Student Info:

Student Name: Nargis Sultana Reshma

Student ID: 24070173

Department Name: B.Sc in CSE

Course Name : Object-Oriented Programming

Batch & Section: 43 (B)

Project Proposal: Security & Logging System

1. Project Title

Security & Logging System

2. Purpose and Problem Statement

The purpose of this project is to develop a secure and efficient Security & Logging System using Java that helps track user access and activity within a specific environment (e.g., office, parking area). This system aims to address the issue of unauthorized access and lack of entry/exit tracking by providing authentication, logging, and report generation.Using Object-Oriented Programming (OOP) concepts such as encapsulation, inheritance, polymorphism, and abstraction makes the project scalable, modular, and easier to maintain.

3. Main Goals and Key Functionalities

- User Authentication & Access Control- Logging Entry & Exit Times- Admin Dashboard- User Parking History- Security & Notifications- Report Generation

4. Tools and Technologies

• Programming Language: Java• Frameworks/Libraries: Swing (for GUI), Java IO, File Handling• Database: Text File (or optionally SQLite for persistence)

5. Use of OOP Concepts

- Encapsulation: All sensitive data will be hidden and accessed via getters and setters.- Inheritance: Different roles such as Admin and User will inherit from a common base class.- Polymorphism: Method overloading and overriding for flexible logging and reporting.- Abstraction: Abstract classes/interfaces to define core features like authentication and logging.

6. Project Phases & Timelines

- Phase 1: Requirements Analysis & Design (2 days)- Phase 2: Core Feature Implementation - Authentication & Logging (4 days)- Phase 3: GUI & Admin Panel Development (3 days)- Phase 4: Testing & Debugging (2 days)- Phase 5: Documentation & Submission (1 day)

7. Final Product Outcome

The final product will be a fully functioning Security & Logging System that provides efficient access control and monitoring. It will benefit users by increasing security and generating useful data for analysis.

8. Summary and Impact

This project demonstrates the practical application of Java and OOP principles in building a real-world system. It enhances learning and provides a meaningful solution to a common security concern.

9. References

- https://www.w3schools.com/java/- Java Documentation: https://docs.oracle.com/javase/8/docs/

10. Development So Far

• Completed:- Basic class structure- User Authentication (with dummy data)- Entry Logging feature (text file-based)• Screenshots: (to be attached manually by student)• To Do:- GUI implementation- Admin Dashboard- Parking history tracking- Notifications and report generation• Challenges:- GUI alignment issues- Handling file reading/writing correctly- Designing role-based access flow

Code :

```
// Main.java

import java.util.*;

// Class to represent a User

class User {

    String username;

    String password;

    List<String> parkingHistory = new ArrayList<>();

    User(String username, String password) {

        this.username = username;

        this.password = password;

    }

}

// Class to represent a log entry (entry/exit time)

class LogEntry {

    String username;

    Date entryTime;

    Date exitTime;

    LogEntry(String username) {

        this.username = username;

        this.entryTime = new Date();
```

```
    }
    void logExit() {
        this.exitTime = new Date();
    }
    @Override
    public String toString() {
        return "User: " + username + ", Entry: " + entryTime + ", Exit: " + (exitTime != null ? exitTime :
"Still Inside");
    }
}
// Main system class
class SecurityLoggingSystem {
    Map<String, User> users = new HashMap<>();
    Map<String, LogEntry> activeLogs = new HashMap<>();
    List<LogEntry> logHistory = new ArrayList<>();
    String adminPassword = "admin123";
    void registerUser(String username, String password) {
        if (!users.containsKey(username)) {
            users.put(username, new User(username, password));
            System.out.println("User registered successfully.");
        } else {
            System.out.println("Username already exists.");
        }
    }
    boolean authenticate(String username, String password) {
        User user = users.get(username);
        return user != null && user.password.equals(password);
    }
    void logEntry(String username) {
        if (!activeLogs.containsKey(username)) {
            LogEntry entry = new LogEntry(username);
```

```
            activeLogs.put(username, entry);
            users.get(username).parkingHistory.add("Entered at: " + entry.entryTime);
            System.out.println("Entry logged.");
        } else {
            System.out.println("User already inside.");
        }
    }
    void logExit(String username) {
        LogEntry entry = activeLogs.get(username);
        if (entry != null) {
            entry.logExit();
            logHistory.add(entry);
            users.get(username).parkingHistory.add("Exited at: " + entry.exitTime);
            activeLogs.remove(username);
            System.out.println("Exit logged.");
        } else {
            System.out.println("No active entry found.");
        }
    }
    void viewAdminDashboard(String password) {
        if (password.equals(adminPassword)) {
            System.out.println("--- Admin Dashboard ---");
            System.out.println("Active Users:");
            for (String user : activeLogs.keySet()) {
                System.out.println(user);
            }
        } else {
            System.out.println("Invalid admin password.");
        }
    }
```

```
    void viewUserHistory(String username) {
        User user = users.get(username);
        if (user != null) {
            System.out.println("--- Parking History for " + username + " ---");
            for (String record : user.parkingHistory) {
                System.out.println(record);
            }
        } else {
            System.out.println("User not found.");
        }
    }
    void generateReport() {
        System.out.println("--- Full Entry/Exit Report ---");
        for (LogEntry log : logHistory) {
            System.out.println(log);
        }
    }
    void sendSecurityNotification(String username, String message) {
        System.out.println("Security Alert for " + username + ": " + message);
    }
}
// Main class with main() function
public class Main {
    public static void main(String[] args) {
        SecurityLoggingSystem system = new SecurityLoggingSystem();
        // Register users
        system.registerUser("john", "1234");
        system.registerUser("alice", "abcd");
        // Authenticate and log entry
        if (system.authenticate("john", "1234")) {
```

```
        system.logEntry("john");
    }
    if (system.authenticate("alice", "abcd")) {
        system.logEntry("alice");
    }
    // Log exit
    system.logExit("john");
    // View admin dashboard
    system.viewAdminDashboard("admin123");
    // View user history
    system.viewUserHistory("john");

    // Generate report
    system.generateReport();

    // Send security notification
    system.sendSecurityNotification("alice", "You forgot to log out!");
  }
}
```

Output of the Code :