EMPLOYEE PAYROLL MANAGEMENT SYSTEM

MINI-PROJECT REPORT

Submitted by

VIRESHWER C A	**241901126**
SIDHARTH REDDY C	**241901107**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **"EMPLOYEE PAYROLL MANAGEMENT SYSTEM"** is the bonafide work of **"VIRESHWER C A(241901126), SIDHARTH REDDY C (241901107)"** who carried out the project work under my supervision.

SIGNATURE

Ms.R.Rupmala

ASSISTANT PROFESSOR

Department of Computer Science and
Engineering(Cyber Security)
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on _____

NTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the mini project report **EMPLOYEE PAYROLL**, submitted as part of the curriculum requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work carried out by us under the supervision of Ms. R. Rupmala, Assistant Professor, Department of Computer Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.

This submission represents our ideas in our own words, and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data, idea, fact, or source in our submission. We understand that any violation of the above will be grounds for disciplinary action by the institute and/or the University and may also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Rajalakshmi Engineering College, **VIRESHWER C A**

Chennai **SIDHARTH REDDY C**

November 2025

ABSTRACT

The **Employee Payroll Management System** is designed to streamline, automate, and secure the process of calculating and managing employee salaries within an organization. Traditional payroll processes often involve manual calculations, repetitive data entry, and increased chances of errors, resulting in inefficiency and compliance risks. This system provides a centralized platform to mana employee records, attendance, salary components, tax deductions, allowances, and statutory compliance.

Keywords: Inventory Management System, JavaFX, SQL Database, Product Tracking, Stock Monitoring, Local Businesses, Real-Time Inventory, Database Application.

ACKNOWLEDGEMENT

We like to convey our sincere appreciation to all who have supported and mentored us during the successful completion of this project work.

We express our profound gratitude to **Mr. Benedict J.N**., Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project.

We express our profound gratitude to **Ms. R. Rupmala,** Assistant Professor in the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for her unwavering support, invaluable insights, and collaboration throughout our endeavor.

We would like to convey our gratitude to our faculty members and colleagues for their valuable feedback and encouragement.

We express our gratitude to our families and friends for their steadfast support, patience and encouragement, which were instrumental in the effective execution of this seminar

VIRESHWER CA

SIDHARTH REDDY C

Table Of Content

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The **Employee Payroll Management System** is a software solution designed to automate and streamline the process of calculating employee salaries, maintaining employee records, managing attendance, deductions, allowances, taxes, and generating payslips. The system eliminates manual calculations and reduces human errors, ensuring accuracy, efficiency, and transparency in payroll processing.

Organizations with large workforces face challenges in maintaining accurate records of employee salaries, attendance, and statutory deductions. Manual payroll processing is time-consuming The **Employee Payroll Management System** provides centralized, automated platform for manag payroll operations efficiently.

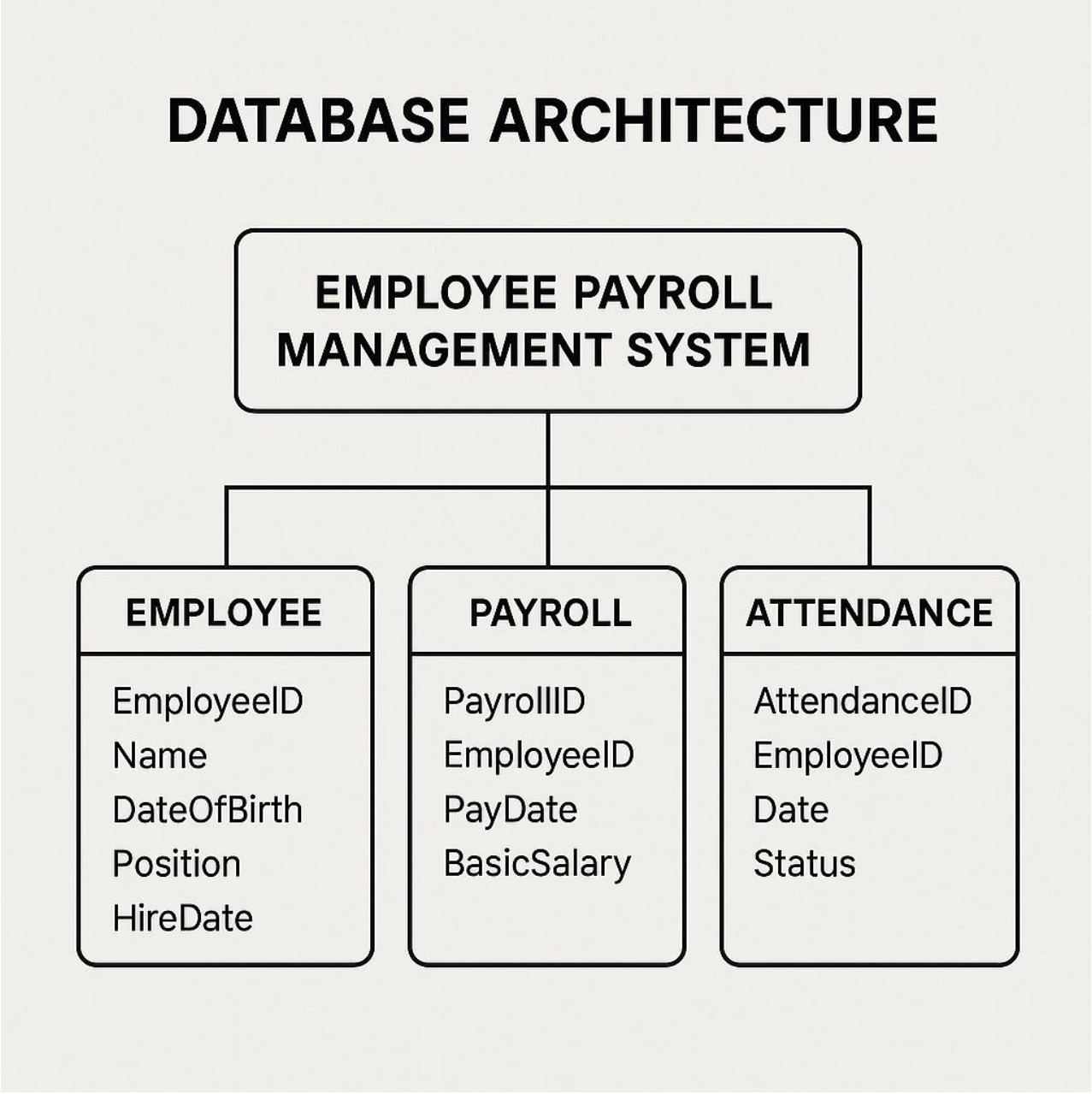

FIG.1.1.DATABASE ARCHITECTURE

1.2 Scope of the Work

The scope of this project defines the boundaries, functionalities, and tasks that the Employee Payroll Management System will cover. It outlines what the system will do, who will use it, and the processes included within the project.

Key features include:

1.2.1 Employee Management

- Add, update, view, and delete employee profiles.
- Store personal, job, bank, and salary-related information.
- Maintain department and designation details.

1.2.2 Attendance & Leave Tracking

- Record daily attendance (Present/Absent/Half-day/Leave).
- Maintain leave requests and approvals.
- Attendance automatically linked to salary calculations.

1.2.3 Automated Payroll Processing

- Auto-calculate monthly salaries based on:
    - Basic pay
    - Allowances (HRA, DA, TA, etc.)
    - Deductions (PF, ESI, Tax, PT, loans)
    - Leaves and overtime
- Eliminates manual errors and ensures accuracy.

1.3 Problem Statement

Managing employee payroll manually is a complex, time-consuming, and error-prone process for most organizations. HR departments face difficulties in maintaining accurate employee records, calculating salaries, managing attendance, and ensuring compliance with statutory regulations such as PF, ESI, and taxes. Manual calculations often lead to discrepancies, delays in salary disbursement, data redundancy, and lack of transparency.

There is a need for an automated, reliable, and efficient system that can handle employee information, attendance data, salary structures, and payroll generation in an integrated manner. The system should minimize human errors, improve accuracy in salary computation, ensure timely payslip generation, and provide secure access to data.

1.4Aim and Objectives of the Project

The aim of the Employee Payroll Management System is to design and develop an automated software solution that efficiently manages employee records, attendance, salary calculations, and payroll generation while ensuring accuracy, security, and ease of access.

---

1. Automate Payroll Processing

- Eliminate manual calculations by automatically computing salaries, allowances, deductions, and taxes.

2. Maintain Centralized Employee Records

- Store complete employee information including personal details, job data, salary structure, and attendance.

3. Improve Accuracy and Reduce Errors

- Minimize human mistakes in salary computation, data entry, and statutory deductions.

4. Streamline Attendance and Leave Management

- Link attendance and leave details directly with payroll to ensure correct compensation.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front – End	:	Java
Back - End	:	MySql
Language	:	Java,SQL

CHAPTER 3

MODULE DESCRIPTION

- The Employee Payroll Management System is designed using a modular architecture, where each module handles a specific functional area. This improves maintainability, scalability, and security while allowing parallel development of components.

---

1. Employee Management Module
Purpose:
Handles employee-related data and acts as the core master module of the system.
Key Functions:

- Add, update, delete employee information
- Store personal details, contact information, job role, department
- Maintain salary structure and bank details
- Retrieve employee profile data for payroll processing

---

2. Attendance & Leave Management Module
Purpose:
Tracks employee presence and leave details connected to payroll calculations.
Key Functions:

- Record daily attendance (Present/Absent/Half-Day/Leave)
- Maintain leave types (Casual Leave, Sick Leave, etc.)
- Track approved leaves and pending requests
- Generate attendance reports

---

3. Salary Setup Module
Purpose:
Defines all payroll components and salary structures.
Key Functions:

- Configure basic salary and grade pay
- Set allowances (HRA, DA, TA, Bonus, etc.)
- Set deductions (PF, ESI, TDS, PT, Loans)
- Assign custom salary structures employee-wise

---

4. Payroll Processing Module
Purpose:
Automatically calculates monthly salaries and prepares payroll data.
Key Functions:

- Calculate gross salary, deductions, net salary

- Apply attendance-based calculations
- Calculate taxes and statutory deductions
- Handle overtime, shift allowances, and bonuses

---

5. Payslip Generation Module
Purpose:
Generates detailed employee salary slips for every payroll cycle.
Key Functions:
- Create printable/downloadable monthly payslips
- Include salary summary and deduction details
- Auto-generate payslip IDs
- Store payslips in the database for future reference

---

6. User Authentication & Authorization Module
Purpose:
Ensures secure access through user roles.
Key Functions:
- Login/logout system
- Role-based access (Admin, HR, Accountant, Employee)
- Encrypt passwords using hashing (e.g., BCrypt)
- Maintain session control and access privileges

---

7. Modular Encryption & Security Module
Purpose:
Protects sensitive payroll and employee information.
Key Functions:
- Encrypt bank details, salary data, and payslip files (AES-GCM)
- Hash passwords securely
- Manage encryption keys (Key Manager)
- Provide audit logs for encryption operations

CHAPTER 4

SAMPLE CODING

```
 package dao;


import model.Employee;

import java.sql.*;

import java.util.ArrayList;

import java.util.List;


public class EmployeeDAO {

    private Connection conn;


    public EmployeeDAO(Connection conn) {

        this.conn = conn;

    }


    public void addEmployee(Employee emp) throws SQLException {

        String sql = "INSERT INTO Employee (name, department, designation, salary) VALUES (?, ?, ?,
?)";

        PreparedStatement stmt = conn.prepareStatement(sql);

        stmt.setString(1, emp.getName());

        stmt.setString(2, emp.getDepartment());

        stmt.setString(3, emp.getDesignation());

        stmt.setDouble(4, emp.getSalary());
```

```
        stmt.executeUpdate();
    }

    public void updateEmployee(Employee emp) throws SQLException {
        String sql = "UPDATE Employee SET name=?, department=?, designation=?, salary=? WHERE
id=?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(1, emp.getName());
        stmt.setString(2, emp.getDepartment());
        stmt.setString(3, emp.getDesignation());
        stmt.setDouble(4, emp.getSalary());
        stmt.setInt(5, emp.getId());
        stmt.executeUpdate();
    }

    public void deleteEmployee(int id) throws SQLException {
        String sql = "DELETE FROM Employee WHERE id=?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setInt(1, id);
        stmt.executeUpdate();
    }

    public List<Employee> getAllEmployees() throws SQLException {
        List<Employee> list = new ArrayList<>();
```

```
    String sql = "SELECT * FROM Employee";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        Employee emp = new Employee(
                rs.getInt("id"),
                rs.getString("name"),
                rs.getString("department"),
                rs.getString("designation"),
                rs.getDouble("salary")
        );
        list.add(emp);
    }
    return list;
}

public boolean validateLogin(String username, String password) throws SQLException {
    String sql = "SELECT * FROM users WHERE username=? AND password=?";
    PreparedStatement stmt = conn.prepareStatement(sql);
    stmt.setString(1, username);
    stmt.setString(2, password);
    ResultSet rs = stmt.executeQuery();
    return rs.next(); // returns true if a match is found
}
```

```
}
package db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    public static Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/payroll";
        String user = "root"; // replace with your MySQL username
        String password = "Qwe!asd@zxc#poi$lkj%mnb"; // replace with your password
        return DriverManager.getConnection(url, user, password);
    }
}

public class LoginTester {
    public static void main(String[] args) {
        try (Connection conn = DBConnection.getConnection()) {
            String sql = "SELECT * FROM users WHERE username=? AND password=?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, "admin");
            stmt.setString(2, "admin123");
```

```
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                System.out.println("✅ Login successful!");
            } else {
                System.out.println("❌ Login failed.");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
package model;

public class Employee {
    private int id;
    private String name;
    private String department;
    private String designation;
    private double salary;

    public Employee(int id, String name, String department, String designation, double salary) {
```

```
        this.id = id;
        this.name = name;
        this.department = department;
        this.designation = designation;
        this.salary = salary;
    }

    public int getId() { return id; }
    public String getName() { return name; }
    public String getDepartment() { return department; }
    public String getDesignation() { return designation; }
    public double getSalary() { return salary; }

    public double getNetSalary() {
        double tax = salary * 0.1;
        double pf = salary * 0.05;
        return salary - tax - pf;
    }
}
```

CHAPTER 5

SCREEN SHOTS

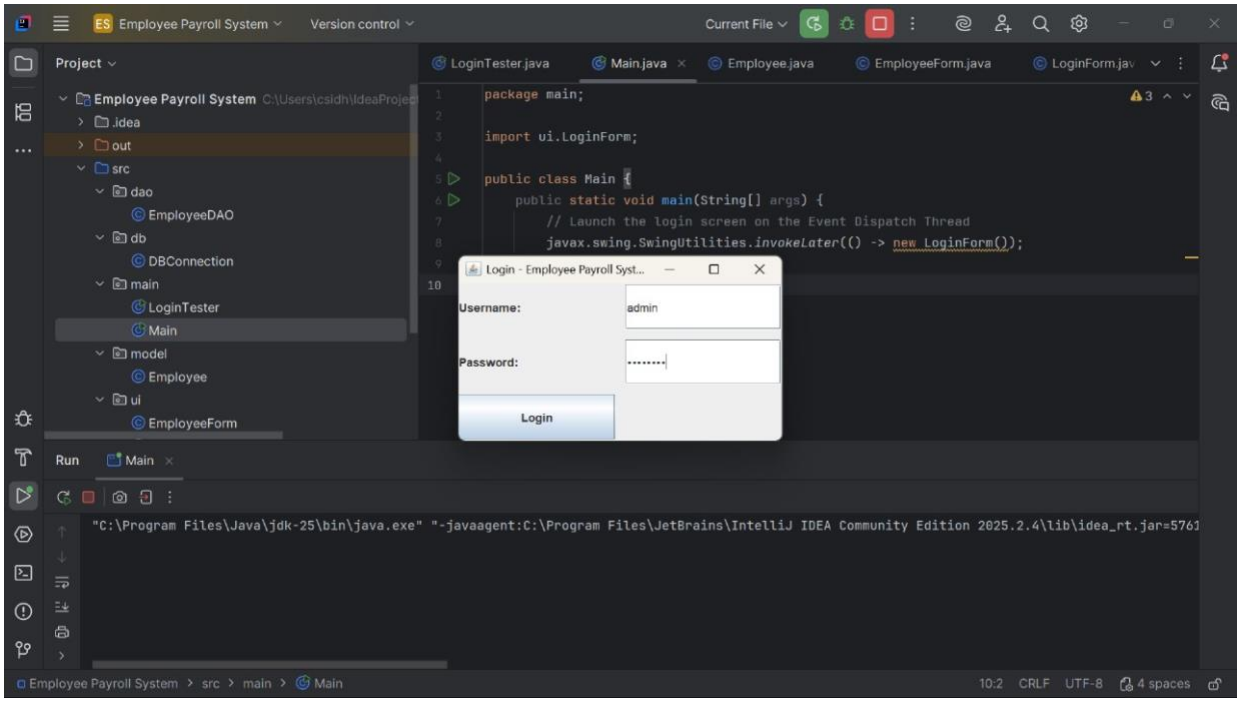

Fig 5.1 USER INTERFACE

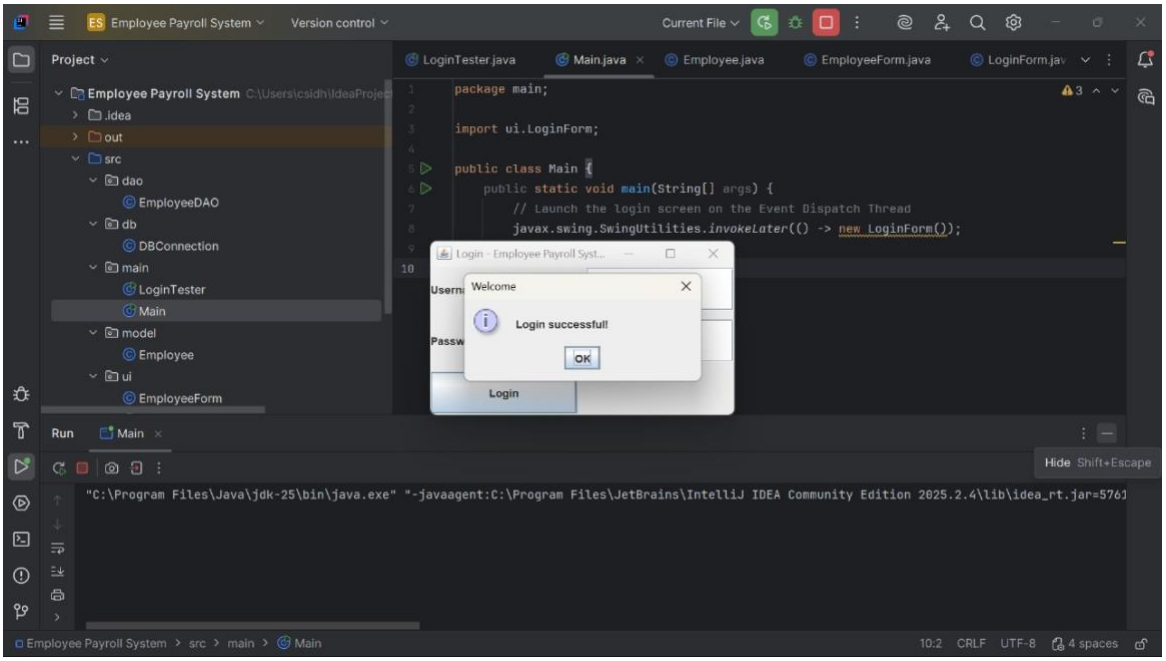

Fig 5.2 LOGIN PAGE

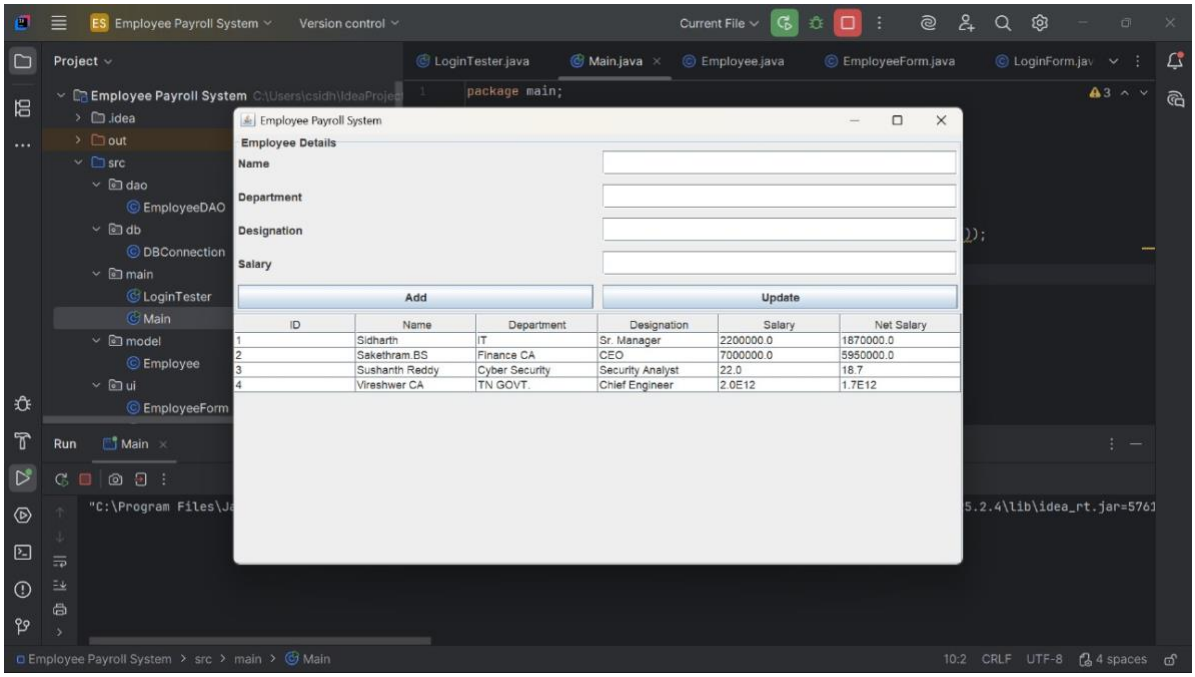

Fig 5.3 SALARY DETAILS

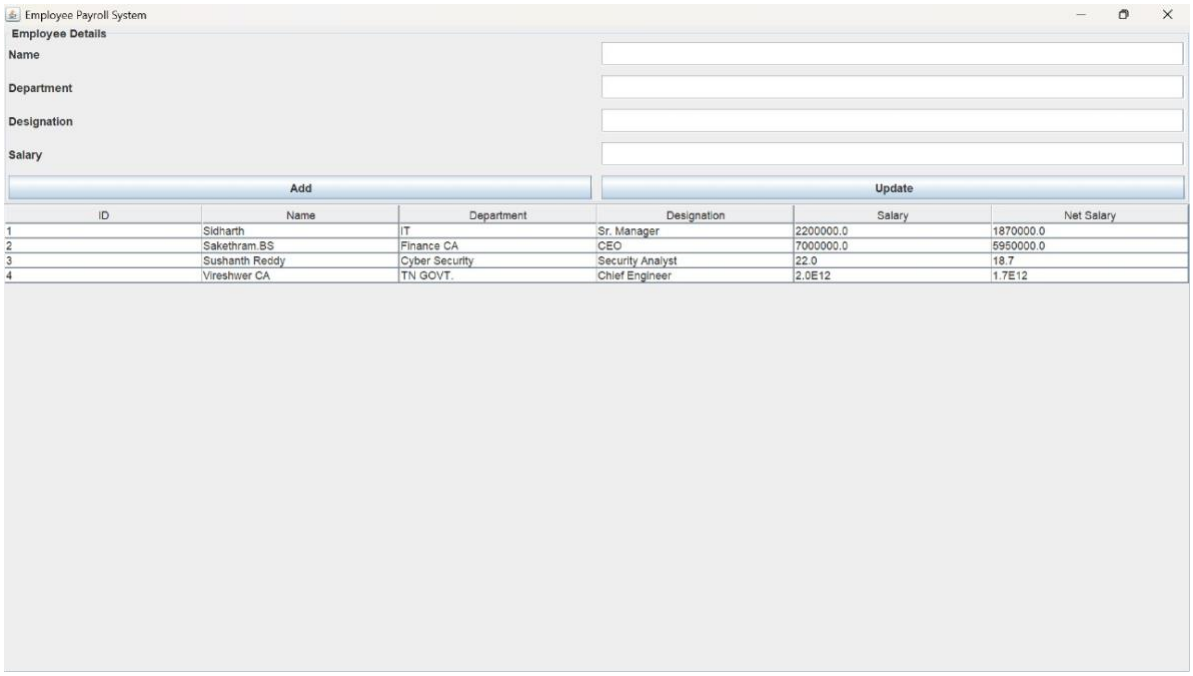

Fig 5.4 EMPLOYEE DETAILS

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The **Employee Payroll Management System** successfully automates and streamlines the payroll process by integrating employee management, attendance tracking, salary computation, and payslip generation into a unified platform. The system minimizes manual calculations, reduces human errors, and ensures timely and accurate salary disbursement.

By incorporating features such as role-based access control, encrypted data storage, and automated payroll calculations, the system enhances data security, operational efficiency, and reliability. It also provides a clear, organized way of maintaining employee records and generating various reports required for audits and management decision-making.

Overall, the project achieves its goal of providing a robust, user-friendly, and efficient solution that replaces traditional manual payroll processing with a modern automated system

6.2 Future Enhancements

Although the system fulfills the essential requirements, several improvements can be integrated in the future to enhance its functionality, scalability, and user convenience:

1. Integration with Biometric Attendance Devices

- Automatic attendance capture using fingerprint/RFID/face recognition machines.
- Reduces manual attendance entry and increases accuracy.

2. Cloud-Based Deployment

- Allow access from multiple branches or remote locations.
- Improve scalability, availability, and data synchronization.

3. Mobile Application for Employees

- Employees can view attendance, payslips, and apply for leave via a mobile app.
- Push notifications for payroll updates and leave approvals.

4. Online Salary Payment Integration

- Integrate with banking APIs for direct salary transfers.
- Automate payment confirmation and reconciliation logs.

5. AI-Based Salary Prediction & Analytics

- Predict salary expenses for future months.
- Identify abnormal patterns and send alerts for anomalies.

6. Advanced Security Enhancements

- Implement multi-factor authentication (MFA).
- **Use Hardware Security Modules (HSM) or KMS for key management.**
- **Apply end-to-end encryption for highly sensitive data.**

REFERENCES

1. https://www.w3schools.com/sql/
2. https://www.tutorialspoint.com/sqlite/index.htm
3. https://www.wikipedia.org/
4. https://www.learnpython.org/
5. https://www.codecademy.com/learn/learn-python