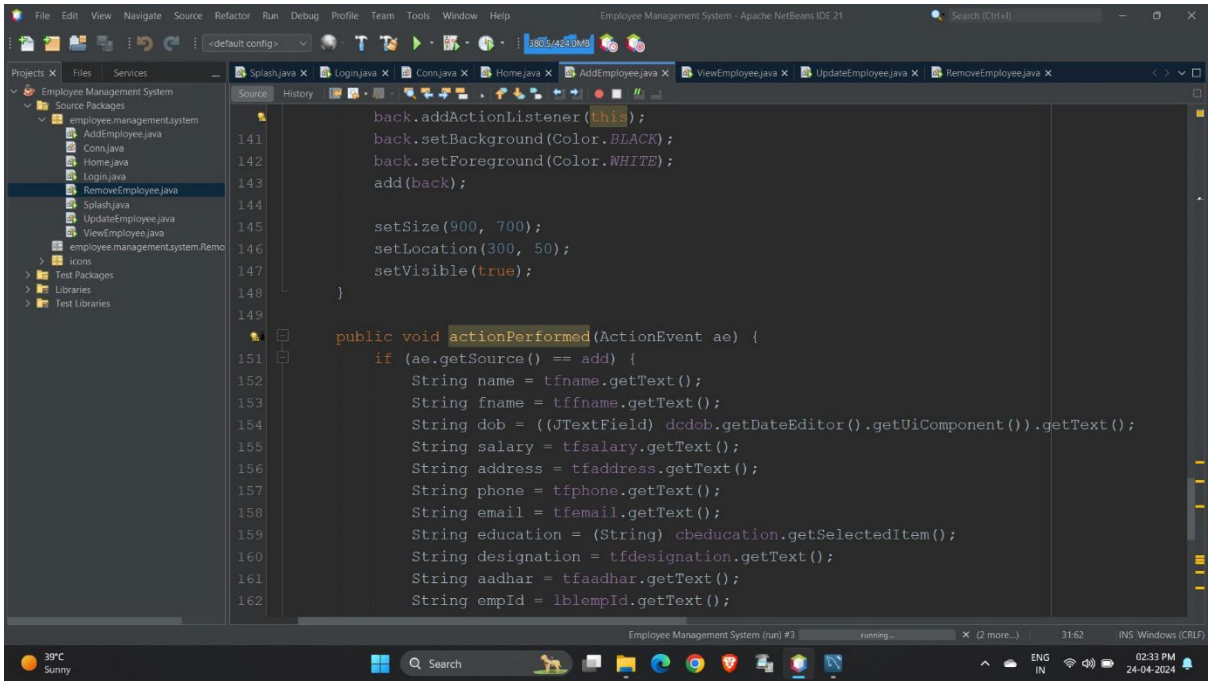
```
70          tfaddress = new JTextField();
71          tfaddress.setBounds(200, 250, 150, 30);
72          add(tfaddress);
73
74          JLabel labelphone = new JLabel("Phone");
75          labelphone.setBounds(400, 250, 150, 30);
76          labelphone.setFont(new Font("serif", Font.PLAIN, 20));
77          add(labelphone);
78
79          tfphone = new JTextField();
80          tfphone.setBounds(600, 250, 150, 30);
81          add(tfphone);
82
83          JLabel labelemail = new JLabel("Email");
84          labelemail.setBounds(50, 300, 150, 30);
85          labelemail.setFont(new Font("serif", Font.PLAIN, 20));
86          add(labelemail);
87
88          tfemail = new JTextField();
89          tfemail.setBounds(200, 300, 150, 30);
90          add(tfemail);
91
92          JLabel labeleducation = new JLabel("Higest Education");
```



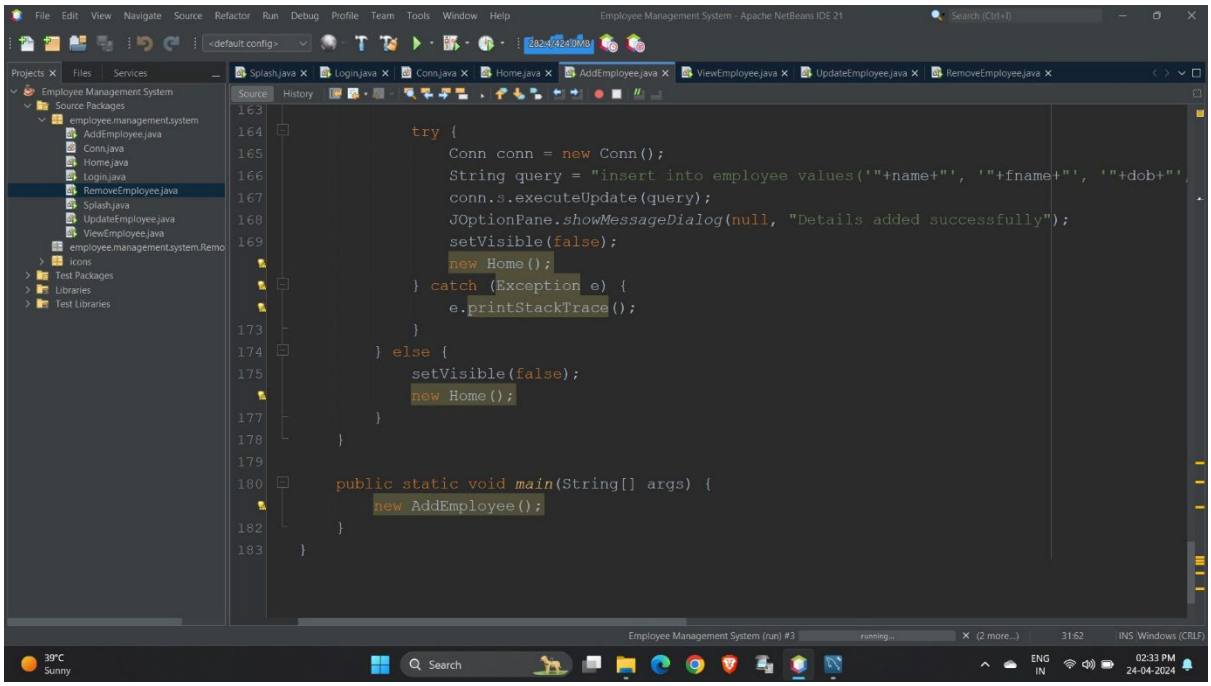
```
labeleducation.setBounds(400, 300, 150, 30);
labeleducation.setFont(new Font("serif", Font.PLAIN, 20));
add(labeleducation);

String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA", "MA", "M
cbeducation = new JComboBox(courses);
cbeducation.setBackground(Color.WHITE);
cbeducation.setBounds(600, 300, 150, 30);
add(cbeducation);

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

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

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



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

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

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

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

back = new JButton("Back");
back.setBounds(450, 550, 150, 40);
```

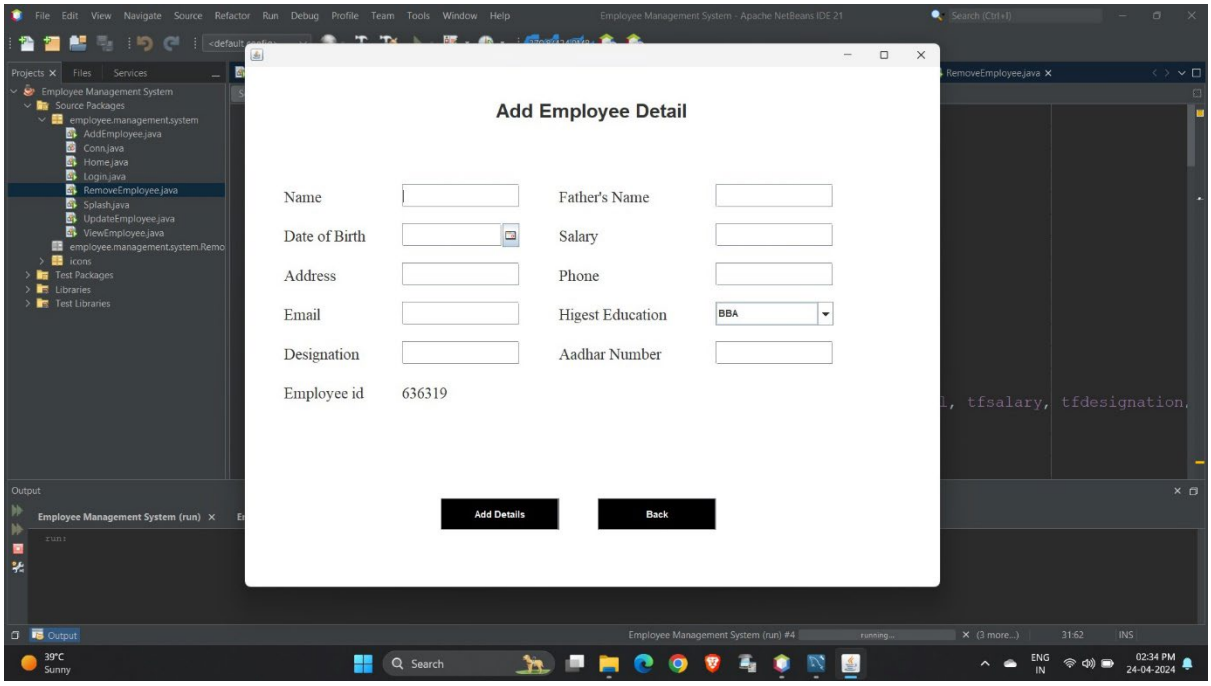


```
            back.addActionListener(this);
            back.setBackground(Color.BLACK);
            back.setForeground(Color.WHITE);
            add(back);

            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();
```

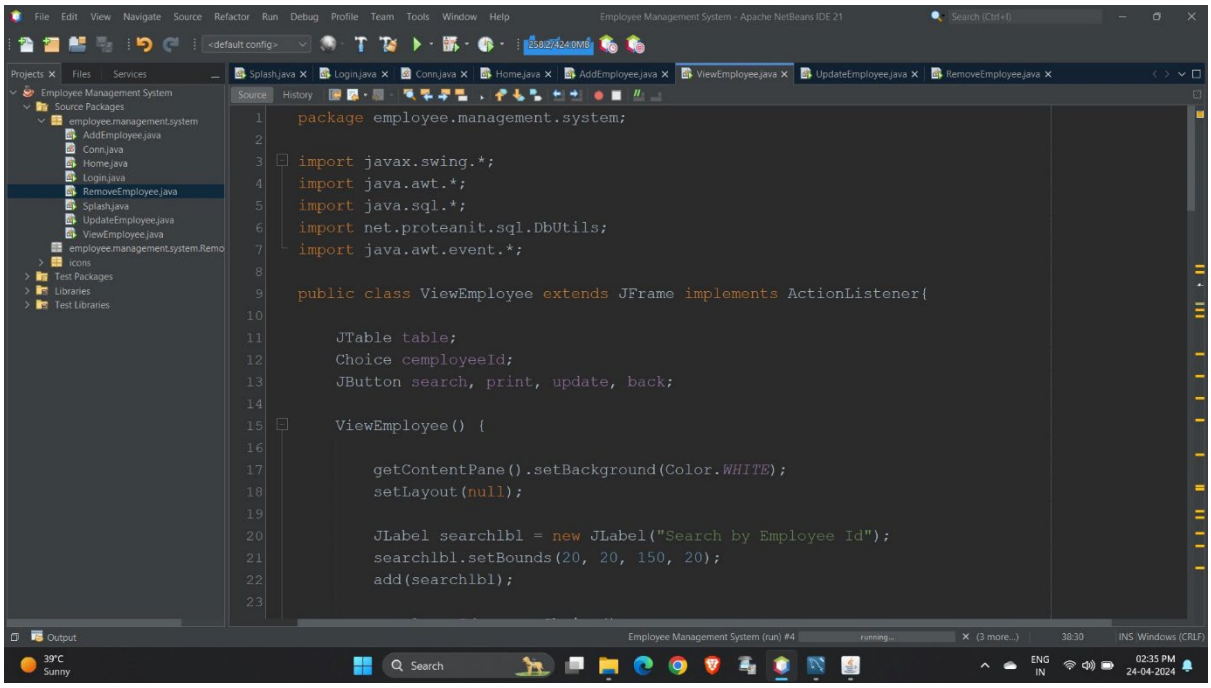


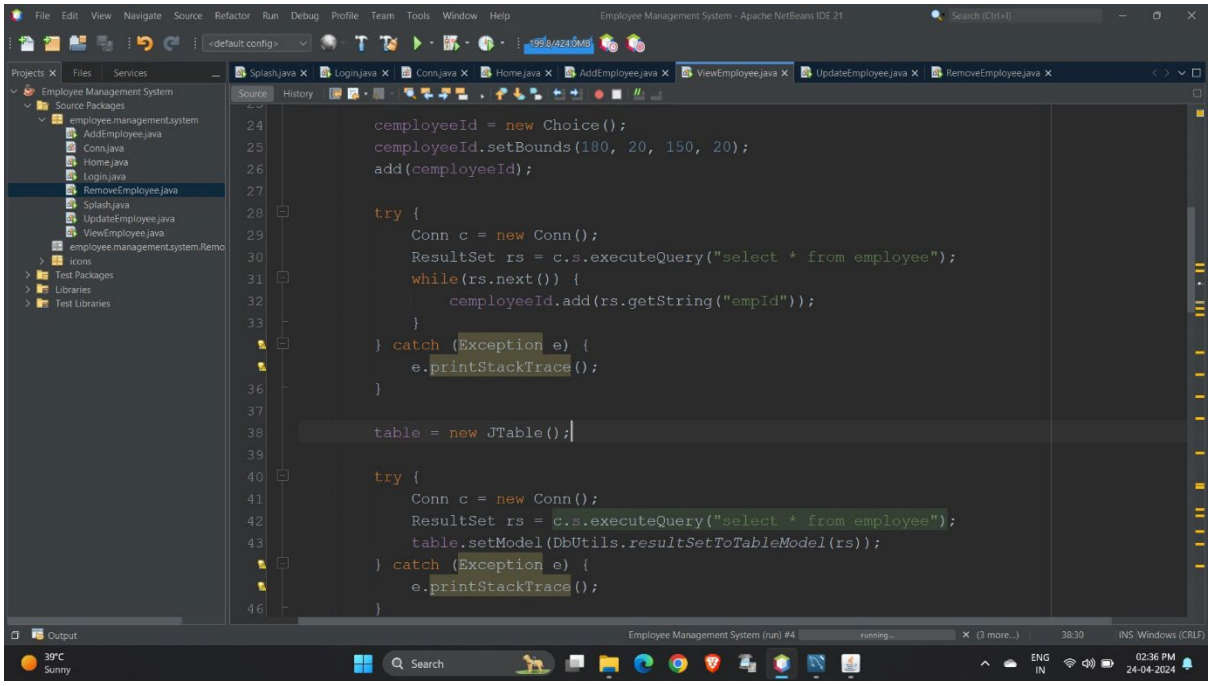
```
                try {
                    Conn conn = new Conn();
                    String query = "insert into employee values('"+name+"', '"+fname+"', '"+dob+"',
                    conn.s.executeUpdate(query);
                    JOptionPane.showMessageDialog(null, "Details added successfully");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

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

✓ Output of Add Employee Page:

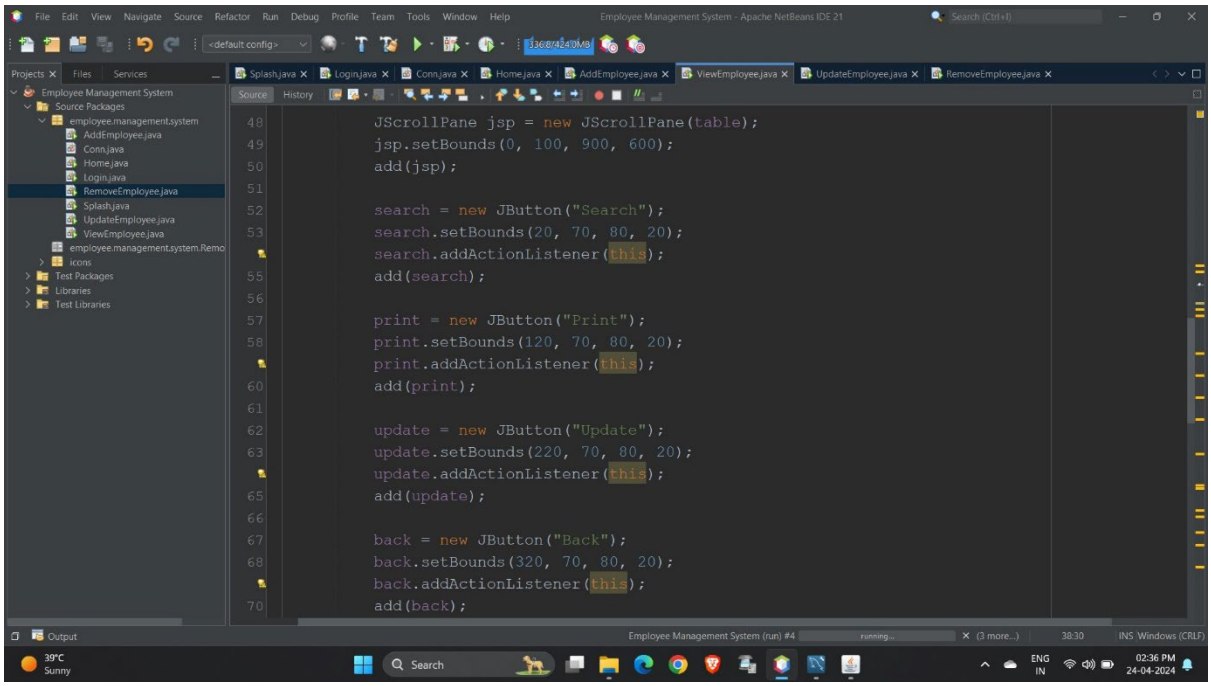


✓ View Employee Page:

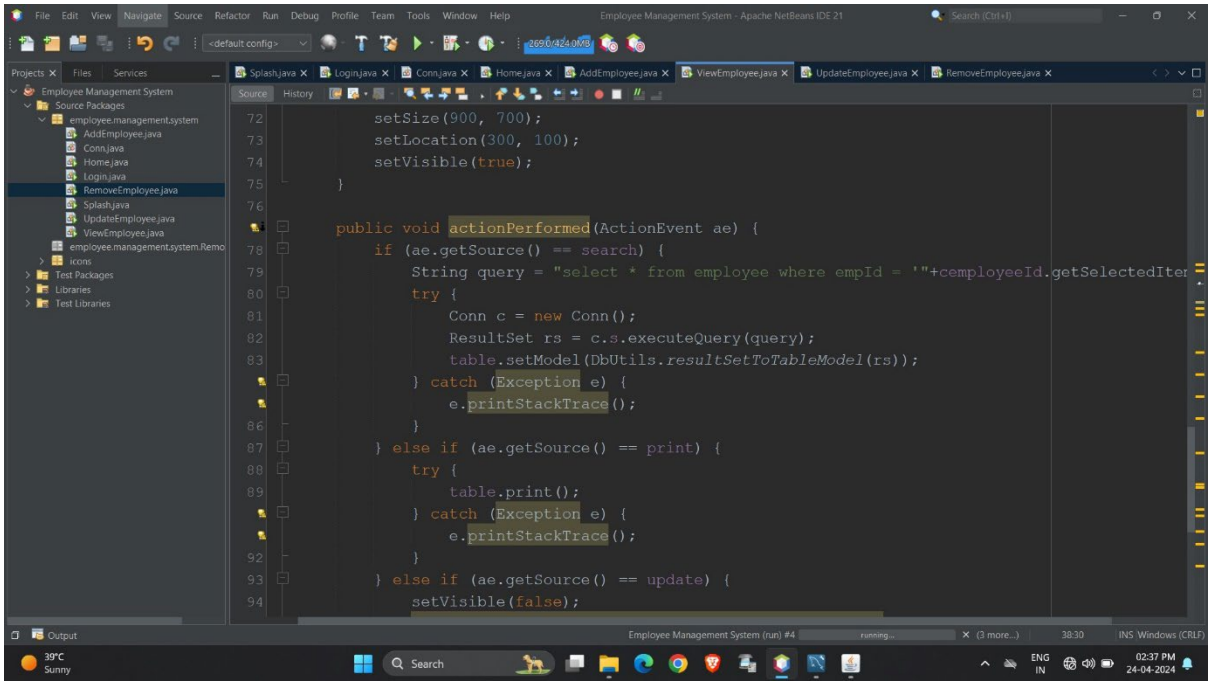




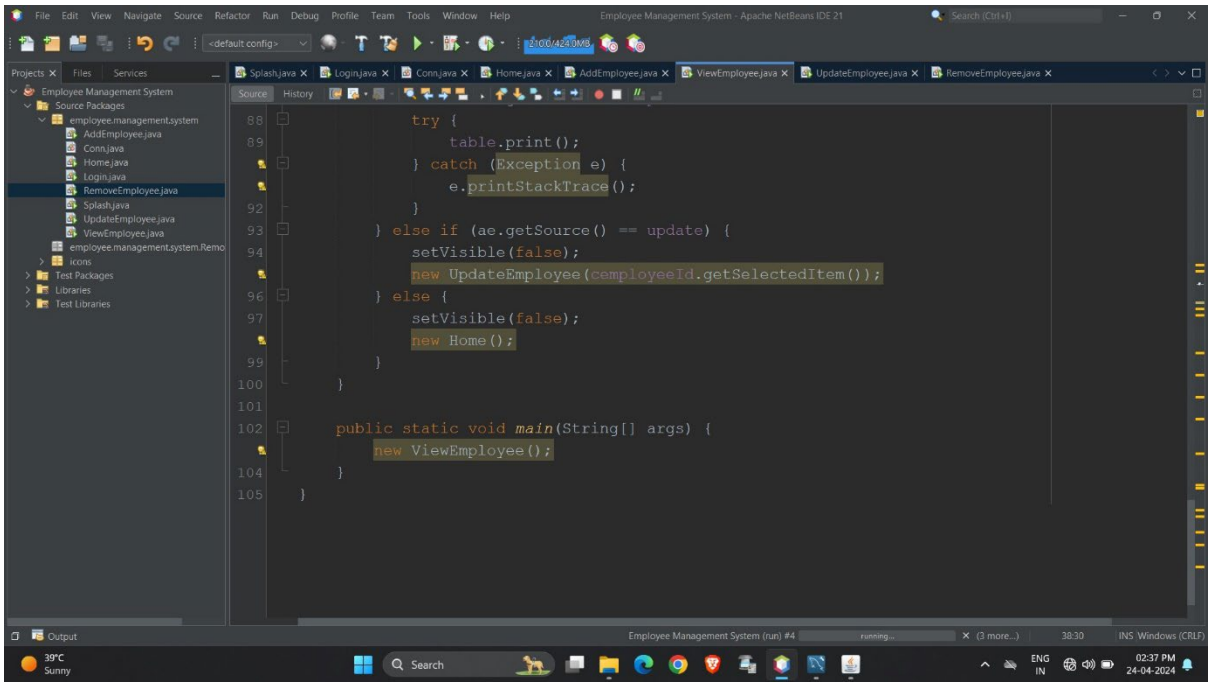
```
        employeeId = new Choice();
        employeeId.setBounds(180, 20, 150, 20);
        add(employeeId);

        try {
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery("select * from employee");
            while(rs.next()) {
                employeeId.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);
        add(jsp);

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

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

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

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

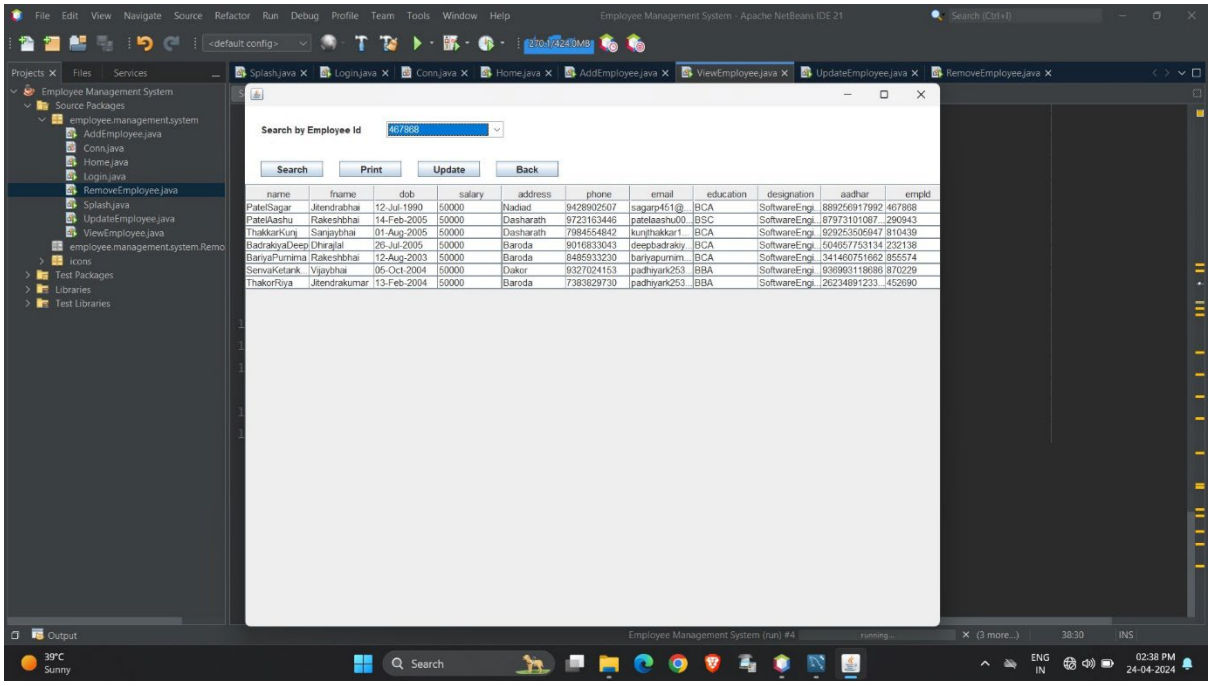


```
            setSize(900, 700);
            setLocation(300, 100);
            setVisible(true);
        }

        public void actionPerformed(ActionEvent ae) {
            if (ae.getSource() == search) {
                String query = "select * from employee where empId = '"+cemployeeId.getSelectedIter
                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);
```

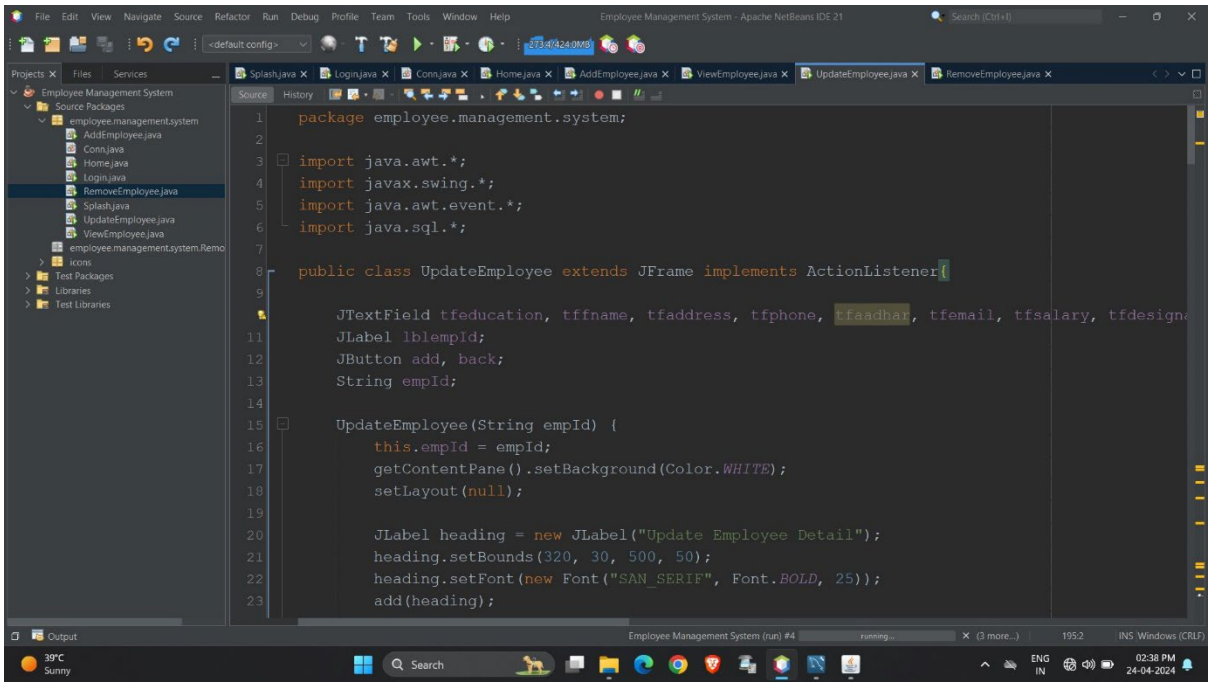


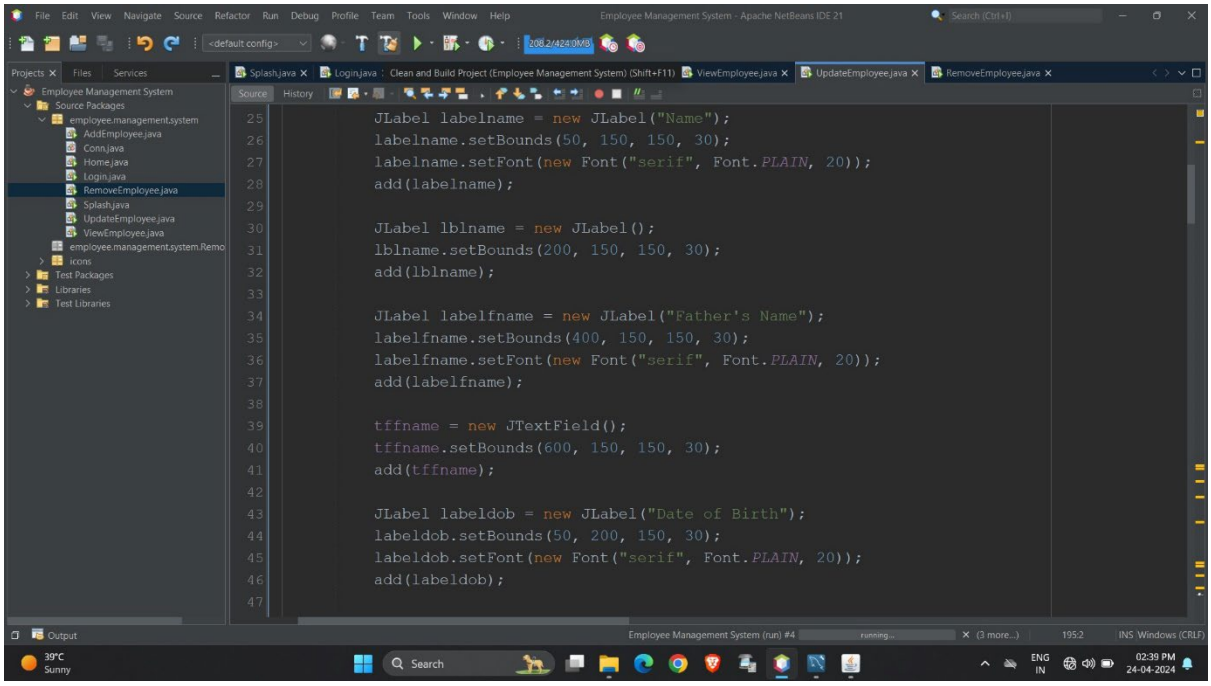
```
                try {
                    table.print();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            } else if (ae.getSource() == update) {
                setVisible(false);
                new UpdateEmployee(cemployeeId.getSelectedItem());
            } else {
                setVisible(false);
                new Home();
            }
        }

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

✓ Output of View Employee Page:

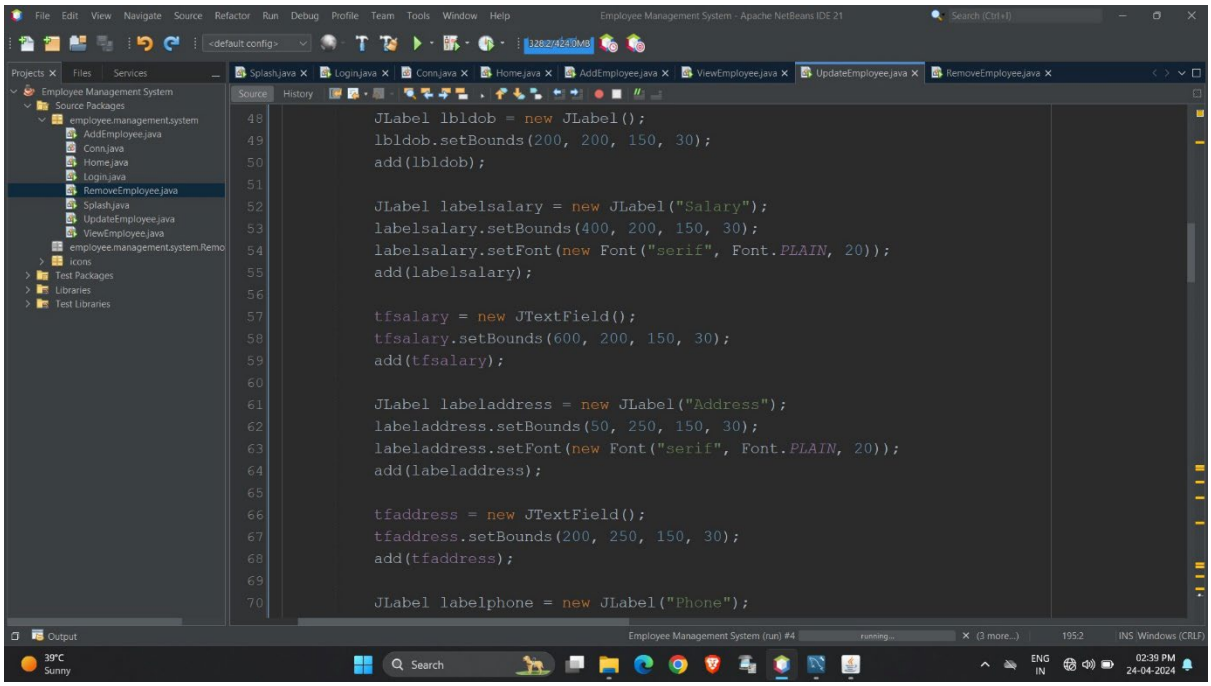


✓ Update Employee Page:

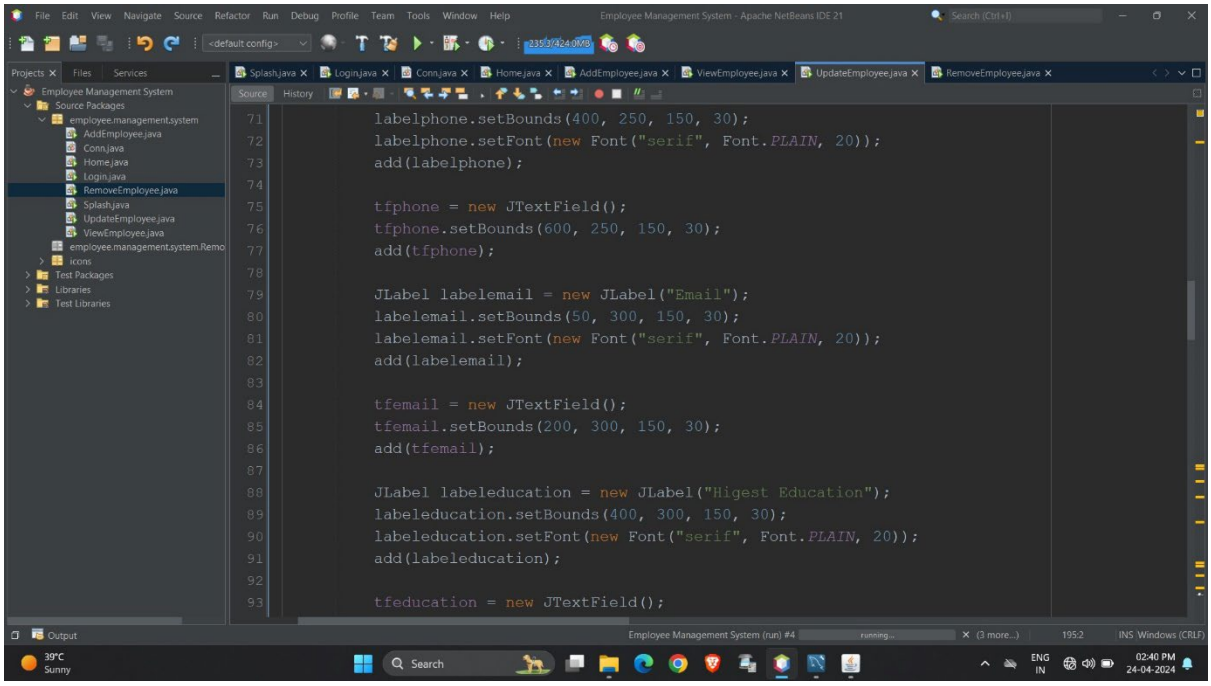




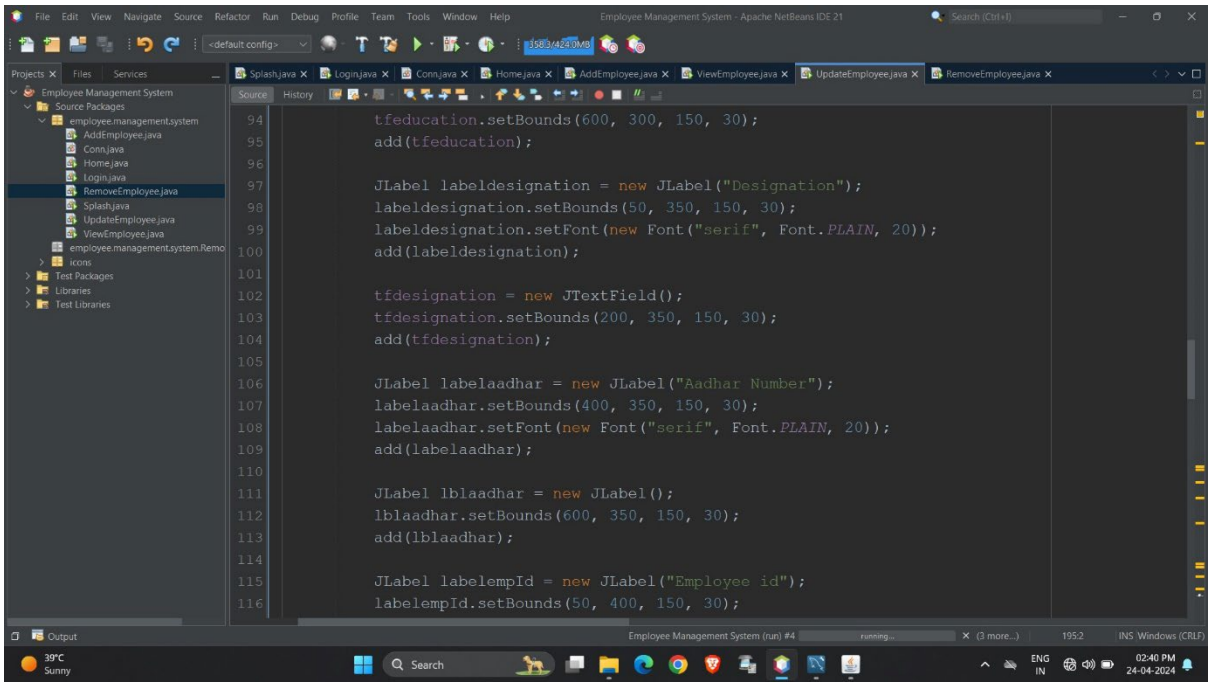
```
25          JLabel labelname = new JLabel("Name");
26          labelname.setBounds(50, 150, 150, 30);
27          labelname.setFont(new Font("serif", Font.PLAIN, 20));
28          add(labelname);
29
30          JLabel lblname = new JLabel();
31          lblname.setBounds(200, 150, 150, 30);
32          add(lblname);
33
34          JLabel labelfname = new JLabel("Father's Name");
35          labelfname.setBounds(400, 150, 150, 30);
36          labelfname.setFont(new Font("serif", Font.PLAIN, 20));
37          add(labelfname);
38
39          tffname = new JTextField();
40          tffname.setBounds(600, 150, 150, 30);
41          add(tffname);
42
43          JLabel labeldob = new JLabel("Date of Birth");
44          labeldob.setBounds(50, 200, 150, 30);
45          labeldob.setFont(new Font("serif", Font.PLAIN, 20));
46          add(labeldob);
47
```



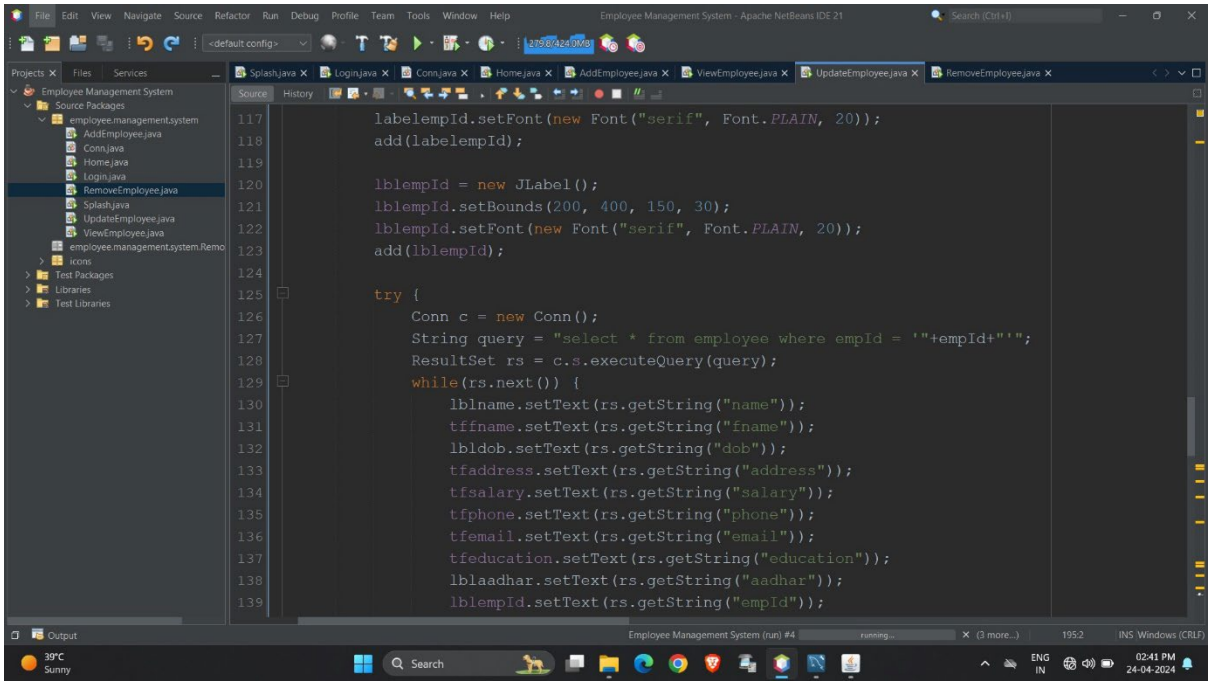
```
48          JLabel lbldob = new JLabel();
49          lbldob.setBounds(200, 200, 150, 30);
50          add(lbldob);
51
52          JLabel labelsalary = new JLabel("Salary");
53          labelsalary.setBounds(400, 200, 150, 30);
54          labelsalary.setFont(new Font("serif", Font.PLAIN, 20));
55          add(labelsalary);
56
57          tfsalary = new JTextField();
58          tfsalary.setBounds(600, 200, 150, 30);
59          add(tfsalary);
60
61          JLabel labeladdress = new JLabel("Address");
62          labeladdress.setBounds(50, 250, 150, 30);
63          labeladdress.setFont(new Font("serif", Font.PLAIN, 20));
64          add(labeladdress);
65
66          tfaddress = new JTextField();
67          tfaddress.setBounds(200, 250, 150, 30);
68          add(tfaddress);
69
70          JLabel labelphone = new JLabel("Phone");
```



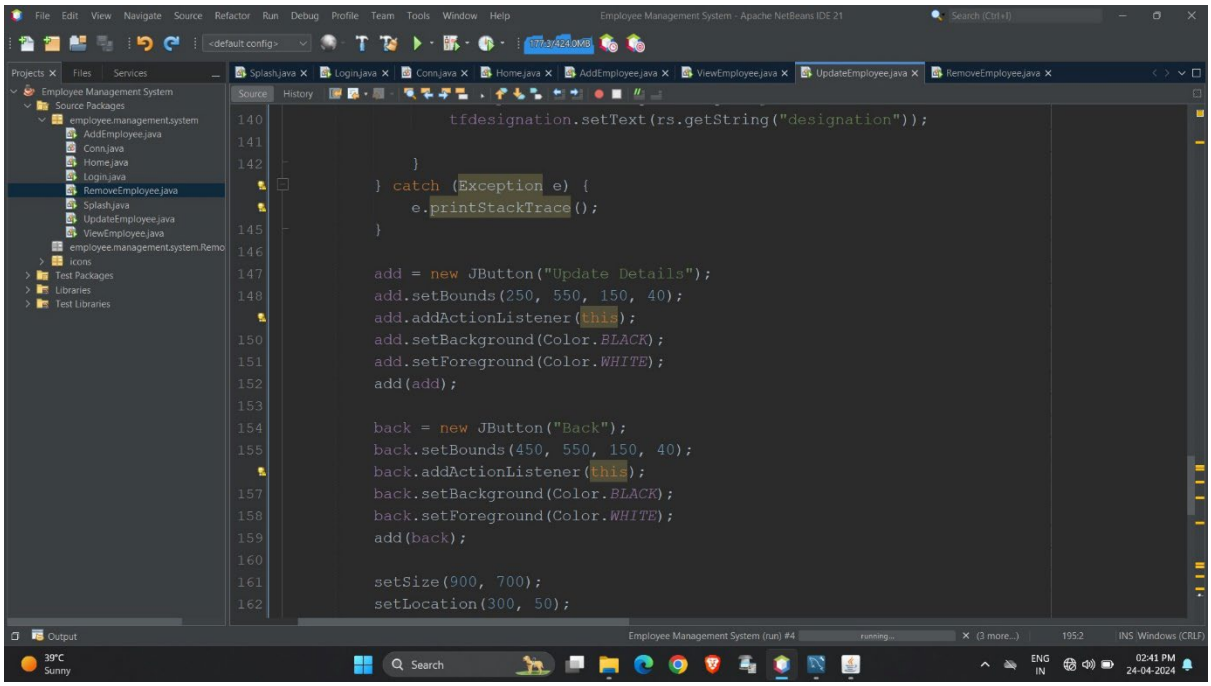
```
labelphone.setBounds(400, 250, 150, 30);
labelphone.setFont(new Font("serif", Font.PLAIN, 20));
add(labelphone);

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

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

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

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

tfeducation = new JTextField();
```



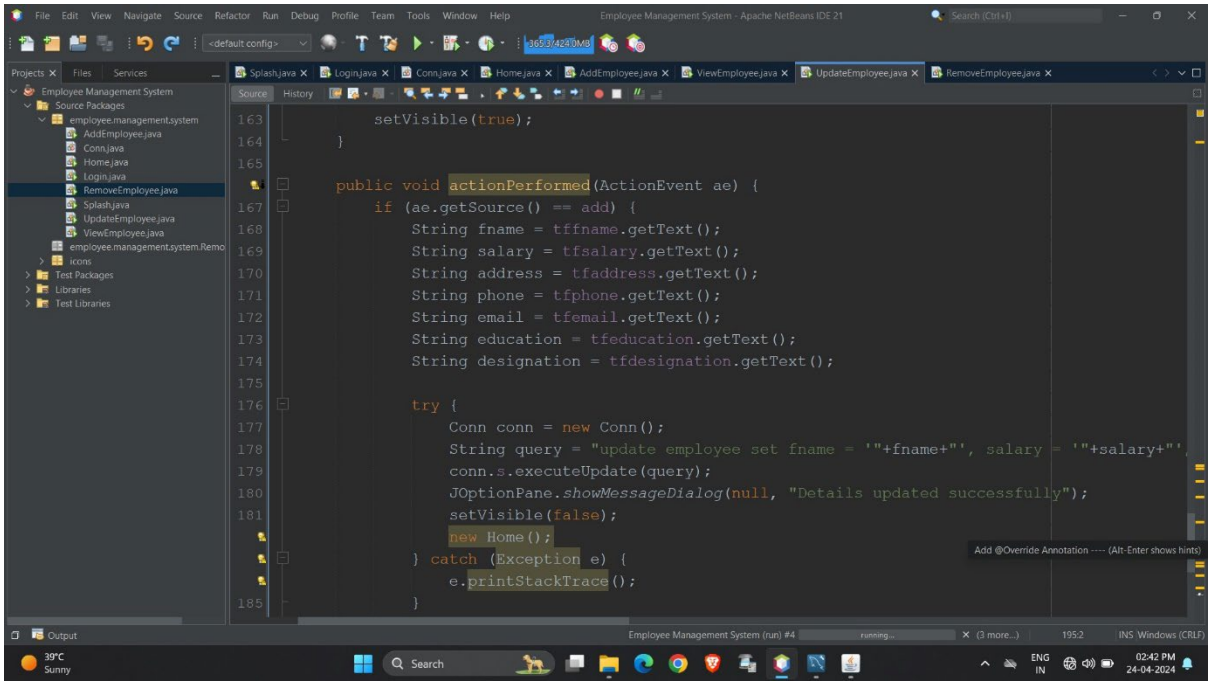
```
tfeducation.setBounds(600, 300, 150, 30);
add(tfeducation);

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

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

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

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

JLabel labelempId = new JLabel("Employee id");
labelempId.setBounds(50, 400, 150, 30);
```

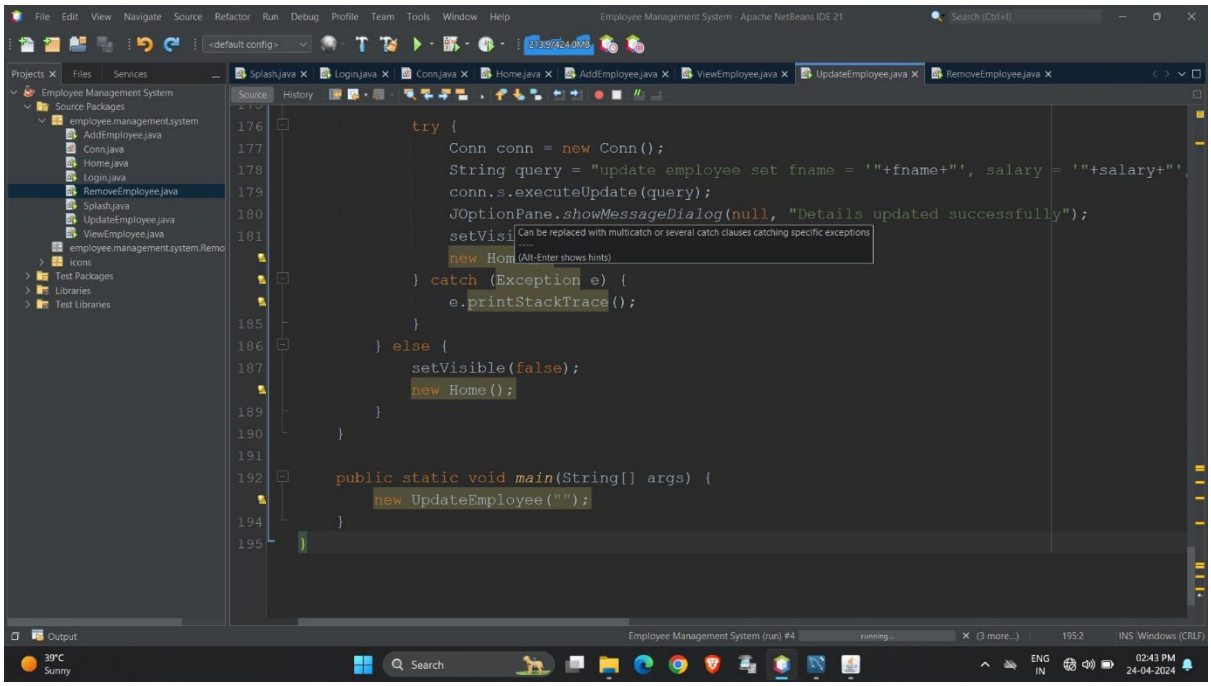


```
        labelempId.setFont(new Font("serif", Font.PLAIN, 20));
        add(labelempId);

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

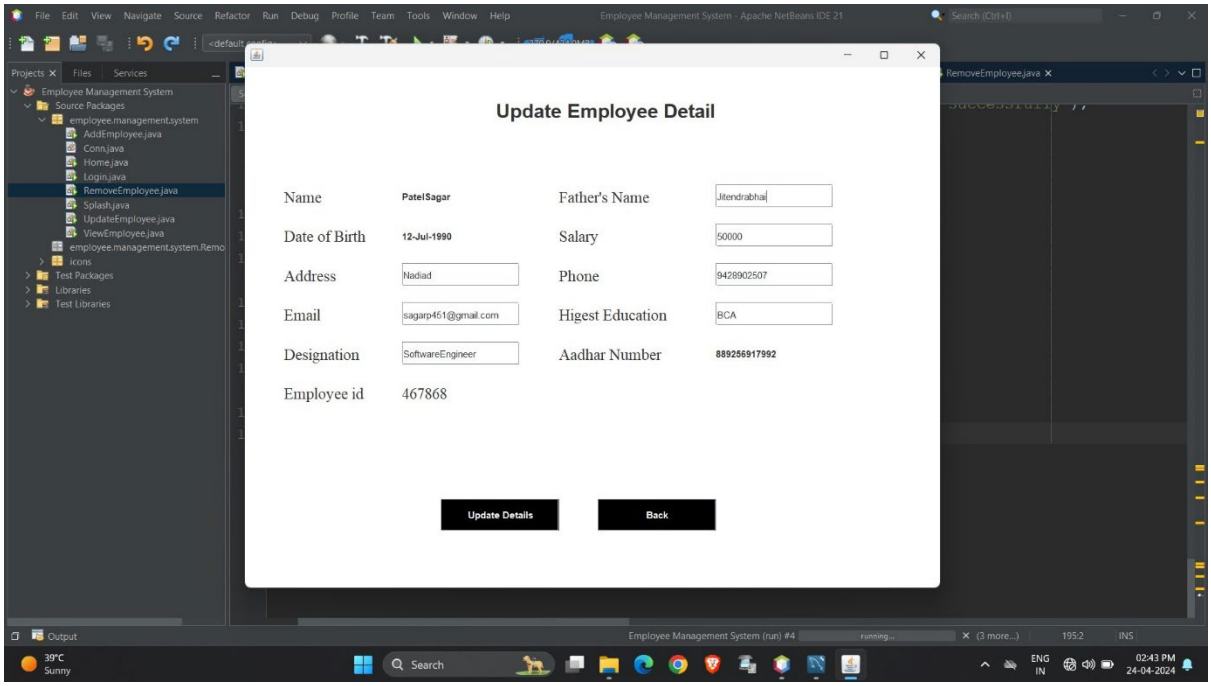


```
                tfdesignation.setText(rs.getString("designation"));

            }
        } catch (Exception e) {
            e.printStackTrace();
        }

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

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

        setSize(900, 700);
        setLocation(300, 50);
```

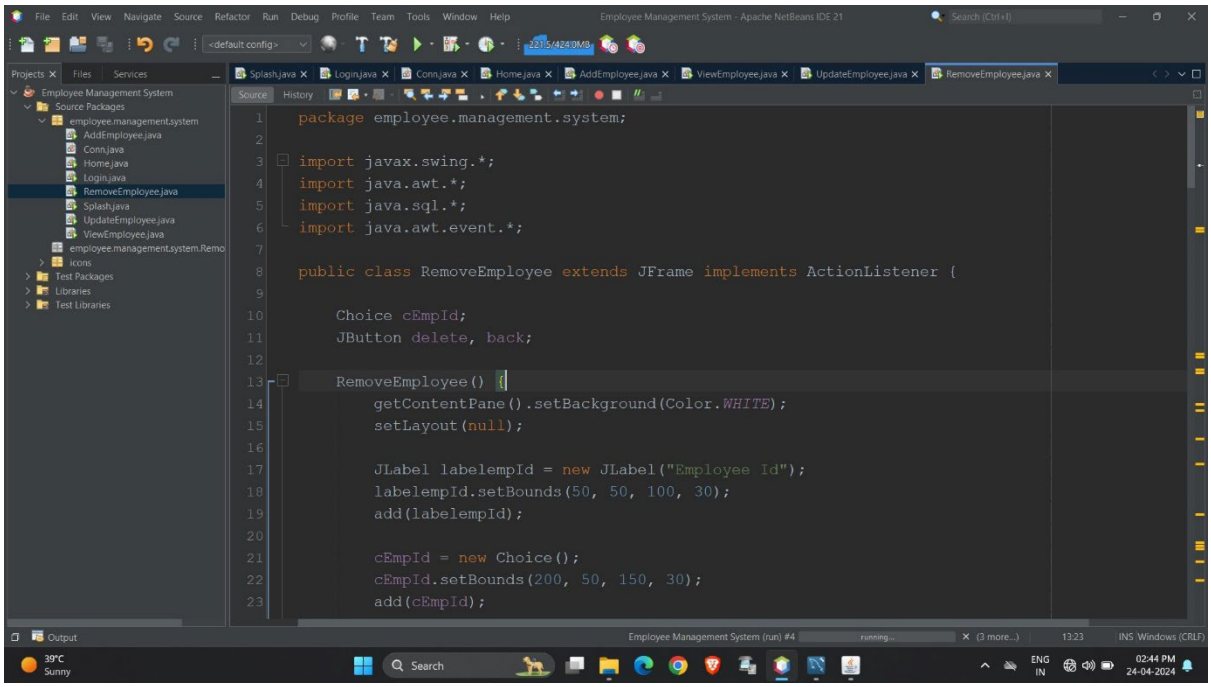

Employee Management System - Apache NetBeans IDE 21
UpdateEmployee.java
setVisible(true);
}
public void actionPerformed(ActionEvent ae) {
if (ae.getSource() == add) {
String fname = tffname.getText();
String salary = tfsalary.getText();
String address = tfaddress.getText();
String phone = tfphone.getText();
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+"',
conn.s.executeUpdate(query);
JOptionPane.showMessageDialog(null, "Details updated successfully");
setVisible(false);
new Home();
} catch (Exception e) {
e.printStackTrace();
}
Add @Override Annotation ---- (Alt-Enter shows hints)

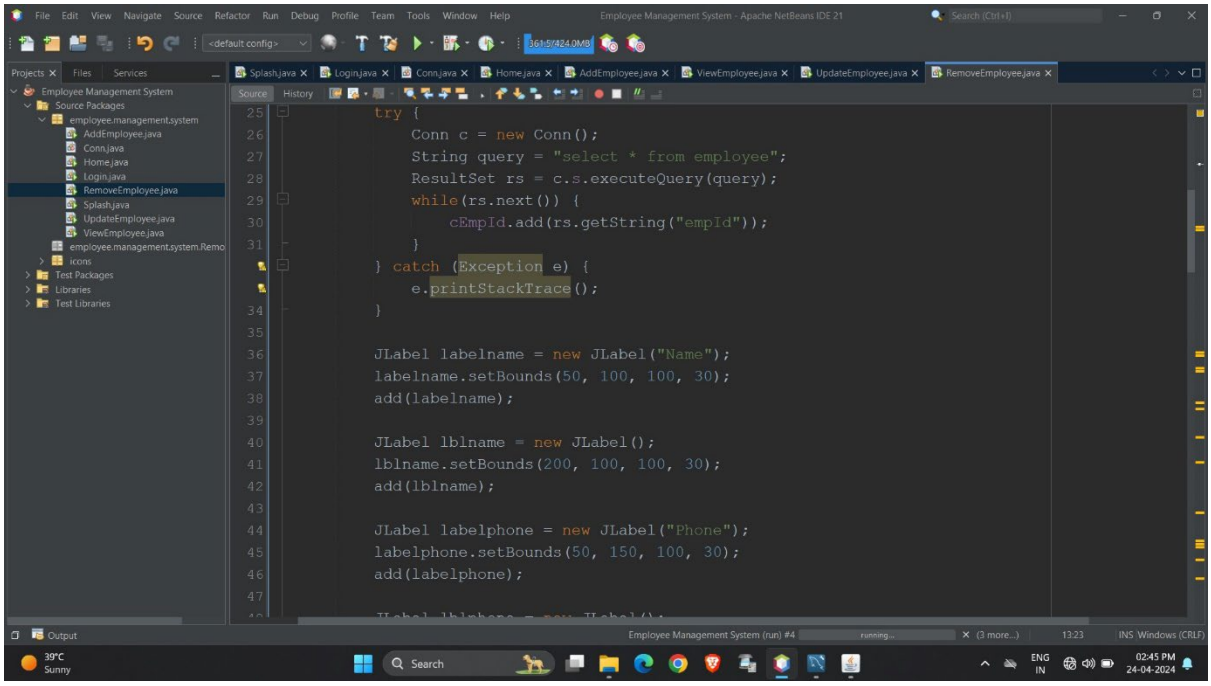

Employee Management System - Apache NetBeans IDE 21
UpdateEmployee.java
try {
Conn conn = new Conn();
String query = "update employee set fname = '"+fname+"', salary = '"+salary+"',
conn.s.executeUpdate(query);
JOptionPane.showMessageDialog(null, "Details updated successfully");
setVisi
new Hom
Can be replaced with multicatch or several catch clauses catching specific exceptions
(Alt-Enter shows hints)
} catch (Exception e) {
e.printStackTrace();
}
} else {
setVisible(false);
new Home();
}
}
public static void main(String[] args) {
new UpdateEmployee("");
}
}

✓ Output of Update Employee Page:

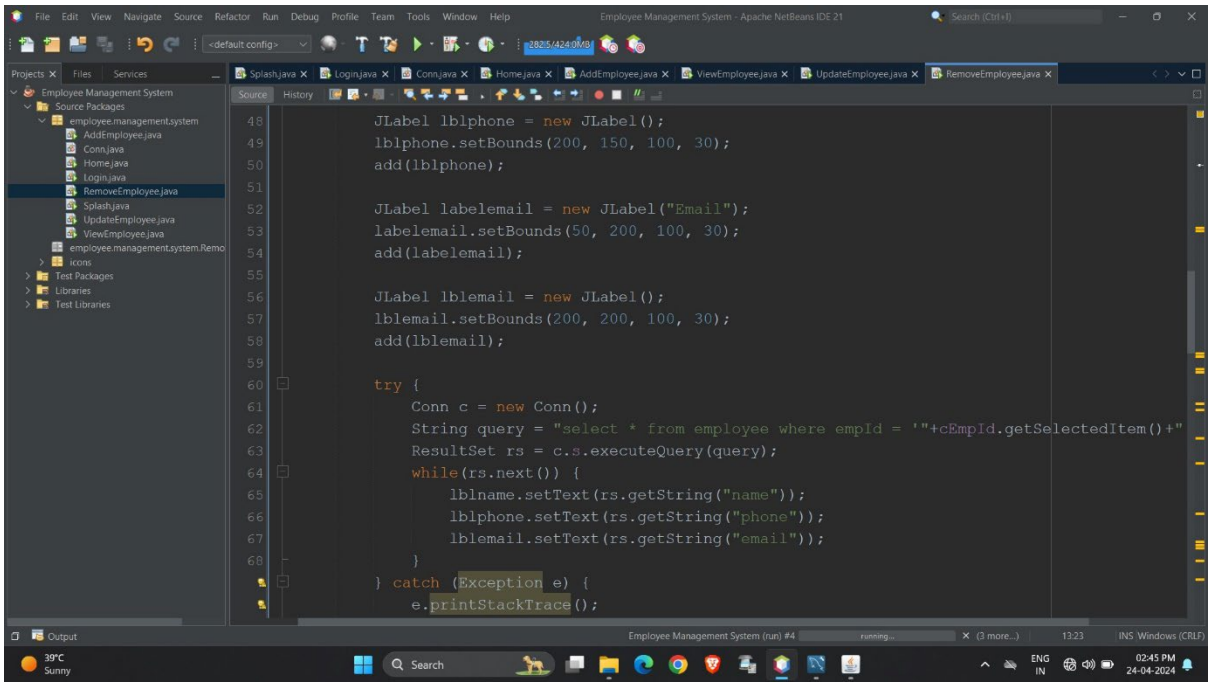


✓ Remove Employee Page:

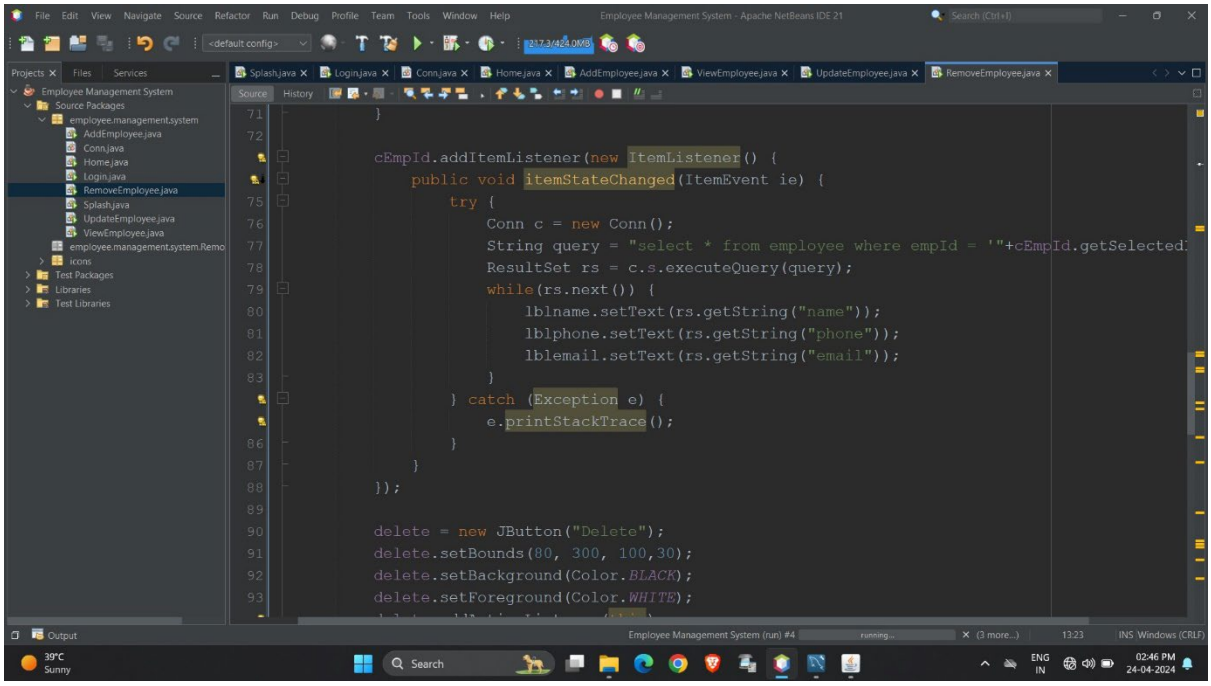




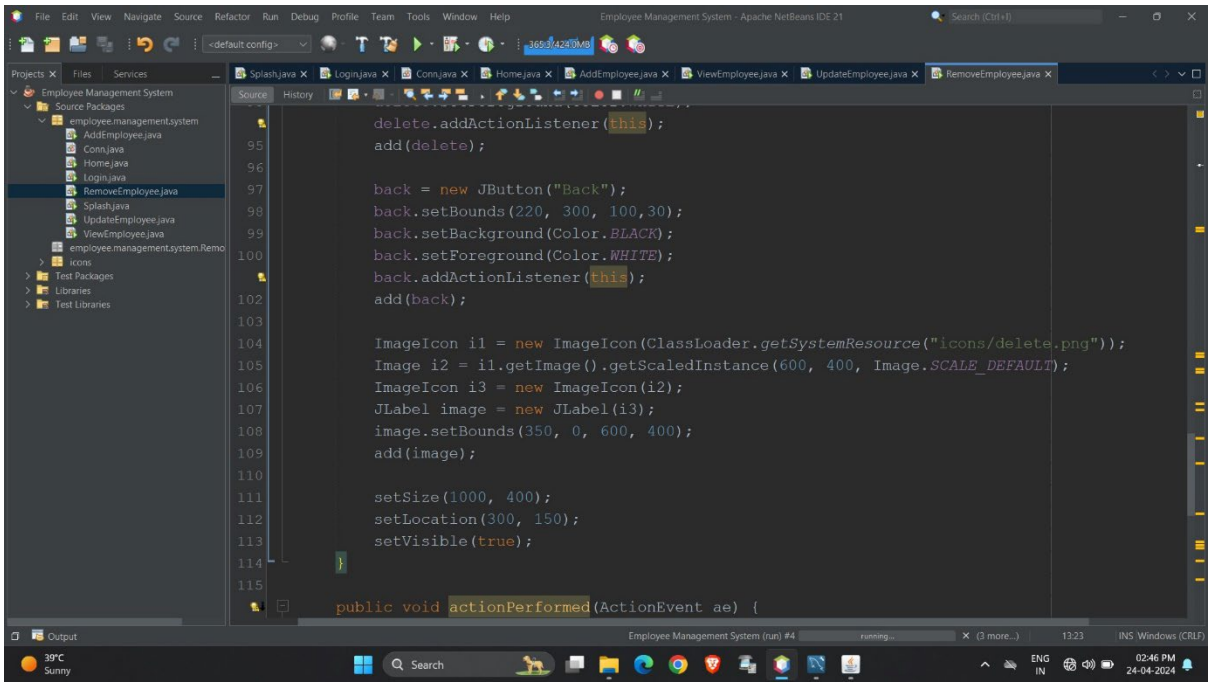
```
try {
    Conn c = new Conn();
    String query = "select * from employee";
    ResultSet rs = c.s.executeQuery(query);
    while(rs.next()) {
        cEmpId.add(rs.getString("empId"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

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);
```



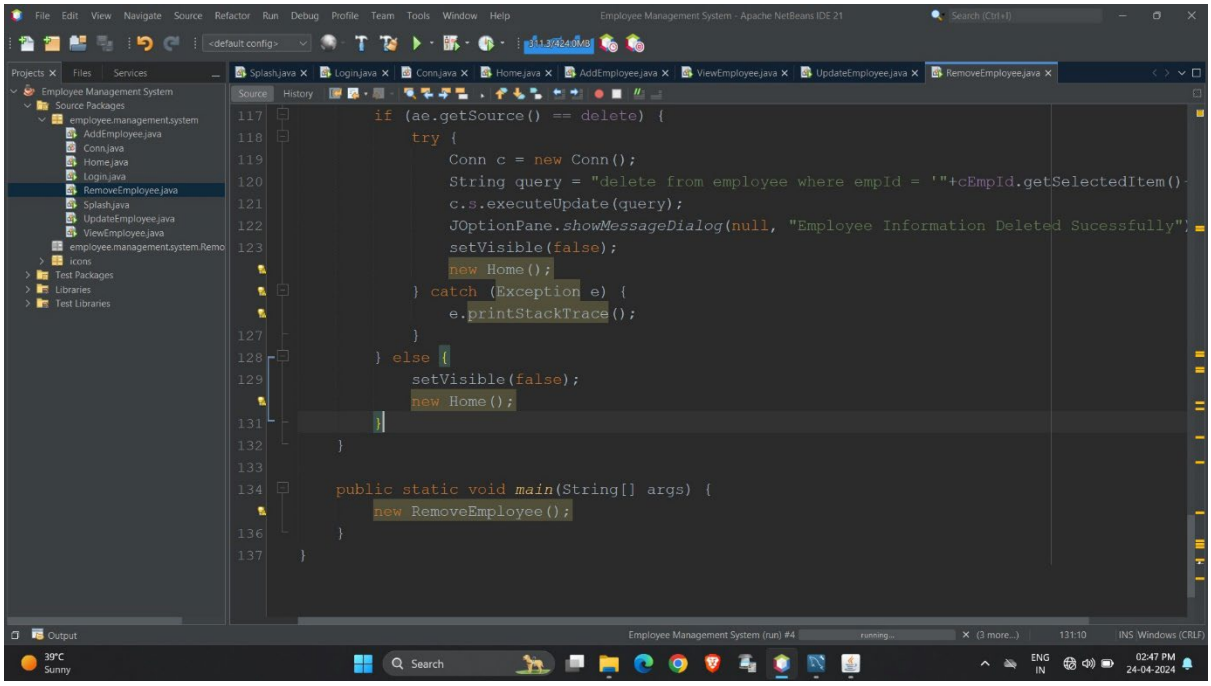
```
JLabel lblphone = new JLabel();
lblphone.setBounds(200, 150, 100, 30);
add(lblphone);

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();
```



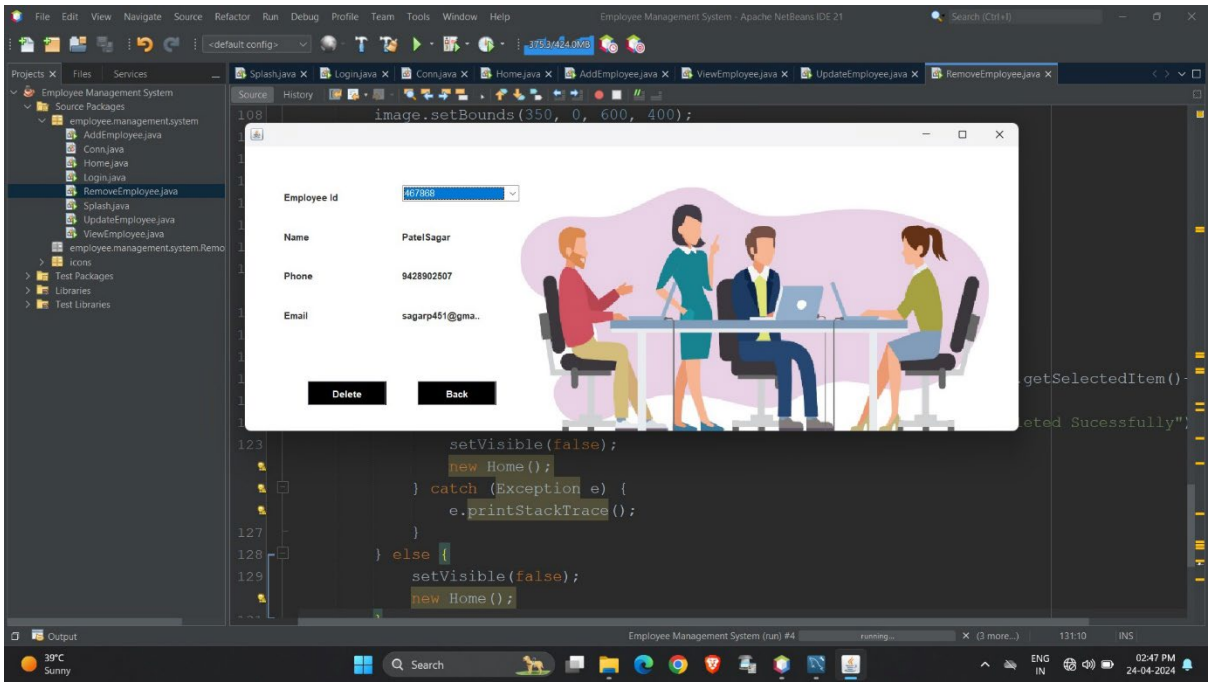
```
        }

        cEmpId.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ie) {
                try {
                    Conn c = new Conn();
                    String query = "select * from employee where empId = '"+cEmpId.getSelected
                    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.getSystemResource("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, 400);
        setLocation(300, 150);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
```



✓ Output of Remove Employee Page:



CHAPTER – 11

➢ CONCLUSION:

The goal of the initiative is to digitize 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.

CHAPTER – 12

➢ 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.

CHAPTER – 13

➢ REFERENCES:

1. Renae Broderick, John W. Boudreau, "Human resource management, information technology, and the competitive edge", Academy of Management Executive, 1992 Vol. 6 No. 2
2. Julie Bulmash, "Human Resource Management and Technology", Chapter 3.
3. Ian Sommerville, "Software Engineering", 9th Edition, Addison-Wesley, 2011.
4. Avison, D. and Fitzgerald, G. (2003). Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire.
5. Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010.
6. Deitel, PJ & Deitel, HM, 2008 Internet & World Wide Web How To Program, Dorling Kindersley, India.

CHAPTER – 14

➢ BIBLIOGRAPHY:

- ✓ www.google.com
- ✓ www.geekforgeeks.com
- ✓ www.wikipedia.com