**Employee Management System**

**Table of Contents:**

* Introduction
* System Requirements
* System Architecture
* Features
* Technical Specifications
* Database Design
* User Interface
* Installation and Setup
* Future Enhancements
* Execution Screenshots
* Conclusion

**1. Introduction**

The Employee Management System (EMS) is a comprehensive desktop application designed to manage employee data efficiently. The system provides functionalities for adding, viewing, updating, and deleting employee records along with user authentication. This document provides an in-depth overview of the EMS, detailing its features, architecture, and technical specifications.

**2. System Requirements**

### **Hardware Requirements**

* Standard PC or laptop

### **Software Requirements**

* Java Development Kit (JDK) 8 or higher
* Eclipse IDE
* MySQL Database
* JDBC Driver for MySQL

**3. System Architecture**

### **Overview**

The Employee Management System follows a desktop application architecture with Swing for the user interface and JDBC for database connectivity.

### **Components**

* Frontend: Java Swing
* Backend: JDBC for database interaction
* Database: MySQL

### **Data Flow**

* User interacts with the Swing frontend.
* Requests are processed by the Java backend.
* Backend communicates with MySQL database for CRUD operations.
* The frontend updates the User Interface based on the response from the backend.

## **4. Features**

## **Employee Management**

* Add, view, update, delete employee records

**Authentication** Login with username and password.

**5. Technical Specifications**

### **Frontend (Java Swing)**

* Swing components for UI design
* Event-driven programming model

### **Backend (Java JDBC)**

* JDBC API for database connectivity
* Prepared statements for SQL queries

### **Database (MySQL)**

* **Tables:**
  + users: username, password
  + employees: id (Primary Key), name, email, position, department, salary, date\_hired

## **6. Database Design**

### **Tables**

* **Users**
  + username (Primary Key)
  + Password
* **Employees**
  + id (Primary Key)
  + name
  + email
  + position
  + department
  + salary
  + Date\_hired

**7. User Interface**

### **Screens**

* Login Frame: Fields for username and password, login button
* Main Dashboard: Buttons for adding, removing, viewing/updating employees, and generating reports.
* Add Employee Frame: Fields for employee details and add button.
* Remove Employee Frame: Field for employee ID and remove button.
* Report Frame: Chart displaying employee count by department.

## **8. Installation and Setup**

### **Steps**

* Download and install Eclipse IDE.
* Clone the Java project repository.
* Import the project into Eclipse.
* Set up MySQL database and import the schema.
* Update database credentials in the Java code.
* Compile and run the LoginFrame.java file in Eclipse.

## **9. Future Enhancements**

* Implement encrypted storage for passwords.
* Add search functionality for employee records.
* Enhance the UI with more detailed employee data views.
* Integrate role-based access control.

## **10. Execution Screenshots**

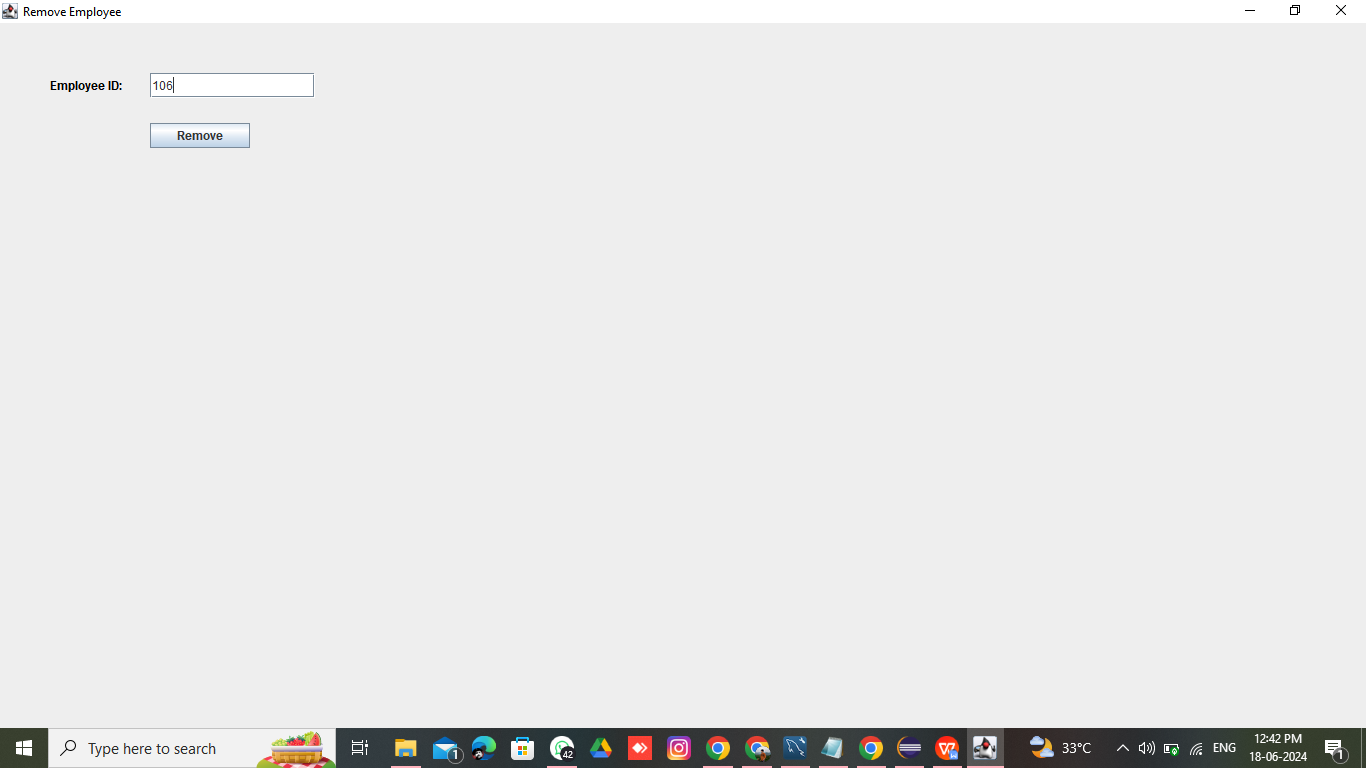
## **User Authentication**

## 

## **Main DashBoard**

## **Add Employee**

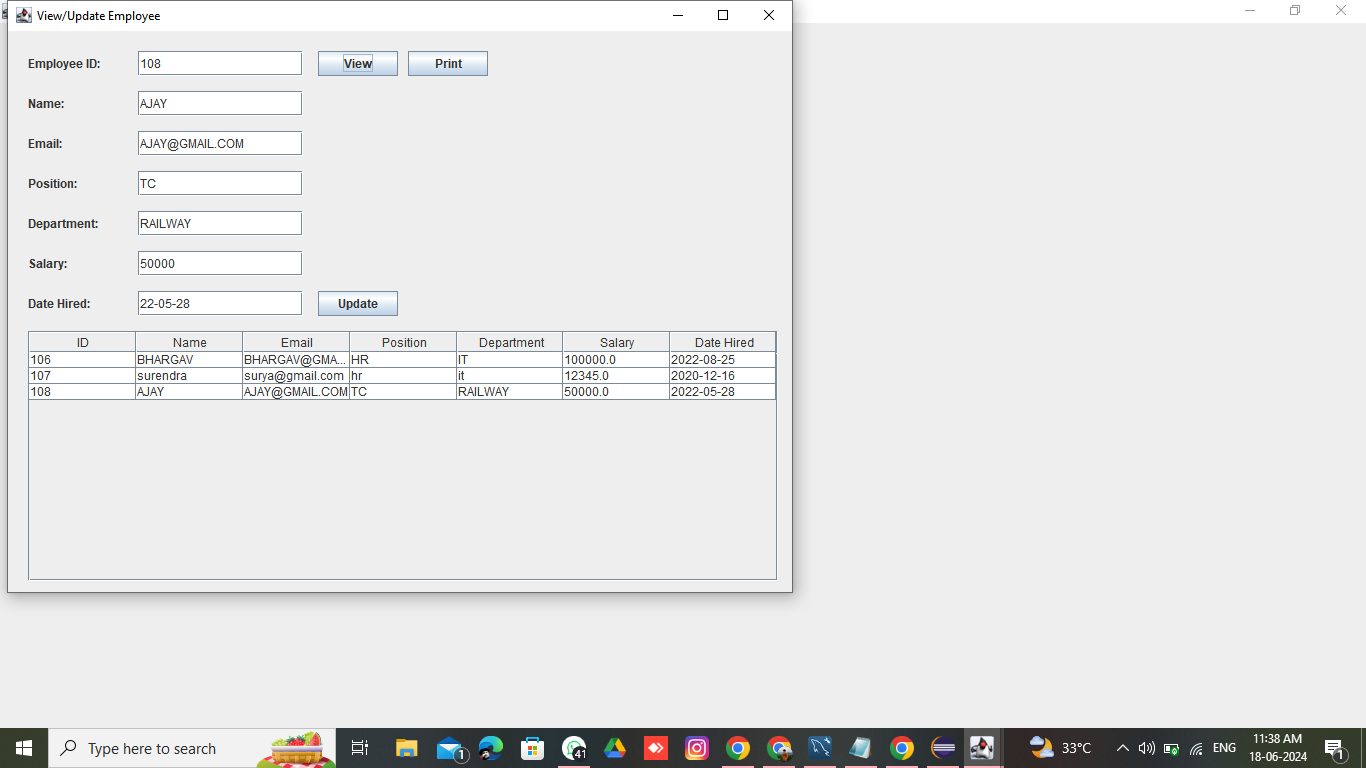
**Remove Employee**

****

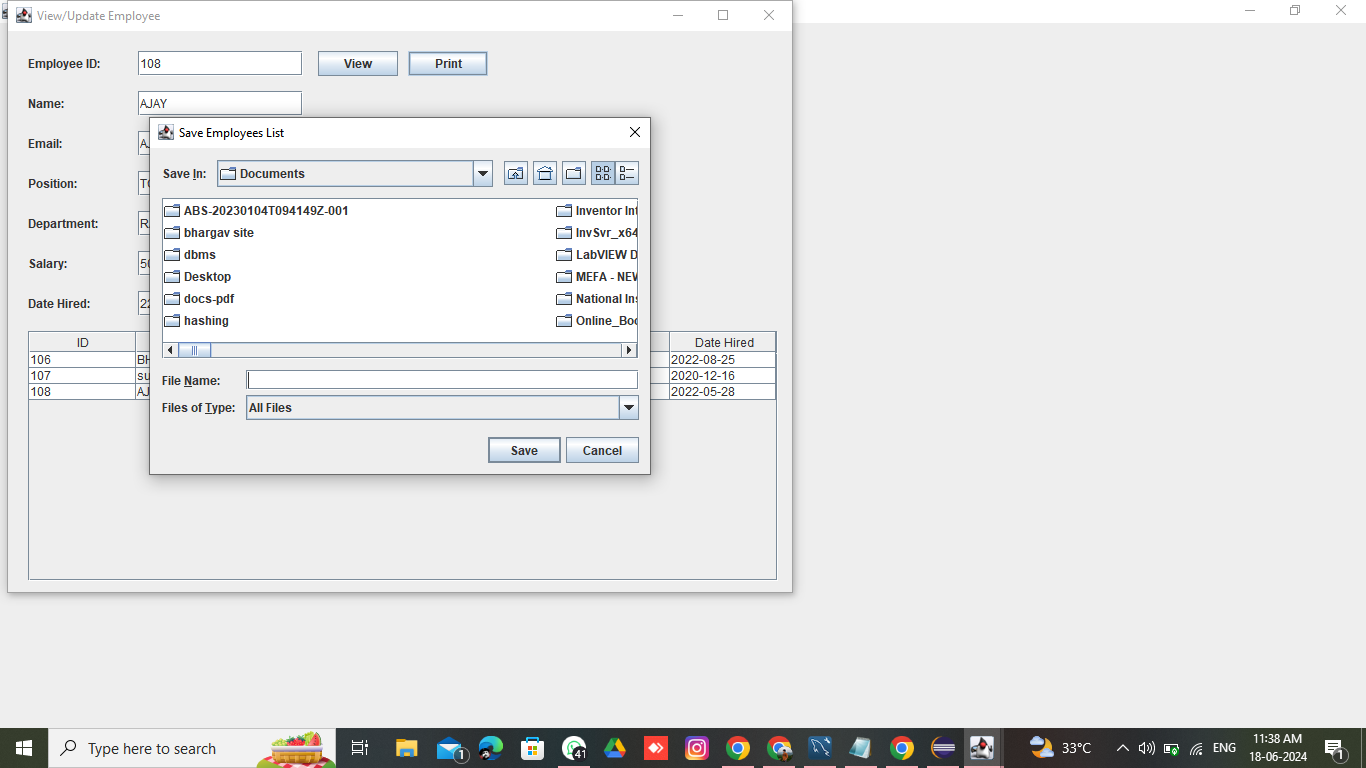
## **View Employee**

## 

## **Update Employee**



**Saving Files In Different Formats**

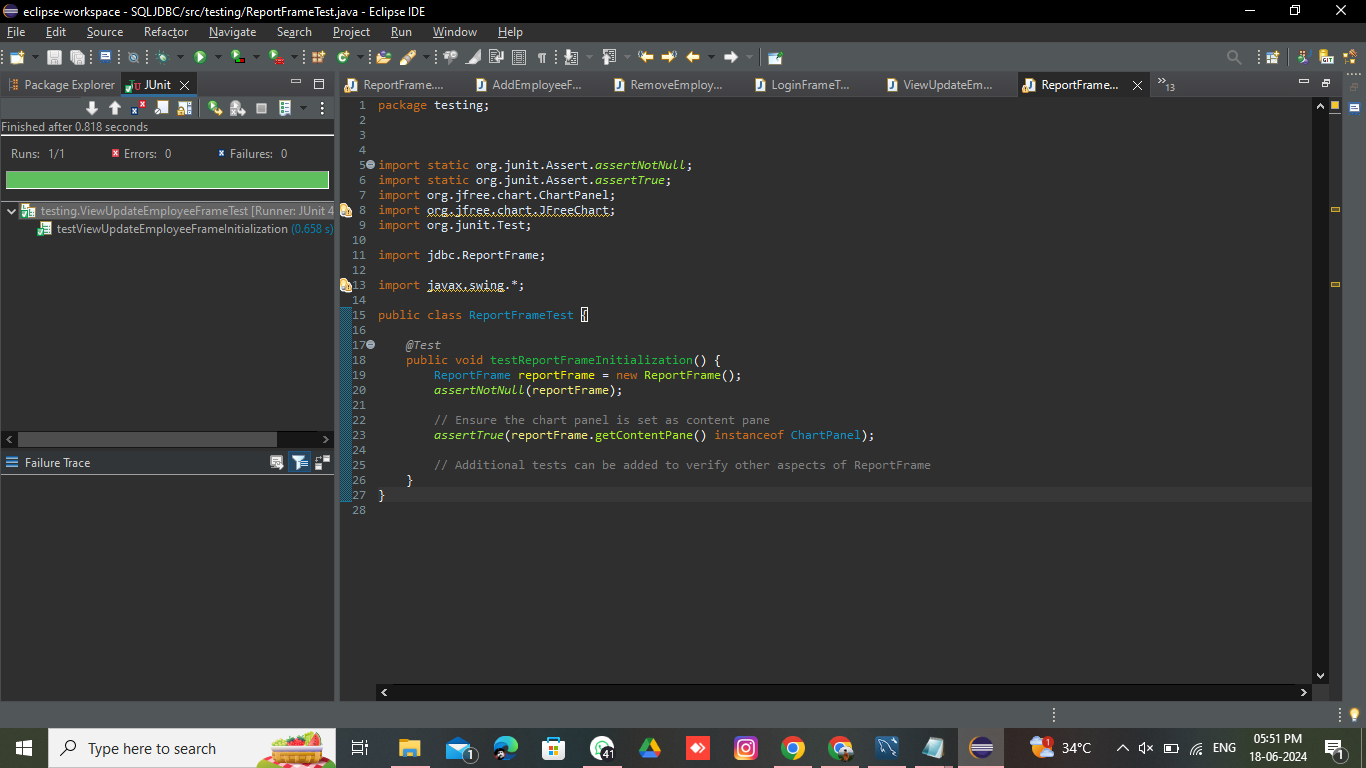
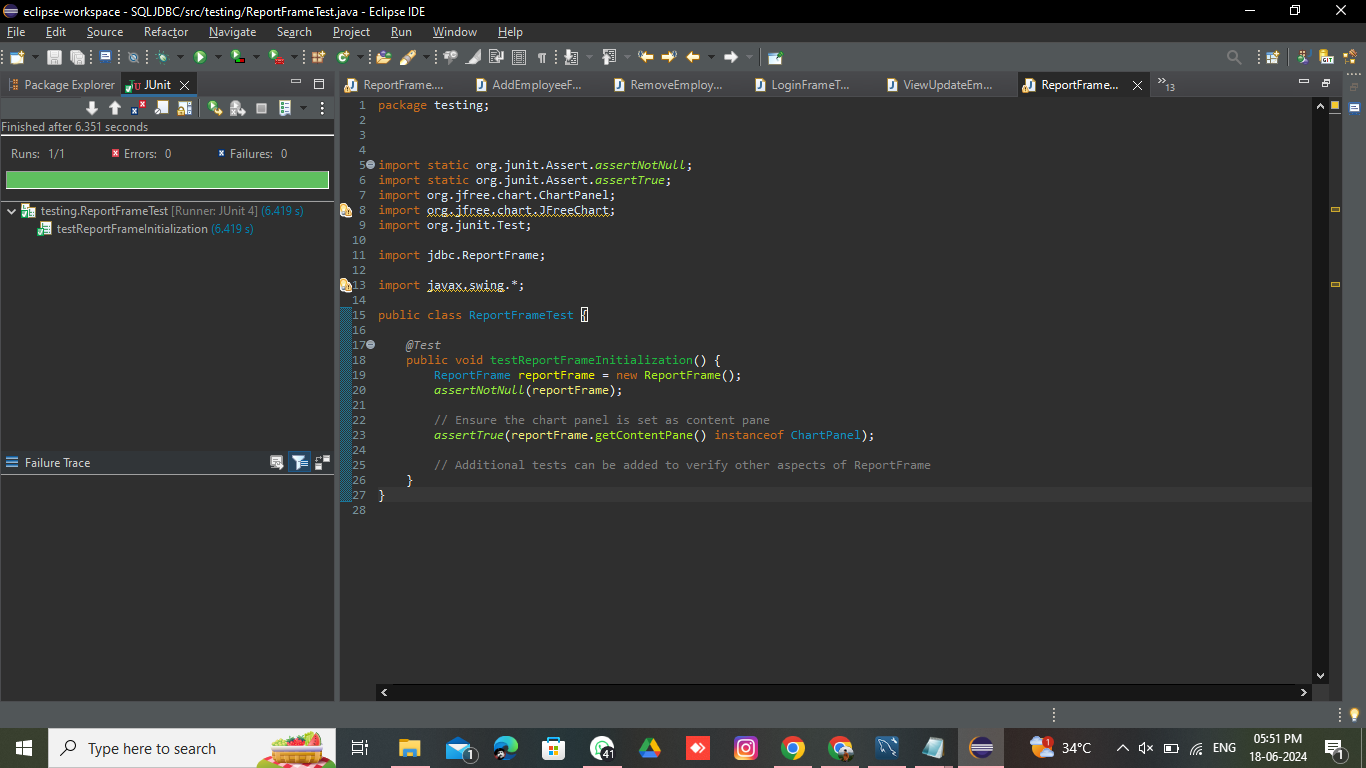
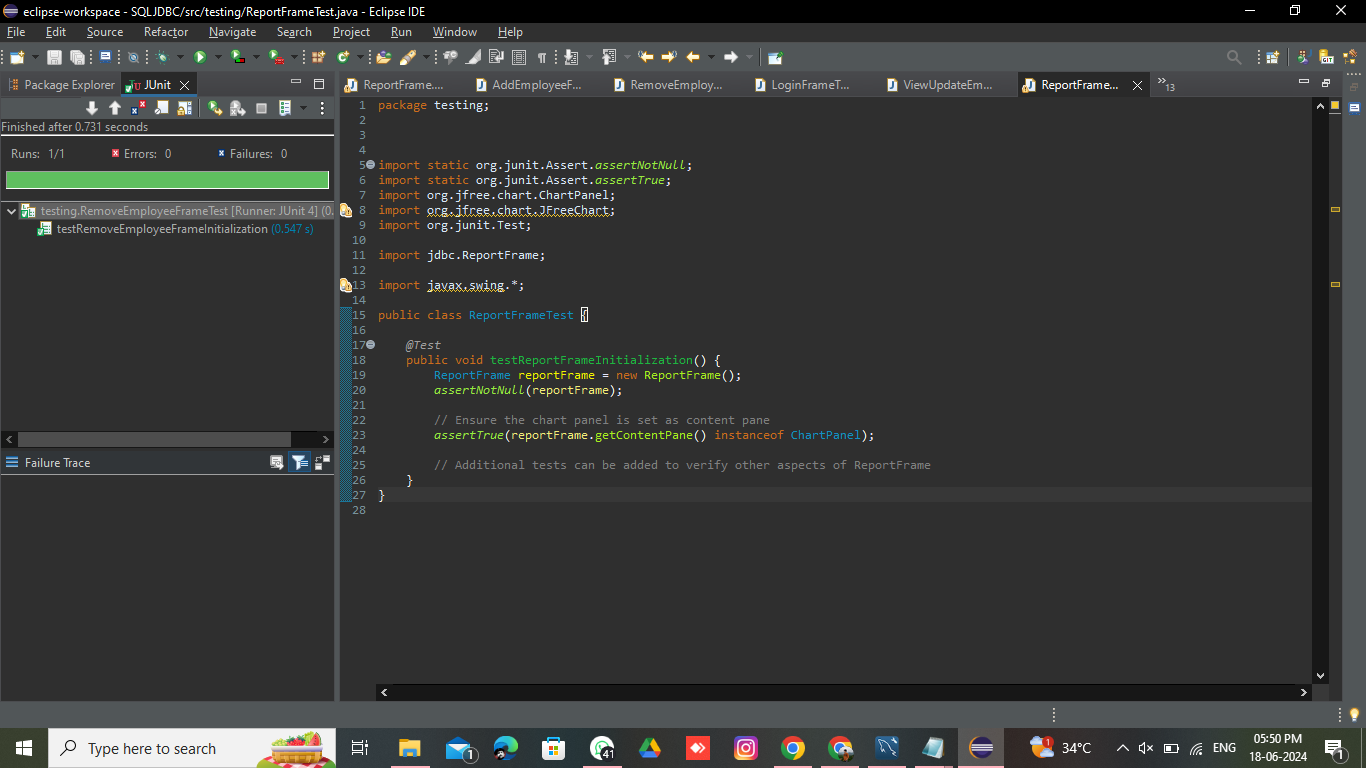
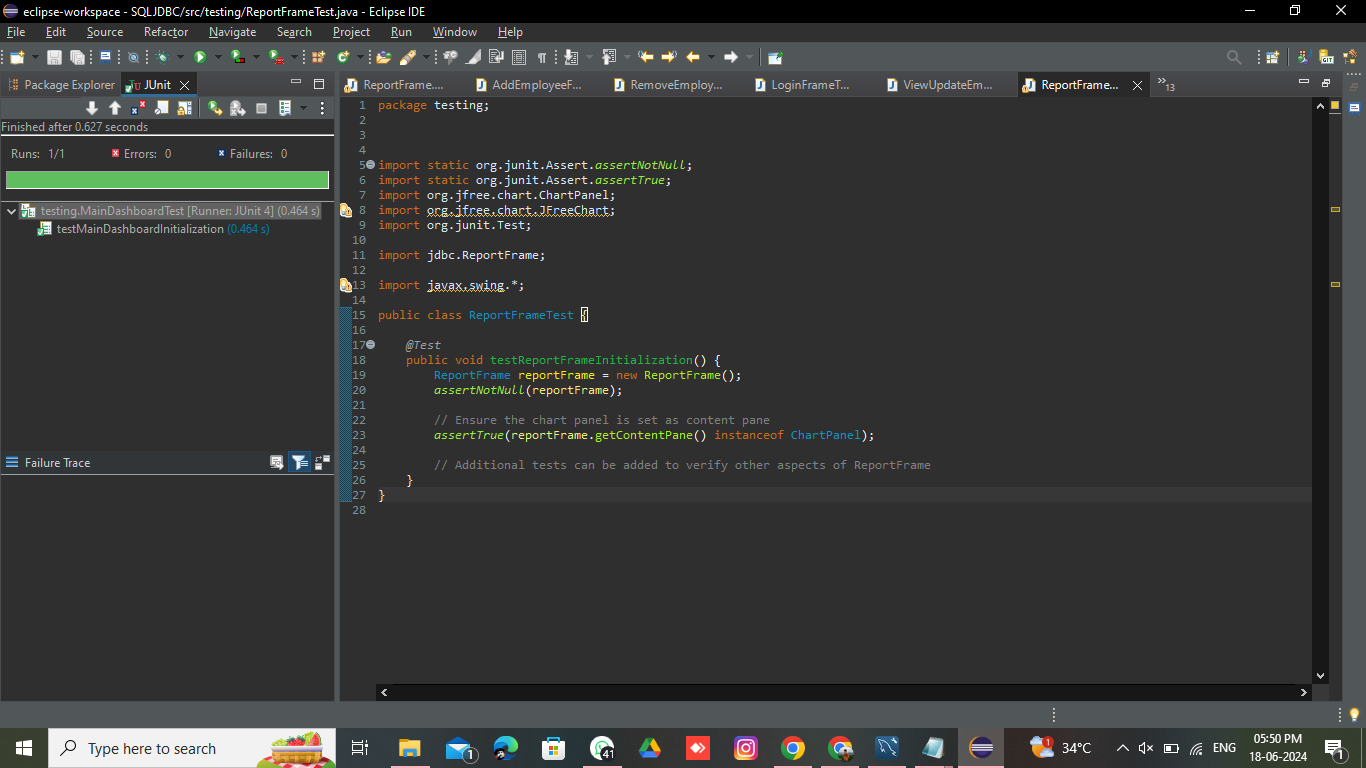
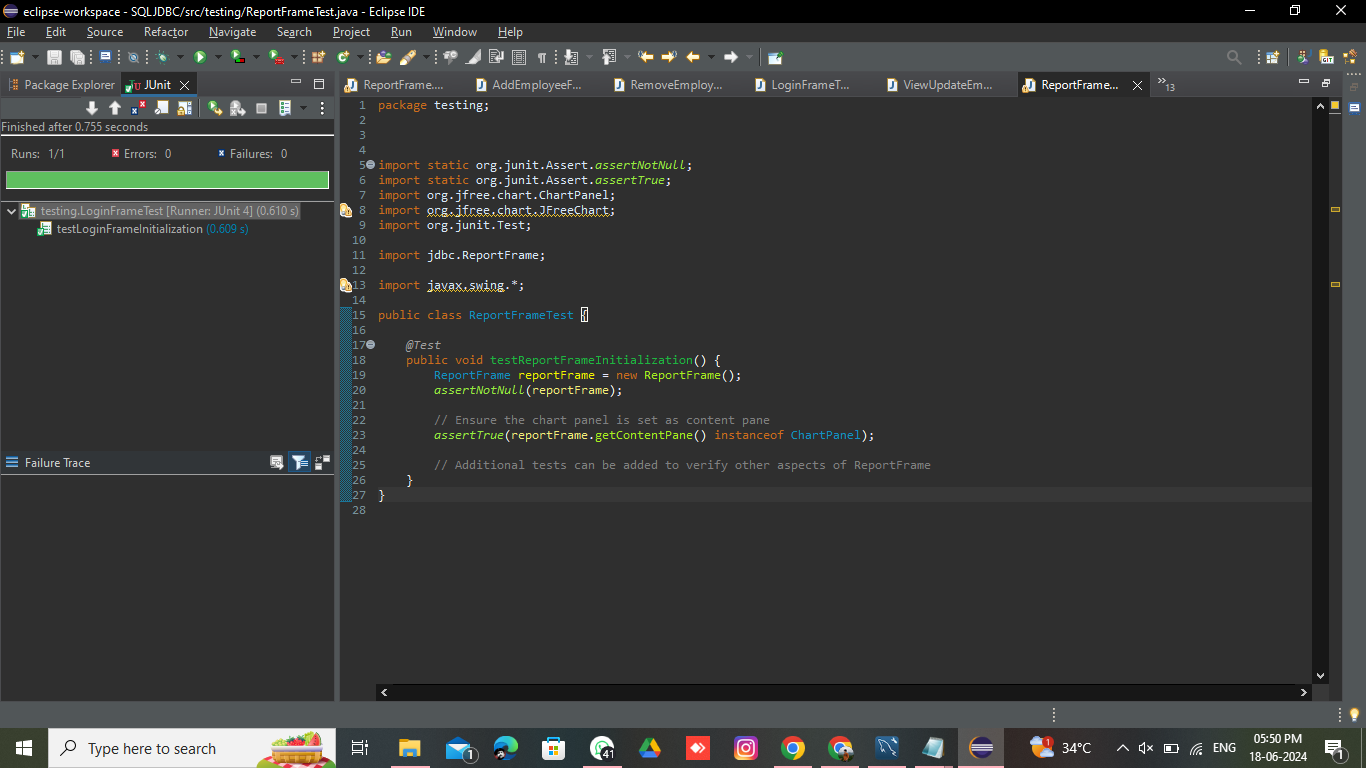
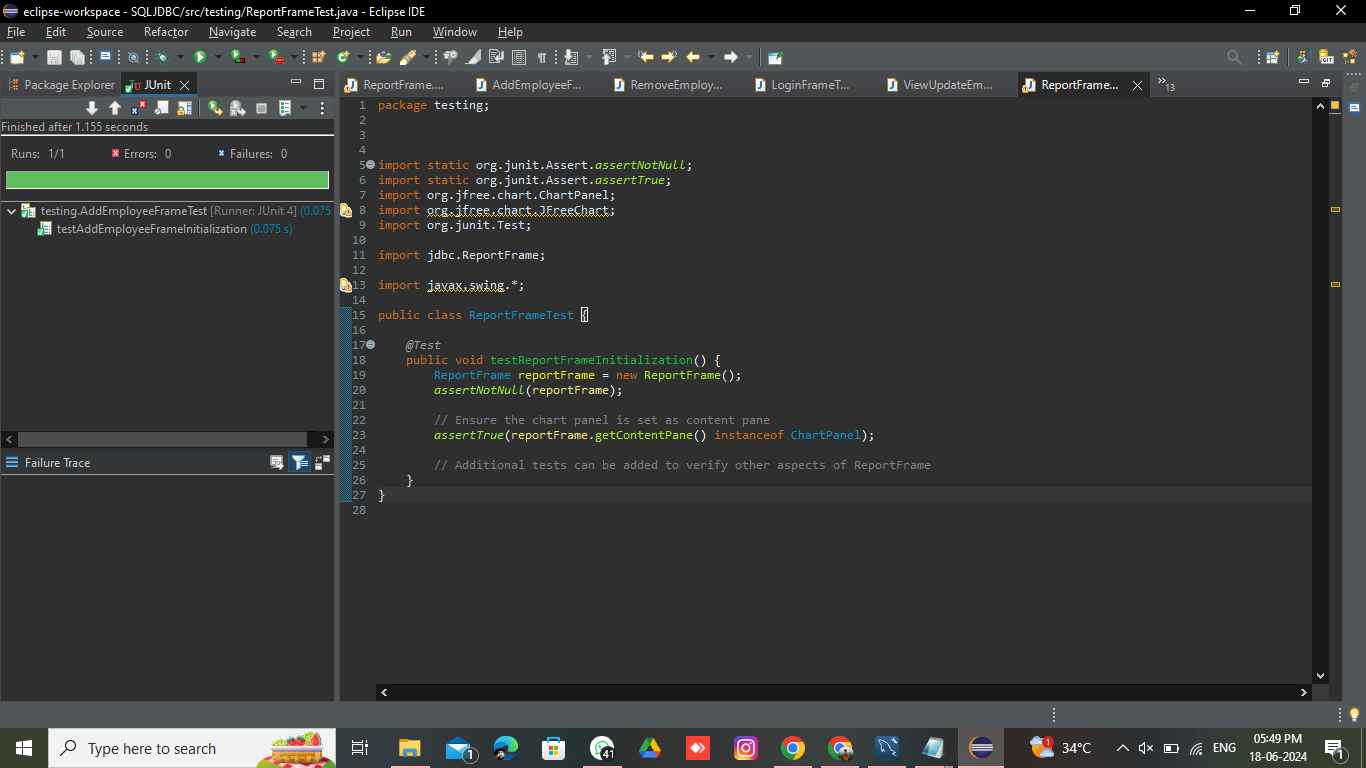
****

## 

## **Generate Report**

## 

## **11. Testing**



## **12. Conclusion**

**The Employee Management System provides a robust solution for managing employee data using Java Swing and JDBC with MySQL. It offers essential functionalities for employee management and user authentication. Future enhancements aim to further improve security and functionality.**