

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.

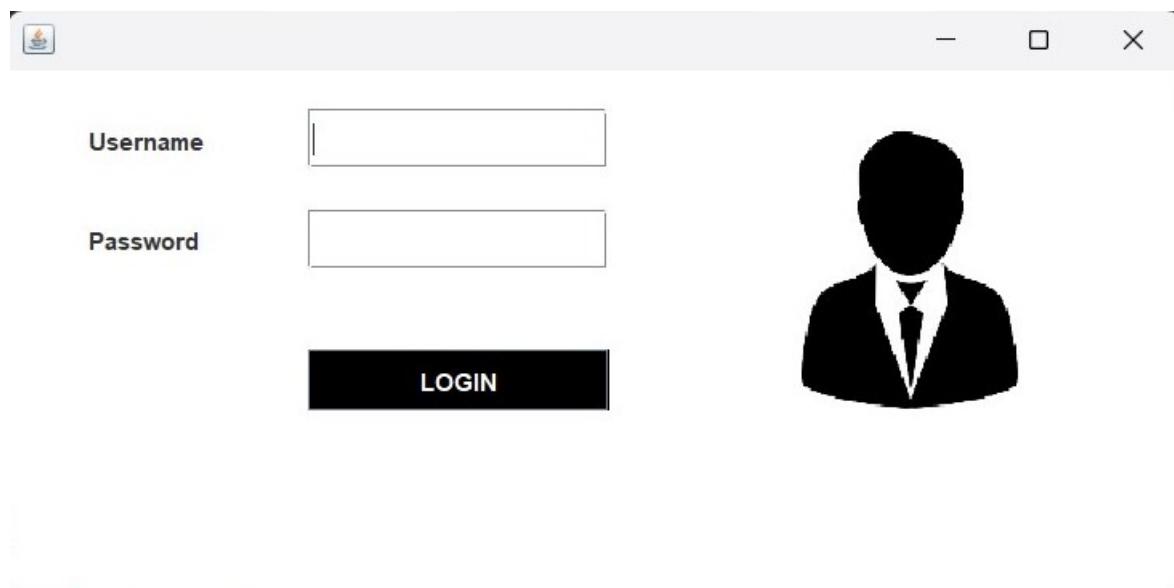
CHAPTER – 7

➤ 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.

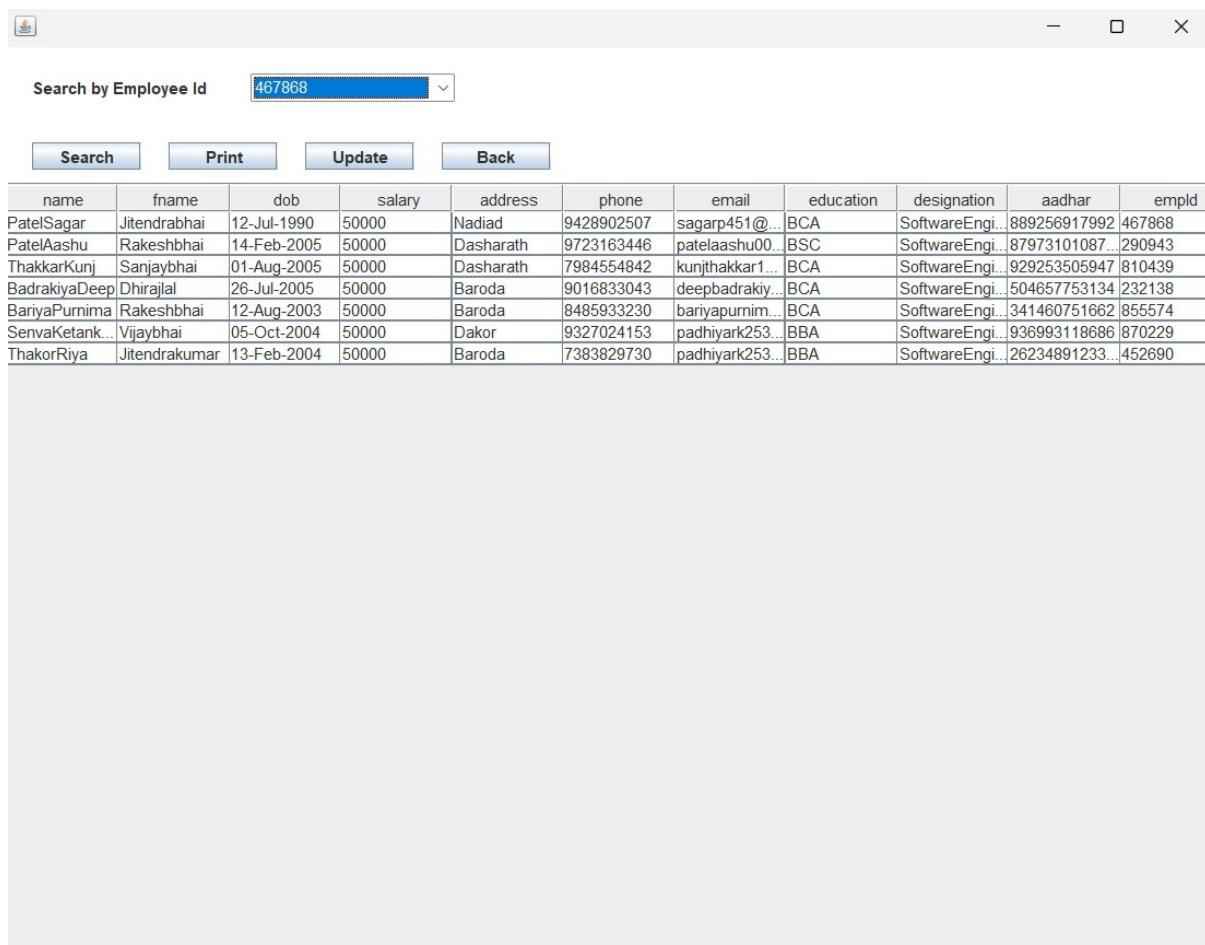
The screenshot shows a Windows-style application window titled "Add Employee Detail". The form contains fields for entering employee information. The fields are arranged in pairs, with labels on the left and input boxes on the right. The fields include Name, Father's Name, Date of Birth, Salary, Address, Phone, Email, Highest Education, Designation, Aadhar Number, and Employee id. The "Highest Education" field contains "BBA" and has a dropdown arrow. The "Employee id" field contains the value "636319". At the bottom, 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"/>	Highest 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.

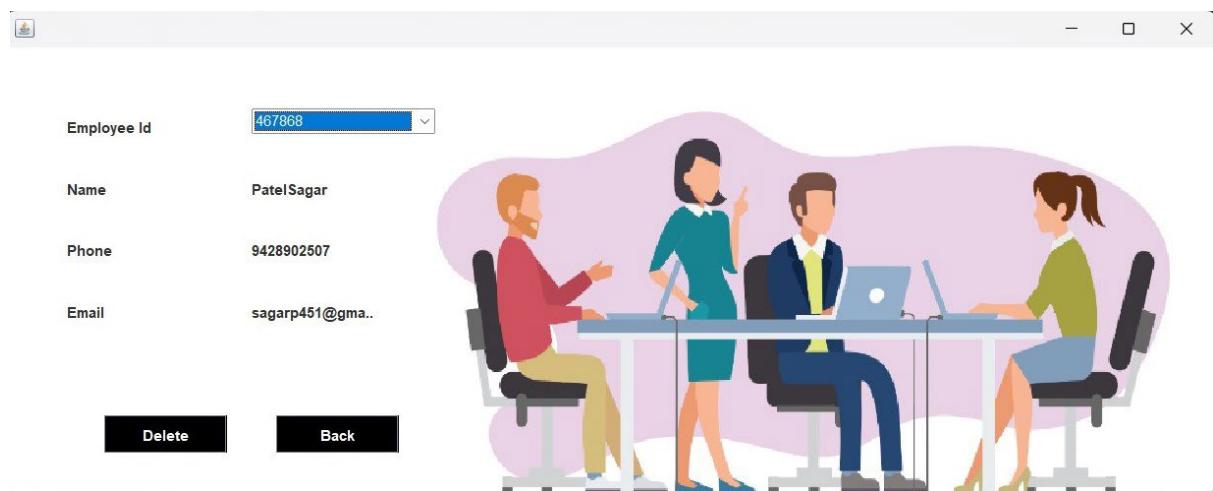
Update Employee Detail

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:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (College, employeemanagementsystem, hospital, sakila, sys, world).
- SQL Editor:** SQL File 3, containing the query: `10
11 • select * from login;
12`.
- Result Grid:** Shows the data from the login table:

username	password
sagarpatel	9428902507
aashupatel	9723163446
kunjthakkar	7984554842
- Toolbar:** Result Grid, Form Editor, Field Types, Query Stats, Execution Plan.
- Session Bar:** Object Info, Session, login 2.
- System Bar:** Read Only, 28°C, Smoke, ENG IN, 05:40 AM, 25-04-2024.

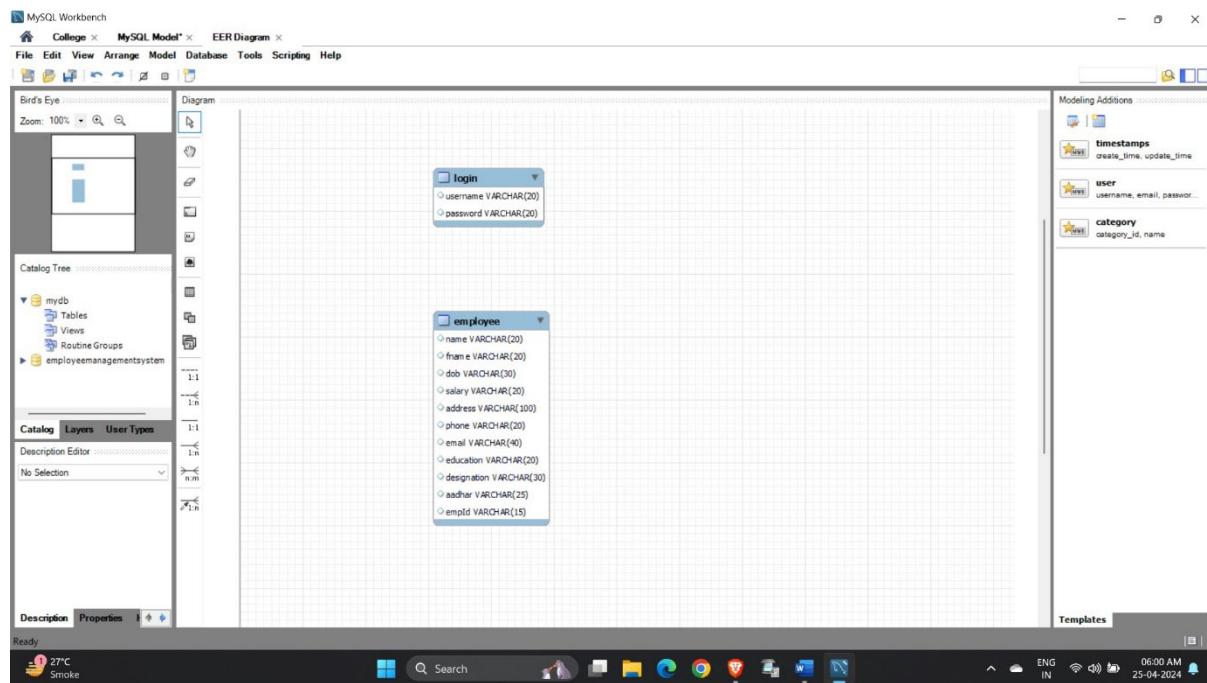
✓ Employee Table:

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (College, employeemanagementsystem, hospital, sakila, sys, world).
- SQL Editor:** SQL File 3, containing the query: `25
26 • select * from employee;`.
- Result Grid:** Shows 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
- Toolbar:** Result Grid, Form Editor, Field Types, Query Stats, Execution Plan.
- Session Bar:** Object Info, Session, employee 3.
- System Bar:** Read Only, 28°C, Smoke, ENG IN, 05:41 AM, 25-04-2024.

✓ ER Diagram:



✓ Splash Page:

The screenshot shows the Apache NetBeans IDE 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, Cut, Copy, Paste, Find, and Run. The central workspace shows a Java class named `Splash` in the `SplashJava` package. The code implements `ActionListener` and contains methods for initializing a `JFrame` with a heading label and an image icon. The left sidebar shows the project structure under "Employee Management System" with packages like `employee.management.system` containing files such as `AddEmployee.java`, `Conn.java`, `Home.java`, `Login.java`, `RemoveEmployee.java`, `Splash.java`, `UpdateEmployee.java`, and `ViewEmployee.java`. Below the project tree are sections for Test Packages, Libraries, and Test Libraries.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Employee Management System - Apache NetBeans IDE 21 Search (Ctrl + F) 222.1/4000 MBs Projects Files Services Employee Management System Source Packages <default config> SplashJava LoginJava ConnJava HomeJava AddEmployee.java ViewEmployee.java UpdateEmployee.java RemoveEmployee.java Source History H 1 package employee.management.system; 2 3 import javax.swing.*; 4 import java.awt.*; 5 import java.awt.event.*; 6 7 public class Splash extends JFrame implements ActionListener { 8 9     Splash() { 10         getContentPane().setBackground(Color.WHITE); 11         setLayout(null); 12 13         JLabel heading = new JLabel("EMPLOYEE MANAGEMENT SYSTEM"); 14         heading.setBounds(80, 30, 1200, 60); 15         heading.setFont(new Font("serif", Font.PLAIN, 60)); 16         heading.setForeground(Color.BLUE); 17         add(heading); 18 19         ImageIcon i1 = new ImageIcon(getClass().getResource("icons/front.jpg")); 20         Image i2 = i1.getImage().getScaledInstance(1100, 700, Image.SCALE_DEFAULT); 21         ImageIcon i3 = new ImageIcon(i2); 22         JLabel image = new JLabel(i3); 23         image.setBounds(50, 100, 1050, 500); 24     } 25 }
```

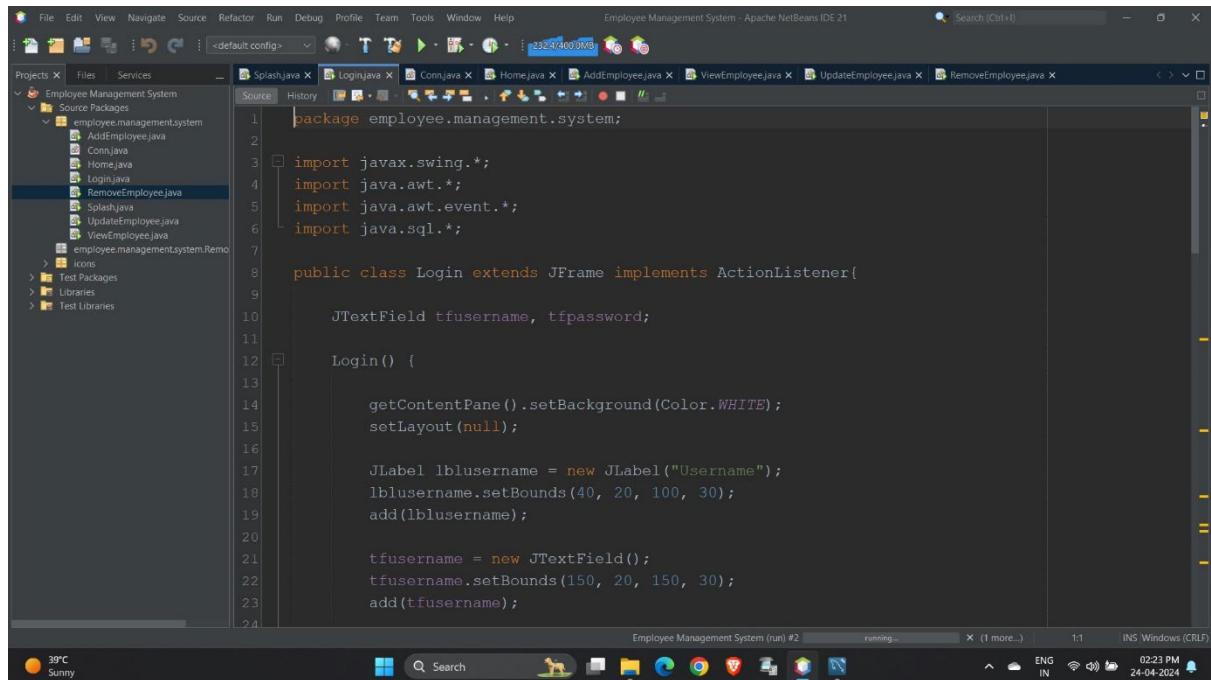
The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project Tree:** Employee Management System > Source Packages > employee.management.system > Splash.java
- Code Editor:** The code for `Splash.java` is displayed. It includes imports for `java.awt`, `java.awt.event`, and `java.net`. The class `Splash` contains a constructor that initializes a frame with a button labeled "CLICK HERE TO CONTINUE". The button's action listener calls the `Login()` method. The `main` method creates and runs a new `Splash` object.
- Output:** Shows the output of the application running as "Employee Management System (run)".
- System Tray:** Displays weather (39°C, Sunny), network status, battery level (38%), and system date/time (24-04-2024, 02:22 PM).

✓ Output of Splash Page:



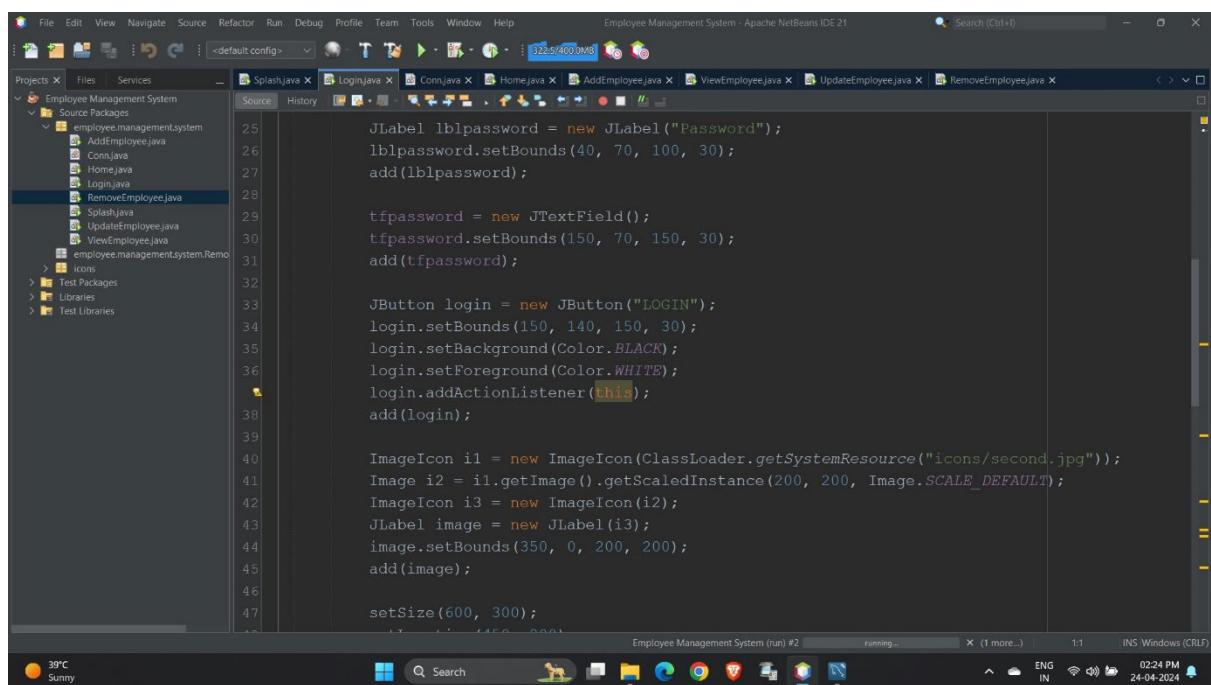
✓ Login Page:



The screenshot shows the Apache NetBeans IDE interface with the 'Employee Management System' project open. The 'Source Packages' tree view shows several Java files under the 'employee.management.system' package, including Login.java, which is currently selected and displayed in the main editor window. The code in the editor is as follows:

```
1 package employee.management.system;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6 import java.sql.*;
7
8 public class Login extends JFrame implements ActionListener{
9
10    JTextField tfusername, tfpassword;
11
12    Login() {
13
14        getContentPane().setBackground(Color.WHITE);
15        setLayout(null);
16
17        JLabel lblusername = new JLabel("Username");
18        lblusername.setBounds(40, 20, 100, 30);
19        add(lblusername);
20
21        tfusername = new JTextField();
22        tfusername.setBounds(150, 20, 150, 30);
23        add(tfusername);
24}
```

The status bar at the bottom indicates the application is running, the screen resolution is 1324x400, and the system date and time are 24-04-2024.



This screenshot shows the continuation of the Login.java code from the previous screenshot. The code adds a password field and a login button, and includes an ImageIcon for a logo. The code is as follows:

```
25    JLabel lblpassword = new JLabel("Password");
26    lblpassword.setBounds(40, 70, 100, 30);
27    add(lblpassword);
28
29    tfpassword = new JTextField();
30    tfpassword.setBounds(150, 70, 150, 30);
31    add(tfpassword);
32
33    JButton login = new JButton("LOGIN");
34    login.setBounds(150, 140, 150, 30);
35    login.setBackground(Color.BLACK);
36    login.setForeground(Color.WHITE);
37    login.addActionListener(this);
38    add(login);
39
40    ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/second.jpg"));
41    Image i2 = i1.getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT);
42    ImageIcon i3 = new ImageIcon(i2);
43    JLabel image = new JLabel(i3);
44    image.setBounds(350, 0, 200, 200);
45    add(image);
46
47    setSize(600, 300);
```

The status bar at the bottom indicates the application is running, the screen resolution is 1324x400, and the system date and time are 24-04-2024.

The screenshot shows the Apache NetBeans IDE 21 interface with the following details:

- Project:** Employee Management System
- Source Packages:** employee.management.system
- File:** Login.java
- Code:** The code implements a ActionListener for a JButton. It retrieves user input from JTextField tfusername and JPasswordField tfpassword. It then connects to a database using Conn.java and executes a query to check if the credentials are valid. If successful, it creates a new Home window; otherwise, it displays an invalid username or password message and keeps the current window visible.

```
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();
    }
}
```

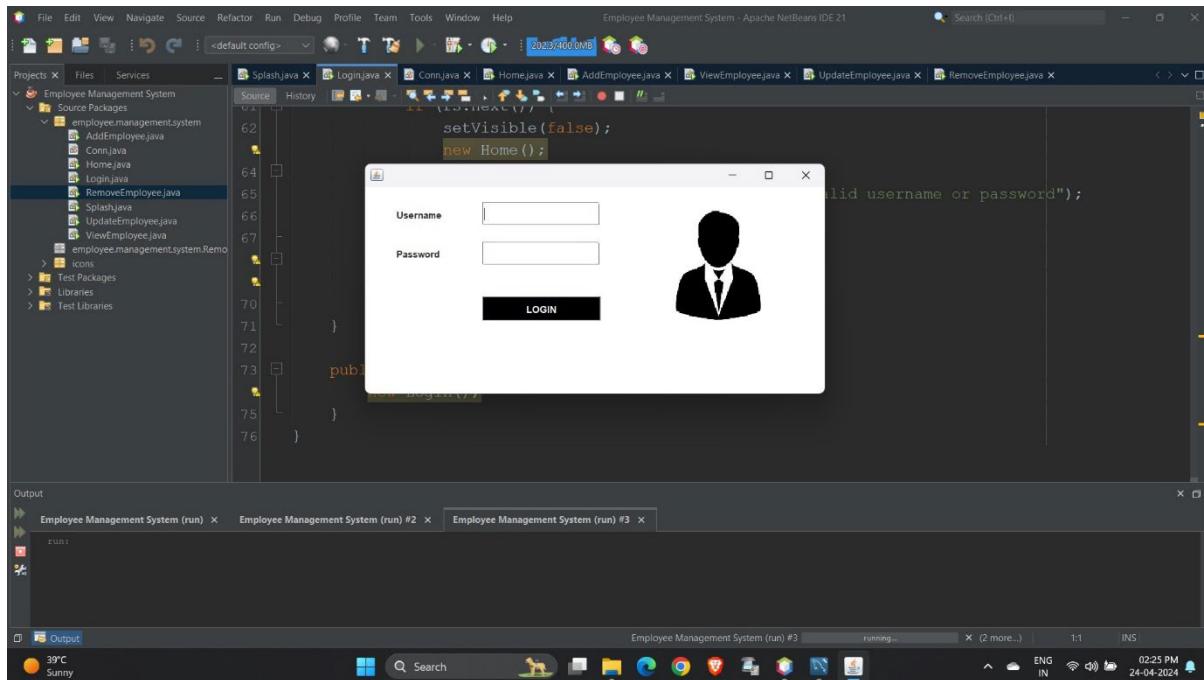
The screenshot shows the Apache NetBeans IDE 21 interface with the following details:

- Project:** Employee Management System
- Source Packages:** employee.management.system
- File:** Login.java
- Code:** The code implements a ActionListener for a JButton. It retrieves user input from JTextField tfusername and JPasswordField tfpassword. It then connects to a database using Conn.java and executes a query to check if the credentials are valid. If successful, it creates a new Home window; otherwise, it displays an invalid username or password message and keeps the current window visible. Additionally, the main method is defined to start the application by creating a new Login window.

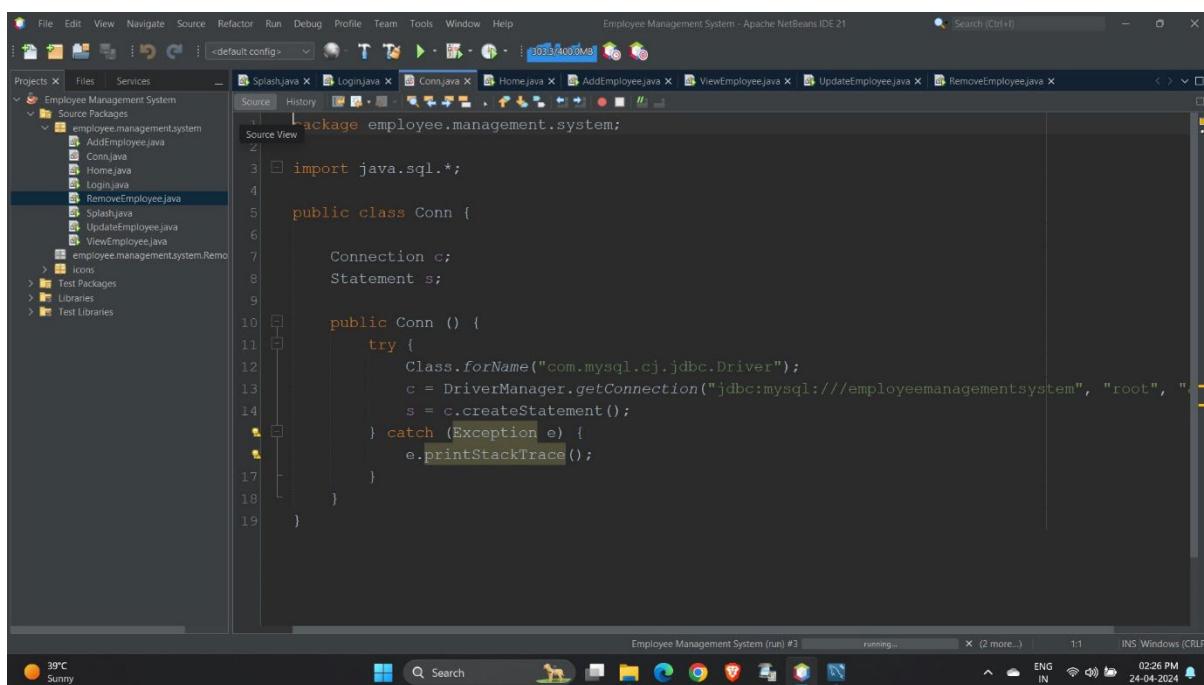
```
setVisible(false);
new Home();
} else {
    JOptionPane.showMessageDialog(null, "Invalid username or password");
    setVisible(false);
}
} catch (Exception e) {
    e.printStackTrace();
}

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

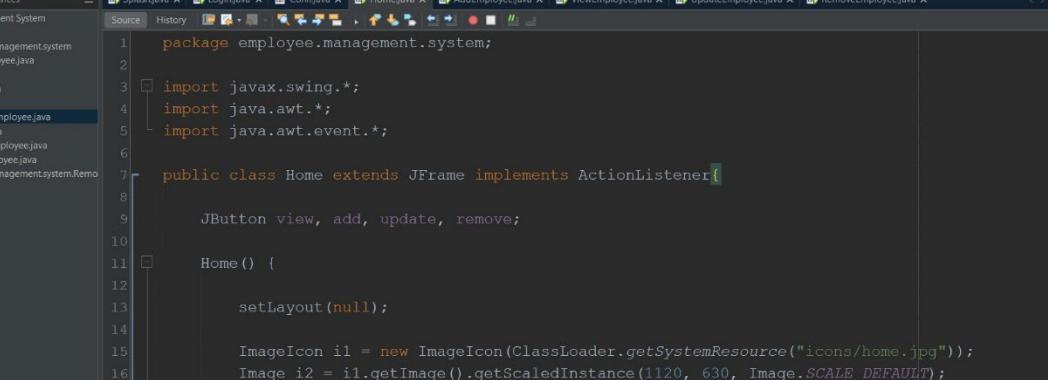
✓ Output of Login Page:



✓ Connection Page:



✓ Home Page:



The screenshot shows the Apache NetBeans IDE interface with the title "Employee Management System - Apache NetBeans IDE 21". The left sidebar displays the project structure under "Employee Management System" with files like AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. The main editor window shows the Java code for "Home.java". The code initializes a JFrame with a background image and a heading label.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Employee Management System - Apache NetBeans IDE 21 Search (Ctrl+) Projects Files Services Employee Management System + employee.management.system AddEmployee.java Conn.java Home.java Login.java RemoveEmployee.java Splash.java UpdateEmployee.java ViewEmployee.java employee.management.system.Remo icons Test Packages Libraries Test Libraries Source History 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(getClass().getResource("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));
```

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:** Standard NetBeans icons for file operations.
- Project Explorer:** Shows the "Employee Management System" project with packages like "employee-management-system" containing classes such as AddEmployee.java, Home.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, ViewEmployee.java, and Conn.java. It also lists "icons" and "Test Packages".
- Code Editor:** Displays Java code for a class that creates a graphical user interface (GUI) using Swing components like JButton. The code defines buttons for adding, viewing, updating, and removing employees, and sets their positions and sizes. It includes imports for java.awt, javax.swing, and java.awt.event.
- Bottom Status Bar:** Shows the application name "Employee Management System (run) #3", status "running...", and other system information like "39°C Sunny" and "02:27 PM IN 71:2 INS Windows (CRLF)".

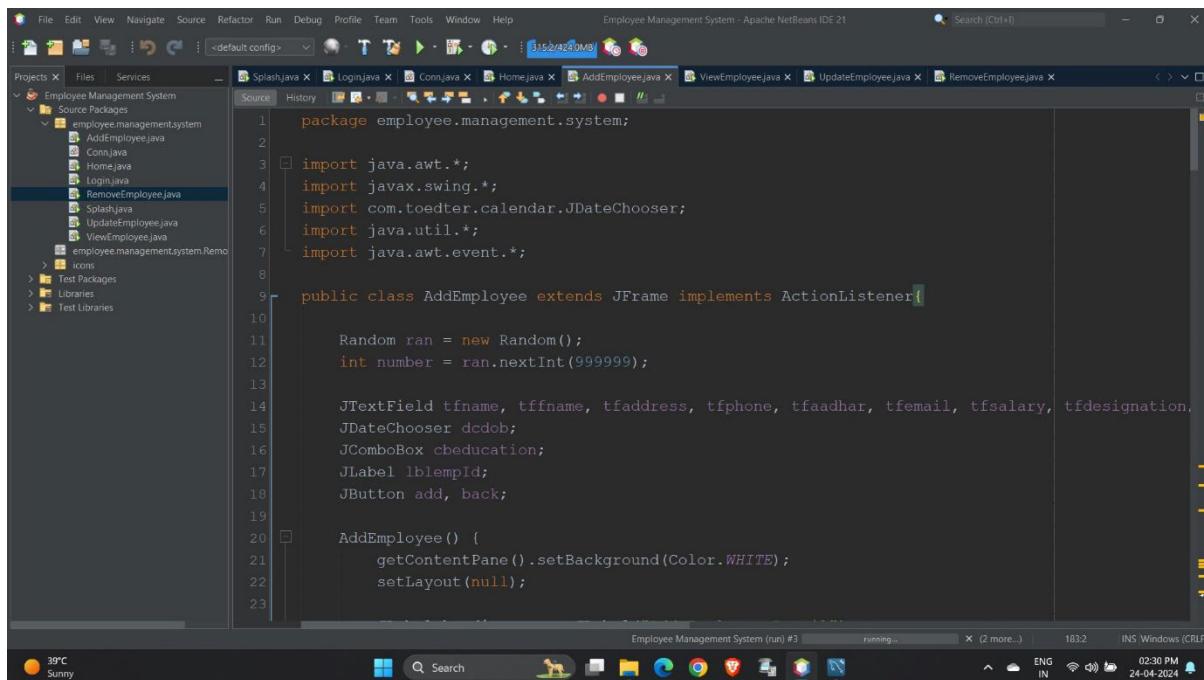
The screenshot shows the Apache NetBeans IDE interface. The left pane displays the project structure under 'Employee Management System' with 'Source Packages' expanded, showing files like Splash.java, Login.java, Conn.java, Home.java, AddEmployee.java, ViewEmployee.java, UpdateEmployee.java, and RemoveEmployee.java. The right pane shows the source code for Home.java. The code includes methods for setting location and visibility, handling action events for different menu items (Add Employee, View Employees, Update Employee, Remove Employee), and a main method that creates a new Home object.

```
4.8 setLocation(250, 100);
4.9 setVisible(true);
5.0 }
5.1
5.2 public void actionPerformed(ActionEvent ae) {
5.3     if (ae.getSource() == add) {
5.4         setVisible(false);
5.5         new AddEmployee();
5.6     } else if (ae.getSource() == view) {
5.7         setVisible(false);
5.8         new ViewEmployee();
5.9     } else if (ae.getSource() == update) {
5.10        setVisible(false);
5.11        new ViewEmployee();
5.12    } else {
5.13        setVisible(false);
5.14        new RemoveEmployee();
5.15    }
5.16 }
5.17
5.18 public static void main(String[] args) {
5.19     new Home();
5.20 }
```

✓ Output of Home Page:

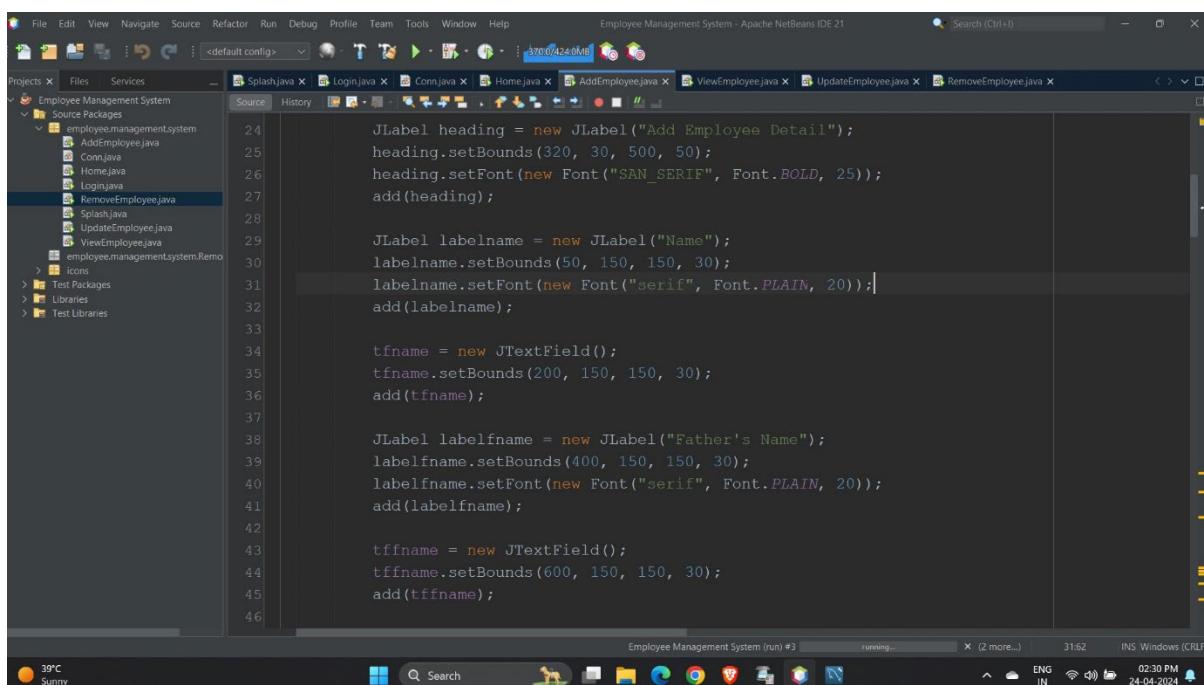


✓ Add Employee Page:



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

- Project:** Employee Management System
- Source Packages:** employee.management.system
- Selected File:** AddEmployee.java
- Code Preview:** The code defines the AddEmployee class, which extends JFrame and implements ActionListener. It includes imports for Java AWT, Swing, and Util, along with a JDateChooser and JComboBox.
- Toolbars and Menus:** Standard NetBeans menus like File, Edit, View, Navigate, Source, Refactor, Run, etc., are visible at the top.
- Status Bar:** Shows the project name "Employee Management System (run) #3", running status, and system information like time (24-04-2024), date (24-04-2024), and battery level (39°C Sunny).



The screenshot shows the continuation of the AddEmployee.java code from the previous screenshot:

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

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

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

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

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

This part of the code sets up labels and text fields for the employee's name and father's name.

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

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

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

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

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

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

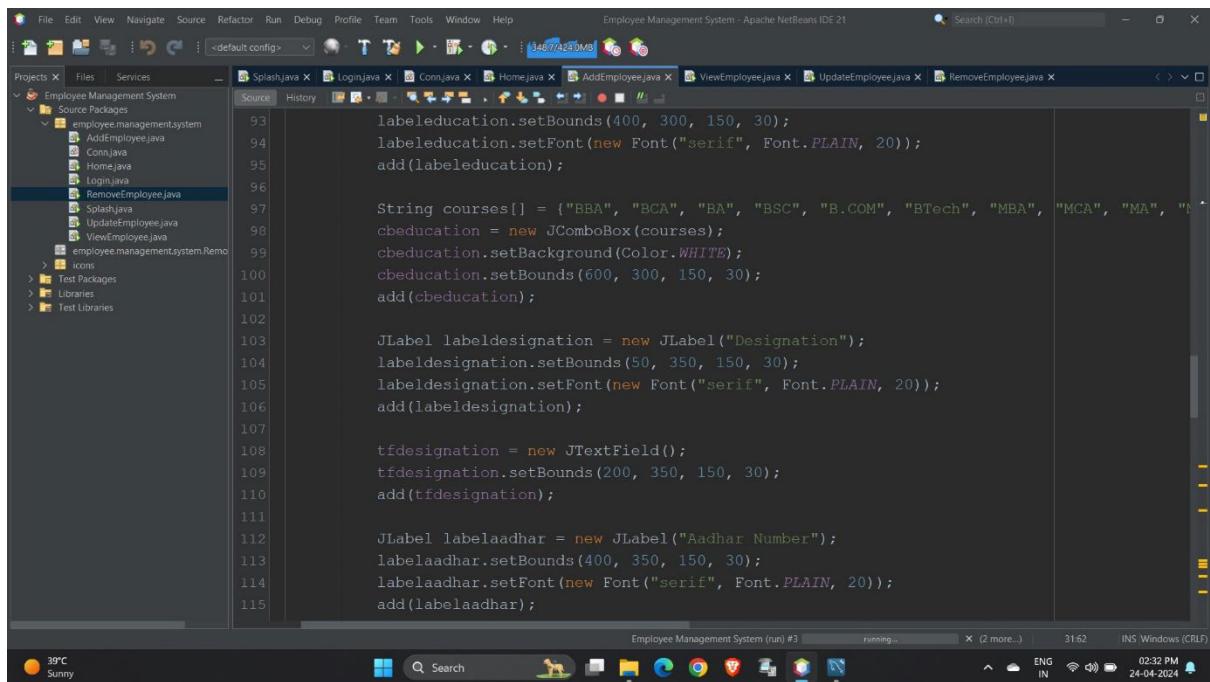
JLabel labelphone = new JLabel("Phone");
labelphone.setBounds(400, 250, 150, 30);
labelphone.setFont(new Font("serif", Font.PLAIN, 20));
add(labelphone);

JTextField 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);

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

JLabel labeleducation = new JLabel("Higest Education");
```



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

String courses[] = {"BBA", "BCA", "BA", "BSC", "B.COM", "BTech", "MBA", "MCA", "MA", "M.Tech"};
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);

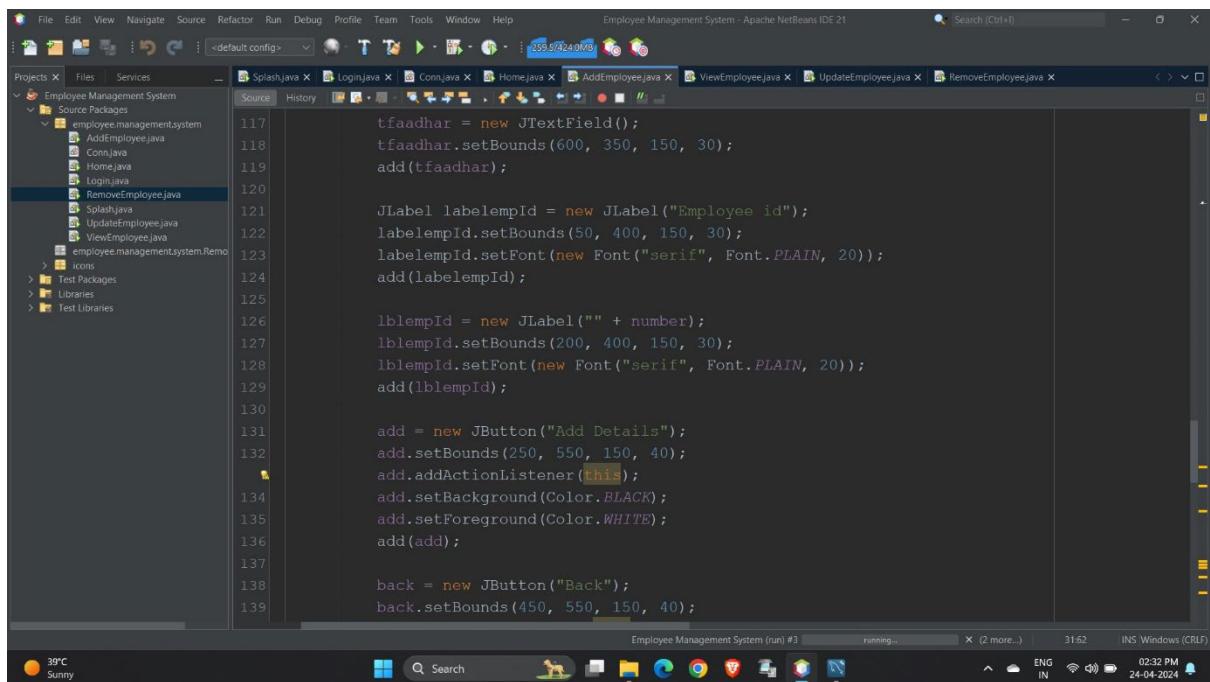
tfaadhar = new JTextField();
tfaadhar.setBounds(600, 350, 150, 30);
add(tfaadhar);

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);
```



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

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

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

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

134 back = new JButton("Back");
135 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, and Build. The Projects tab shows a single project named "Employee Management System" with a source package "employee.management.system" containing files: AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, ViewEmployee.java, and employee.management.system.Rem. The Source tab displays the code for AddEmployee.java, specifically the actionPerformed method which handles form submission to insert data into the database.

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

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

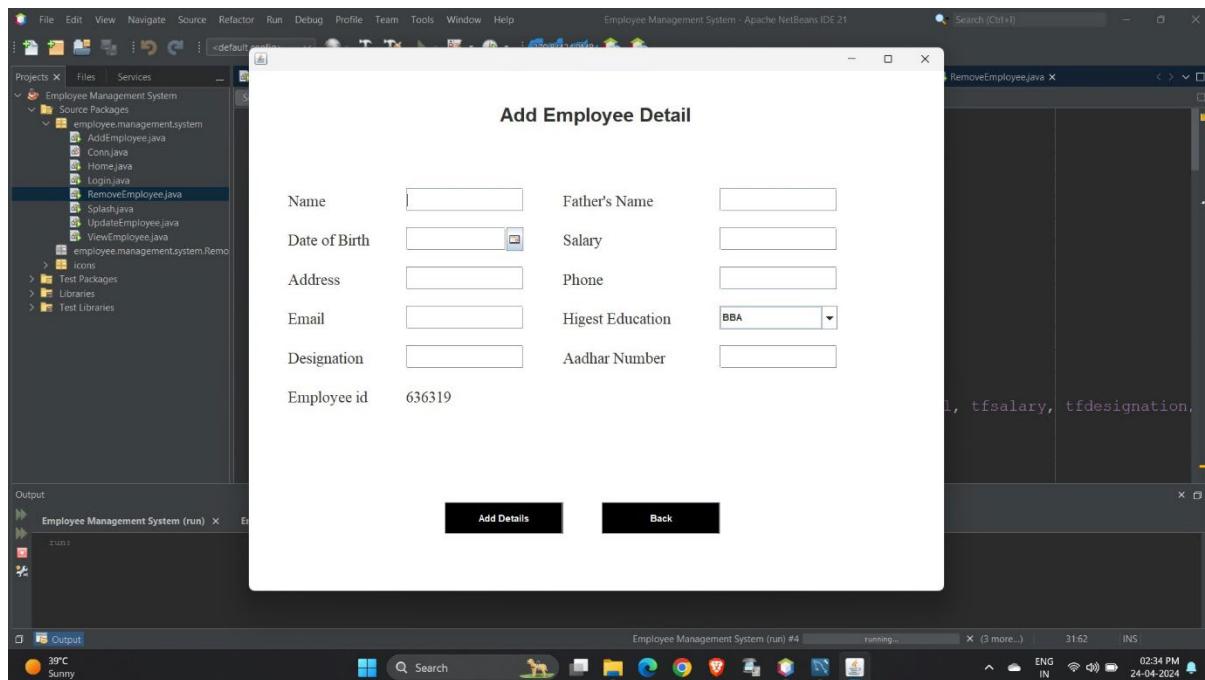
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == add) {
        String name = tfname.getText();
        String fname = tffname.getText();
        String dob = ((JText Field) dcdob.getDateEditor().getUiComponent()).getText();
        String salary = tfsalary.getText();
        String address = tfaddress.getText();
        String phone = tfphone.getText();
        String email = tfeemail.getText();
        String education = (String) cbeducation.getSelectedItem();
        String designation = tfdesignation.getText();
        String aadhar = tfaadhar.getText();
        String empId = lblempId.getText();
```

This screenshot shows the same Apache NetBeans IDE session with the code for AddEmployee.java. The code has been modified to include a try-catch block for handling database insertion errors. The main method is also present at the bottom of the code.

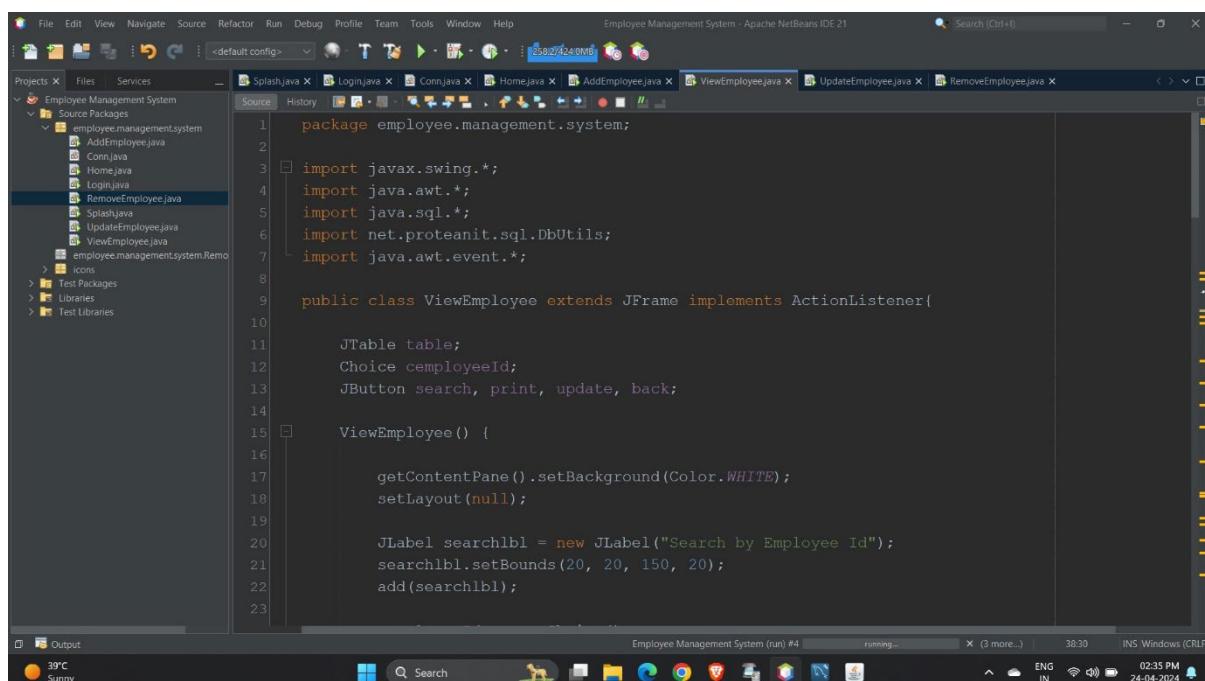
```
try {
    Conn conn = new Conn();
    String query = "insert into employee values('"+name+"', '"+fname+"', '"+dob+"', '"+salary+"', '"+address+"', '"+phone+"', '"+email+"', '"+education+"', '"+designation+"', '"+aadhar+"', '"+empId+"')";
    conn.s.executeUpdate(query);
    JOptionPane.showMessageDialog(null,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

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

✓ Output of Add Employee Page:



✓ View Employee Page:



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 single project named "Employee Management System" with a source package "employee.management.system" containing files AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. The Source tab displays Java code for the RemoveEmployee.java file:

```
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();
```

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 single project named "Employee Management System" with a source package "employee.management.system" containing files AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, and ViewEmployee.java. The Source tab displays Java code for the RemoveEmployee.java file, continuing from the previous snippet:

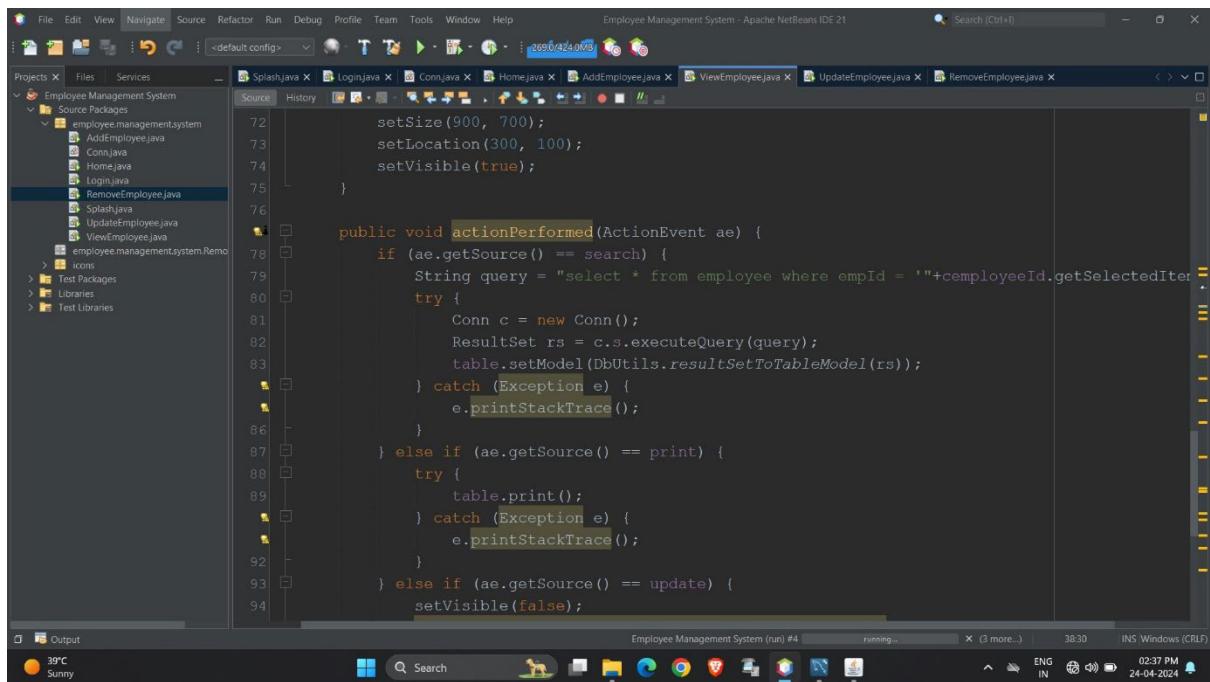
```
JScrollPane jsp = new JScrollPane(table);
jsp.setBounds(0, 100, 900, 600);
add(jsp);

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);

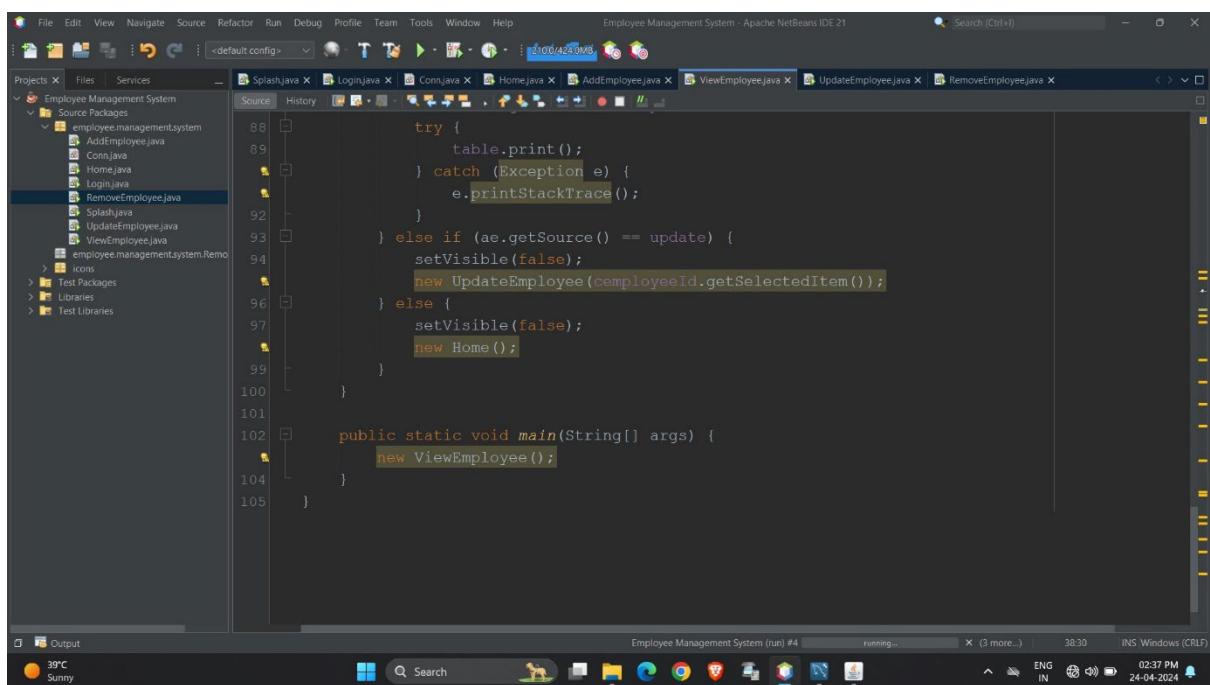
update = new JButton("Update");
update.setBounds(220, 70, 80, 20);
update.addActionListener(this);
add(update);

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



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

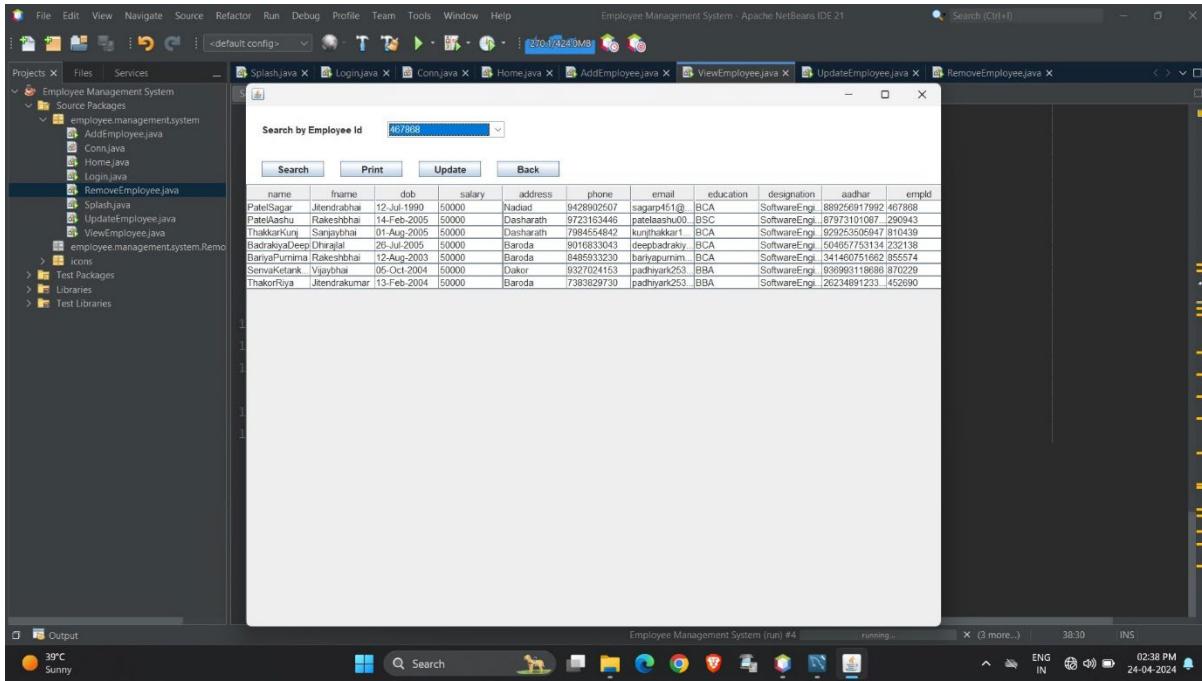
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == search) {
        String query = "select * from employee where empId = '" + employeeId.getSelectedItem();
        try {
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery(query);
            table.setModel(DbUtils.resultSetToTableModel(rs));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



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

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

✓ Output of View Employee Page:



✓ Update Employee Page:

```
package employee.management.system;

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

public class UpdateEmployee extends JFrame implements ActionListener {
    JTextField tffname, tfaddress, tfphone, tfaadhar, tfemail, tfsalary, tfdesignation;
    JLabel lblempId;
    JButton add, back;
    String empId;
    UpdateEmployee(String empId) {
        this.empId = empId;
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
        JLabel heading = new JLabel("Update Employee Detail");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        add(heading);
```

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 Build. 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, Conn.java, Home.java, Login.java, RemoveEmployee.java, Splash.java, UpdateEmployee.java, ViewEmployee.java, and icons. The Files tab is selected, displaying the content of RemoveEmployee.java. The code defines several JLabel and JTextField components for a user interface, setting their bounds and fonts. The code block starts at line 25:

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

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

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

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

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

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 Files tab is selected, displaying the content of RemoveEmployee.java. The code defines several JLabel and JTextField components for a user interface, setting their bounds and fonts. The code block starts at line 48:

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

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

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

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

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

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+F). The toolbar has icons for file operations like Open, Save, and Build. The Projects tab shows a tree view of the "Employee Management System" project, including packages like "employee.management.system" containing files such as AddEmployee.java, Conn.java, Home.java, Login.java, RemoveEmployee.java, SplashJava.java, UpdateEmployee.java, and ViewEmployee.java. The Source tab displays the code for RemoveEmployee.java, which handles the removal of an employee from the system. The code uses Java Swing components like JLabel, JTextField, and JPanel to create a user interface for entering employee details.

```
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();
```

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 code for RemoveEmployee.java, which continues from the previous snippet. It includes code for setting up labels and text fields for designation, Aadhar number, and Employee ID, and adds them to a panel.

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

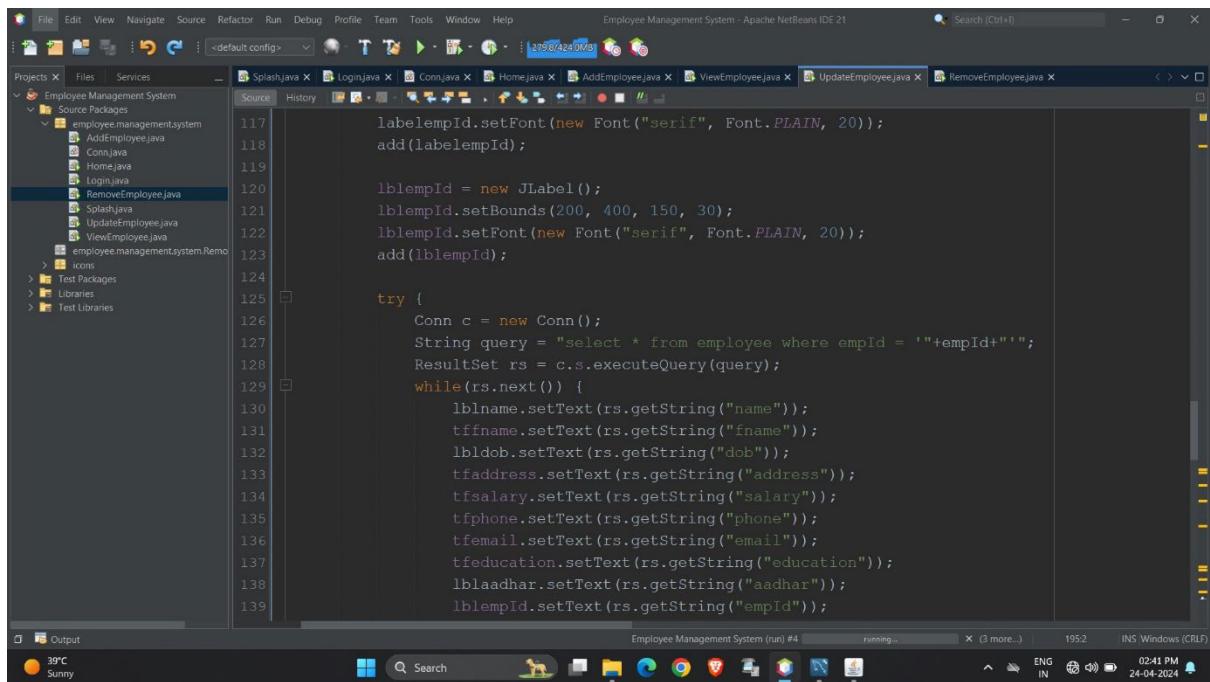
JLabel labeldesignation = new JLabel("Designation");
labeldesignation.setBounds(50, 350, 150, 30);
labeldesignation.setFont(new Font("serif", Font.PLAIN, 20));
add(labeldesignation);

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

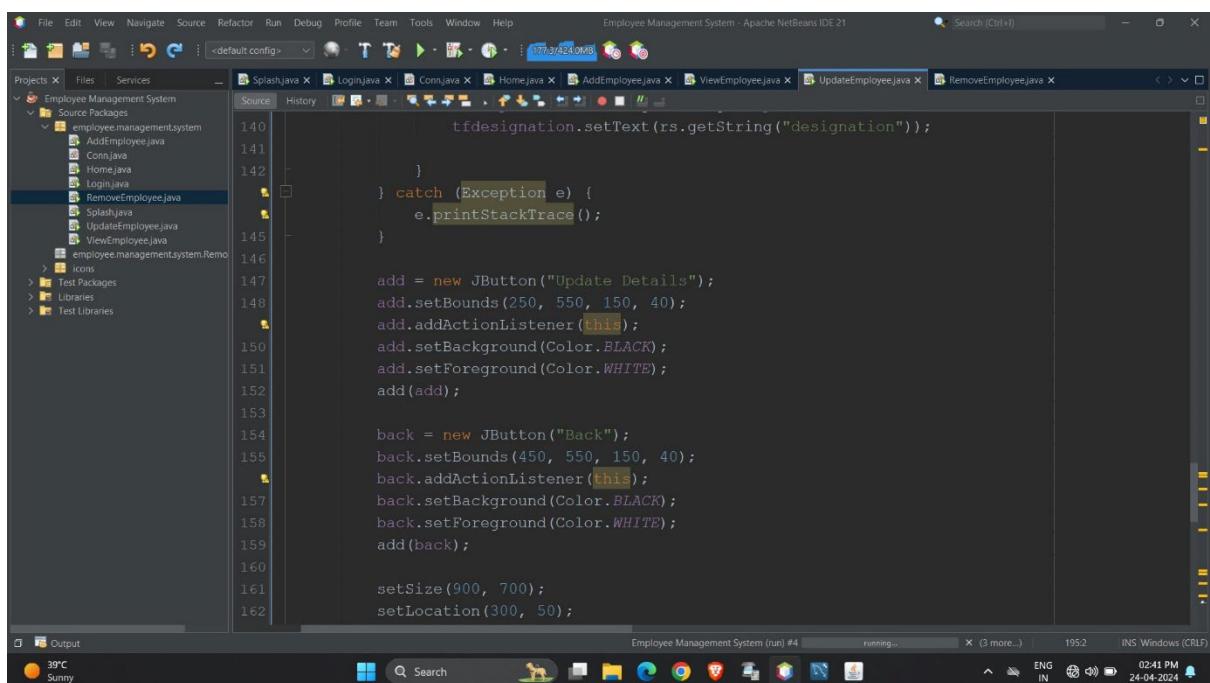
JLabel labelaadhar = new JLabel("Aadhar Number");
labelaadhar.setBounds(400, 350, 150, 30);
labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));
add(labelaadhar);

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

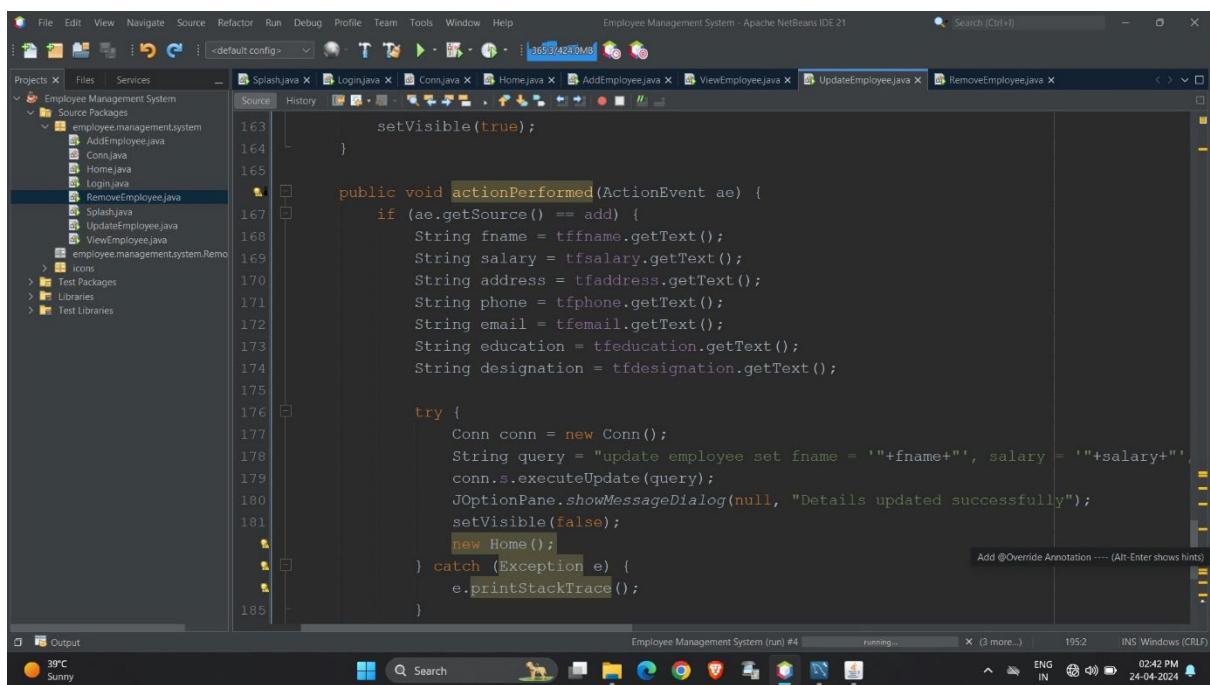
JLabel labelempId = new JLabel("Employee id");
labelempId.setBounds(50, 400, 150, 30);
```



```
Employee Management System - Apache NetBeans IDE 21
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+F)
Projects Files Services
Employee Management System
Source Packages
employee.management.system
AddEmployee.java Conn.java Home.java AddEmployee.java ViewEmployee.java UpdateEmployee.java RemoveEmployee.java
employee.management.system.Remo
Test Packages Libraries Test Libraries
117 lblempId.setFont(new Font("serif", Font.PLAIN, 20));
118 add(lblempId);
119
120 lblempId = new JLabel();
121 lblempId.setBounds(200, 400, 150, 30);
122 lblempId.setFont(new Font("serif", Font.PLAIN, 20));
123 add(lblempId);
124
125 try {
126     Conn c = new Conn();
127     String query = "select * from employee where empId = '" + empId + "'";
128     ResultSet rs = c.s.executeQuery(query);
129     while(rs.next()) {
130         lblname.setText(rs.getString("name"));
131         tffname.setText(rs.getString("fname"));
132         lbldob.setText(rs.getString("dob"));
133         tfaddress.setText(rs.getString("address"));
134         tfsalary.setText(rs.getString("salary"));
135         tfphone.setText(rs.getString("phone"));
136         tfemail.setText(rs.getString("email"));
137         tfeducation.setText(rs.getString("education"));
138         lblaadhar.setText(rs.getString("aadhar"));
139         lblempId.setText(rs.getString("empId"));
140     }
141 }
142 } catch (Exception e) {
143     e.printStackTrace();
144 }
145
146 add = new JButton("Update Details");
147 add.setBounds(250, 550, 150, 40);
148 add.addActionListener(this);
149 add.setBackground(Color.BLACK);
150 add.setForeground(Color.WHITE);
151 add(add);
152
153 back = new JButton("Back");
154 back.setBounds(450, 550, 150, 40);
155 back.addActionListener(this);
156 back.setBackground(Color.BLACK);
157 back.setForeground(Color.WHITE);
158 add(back);
159
160 setSize(900, 700);
161 setLocation(300, 50);
162
```



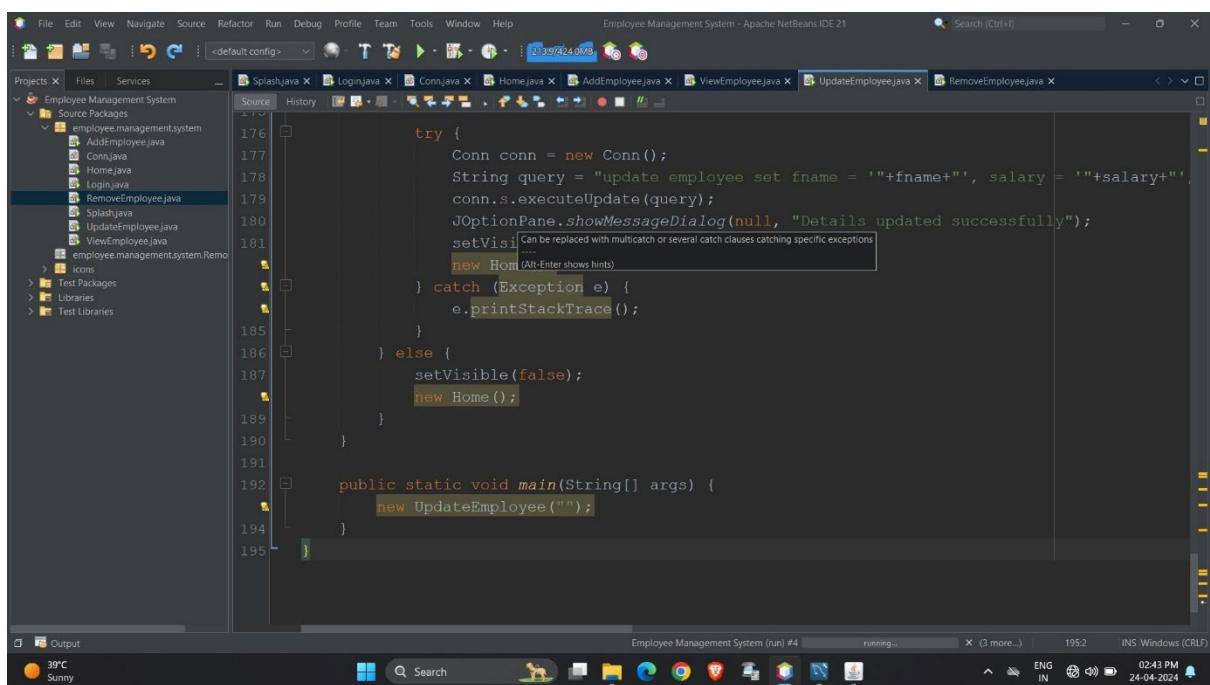
```
Employee Management System - Apache NetBeans IDE 21
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+F)
Projects Files Services
Employee Management System
Source Packages
employee.management.system
AddEmployee.java Conn.java Home.java AddEmployee.java ViewEmployee.java UpdateEmployee.java RemoveEmployee.java
employee.management.system.Remo
Test Packages Libraries Test Libraries
140     tfdesignation.setText(rs.getString("designation"));
141 }
142 }
143 } catch (Exception e) {
144     e.printStackTrace();
145 }
146
147 add = new JButton("Update Details");
148 add.setBounds(250, 550, 150, 40);
149 add.addActionListener(this);
150 add.setBackground(Color.BLACK);
151 add.setForeground(Color.WHITE);
152 add(add);
153
154 back = new JButton("Back");
155 back.setBounds(450, 550, 150, 40);
156 back.addActionListener(this);
157 back.setBackground(Color.BLACK);
158 back.setForeground(Color.WHITE);
159 add(back);
160
161 setSize(900, 700);
162 setLocation(300, 50);
163
```



```
setVisible(true);

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == add) {
            String fname = tffname.getText();
            String salary = tfssalary.getText();
            String address = tfaddress.getText();
            String phone = tfphone.getText();
            String email = tfemail.getText();
            String education = tfeducation.getText();
            String designation = tfdesignation.getText();

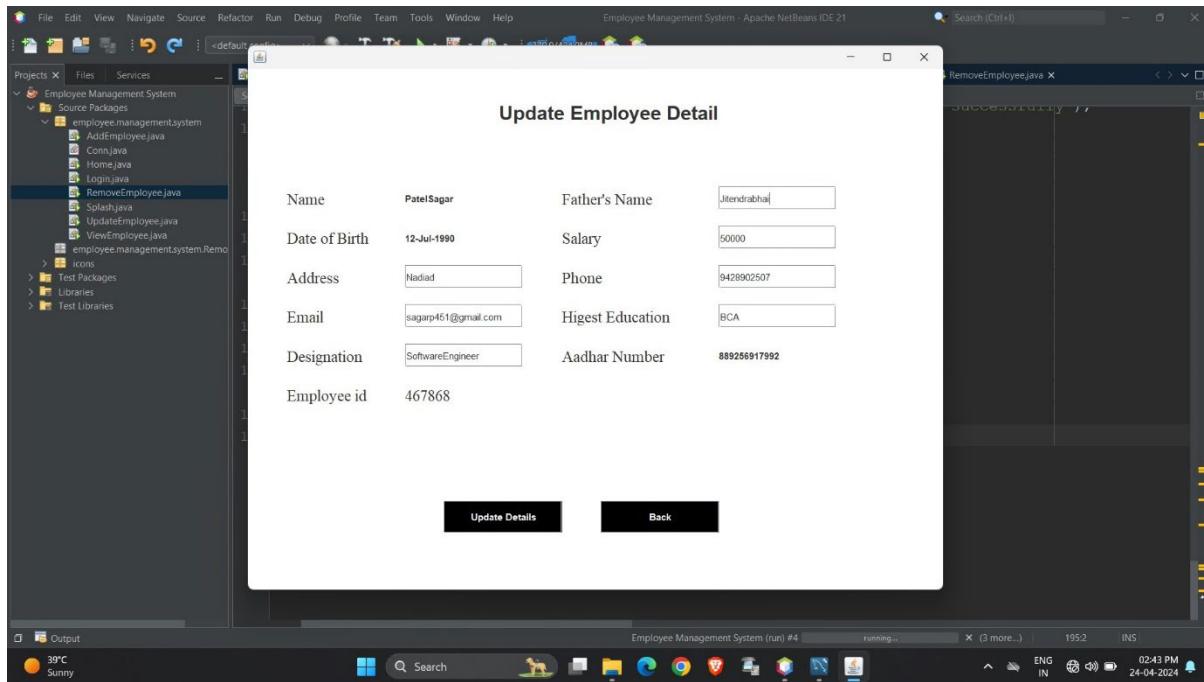
            try {
                Conn conn = new Conn();
                String query = "update employee set fname = '" + fname + "', salary = '" + salary + "'";
                conn.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null, "Details updated successfully");
                setVisible(false);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



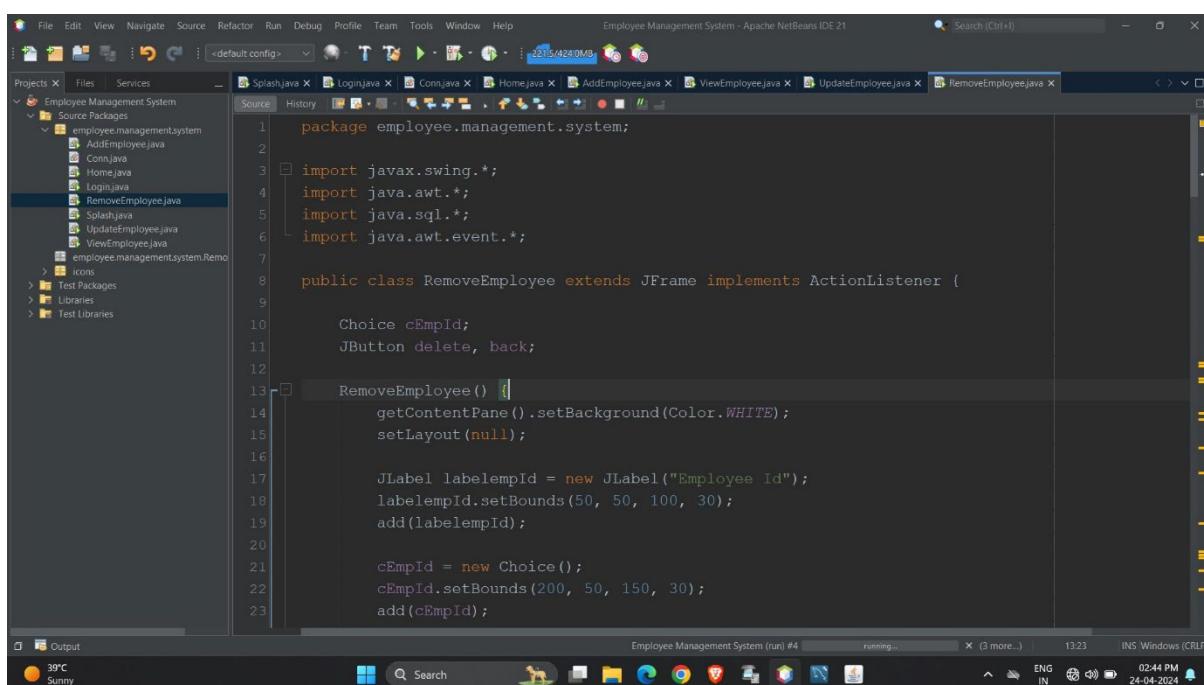
```
try {
    Conn conn = new Conn();
    String query = "update employee set fname = '" + fname + "', salary = '" + salary + "'";
    conn.s.executeUpdate(query);
    JOptionPane.showMessageDialog(null, "Details updated successfully");
    setVisible(...);
    new Home();
} catch (Exception e) {
    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 lblphone = new JLabel();
lblphone.setBounds(200, 150, 100, 30);
add(lblphone);

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

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

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

```
    }
    cEmpId.addItemListener(new ItemListener() {
        public void itemStateChanged(ItemEvent ie) {
            try {
                Conn c = new Conn();
                String query = "select * from employee where empId = '" + cEmpId.getSelectedItem();
                ResultSet rs = c.s.executeQuery(query);
                while(rs.next()) {
                    lblname.setText(rs.getString("name"));
                    lblemail.setText(rs.getString("email"));
                    lblphone.setText(rs.getString("phone"));
                }
            } 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 ii = new ImageIcon(ClassLoader.getSystemResource("icons/delete.png"));
Image i2 = ii.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) {

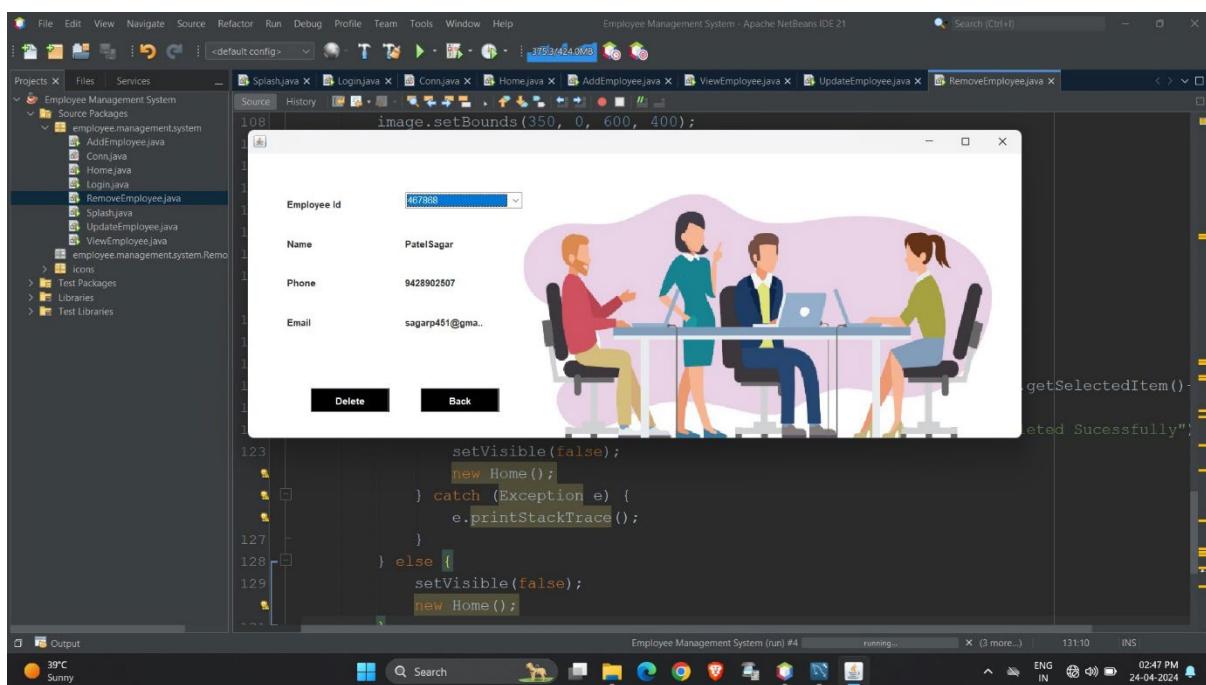
```

The screenshot shows the Apache NetBeans IDE 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 New, 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 main editor window displays the RemoveEmployee.java code:

```
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