

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

No	Description	Page No.
1.	A Project Report	01
2.	Certificate of the Guide	03
3.	Acknowledgement	04
4.	Declaration	05
5.	Evaluation	06
6.	Introduction	08
7.	Objective	09
8.	Methodology	10
9.	Feasibility Study	12
10.	Design	13
11.	Describe How to Run A Program?	14
12.	Implementation and Results	15
13.	Software & Hardware Requirement Specification	21
14.	Testing	22
15.	Code Screenshot	23
16.	Conclusion	48
17.	Future Scope	49
18.	References	50
19.	Bibliography	51

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.

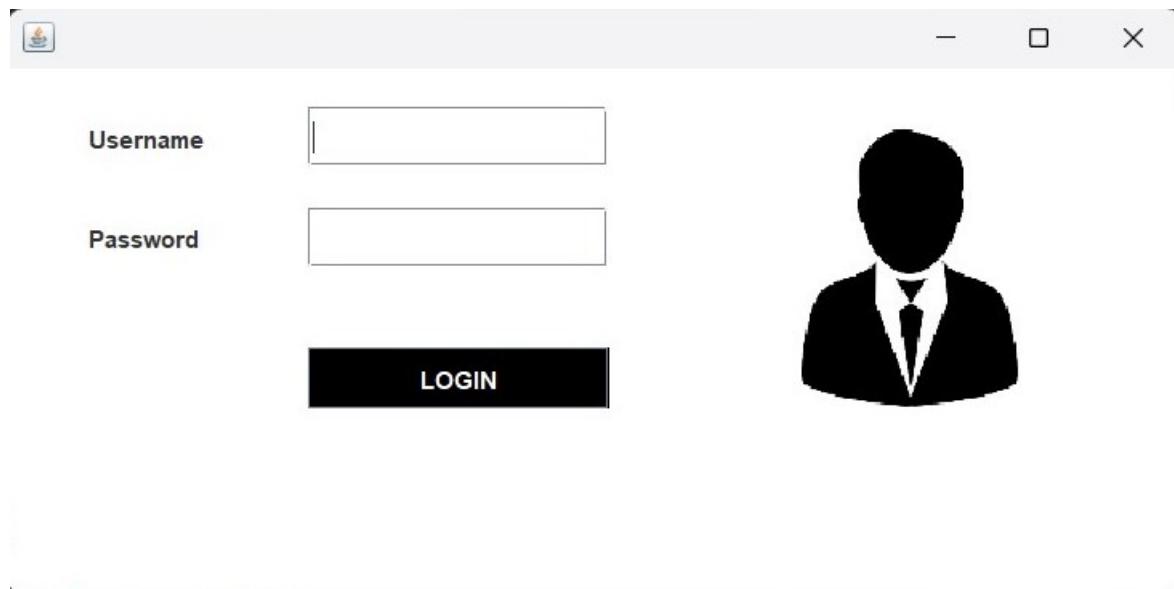
CHAPTER – 7

➤ 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.

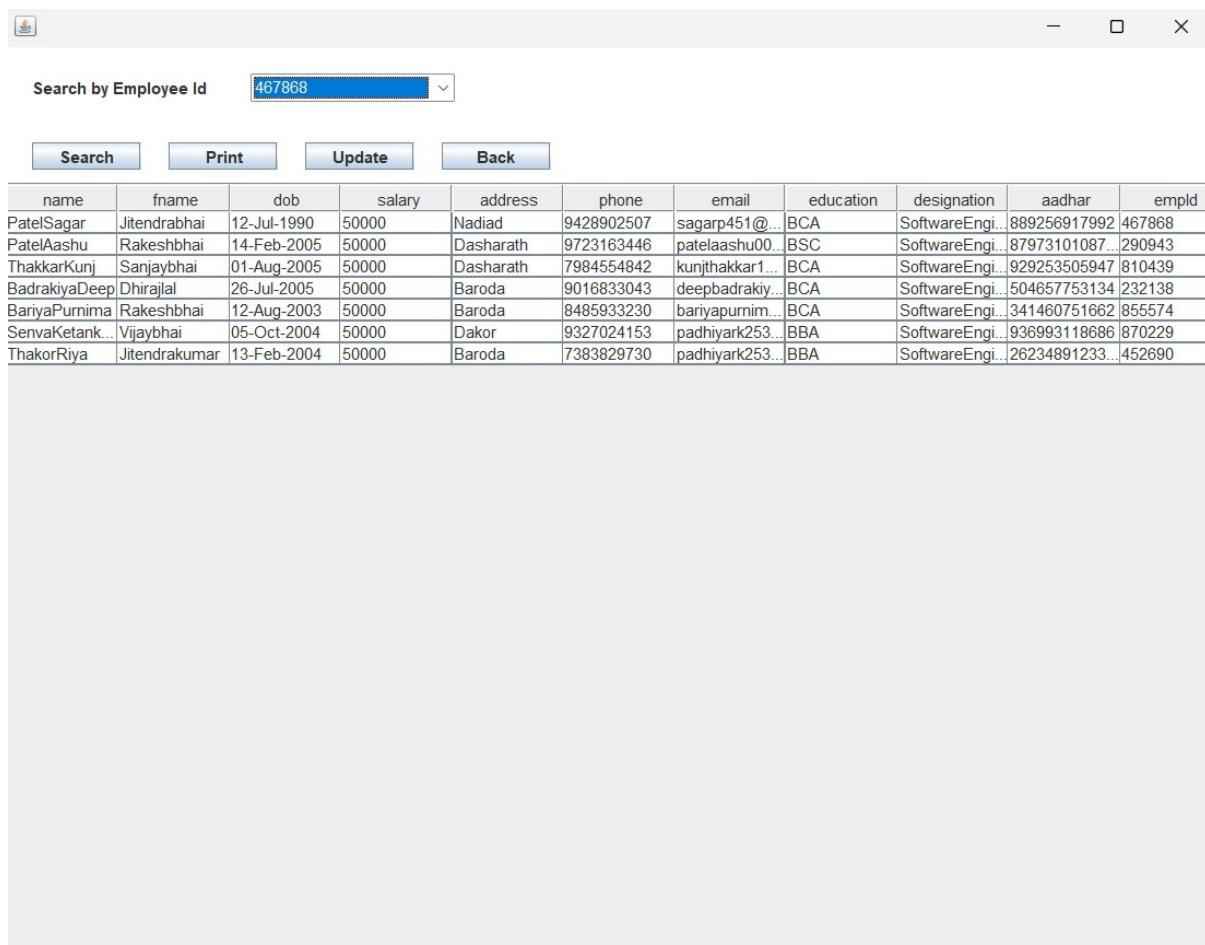
The screenshot shows a Windows-style application window titled "Add Employee Detail". The form contains fields for entering employee information. The fields are arranged in pairs, separated by a vertical line. The first pair includes "Name" and "Father's Name". The second pair includes "Date of Birth" with a calendar icon and "Salary". The third pair includes "Address" and "Phone". The fourth pair includes "Email" and "Higest Education" with a dropdown menu showing "BBA". The fifth pair includes "Designation" and "Aadhar Number". Below the form, there are two buttons: "Add Details" and "Back".

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"/>	Higest Education	<input type="text"/> BBA
Designation	<input type="text"/>	Aadhar Number	<input type="text"/>
Employee id	636319		

Add Details **Back**

4. View Employee:

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



name	fname	dob	salary	address	phone	email	education	designation	aadhar	empld
PatelSagar	Jitendrabhai	12-Jul-1990	50000	Nadiad	9428902507	sagarp451@...	BCA	SoftwareEngi...	889256917992	467868
PatelAashu	Rakeshbhai	14-Feb-2005	50000	Dasharath	9723163446	patealaashu00...	BSC	SoftwareEngi...	87973101087...	290943
ThakkarKunj	Sanjaybhai	01-Aug-2005	50000	Dasharath	7984554842	kunjthakkar1...	BCA	SoftwareEngi...	929253505947	810439
BadrakiyaDeep	Dhirajlal	26-Jul-2005	50000	Baroda	9016833043	deepbadrakiy...	BCA	SoftwareEngi...	504657753134	232138
BariyaPurnima	Rakeshbhai	12-Aug-2003	50000	Baroda	8485933230	bariyapurnim...	BCA	SoftwareEngi...	341460751662	855574
SenvaKetank...	Vijaybhai	05-Oct-2004	50000	Dakor	9327024153	padhiyark253...	BBA	SoftwareEngi...	936993118686	870229
ThakorRiya	Jitendrakumar	13-Feb-2004	50000	Baroda	7383829730	padhiyark253...	BBA	SoftwareEngi...	26234891233...	452690

5. Update Employee:

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

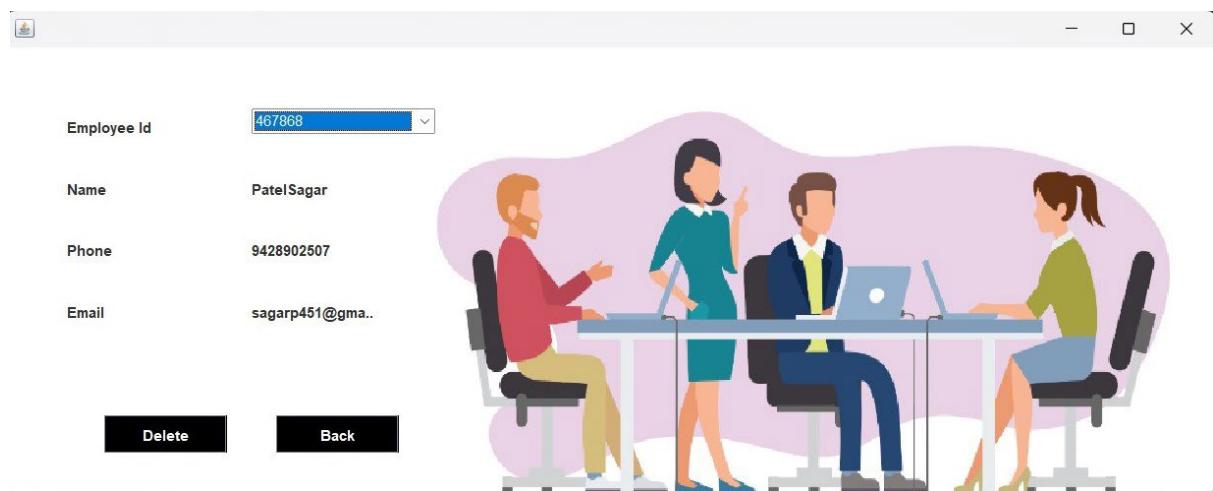
Update Employee Detail

Name	PatelSagar	Father's Name	Jitendrabhai
Date of Birth	12-Jul-1990	Salary	50000
Address	Nadiad	Phone	9428902507
Email	sagarp451@gmail.com	Higest Education	BCA
Designation	SoftwareEngineer	Aadhar Number	889256917992
Employee id	467868		

Update Details **Back**

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:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `employeeemanagementsystem` containing tables `college`, `employee`, `news`, `stored Procedures`, `hospital`, `sakila`, `sys`, and `world`.
- SQL File 3:** Displays the following SQL code:

```
1 • create database employeeemanagementsystem;
2
3 • use employeeemanagementsystem;
4
5 • create table login(username varchar(20), password varchar(20));
6
7 • insert into login values('sagarpatel', '9428902507');
8 • insert into login values('aashupatel', '9723163446');
9 • insert into login values('kunjthakkar', '7984554842');
10
11 • select * from login;
12
```
- Output:** Shows the execution results:

Action	Message	Duration / Fetch
3 05:38:06 create table login(username varchar(20), password varchar(20))	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
4 05:38:12 create table login(username varchar(20), password varchar(20))	0 row(s) affected	0.094 sec
5 05:38:19 insert into login values('sagarpatel', '9428902507')	1 row(s) affected	0.015 sec
6 05:38:19 insert into login values('aashupatel', '9723163446')	1 row(s) affected	0.015 sec
7 05:38:19 insert into login values('kunjthakkar', '7984554842')	1 row(s) affected	0.015 sec
8 05:38:24 select * from login LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `employeeemanagementsystem` containing tables `college`, `hospital`, `sakila`, `sys`, and `world`.
- SQL File 3:** Displays the following SQL code:

```
13 • create table employee
14   (
15     name varchar(20),
16     fname varchar(20),
17     dob varchar(30),
18     salary varchar(20),
19     address varchar(100),
20     phone varchar(20),
21     email varchar(40),
22     education varchar(20),
23     designation varchar(30),
24     aadhar varchar(25),
25     empld varchar(15));
26 • select * from employee;
```

✓ Login Table:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the `employeeManagementSystem` database selected.
- SQL File 3:** Contains the SQL query: `10
11 • select * from login;`
- Result Grid:** Displays the data from the `login` table:

username	password
sagarpatel	9428902507
aashupatel	9723163446
kunjthakkar	7984554842
- Session:** Shows the session information: `login 2`, `Object Info`, `Session`.
- System Bar:** Shows the system tray with icons for battery, network, and date/time: `28°C Smoke`, `ENG IN`, `05:40 AM 25-04-2024`.

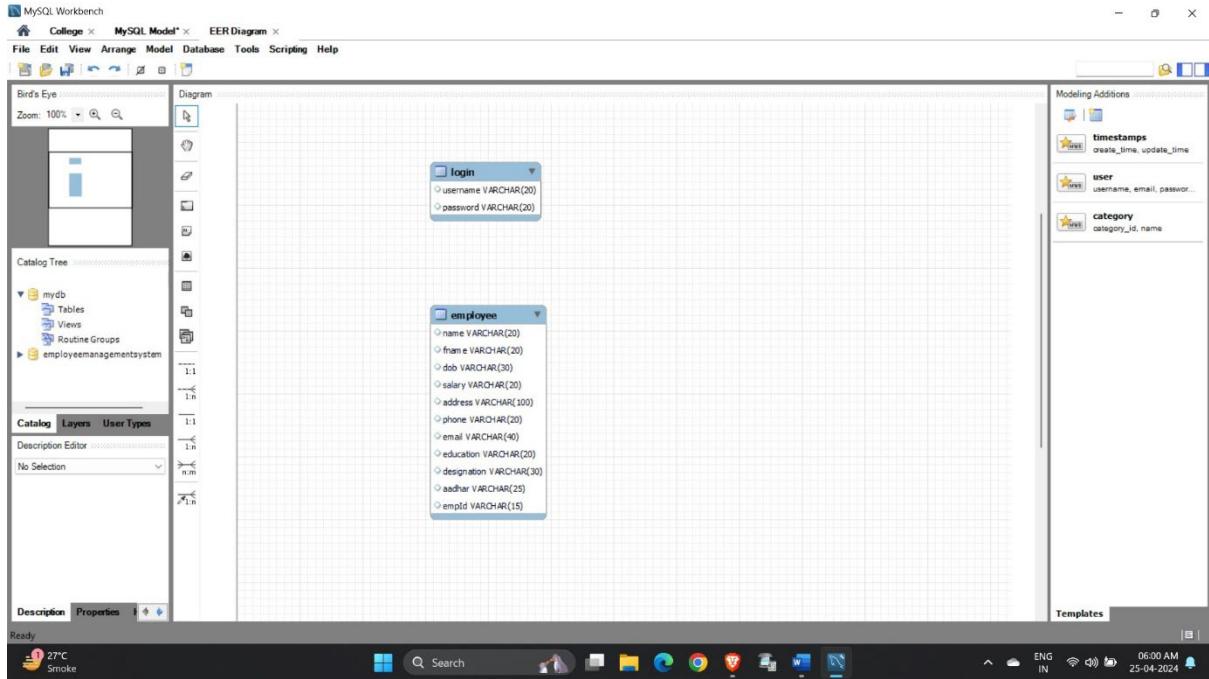
✓ Employee Table:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the `employeeManagementSystem` database selected.
- SQL File 3:** Contains the SQL query: `25
26 • select * from employee;`
- Result Grid:** Displays the data from the `employee` table:

name	fname	dob	salary	address	phone	email
PatelSagar	Jitendrabhai	12-Jul-1990	50000	Nadiad	9428902507	sagarp451@gmail.com
PatelAashu	Rakeshbhai	14-Feb-2005	50000	Dasharath	9723163446	patelaashu004@gmail.com
ThakkarKunj	Sanjaybhai	01-Aug-2005	50000	Dasharath	7984554842	kunjthakkar185@gmail.com
BadrakiyaDeep	Dhirajlal	26-Jul-2005	50000	Baroda	9016833043	deepbadrakiya@gmail.com
Bariyapurnima	Rakeshbhai	12-Aug-2003	50000	Baroda	8485933230	bariyapurnima2@gmail.com
Senvaketankumar	Vijaybhai	05-Oct-2004	50000	Dakor	9327024153	padhiyark253@gmail.com
ThakorRiya	Jitendrakumar	13-Feb-2004	50000	Baroda	7383829730	padhiyark253@gmail.com
- Session:** Shows the session information: `employee 3`, `Object Info`, `Session`.
- System Bar:** Shows the system tray with icons for battery, network, and date/time: `28°C Smoke`, `ENG IN`, `05:41 AM 25-04-2024`.

✓ ER Diagram:



✓ Splash Page:

```
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);
        setLayout(null);

        JLabel heading = new JLabel("EMPLOYEE MANAGEMENT SYSTEM");
        heading.setBounds(80, 30, 1200, 60);
        heading.setFont(new Font("serif", Font.PLAIN, 60));
        heading.setForeground(Color.BLUE);
        add(heading);

        ImageIcon i1 = new ImageIcon(getClass().getResource("icons/front.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1100, 700, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(50, 100, 1050, 500);
    }

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

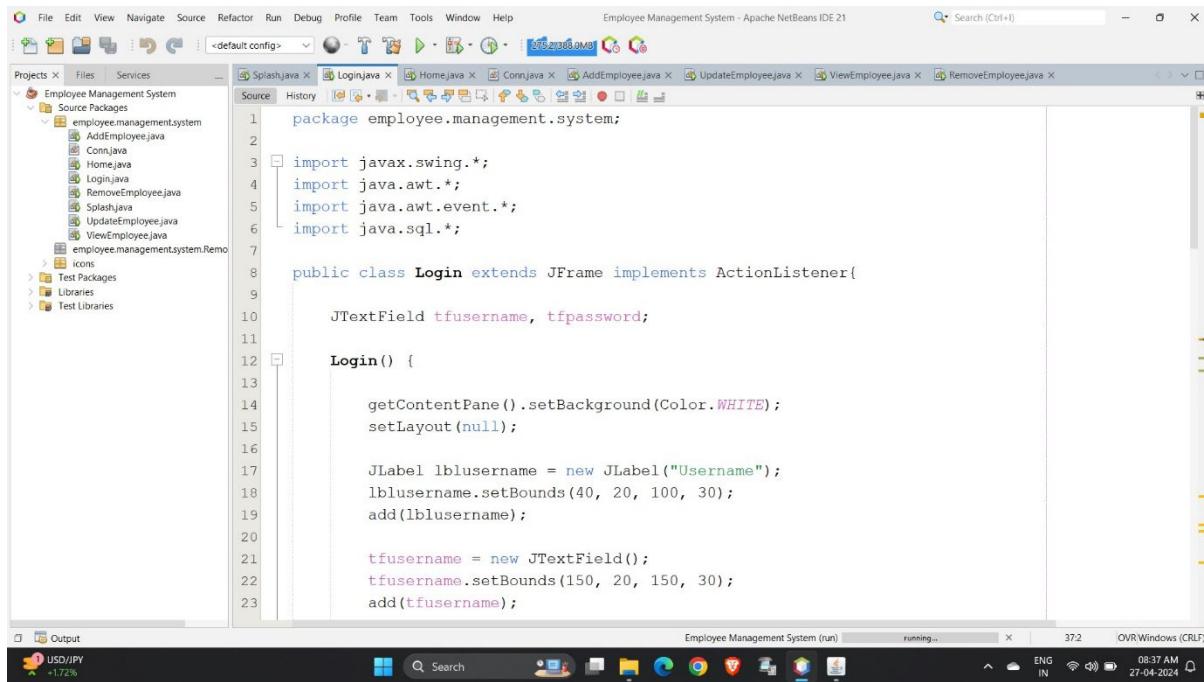
The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, and Print. The Projects tab shows a single project named "Employee Management System" with a Source Package containing files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, ViewEmployee.java, and employee.management.system.Rem. The Files tab shows icons for Test Packages, Libraries, and Test Libraries. The Services tab is empty. The central editor pane displays the code for SplashJava.java:

```
26     add(image);  
27  
28     JButton clickhere = new JButton("CLICK HERE TO CONTINUE");  
29     clickhere.setBounds(400, 400, 300, 70);  
30     clickhere.setBackground(Color.BLACK);  
31     clickhere.setForeground(Color.WHITE);  
32     clickhere.addActionListener(this);  
33     image.add(clickhere);  
34  
35     setSize(1170, 650);  
36     setLocation(200, 50);  
37     setVisible(true);  
38 }  
39  
40 public void actionPerformed(ActionEvent ae) {  
41     setVisible(false);  
42     new Login();  
43 }  
44  
45 public static void main(String args[]) {  
46     new Splash();  
47 }
```

✓ Output of Splash Page:



✓ Login Page:



The screenshot shows the Apache NetBeans IDE interface with the 'Employee Management System' project open. The 'Source Packages' tree view shows the package structure under 'employee.management.system'. The 'Login.java' file is selected and displayed in the central code editor. The code implements a JFrame for a login interface, setting up labels for 'Username' and 'Password', and a 'LOGIN' button.

```
package employee.management.system;

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

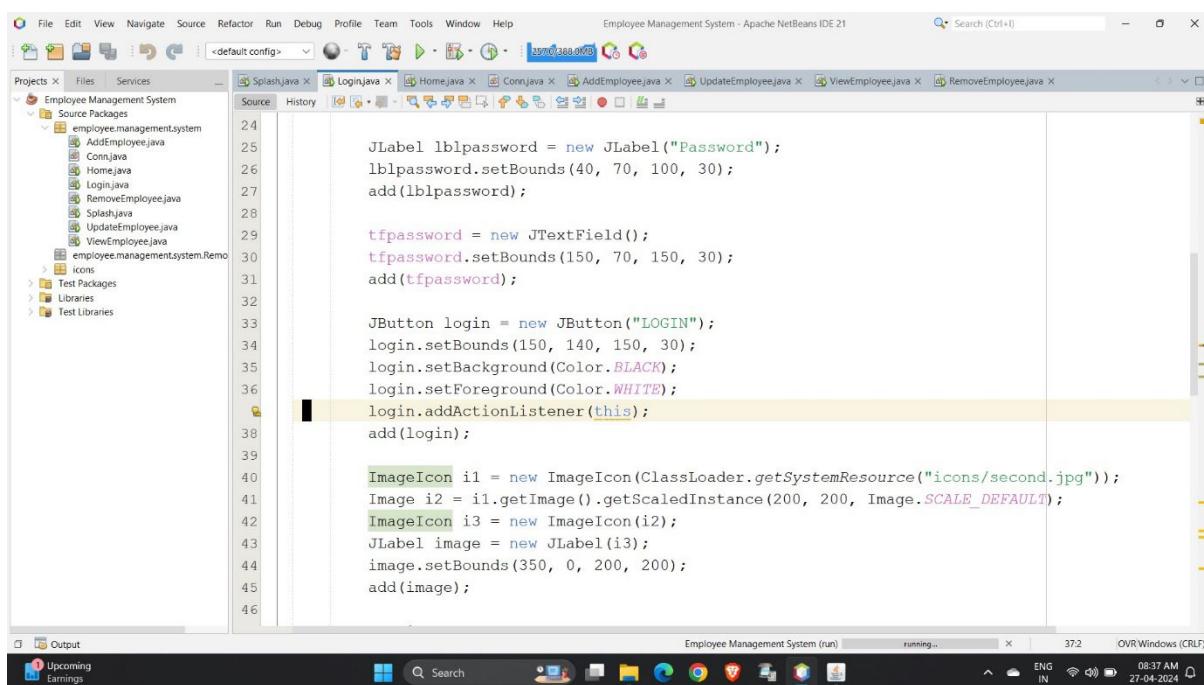
public class Login extends JFrame implements ActionListener{

    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);
    }
}
```



The screenshot shows the continuation of the 'Login.java' code in the Apache NetBeans IDE. The code adds a 'Password' label and field, and a 'LOGIN' button. It also includes code to load an image icon named 'second.jpg' and display it as a JLabel.

```
    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);
}
```

The screenshot shows the Apache NetBeans IDE 21 interface. The left pane displays the project structure for 'Employee Management System' under 'Source Packages'. The right pane shows the source code for 'Login.java'. The code handles user login by connecting to a database and checking if the provided username and password match. If successful, it creates a new 'Home' window; if not, it shows an error dialog.

```
    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();
}
```

This screenshot is identical to the one above, showing the same Java code for 'Login.java' in the Apache NetBeans IDE 21 interface. It performs the same function of handling user login and displaying a new 'Home' window if successful.

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

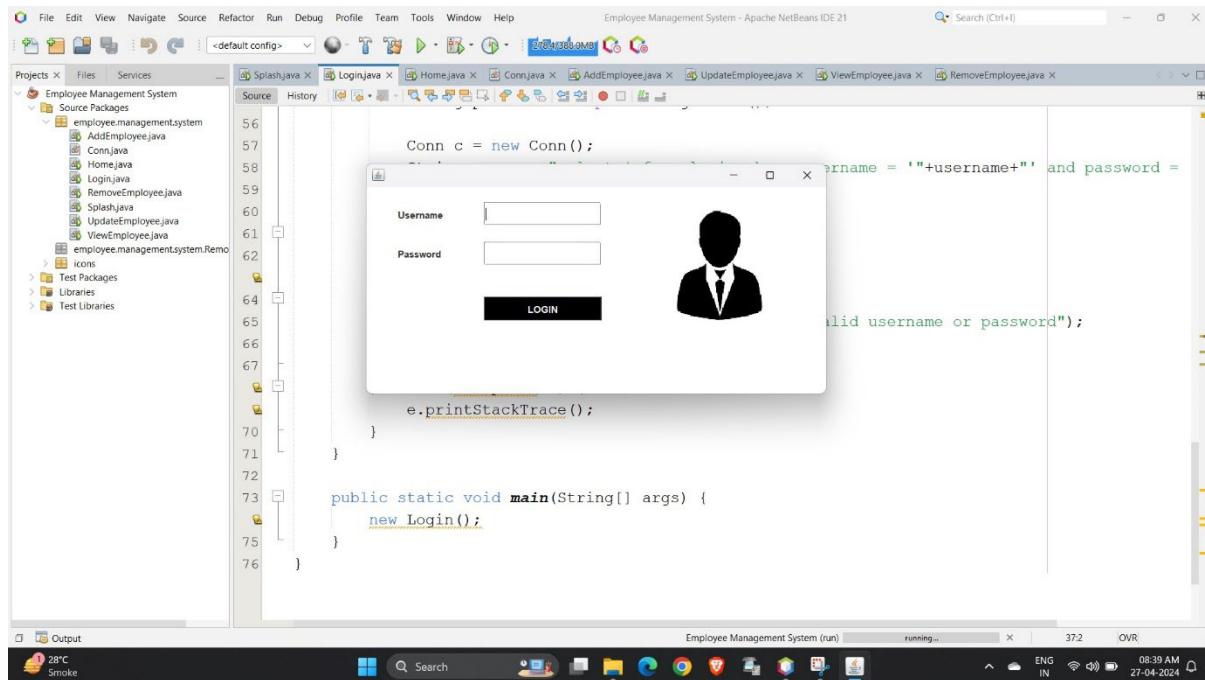
public void actionPerformed(ActionEvent ae) {
    try {
        String username = tfusername.getText();
        String password = tfpassword.getText();

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

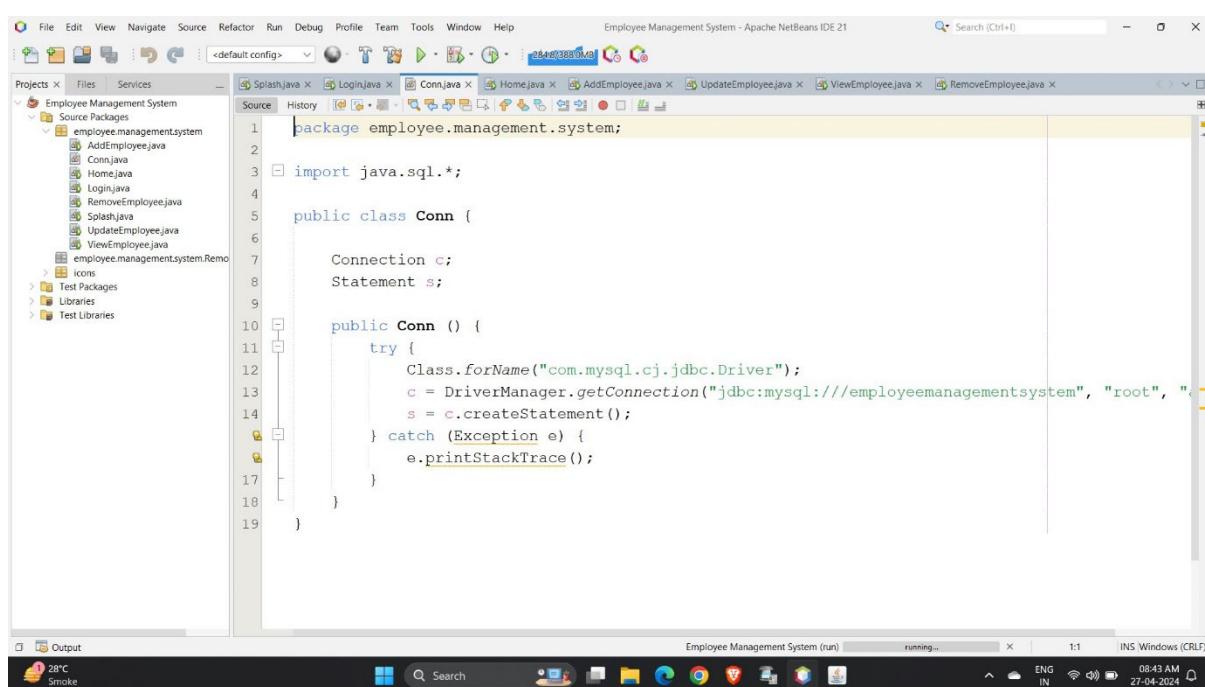
        ResultSet rs = c.s.executeQuery(query);
        if (rs.next()) {
            setVisible(false);
            new Home();
        } else {
            JOptionPane.showMessageDialog(null, "Invalid username or password");
            setVisible(false);
        }
    } catch (Exception e) {
        e.printStackTrace();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

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

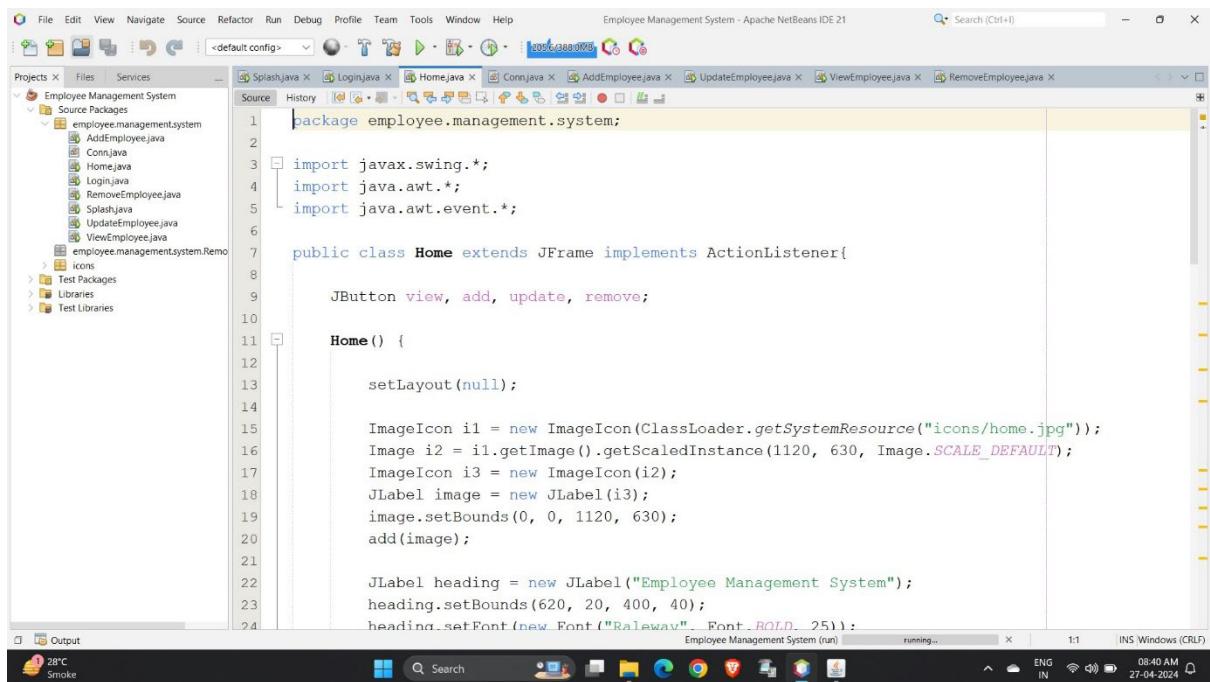
✓ Output of Login Page:



✓ Connection Page:



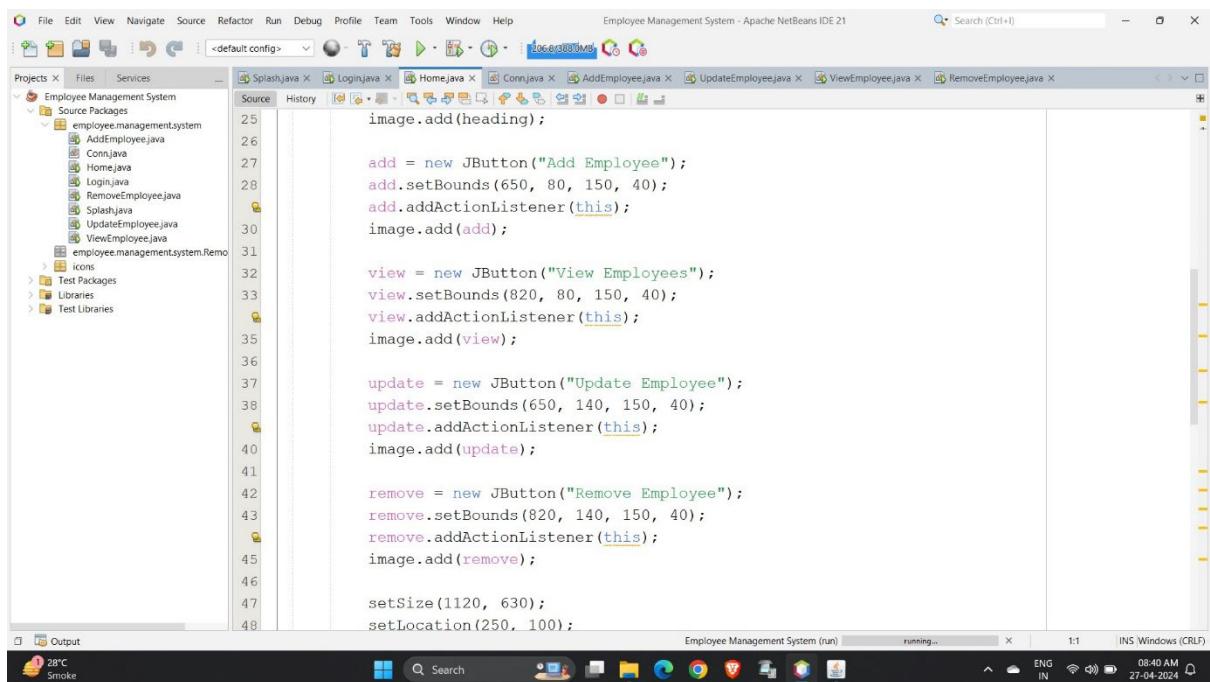
✓ Home Page:



The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project:** Employee Management System
- Source Package:** employee.management.system
- File:** Home.java
- Code Content:** The code defines a `Home` class that extends `JFrame` and implements `ActionListener`. It sets the layout to null, adds a background image, and adds several buttons for managing employees.

```
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6
7 public class Home extends JFrame implements ActionListener{
8
9     JButton view, add, update, remove;
10
11    Home() {
12
13        setLayout(null);
14
15        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/home.jpg"));
16        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
17        ImageIcon i3 = new ImageIcon(i2);
18        JLabel image = new JLabel(i3);
19        image.setBounds(0, 0, 1120, 630);
20        add(image);
21
22        JLabel heading = new JLabel("Employee Management System");
23        heading.setBounds(620, 20, 400, 40);
24        heading.setFont(new Font("Raleway", Font.BOLD, 25));
25
26        image.add(heading);
27
28        add = new JButton("Add Employee");
29        add.setBounds(650, 80, 150, 40);
30        add.addActionListener(this);
31        image.add(add);
32
33        view = new JButton("View Employees");
34        view.setBounds(820, 80, 150, 40);
35        view.addActionListener(this);
36        image.add(view);
37
38        update = new JButton("Update Employee");
39        update.setBounds(650, 140, 150, 40);
40        update.addActionListener(this);
41        image.add(update);
42
43        remove = new JButton("Remove Employee");
44        remove.setBounds(820, 140, 150, 40);
45        remove.addActionListener(this);
46        image.add(remove);
47
48        setSize(1120, 630);
49        setLocation(250, 100);
50    }
51
52    public void actionPerformed(ActionEvent e) {
53        if (e.getSource() == add) {
54            AddEmployee();
55        } else if (e.getSource() == update) {
56            UpdateEmployee();
57        } else if (e.getSource() == remove) {
58            RemoveEmployee();
59        } else if (e.getSource() == view) {
60            ViewEmployee();
61        }
62    }
63
64    private void AddEmployee() {
65        Conn conn = new Conn();
66        String id = JOptionPane.showInputDialog("Enter Employee ID");
67        String name = JOptionPane.showInputDialog("Enter Employee Name");
68        String address = JOptionPane.showInputDialog("Enter Employee Address");
69        String phone = JOptionPane.showInputDialog("Enter Employee Phone");
70
71        try {
72            String query = "insert into employee values ('" + id + "','" + name + "','" + address + "','" + phone + "')";
73            conn.s.executeUpdate(query);
74            JOptionPane.showMessageDialog(null, "Employee Added");
75        } catch (Exception e1) {
76            JOptionPane.showMessageDialog(null, "Employee Not Added");
77        }
78    }
79
80    private void UpdateEmployee() {
81        Conn conn = new Conn();
82        String id = JOptionPane.showInputDialog("Enter Employee ID");
83        String name = JOptionPane.showInputDialog("Enter Employee Name");
84        String address = JOptionPane.showInputDialog("Enter Employee Address");
85        String phone = JOptionPane.showInputDialog("Enter Employee Phone");
86
87        try {
88            String query = "update employee set name = '" + name + "', address = '" + address + "', phone = '" + phone + "' where id = '" + id + "'";
89            conn.s.executeUpdate(query);
90            JOptionPane.showMessageDialog(null, "Employee Updated");
91        } catch (Exception e1) {
92            JOptionPane.showMessageDialog(null, "Employee Not Updated");
93        }
94    }
95
96    private void RemoveEmployee() {
97        Conn conn = new Conn();
98        String id = JOptionPane.showInputDialog("Enter Employee ID");
99
100       try {
101           String query = "delete from employee where id = '" + id + "'";
102           conn.s.executeUpdate(query);
103           JOptionPane.showMessageDialog(null, "Employee Removed");
104       } catch (Exception e1) {
105           JOptionPane.showMessageDialog(null, "Employee Not Removed");
106       }
107   }
108
109   private void ViewEmployee() {
110      Conn conn = new Conn();
111      String id = JOptionPane.showInputDialog("Enter Employee ID");
112
113      try {
114          String query = "select * from employee where id = '" + id + "'";
115          conn.s.executeQuery(query);
116
117          ResultSet rs = conn.s.getResultSet();
118
119          while (rs.next()) {
120              String name = rs.getString("name");
121              String address = rs.getString("address");
122              String phone = rs.getString("phone");
123
124              JOptionPane.showMessageDialog(null, "Employee Name: " + name + "\nAddress: " + address + "\nPhone: " + phone);
125          }
126      } catch (Exception e1) {
127          JOptionPane.showMessageDialog(null, "Employee Not Found");
128      }
129  }
130}
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project:** Employee Management System
- Source Package:** employee.management.system
- File:** Home.java
- Code Content:** The continuation of the `Home` class code, which includes the `actionPerformed` method and the `AddEmployee`, `UpdateEmployee`, `RemoveEmployee`, and `ViewEmployee` methods.

```
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6
7 public class Home extends JFrame implements ActionListener{
8
9     JButton view, add, update, remove;
10
11    Home() {
12
13        setLayout(null);
14
15        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/home.jpg"));
16        Image i2 = i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
17        ImageIcon i3 = new ImageIcon(i2);
18        JLabel image = new JLabel(i3);
19        image.setBounds(0, 0, 1120, 630);
20        add(image);
21
22        JLabel heading = new JLabel("Employee Management System");
23        heading.setBounds(620, 20, 400, 40);
24        heading.setFont(new Font("Raleway", Font.BOLD, 25));
25
26        image.add(heading);
27
28        add = new JButton("Add Employee");
29        add.setBounds(650, 80, 150, 40);
30        add.addActionListener(this);
31        image.add(add);
32
33        view = new JButton("View Employees");
34        view.setBounds(820, 80, 150, 40);
35        view.addActionListener(this);
36        image.add(view);
37
38        update = new JButton("Update Employee");
39        update.setBounds(650, 140, 150, 40);
40        update.addActionListener(this);
41        image.add(update);
42
43        remove = new JButton("Remove Employee");
44        remove.setBounds(820, 140, 150, 40);
45        remove.addActionListener(this);
46        image.add(remove);
47
48        setSize(1120, 630);
49        setLocation(250, 100);
50    }
51
52    public void actionPerformed(ActionEvent e) {
53        if (e.getSource() == add) {
54            AddEmployee();
55        } else if (e.getSource() == update) {
56            UpdateEmployee();
57        } else if (e.getSource() == remove) {
58            RemoveEmployee();
59        } else if (e.getSource() == view) {
60            ViewEmployee();
61        }
62    }
63
64    private void AddEmployee() {
65        Conn conn = new Conn();
66        String id = JOptionPane.showInputDialog("Enter Employee ID");
67        String name = JOptionPane.showInputDialog("Enter Employee Name");
68        String address = JOptionPane.showInputDialog("Enter Employee Address");
69        String phone = JOptionPane.showInputDialog("Enter Employee Phone");
70
71        try {
72            String query = "insert into employee values ('" + id + "','" + name + "','" + address + "','" + phone + "')";
73            conn.s.executeUpdate(query);
74            JOptionPane.showMessageDialog(null, "Employee Added");
75        } catch (Exception e1) {
76            JOptionPane.showMessageDialog(null, "Employee Not Added");
77        }
78    }
79
80    private void UpdateEmployee() {
81        Conn conn = new Conn();
82        String id = JOptionPane.showInputDialog("Enter Employee ID");
83        String name = JOptionPane.showInputDialog("Enter Employee Name");
84        String address = JOptionPane.showInputDialog("Enter Employee Address");
85        String phone = JOptionPane.showInputDialog("Enter Employee Phone");
86
87        try {
88            String query = "update employee set name = '" + name + "', address = '" + address + "', phone = '" + phone + "' where id = '" + id + "'";
89            conn.s.executeUpdate(query);
90            JOptionPane.showMessageDialog(null, "Employee Updated");
91        } catch (Exception e1) {
92            JOptionPane.showMessageDialog(null, "Employee Not Updated");
93        }
94    }
95
96    private void RemoveEmployee() {
97        Conn conn = new Conn();
98        String id = JOptionPane.showInputDialog("Enter Employee ID");
99
100       try {
101           String query = "delete from employee where id = '" + id + "'";
102           conn.s.executeUpdate(query);
103           JOptionPane.showMessageDialog(null, "Employee Removed");
104       } catch (Exception e1) {
105           JOptionPane.showMessageDialog(null, "Employee Not Removed");
106       }
107   }
108
109   private void ViewEmployee() {
110      Conn conn = new Conn();
111      String id = JOptionPane.showInputDialog("Enter Employee ID");
112
113      try {
114          String query = "select * from employee where id = '" + id + "'";
115          conn.s.executeQuery(query);
116
117          ResultSet rs = conn.s.getResultSet();
118
119          while (rs.next()) {
120              String name = rs.getString("name");
121              String address = rs.getString("address");
122              String phone = rs.getString("phone");
123
124              JOptionPane.showMessageDialog(null, "Employee Name: " + name + "\nAddress: " + address + "\nPhone: " + phone);
125          }
126      } catch (Exception e1) {
127          JOptionPane.showMessageDialog(null, "Employee Not Found");
128      }
129  }
130}
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** Employee Management System - Apache NetBeans IDE 21.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search bar.
- Project Explorer:** Shows the project structure under "Employee Management System".
- Code Editor:** Displays the `Home.java` file with the following code:

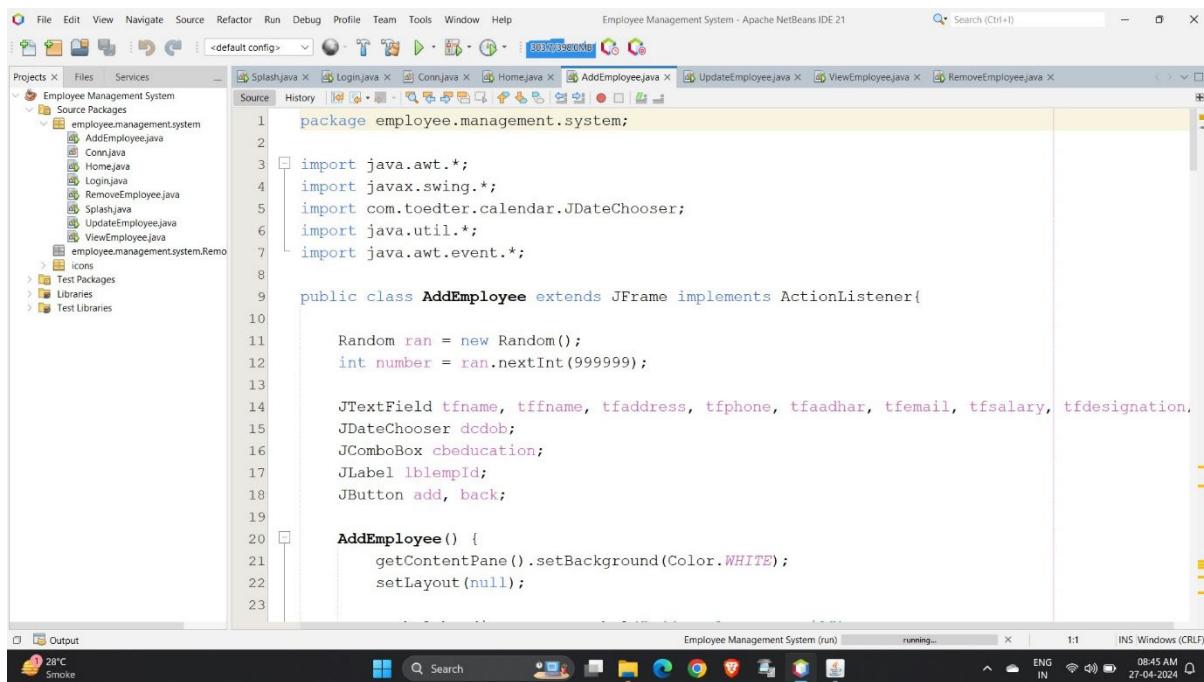
```
49     setVisible(true);
50 }
51
52 public void actionPerformed(ActionEvent ae) {
53     if (ae.getSource() == add) {
54         setVisible(false);
55         new AddEmployee();
56     } else if (ae.getSource() == view) {
57         setVisible(false);
58         new ViewEmployee();
59     } else if (ae.getSource() == update) {
60         setVisible(false);
61         new ViewEmployee();
62     } else {
63         setVisible(false);
64         new RemoveEmployee();
65     }
66 }
67
68 public static void main(String[] args) {
69     new Home();
70 }
71 }
```

- Output Tab:** Shows the output of the application running.
- System Tray:** Shows weather (28°C), battery status, and system date (27-04-2024).

✓ Output of Home Page:



✓ Add Employee Page:



The screenshot shows the Apache NetBeans IDE interface with the 'Employee Management System - Apache NetBeans IDE 21' window open. The 'Source' tab is selected in the AddEmployee.java editor. The code implements the ActionListener interface for the 'add' button. It includes imports for Java AWT and Swing components, and defines several fields like JTextField and JComboBox. The constructor 'AddEmployee()' sets the content pane's background to white and layout to null.

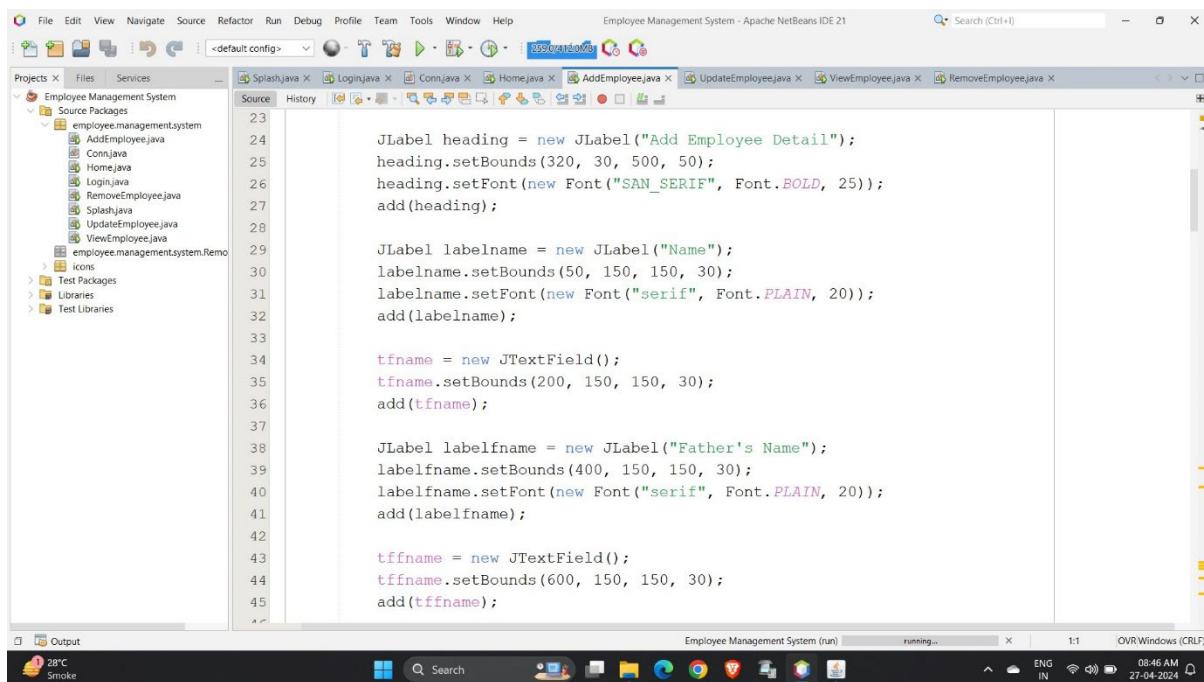
```
package employee.management.system;
import java.awt.*;
import javax.swing.*;
import com.toedter.calendar.JDateChooser;
import java.util.*;
import java.awt.event.*;

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);
    }
}
```



The screenshot shows the continuation of the AddEmployee.java code. It adds labels and text fields for employee details. Labels include 'Add Employee Detail', 'Name', 'Father's Name', and 'Father's Name'. Text fields correspond to these labels. The code uses setBounds and setFont methods to define the appearance of each component.

```
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);
```

The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like New, Open, Save, and Run. The Projects tab shows a tree view of the "Employee Management System" project, including source packages like "employee.management.system" containing files such as AddEmployee.java, ConnJava, Home.java, Login.java, RemoveEmployee.java, SplashJava, UpdateEmployee.java, ViewEmployee.java, and others. The Files tab is selected, displaying the source code for AddEmployee.java. The code is as follows:

```
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     JDateChooser 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     JTextField 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 
```

This screenshot continues the Java code from the previous one, showing the completion of the AddEmployee.java class. The code adds labels and text fields for address, phone, and email.

```
69
70     JTextField 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     JTextField 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     JTextField tfEmail = new JTextField();
89     tfEmail.setBounds(200, 300, 150, 30);
90     add(tfEmail);

91 
```

The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, and Run. The Projects tab shows a tree view of the "Employee Management System" project, including source packages like "employee.management.system" containing files such as AddEmployee.java, ConnJava, Home.java, Login.java, RemoveEmployee.java, SplashJava, UpdateEmployee.java, ViewEmployee.java, and "employee.management.system.Remo". The Source tab displays the Java code for AddEmployee.java, which handles the creation of GUI components like JLabels, JComboBoxes, and JTextFieldes for education, designation, and Aadhar number.

```
92     JLabel labelEducation = new JLabel("Higest Education");
93     labelEducation.setBounds(400, 300, 150, 30);
94     labelEducation.setFont(new Font("serif", Font.PLAIN, 20));
95     add(labelEducation);
96
97     String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA", "MA", "N";
98     JComboBox cbEducation = new JComboBox(courses);
99     cbEducation.setBackground(Color.WHITE);
100    cbEducation.setBounds(600, 300, 150, 30);
101    add(cbEducation);
102
103    JLabel labelDesignation = new JLabel("Designation");
104    labelDesignation.setBounds(50, 350, 150, 30);
105    labelDesignation.setFont(new Font("serif", Font.PLAIN, 20));
106    add(labelDesignation);
107
108    JTextField tfDesignation = new JTextField();
109    tfDesignation.setBounds(200, 350, 150, 30);
110    add(tfDesignation);
111
112    JLabel labelAadhar = new JLabel("Aadhar Number");
113    labelAadhar.setBounds(400, 350, 150, 30);
114    labelAadhar.setFont(new Font("serif", Font.PLAIN, 20));
```

This screenshot is identical to the one above, showing the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar, toolbar, and Projects tab are the same. The Source tab displays the Java code for AddEmployee.java, which continues from the previous snippet. It includes code for creating a JTextField for Aadhar number, a JLabel for Employee ID, and a JButton for adding details. The code uses various Java Swing components and their properties like setBounds and setFont.

```
117    JTextField tfaadhar = new JTextField();
118    tfaadhar.setBounds(600, 350, 150, 30);
119    add(tfaadhar);
120
121    JLabel labelEmpId = new JLabel("Employee id");
122    labelEmpId.setBounds(50, 400, 150, 30);
123    labelEmpId.setFont(new Font("serif", Font.PLAIN, 20));
124    add(labelEmpId);
125
126    JLabel lblEmpId = new JLabel("" + number);
127    lblEmpId.setBounds(200, 400, 150, 30);
128    lblEmpId.setFont(new Font("serif", Font.PLAIN, 20));
129    add(lblEmpId);
130
131    JButton add = new JButton("Add Details");
132    add.setBounds(250, 550, 150, 40);
133    add.addActionListener(this);
134    add.setBackground(Color.BLACK);
135    add.setForeground(Color.WHITE);
136    add(add);
137
138    JButton back = new JButton("Back");
139    back.setBounds(450, 550, 150, 40);
```

The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, Find, and Run. The Projects tab shows a single project "Employee Management System" with a source package "employee.management.system" containing files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, ViewEmployee.java, and icons. The Files tab shows various Java files listed. The Services tab is empty. The Source tab displays the code for AddEmployee.java:

```
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();
```

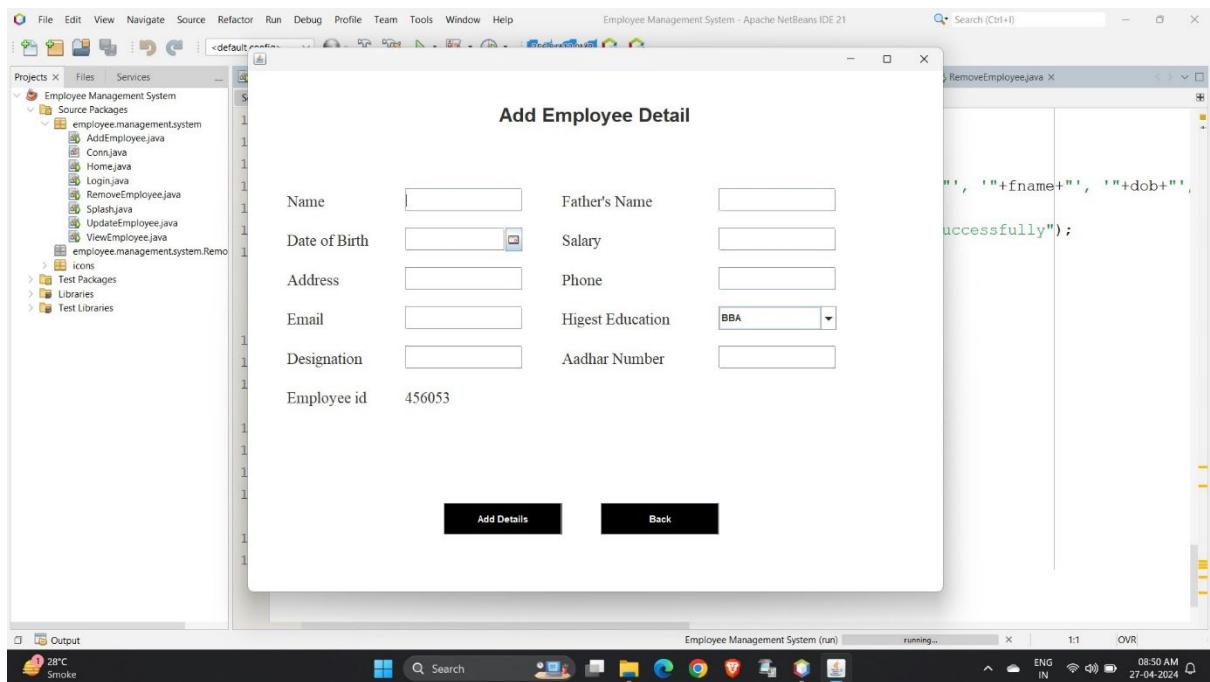
The screenshot continues from the previous one, showing the code for AddEmployee.java in the Source tab of the Apache NetBeans IDE. The code includes a try-catch block for database insertion and a main method to run the application. The code is as follows:

```
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();
}

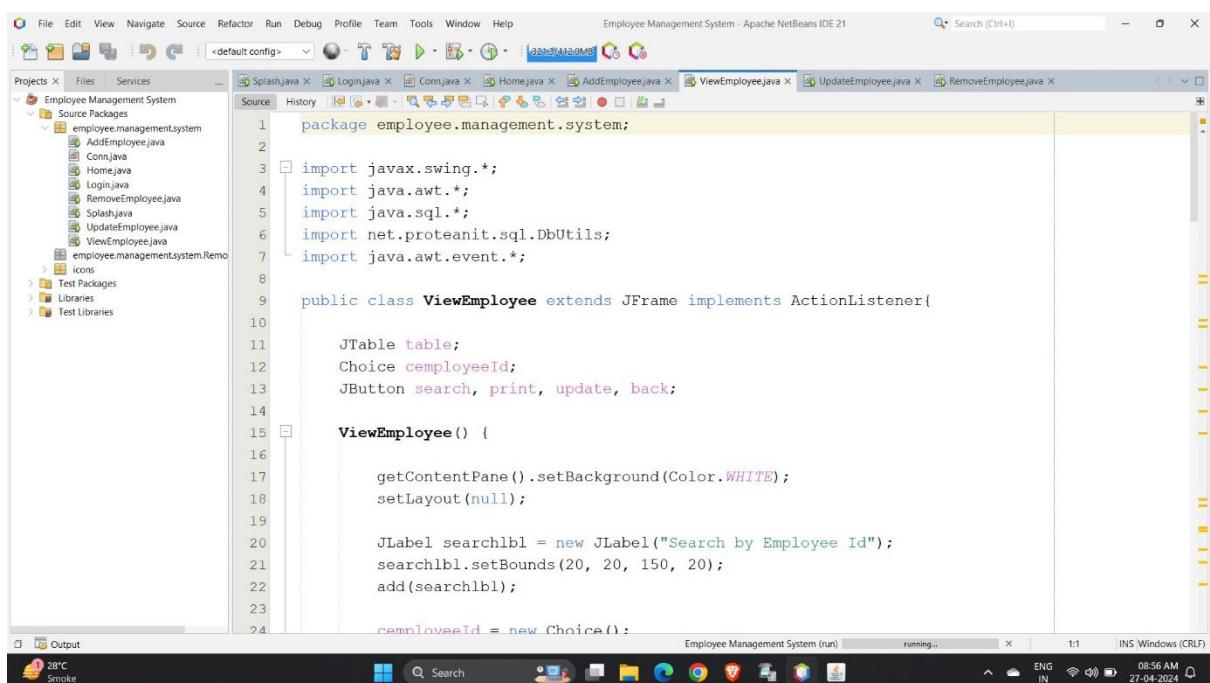
}

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

✓ Output of Add Employee Page:



✓ View Employee Page:



The screenshot shows the Apache NetBeans IDE 21 interface. The left pane displays the project structure for 'Employee Management System' under 'Source Packages'. The right pane shows the source code for 'ViewEmployee.java'. The code retrieves employee data from a database and populates a JTable.

```
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();
}
```

The screenshot shows the continuation of the Java code for 'ViewEmployee.java'. It adds components like JScrollPane, JButton, and JButton to a panel.

```
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);
```

The screenshot shows the Apache NetBeans IDE 21 interface. The left pane displays the project structure under 'Employee Management System' with several source files listed. The right pane shows the source code for one of the files. The code is a Java class with methods for actionPerformed and main.

```
    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.getSelectedItem();
        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);
    }
}

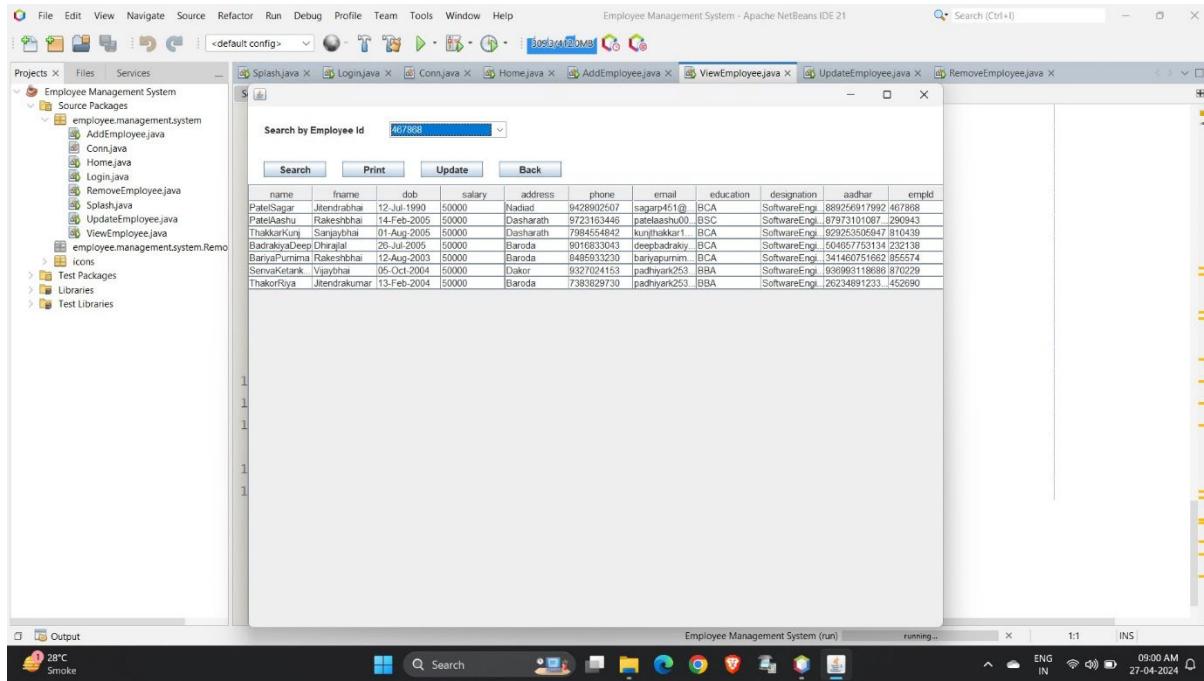
public static void main(String[] args) {
    new ViewEmployee();
}
```

The screenshot shows the Apache NetBeans IDE 21 interface. The left pane displays the project structure under 'Employee Management System' with several source files listed. The right pane shows the source code for one of the files. The code is a Java class with methods for actionPerformed and main.

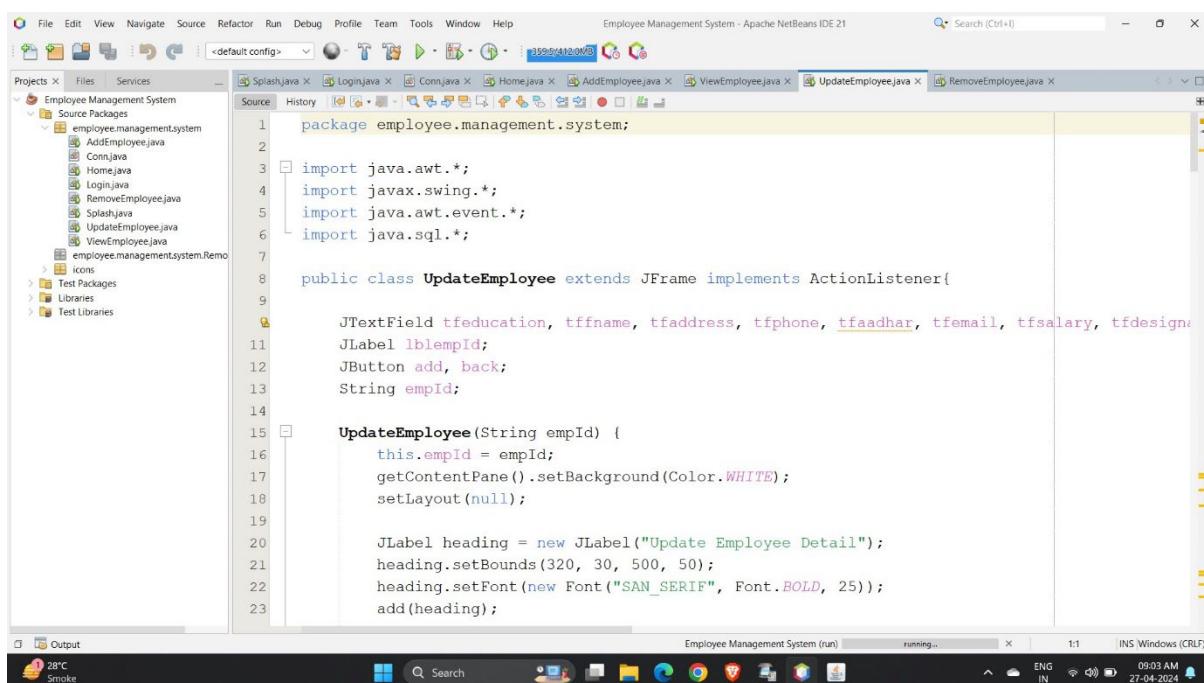
```
    try {
        table.print();
    } catch (Exception e) {
        e.printStackTrace();
    }
} else if (ae.getSource() == update) {
    setVisible(false);
    new UpdateEmployee(cemployeeId.getSelectedItem());
} else {
    setVisible(false);
    new Home();
}
}

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

✓ Output of View Employee Page:



✓ Update Employee Page:



The screenshot shows the Apache NetBeans IDE interface for an 'Employee Management System' project. The code editor displays Java code for the 'UpdateEmployee.java' file. The code creates several GUI components like JLabels and JTextField objects, setting their bounds and fonts. The code editor has a vertical scrollbar on the right. Below the editor is a status bar showing 'Employee Management System (run)' and 'running...'. The system tray at the bottom shows the date and time as '27-04-2024 09:04 AM'.

```
24     JLabel labelname = new JLabel("Name");
25     labelname.setBounds(50, 150, 150, 30);
26     labelname.setFont(new Font("serif", Font.PLAIN, 20));
27     add(labelname);
28
29     JLabel lblname = new JLabel();
30     lblname.setBounds(200, 150, 150, 30);
31     add(lblname);
32
33     JLabel labelfname = new JLabel("Father's Name");
34     labelfname.setBounds(400, 150, 150, 30);
35     labelfname.setFont(new Font("serif", Font.PLAIN, 20));
36     add(labelfname);
37
38     JTextField tffname = new JTextField();
39     tffname.setBounds(600, 150, 150, 30);
40     add(tffname);
41
42     JLabel labeldob = new JLabel("Date of Birth");
43     labeldob.setBounds(50, 200, 150, 30);
44     labeldob.setFont(new Font("serif", Font.PLAIN, 20));
45     add(labeldob);
46
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     JTextField 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     JTextField tfaddress = new JTextField();
67     tfaddress.setBounds(200, 250, 150, 30);
68     add(tfaddress);
69
70     JLabel labelphone = new JLabel("Phone");
```

This screenshot shows the same Apache NetBeans IDE interface as the first one, but the code editor now displays Java code for the 'ViewEmployee.java' file. The code creates GUI components for viewing employee details. The code editor has a vertical scrollbar on the right. Below the editor is a status bar showing 'Employee Management System (run)' and 'running...'. The system tray at the bottom shows the date and time as '27-04-2024 09:04 AM'.

```
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     JTextField 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     JTextField tfaddress = new JTextField();
67     tfaddress.setBounds(200, 250, 150, 30);
68     add(tfaddress);
69
70     JLabel labelphone = new JLabel("Phone");
```

The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and Search (Ctrl+I). The toolbar has icons for file operations like New, Open, Save, and Run.

The Projects tab shows a tree view of the "Employee Management System" project, including Source Packages (employee.management.system) containing AddEmployee.java, ConnJava, Home.java, Login.java, RemoveEmployee.java, SplashJava, UpdateEmployee.java, and ViewEmployee.java, along with icons, Test Packages, Libraries, and Test Libraries.

The Source tab displays the Java code for the ViewEmployee.java class:

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

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

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

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

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

88     tfeducation = new JTextField();
89     tfeducation.setBounds(600, 300, 150, 30);
90     add(tfeducation);

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

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

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

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

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

104    tfdesignation.setBounds(200, 350, 150, 30);
105    add(tfdesignation);

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

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

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

The screenshot shows the Apache NetBeans IDE interface with the title bar "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and Search (Ctrl+I). The toolbar has icons for file operations like New, Open, Save, and Run.

The Projects tab shows a tree view of the "Employee Management System" project, including Source Packages (employee.management.system) containing AddEmployee.java, ConnJava, Home.java, Login.java, RemoveEmployee.java, SplashJava, UpdateEmployee.java, and ViewEmployee.java, along with icons, Test Packages, Libraries, and Test Libraries.

The Source tab displays the Java code for the ViewEmployee.java class:

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

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

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

103    tfdesignation.setBounds(200, 350, 150, 30);
104    add(tfdesignation);

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

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

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

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** Employee Management System - Apache NetBeans IDE 21
- Toolbar:** Includes icons for file operations like New, Open, Save, Cut, Copy, Paste, Find, Replace, and others.
- Project Explorer (Left):**
 - Employee Management System (selected)
 - Source Packages
 - employee.managementSystem (selected)
 - AddEmployee.java
 - Conn.java
 - Home.java
 - Login.java
 - RemoveEmployee.java
 - Splash.java
 - UpdateEmployee.java
 - ViewEmployee.java
 - employee.managementSystem.Remote
 - icons
 - Test Packages
 - Libraries
 - Test Libraries
- Code Editor (Center):** Displays Java code for a ViewEmployee.java class. The code retrieves employee information from a database and sets it into various UI components like JLabels and JTextFieldes.
- Output (Bottom Left):** Shows the output of the build process.
- Task List (Bottom Right):** Lists tasks such as "Employee Management System (run)", "running...", "1:1", and "INS Windows (CRLF)".
- System Tray (Bottom Left):** Shows system status including temperature (28°C), battery level (Smoke), and network connection.
- System Icons (Bottom Right):** Standard Windows taskbar icons for search, start, file explorer, browser, and others.

The screenshot shows the Apache NetBeans IDE 21 interface. The title bar reads "Employee Management System - Apache NetBeans IDE 21". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, and Run. The Projects tab shows a single project named "Employee Management System" with a source package "employee.management.system" containing files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. The Files tab is selected, displaying the code for "UpdateEmployee.java". The code is as follows:

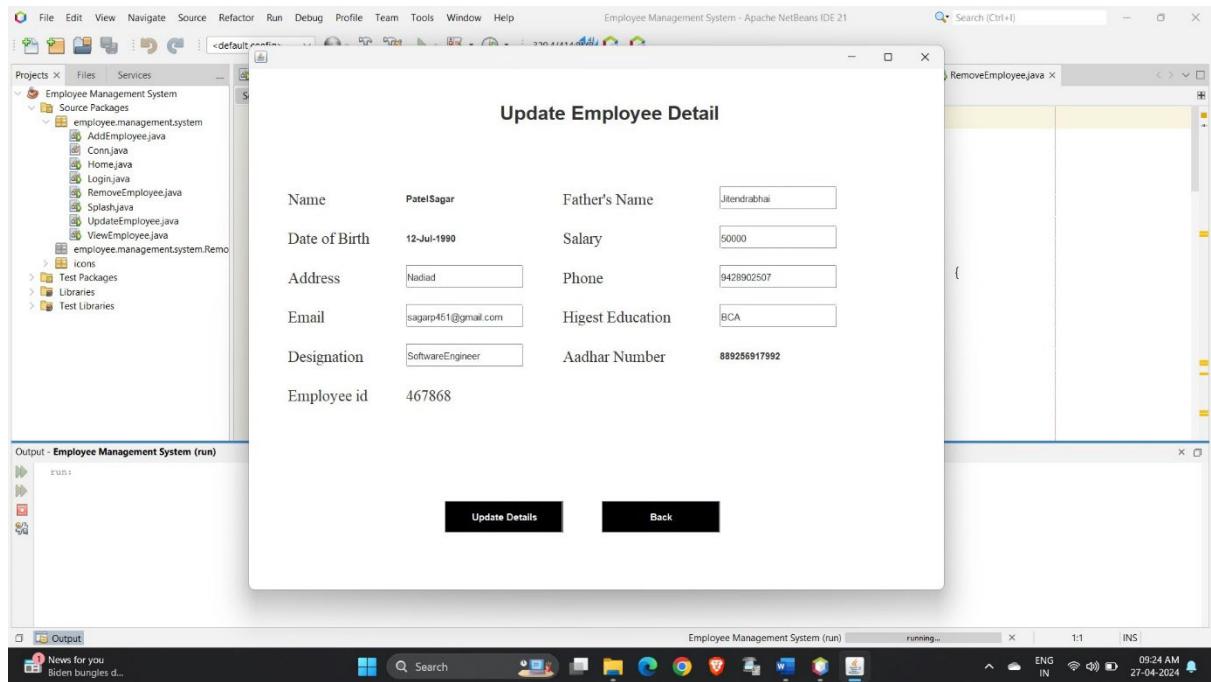
```
163     setVisible(true);
164 }
165 
166 public void actionPerformed(ActionEvent ae) {
167     if (ae.getSource() == add) {
168         String fname = tfname.getText();
169         String salary = tsalary.getText();
170         String address = taddress.getText();
171         String phone = tphone.getText();
172         String email = temail.getText();
173         String education = teducation.getText();
174         String designation = tdesignation.getText();
175 
176         try {
177             Conn conn = new Conn();
178             String query = "update employee set fname = '" + fname + "', salary = '" + salary + "'";
179             conn.s.executeUpdate(query);
180             JOptionPane.showMessageDialog(null, "Details updated successfully");
181             setVisible(false);
182             new Home();
183         } catch (Exception e) {
184             e.printStackTrace();
185         }
186     }
187 }
188 
189 public static void main(String[] args) {
190     new UpdateEmployee("");
191 }
192 
193 }
```

This screenshot is identical to the one above, showing the Apache NetBeans IDE 21 interface with the "UpdateEmployee.java" code. However, there is a noticeable addition at the bottom of the code:

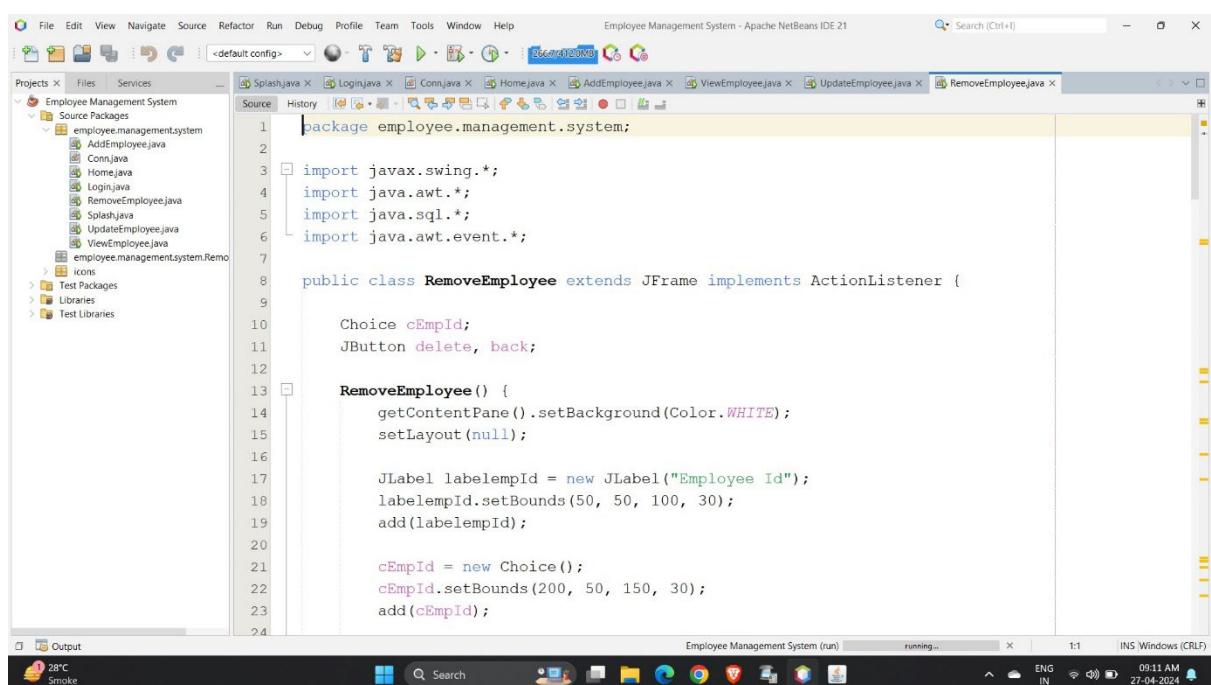
```
192     new UpdateEmployee("");
193 }
```

The rest of the code remains the same as in the first screenshot.

✓ Output of Update Employee Page:



✓ Remove Employee Page:



The screenshot shows the Apache NetBeans IDE 21 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar reads "Employee Management System - Apache NetBeans IDE 21". The left sidebar displays the "Projects" tree, which contains a single project named "Employee Management System" with a "Source Packages" node expanded, showing files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. Below the projects tree are "Test Packages", "Libraries", and "Test Libraries". The main workspace shows a portion of the Java code for a class, likely a JPanel subclass, containing methods for adding labels and a try-catch block for database operations. The code uses standard Java Swing components (JLabel) and JDBC (Conn, ResultSet). The bottom status bar shows the application is running, the date and time (27-04-2024, 09:11 AM), and system information (INS Windows (CRLF)).

```
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);
```

This screenshot is nearly identical to the one above, showing the same version of the Java code for the GUI application. The code is identical, including the try-catch block for database operations and the addition of three JLabel components (labelname, lblname, labelphone) to a panel. The IDE interface, project structure, and system status bar are also identical.

```
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();
```

The screenshot shows the Apache NetBeans IDE 21 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar reads "Employee Management System - Apache NetBeans IDE 21". The left sidebar displays the project structure under "Employee Management System" with "Source Packages" expanded, showing files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. The main editor area shows Java code for a class, likely a button's action listener, which retrieves employee details from a database and sets them to labels. The code uses JDBC and Swing components. The bottom status bar shows the application is running, the date and time (27-04-2024, 09:13 AM), and system information (INS Windows (CRLF)).

```
    }
    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);
```

This screenshot shows the same Apache NetBeans IDE 21 interface as the first one, but the code in the editor has changed. The code now handles the actionPerformed event for the "Delete" button. It creates a new ImageIcon from a file named "delete.png", scales it to 600x400 pixels, and adds it to a JLabel. This JLabel is then added to the JPanel where the other components are located. The code uses Java's Swing library for GUI components.

```
    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 il = new ImageIcon(ClassLoader.getSystemResource("icons/delete.png"));
    Image i2 = il.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) {
```

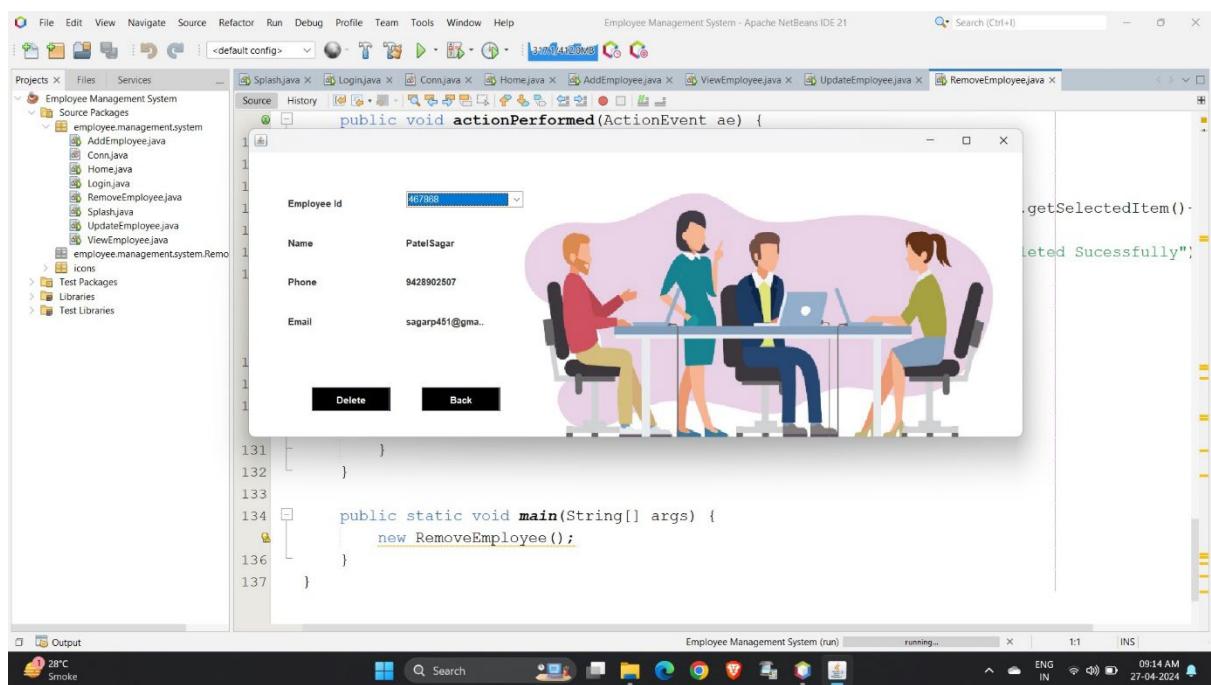
```

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == delete) {
        try {
            Conn c = new Conn();
            String query = "delete from employee where empId = '" + cEmpId.getSelectedItem() -> 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();
}

```

✓ 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