EMPLOYEE MANAGEMENT SYSYEM

MINOR PROJECT REPORT

By

SANKAR M (RA2211031010024)
ANAND BALAJI S N (RA2211031010012)
GIRISAI S (RA2211031010011)

Under the guidance of

Dr. V. Nallarasan

(Assistant Professor, Department of Networking and Communications)

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in COMPUTER SCIENCE AND ENGINEERING

with specialization in INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in **"EMPLOYEE MANAGEMENT SYSTEM"** is the bonafide work of **SANKAR M (RA2211031010024), ANAND BALAJI S N (RA2211031010012) and GIRISAI S (RA2211031010011)** who carried out the work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dept. of Networking and Communications
Kattankulathur
603 203
SRM IST

DR. V.Nallarasan

GUIDE

Assistant Professor

Dept. of Networking and Communications

DR. ANNAPURANI PANAIYAPPAN.K

HEAD OF THE DEPARTMENT

Professor

Dept. of Networking and Communications

INTERNAL EXAMINER

EXTERNAL EXAMINER

Annexure II

Department of Networking and Communications

SRM Institute of Science & Technology

OWN WORK DECLARATION

Degree/ Course: B.Tech/Computer Science Engineering with specialization in Information Technology

Student Name : Sankar M/ Anand Balaji S N/ Girisai S

Registration Number : RA2211031010024/ RA2211031010012/ RA2211031010011

Title of Work : Employee Management System

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own

- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources) .
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my own work, except where indicated by referring, and that I have followed the good academic practices noted above. RA2211031010024 RA2211031010012 RA2211031010011 If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilitiesextended for the project work and his continued support. We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support. We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRMInstitute of Science and Technology, for her support throughout the project work. We are incredibly grateful to our Head of the Department, **Dr K. Annapurani Panaiyappan,** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Balakiruthiga**, Assistant Professor, Networking & Communications, SRM Institute of Science and Technology, for leading and helping us to completeour course. Our inexpressible respect and thanks to my guide, **Dr. V.Nallarasan**, Assistant Professor, Networking & Communications, SRM IST, for providing me with an opportunity to pursue my project under his mentorship. He provided me with the freedom and support to explore the research topics of my interest.

His passion for solving problems and making a difference in the world has always been inspiring. We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

ABSTRACT

An Employee Management System in Java serves as a comprehensive platform for efficiently handling various aspects of employee-related information and operations within an organization. This system embodies a modular and robust architecture to manage the entire employee lifecycle, from onboarding to offboarding, while ensuring data integrity and security. At its core, the system comprises multiple interconnected modules, each designed to address specific functionalities:

Authentication and Authorization Module: Provides secure login functionality to authenticate users and enforce access control based on roles and permissions.

Employee Information Module: Stores and manages comprehensive employee details, including personal information, contact details, job roles, departments, and performance-related data.

Attendance and Leave Management Module: Tracks employee attendance, manages leave requests, and automates the calculation of leave balances and accruals.

Payroll and Benefits Module: Manages salary details, benefits administration, tax deductions, and other financial aspects related to employees.

Performance Evaluation Module: Facilitates the assessment of employee performance through periodic reviews, goal setting, feedback mechanisms, and performance metrics tracking.

Training and Development Module: Identifies training needs, schedules training sessions, tracks employee skill development, and manages certification records.

Employee Self-Service Portal: Provides a user-friendly interface for employees to access and update their information, request leave, view pay stubs, and participate in performance reviews.

Reporting and Analytics Module: Generates comprehensive reports and analytics on various HR metrics, aiding in strategic decision-making and resource optimization.

The system is built on Java technologies, leveraging frameworks like Java Swing for the user interface and JDBC for database connectivity. It interacts with a MySQL database named "EmployeeManagementSystem," ensuring seamless data storage and retrieval.

Key features of this system include a user-friendly graphical interface, robust data validation and exception handling, encrypted data storage for sensitive information, and adherence to industry-standard security protocols.

Through its modular design and comprehensive functionality, the Employee Management System in Java provides organizations with a centralized, efficient, and scalable solution to streamline HR operations, enhance employee engagement , and maintain compliance with regulatorystandards.

TABLE OF CONTENT

1.INTRODUCTION

Motivation

Our drive to create the Employee Management System stems from the aim to revolutionize traditional workforce management, aligning it with the needs of contemporary organizations. We believe in fostering an environment where managing employees is seamless and efficient, empowering businesses to integrate workforce operations seamlessly into their dynamic workflows.

Our motivation is to bridge the gap between conventional employee management methods and the evolving demands of today's workplaces. We envision a system that offers accessibility and convenience, allowing organizations to effortlessly oversee and optimize their workforce, even in the midst of complex and rapidly changing operational landscapes.

Objective

In the development of this web-based tourism management system using the Java programming language within Netbeans environment and MySQL as the chosen database, the overarching goal is to craft a comprehensive and innovative platform. The project aims to go beyond mere functionality, aspiring to create an immersive and seamless experience for both customers and travel agencies alike.To ensure speed and efficiency, the Java programming language will be leveraged to optimize the system's performance, allowing for swift execution of tasks and seamless navigation. Netbeans known for its simplicity and robust features, will serve as the development environment, fostering a collaborative and efficient coding process.

Reliability is a cornerstone of this project, with a focus on robust architecture and stringent testing protocols. By implementing rigorous testing methodologies, the system will undergo thorough scrutiny to identify and rectify any potential issues, ensuring a stable and dependable platform for users.User-friendliness takes center stage, with a commitment to delivering an intuitive and visually appealing interface. The design will prioritize ease of use, enabling customers to effortlessly navigate through the platform. Additionally, personalized user profiles will contribute to a tailored experience, allowing individuals to manage preferences and track their booking history seamlessly.Accessibility is a key consideration, and the responsive design will guarantee a consistent and enjoyable user experience across various devices. Whether accessed from a desktop, tablet, or smartphone, users can expect a visually

optimized and user-friendly interface. The establishment of a versatile communication system holds paramount importance in this Employee Management System project. Real-time updates and notifications will bridge the communication divide between employees and management, fostering dynamic and responsive interaction. Features such as instant task assignment confirmations and timely updates on project status will elevate the overall communication flow.

The employee onboarding process, a critical touchpoint, is poised for a revolution in terms of convenience. Secure channels will be integrated to facilitate seamless and protected data transmissions. Stringent security protocols will safeguard sensitive employee information, instilling confidence in the reliability and security of the platform.

Scalability and adaptability form the core foundation of this project. The architecture is intricately designed to evolve with the growing needs of human resource management, allowing for seamless integration of new features and updates. This forward-thinking approach ensures that the system remains relevant and cutting-edge, adapting to the evolving landscape of employee management.

Problem Statement

Our Employee Management System (EMS) project is designed with the primary objective of modernizing and streamlining traditional human resource operations. Through the incorporation of automation and digitization, our project aims to revolutionize the way organizations manage their workforce, providing benefits for both the HR staff and employees. The following key goals underline our mission:

1.Enhance Operational Efficiency:The central aim of our EMS project is to eliminate the manual and time-consuming processes traditionally associated with employee onboarding, attendance tracking, and performance management. By doing so, we intend to significantly boost the operational efficiency of HR staff, allowing them to allocate more time to strategic tasks and improving the overall HR management experience.

2.mprove Accessibility: We envision an EMS that empowers employees with a user-friendly platform. This platform enables them to effortlessly access HR services, submit leave requests, and view important HR-related information. This improvement in accessibility ensures that HR services are easily accessible, making it simpler for employees to engage with and utilize HR resources.

3.Ensure Accuracy:Manual data entry and record-keeping often invite errors and inconsistencies in employee records. Our EMS project aims to mitigate these issues by maintaining precise and error-free records of employee information, attendance, and performance evaluations. This reliability is essential for HR staff to effectively manage human resources and for employees to trust the information they access.

4.Promote User Experience:Creating an intuitive and user-friendly interface is at the core of our project. We are dedicated to enhancing the user experience for both HR staff and employees. A well-designed system not only simplifies HR tasks but also fosters satisfaction among users. This improved experience is likely to encourage employees to engage more actively with HR services.

5. Enable Real-time Updates:Keeping track of employee attendance, performance, and HR activities in real-time is crucial for modern HR management. Our EMS project includes features that ensure that the information provided is always up-to-date and reliable. This real-time capability saves time and avoids discrepancies for HR staff seeking specific employee information.

6.Facilitate Data Analysis: We recognize the value of data in making informed HR decisions. Our EMS incorporates robust reporting tools that enable HR staff to analyze attendance patterns, identify skill gaps, and understand employee preferences. This data-driven approach not only aids in talent management but also ensures that HR remains a relevant and strategic function within the organization.

7. Ensure Security: In today's digital age, data security and privacy are paramount. Our EMS project takes these concerns seriously by implementing robust security measures. Employee information is protected, ensuring confidentiality and building trust within the organization.

In essence, our Employee Management System project aspires to bring HR management into the modern era, enhancing its efficiency, accessibility, and accuracy. By focusing on the user experience, real-time updates, data analysis, and security, we aim to create an HR environment that not only meets but exceeds the expectations of both HR staff and employees.

1. LITERATURE SURVEY

S.NO	PAPER TITLE	AUTHOR	YEAR	Algorithms/Tools/Techniques Used with drawback
1	Evaluation of knowledge management system to improve the performance of employees at PT Data Citra Mandiri	Citra Mandiri	2017	SmartPLS with SEM: Subject to potential bias due to self-reported questionnaire data.
2	Design of Employee Management Application for Small Medium Enterprise		2021	The system utilized System Development Life Cycle (SDLC) incorporating interviews and literature study, potentially limiting flexibility for dynamic changes.
3	Renae Broderick, John W. Boudreau, "Human resource management, information technology, and the competitive edge", Academy of Management Executive, 1992 Vol. 6 No. 2	Renae Broderick, John W. Boudreau	1992	Predictive analytics in HR may enhance decision-making but can perpetuate biases if based on flawed historical data.
4	ulie Bulmash, "Human Resource Management and Technology", Chapter 3.	ulie Bulmash	2009	Time tracking algorithms: May lead to employee stress due to constant monitoring and lack of privacy.
5	Ian Sommerville, "Software Engineering", 9th Edition, Addison-Wesley, 2011.	Ian Sommerville	2011	"An invaluable guide offering comprehensive insights into software engineering principles and practices, authored by Ian Sommerville."
6	Avison, D. and Fitzgerald, G. (2003).Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire	Avison, D. and Fitzgerald, G.	2003	Oracle applications, UML notation, RUP, RAD, DSDM, SAP, ERP systems, E-commerce, Dreamweaver, package development, component-based design, integrated with social and economic factors.
7	Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010.	Juan Manuel Munoz Palacio	2010	Java with SQL Server backend - potential scalability limitations with increased user load due to SQL Server backend.
8	Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How To Program,Dorling Kindersley, India.	Deitel, PJ & Deitel, HM	2008	The book "Internet & World Wide Web How To Program" by Deitel & Deitel provides comprehensive guidance on web programming and design principles.

2. REQUIREMENTS

Hardware Requirements:

- This is the central processing unit where the program will be executed. It includes the ALU (Arithmetic Logic Unit), registers, and control unit.
- This includes both ROM (Read-Only Memory) for storing the program and RAM (Random Access Memory) for storing data during execution.
- These are used for user interaction. For this project, a simple input device like a keyboard and an output device like a display or LEDs might be sufficient.
- These are necessary to connect various components of the microprocessor, allowing them to communicate.
- A clock generator or crystal oscillator is required to provide clock pulses to the microprocessor, synchronizing its operations.
- A stable power source is crucial for the proper functioning of the microprocessor and associated components.

Software Requirements:

- A software tool that translates assembly language code into machine code (binary) that the 8085 microprocessor can execute. Examples include assemblers like ASEM-85, ASMacro, etc.
- To test the program without the actual hardware, you can use an 8085 simulator or emulator. This allows you to run and debug your program on a computer.
- A simple text editor is needed to write and edit the assembly language code.
- This tool allows you to view and edit the machine code in hexadecimal format, which can be useful for debugging.

- If you're planning to run the code on actual hardware, you might need an interface to connect your computer to the microprocessor.
- For documenting your code, you might need tools like a word processor or LaTeX for creating reports or manuals.

3. ARCHITECTURE AND DESIGN

Certainly! Below is a brief procedure for the Java code you provided, which appears to be an Employee Management System:

Employee Management System Procedure:

1. Splash Screen:

Upon running the application, a splash screen appears with the title "EMPLOYEE MANAGEMENT SYSTEM."- The splash screen contains a clickable button labeled "CLICK HERE TO CONTINUE."

2. Login:

Clicking the "CLICK HERE TO CONTINUE" button opens the login window. The login window prompts for a username and password.Upon successful login, it opens the main application window.

3. Home:

The main application window is the Home screen.It displays the title "Employee Management System" and provides options to:

- Add Employee
- View Employees
- Update Employee
- Remove Employee

4. Add Employee:

Clicking on "Add Employee" in the Home screen opens a form to enter employee details. Employee details include name, father's name, date of birth, salary, address, phone, email, education, designation, Aadhar number, and an automatically generated employee ID. Clicking "Add Details" saves the employee details to the database.

5. View Employee:

Clicking on "View Employees" in the Home screen opens a window to view a table of all employees.Users can search for a specific employee by entering their employee ID. The table displays information such as name, father's name, date of birth, salary, address, phone, email, education, designation, Aadhar number, and employee ID. Options to print the table, search for a specific employee, update an employee's details, and go back to the Home screen are provided.

6. Update Employee:

Clicking on "Update Employee" in the View Employee window allows users to update an employee's details. Users can select an employee from the dropdown, and the form is populated with the existing details. They can modify the details and click "Update Details" to save changes.

7. Remove Employee:

Clicking on "Remove Employee" in the Home screen opens a window to remove an employee. Users can select an employee from the dropdown to view details and confirm deletion. Clicking "Delete" removes the employee from the database.

8. Database Connectivity:

The application uses a MySQL database named "employeemanagementsystem.". The `Conn` class handles database connectivity with the JDBC driver. SQL queries are used to interact with the database for various operations.

9. Security:

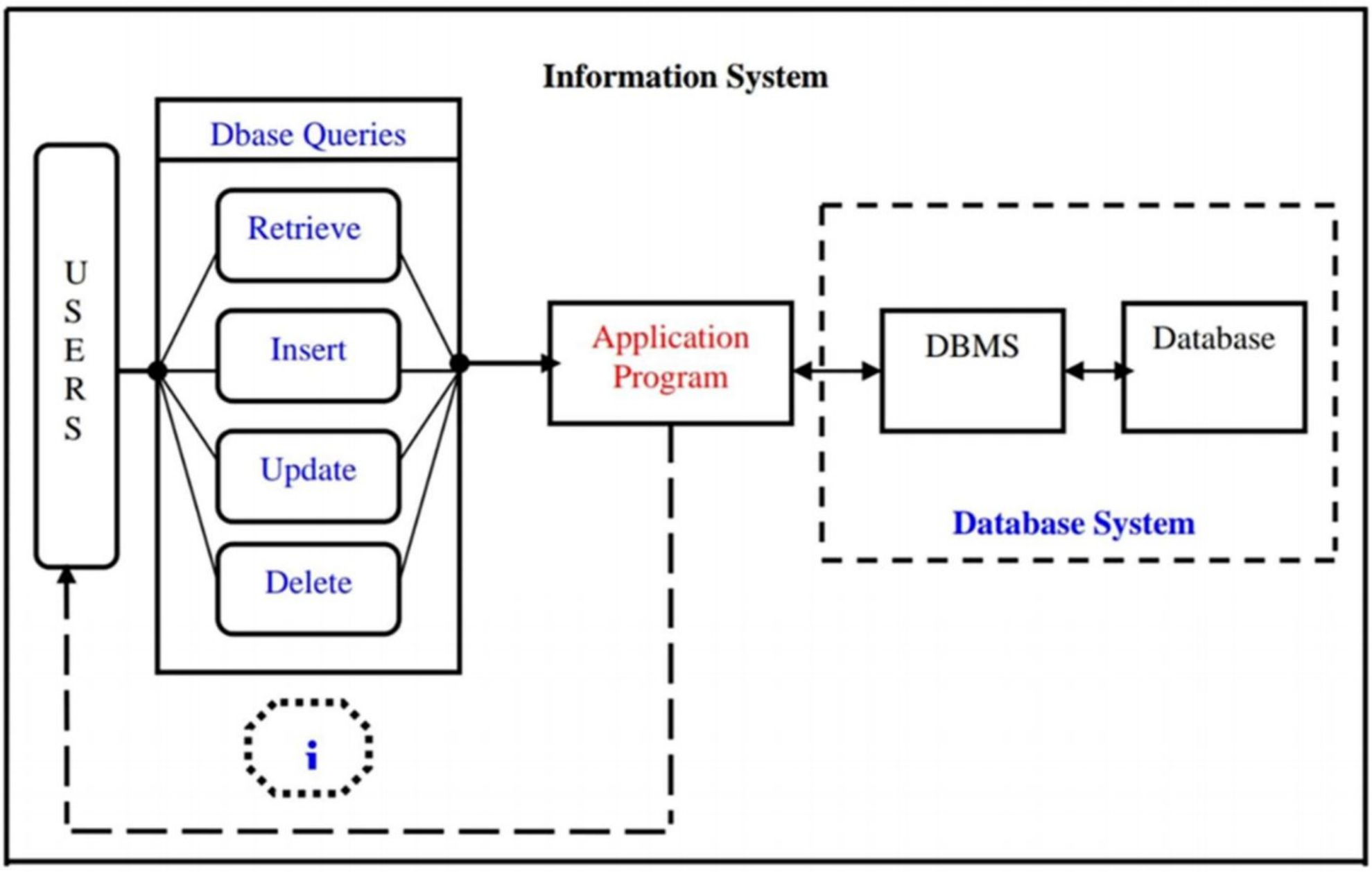
The application includes a login system to ensure secure access. Passwords and usernames are verified against the database.

10. User Interface:

The application uses Swing for GUI components, providing a user-friendly interface.

11. Closing the Application:

Users can close the application using the close (X) button on the window.



5. IMPLEMENTATION

The development of the Employee Management System entailed the utilization of diverse technologies and the creation of specific components tailored to meet the distinct requirements of efficient workforce administration. This section provides an insight into the technical aspects and functionalities of the software.

Technology Stack:

The "Employee Management System" application was implemented using the followingtechnologies:

1.Java Programming Language: The core of the software is built using Java,a versatile and platform-independent language. Java was chosen for its robustness and ability to create cross-platform applications.

2.MySQL Database: In the Medicare app, SQL facilitates secure user authentication. Stored user IDs and passwords are queried using SQL SELECTduring login. The entered credentials are compared with the stored data; a match grants access, ensuring secure user verification.

Component Implementation:

User Interface:

The user interface was designed with a focus on usability and visual appeal. Java Swing components were used to create a responsive and intuitive interface. Users can interact with the software through graphical elements like buttons, text fields, and labels, making it easy to navigate and use the various features.

Testing and Debugging:

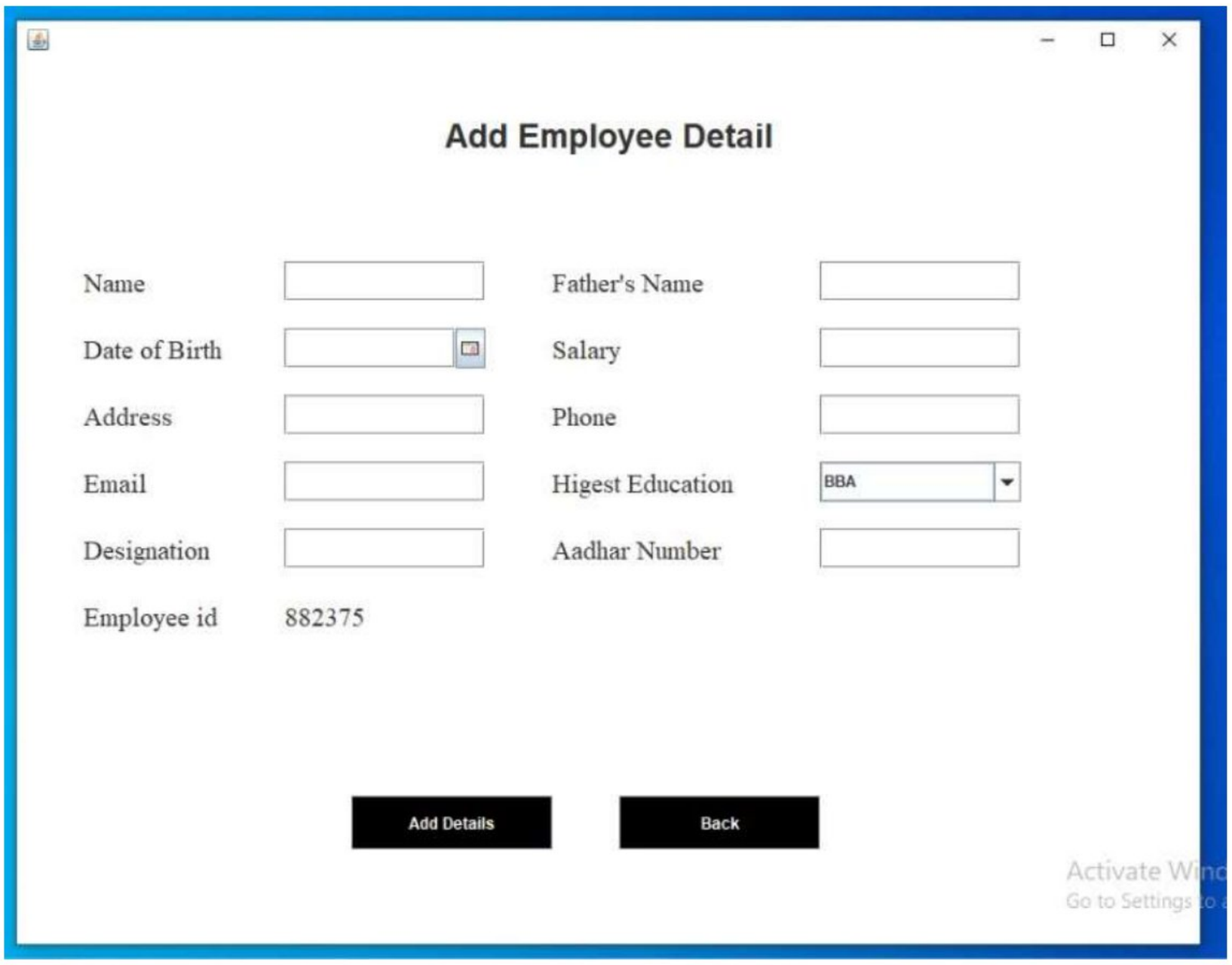
Throughout the implementation phase, rigorous testing and debugging were conducted to identify and rectify any issues. Testing included user experience testing to ensure that the software's features met the needs and expectations of patients.

6.RESULTS AND DISCUSSION

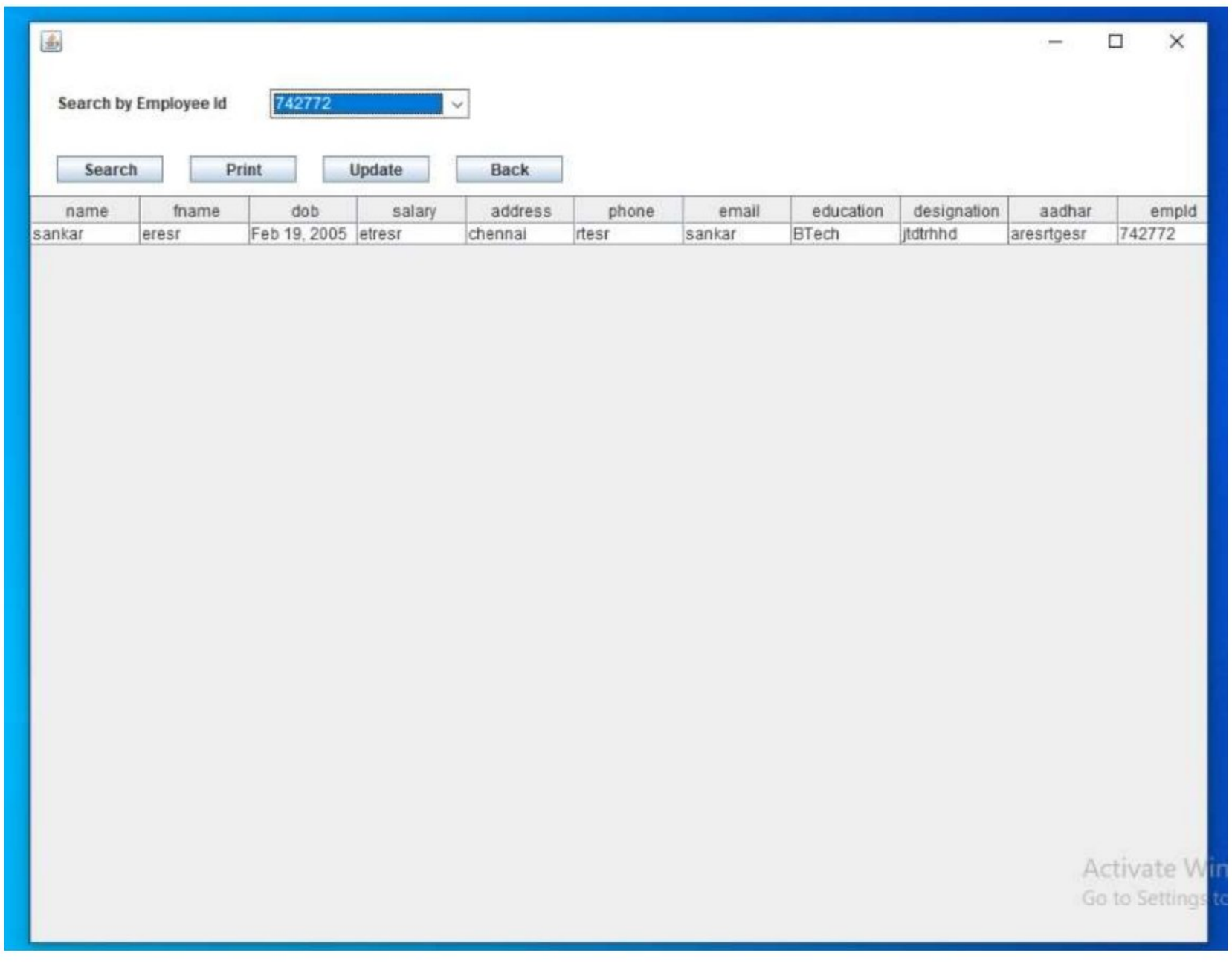
LOGIN PAGE:

HOME PAGE:

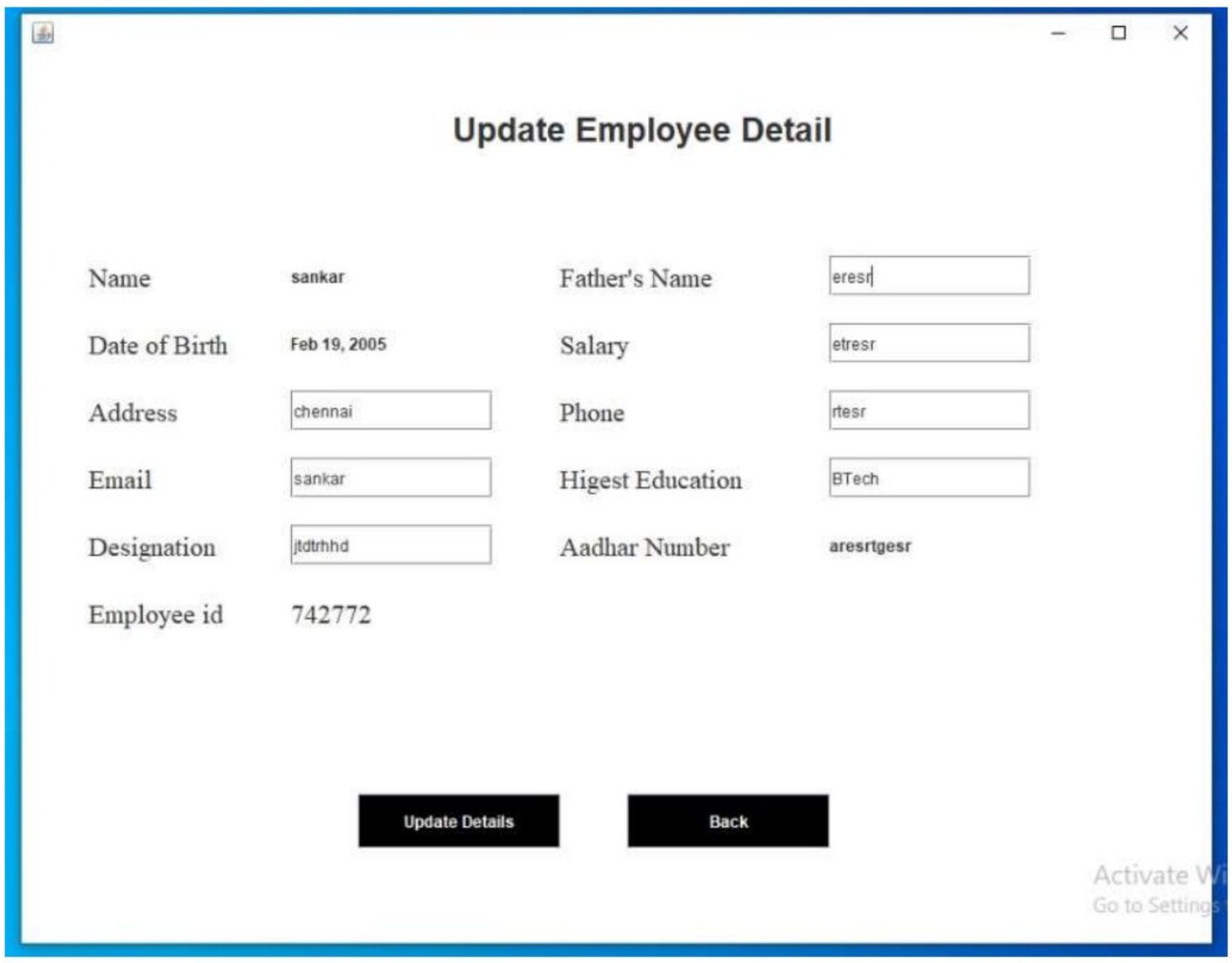
ADD EMPLOYEE:



VIEW EMPLOYEE:



UPDATE EMPLOYEE:



DELETE EMPLOYEE:

7.CONCLUSION

- A Java-based employee management system offers an efficient, secure, and scalable solution for handling workforce resources within organizations.

- Its automated features, cross-platform compatibility, and customizable options empower HR professionals to streamline employee management, adapt to evolving demands, and leverage data analysis and reporting tools.

- With cost-saving functionalities and compliance features, this system becomes a vital asset for companies, ensuring effective personnel management while facilitating informed decision-making.

- The system's versatility caters to a range of organizational needs, making it a reliable tool for enhancing HR processes and strategic planning within diverse business settings.

8.REFERENCES

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

9.APPENDIX

ADD EMPLOYEE PAGE:

```
package employee.management.system;

import java.awt.*;
import javax.swing.*;
import com.toedter.calendar.JDateChooser;
import java.util.*;
import java.awt.event.*;

public class AddEmployee extends JFrame implements ActionListener{

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

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

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

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

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

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

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

    tffname = new JTextField();
```

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

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

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);

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);

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);

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);

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

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

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

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

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

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

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

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

        back = new JButton("Back");
```

```
        back.setBounds(450, 550, 150, 40);
        back.addActionListener(this);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        add(back);

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

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

            try {
                Conn conn = new Conn();
                String query = "insert into employee values('"+name+"', '"+fname+"', '"+dob+"',
'"+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();
    }
}
```

CONNECTION OF JDBC:

```
package employee.management.system;

import java.sql.*;
public class Conn {

    Connection c;
    Statement s;

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

HOME PAGE:

```
package employee.management.system;

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

public class Home extends JFrame implements ActionListener{

    JButton view, add, update, remove;

    Home() {

        setLayout(null);

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

```
add(image);

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

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

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

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

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

    setSize(1120, 630);
    setLocation(250, 100);
    setVisible(true);
}

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

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

LOGIN PAGE:

```
package employee.management.system;

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

public class Login extends JFrame implements ActionListener{

    JTextField tfusername, tfpassword;

    Login() {

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

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

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

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

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

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

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

```
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(350, 0, 200, 200);
        add(image);

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

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

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

DELETE EMPLOYEE PAGE:

```
package employee.management.system;

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

public class RemoveEmployee extends JFrame implements ActionListener {
```

```
Choice cEmpId;
JButton delete, back;

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

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

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

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

    JLabel labelname = new JLabel("Name");
    labelname.setBounds(50, 100, 100, 30);
    add(labelname);

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

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

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

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

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

```
        add(lblemail);

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

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

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

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

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

setSize(1000, 400);
setLocation(300, 150);
setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        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();
    }
}
```

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.RED);
add(heading);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("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);
add(image);

JButton clickhere = new JButton("CLICK HERE TO CONTINUE");
clickhere.setBounds(400, 400, 300, 70);
clickhere.setBackground(Color.BLACK);
clickhere.setForeground(Color.WHITE);
clickhere.addActionListener(this);
image.add(clickhere);

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

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

    }

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

    }
  }
}

public void actionPerformed(ActionEvent ae) {
   setVisible(false);
```

```
        new Login();
    }

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

VIEW EMPLOYEE PAGE:

```
package employee.management.system;

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

public class ViewEmployee extends JFrame implements ActionListener{

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

    ViewEmployee() {

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

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

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

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

        table = new JTable();
```

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

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

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

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

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

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

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

public void actionPerformed(ActionEvent ae) {
      if (ae.getSource() == search) {
          String query = "select * from employee where empId =
'"+cemployeeId.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);
    new UpdateEmployee(cemployeeId.getSelectedItem());
        } else {
            setVisible(false);
            new Home();
        }
    }

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

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

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

JLabel heading = new JLabel("Update 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);

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);

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);

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);

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);

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);

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

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

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

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

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

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

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

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

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

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

lblempId = new JLabel();
```

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

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

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

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

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

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

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

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

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

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