



ATTENDANCE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

SATHIYASEELAN A (2303811724321099)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by
AICTE, New Delhi)

SAMAYAPURAM – 621 112

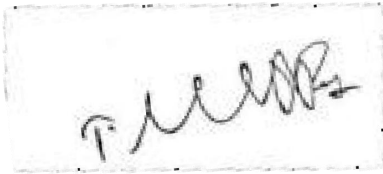
DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**ATTENDANCE MANAGEMENT SYSTEM**” is the bonafide work of **SATHIYASEELAN (2303811724321099)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.



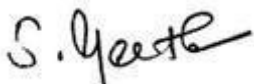
Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



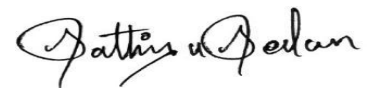
INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**ATTENDANCE MANAGEMENT SYSTEM** ” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVAPROGRAMMING**.



Signature

SATHIYASEELAN A

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Attendance Monitoring and Notification System is designed to streamline the management of student attendance in educational institutions. Using Java Swing and AWT, this system provides a user-friendly graphical interface for students and advisors to interact with the attendance data. Advisors can add, modify, and monitor attendance records, while students can securely view their individual records. The system also includes an automated notification feature that alerts advisors when students' attendance drops below a defined threshold. By utilizing core Java programming concepts such as object-oriented programming, event handling, and collections, the system ensures scalability, accuracy, and ease of use. This project aims to reduce inefficiencies associated with traditional attendance systems and provide a reliable solution for modern educational needs.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	3
3	JAVA PROGRAMMING CONCEPTS	4
	3.1 SWING AND AWT	4
	3.2 CLASSES AND OBJECTS	4
	3.3 COLLECTIONS (ARRAYLIST)	4
	3.4 EVENT HANDLING	5
	3.5 JOPTIONPANE	5
	3.6 CONTROL STRUCTURES:	5
4	MODULE DESCRIPTION	6
	4.1 STUDENT ID	6
	4.2 ADVISOR MODULE	6
	4.3 ADD STUDENTS MODULE	6
	4.4 MODIFY ATTENDANCE MODULE	6
	4.5 SEND NOTIFICATIONS MODULE	7
	4.6 EXIT MODULE	7
5	CONCLUSION	8

	REFERENCES	9
	APPENDICES	10
	APPENDIX A – SOURCE CODE	10
	APPENDIX B – SCREEN SHOTS	15

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

- Attendance monitoring is a crucial aspect of ensuring discipline and accountability in educational institutions. With the increasing student population and the need for more effective management systems, manual attendance tracking methods often prove inefficient and prone to errors. This project, Attendance Monitoring System, aims to digitize and automate the process of tracking and analyzing student attendance using a user-friendly interface.
- The system is designed to assist both students and advisors by providing real-time updates on attendance records, identifying students with attendance below the predefined threshold (e.g., 75%), and sending timely notifications to both parties. It ensures transparency and helps advisors take proactive measures to address issues related to attendance.
- Built using Java and Swing, this project combines a simple graphical user interface (GUI) with logical backend functionalities to make attendance management seamless and efficient. The system is particularly useful for institutions aiming to improve student engagement and compliance with attendance policies. Through this project, the process of monitoring attendance becomes not only accurate but also easily accessible for all stakeholders.

1.2 OBJECTIVE

The primary objective of this project is to develop an Attendance Monitoring and Notification System that provides a comprehensive solution for managing student attendance in educational institutions. The system is designed to ensure secure and efficient handling of attendance records, enabling students to view their individual attendance details while granting advisors the ability to add, modify, and monitor attendance data. Additionally, the system incorporates a notification feature that alerts advisors when a student's attendance falls below a predefined threshold, facilitating timely interventions. By leveraging Java Swing and AWT for a user-friendly graphical interface, the project aims to replace traditional, error-prone manual attendance systems with an automated, scalable, and intuitive alternative that can be easily adapted to meet the needs of institutions of varying sizes.

CHAPTER 2

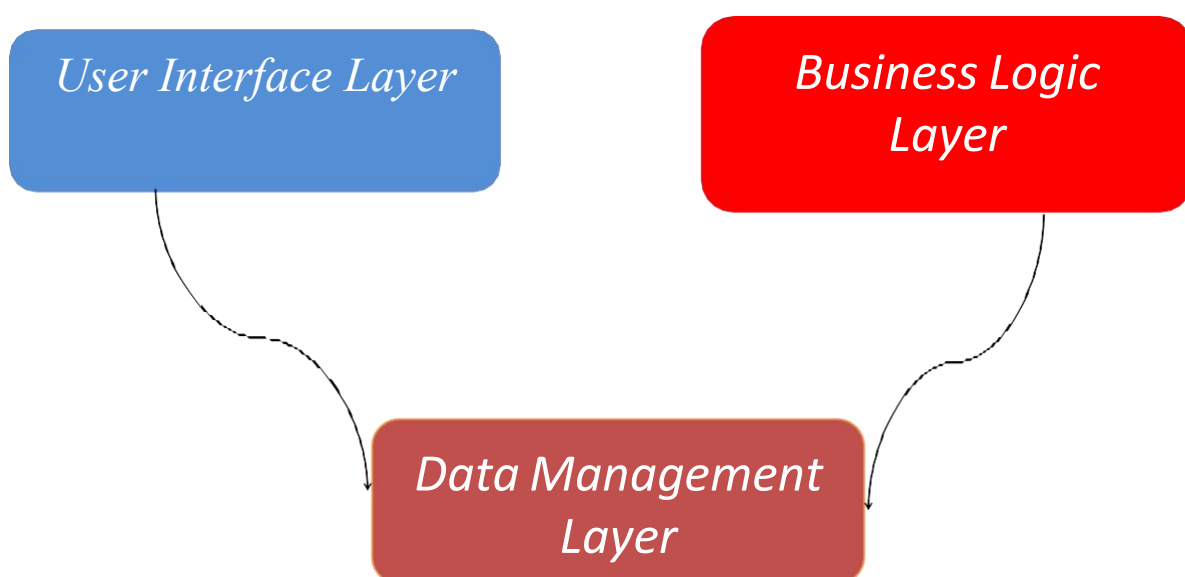
PROJECT METHODOLOGY

2.1 PROPOSED WORK:

The proposed system integrates a graphical user interface with back-end logic to manage attendance data effectively. Key features include:

- Secure student access for viewing attendance records.
- Advisor functionalities to manage data, modify attendance, and send notifications.
- Automated storage of attendance records for scalability.
- Notifications for low attendance thresholds.
- The system's architecture is designed to be extendable for future integrations such as biometric systems and advanced analytics.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 SWING AND AWT (ABSTRACT WINDOW TOOLKIT)

- Swing and AWT are Java libraries used for creating graphical user interfaces (GUIs).
- AWT provides basic GUI components like buttons, text fields, labels, etc., but with a limited set of features. It is platform-dependent and uses native resources of the operating system.
- Swing is built on top of AWT and provides a richer set of components like tables, trees, and text areas. It is platform-independent, meaning it looks the same across different platforms.
- Swing components are more flexible and customizable compared to AWT.

3.2 CLASSES AND OBJECTS

- Classes are blueprints for creating objects, which are instances of classes. A class defines the properties (attributes) and behaviors (methods) that the objects created from it will have.
- For example, a `Student` class can have attributes like `name`, `age`, and `grade`, and methods like `study()` or `attendClass()`. Objects are the specific instances of that class, such as a particular student like "John Doe".

3.3 COLLECTIONS (ARRAYLIST)

- Collections are data structures used to store and manage groups of objects in Java.
- `ArrayList` is a dynamic array, meaning it can grow or shrink in size as needed. Unlike regular arrays, which have a fixed size, an `ArrayList` allows adding and removing elements easily.

- It provides methods to manipulate elements, such as `add()`, `remove()`, and `get()`. It's widely used due to its flexibility and efficiency in managing data.

3.4 EVENT HANDLING

- Event handling in Java is the mechanism by which a program responds to user inputs, such as clicks, key presses, or mouse movements.
- In GUI programming, event listeners are used to detect specific actions. For example, a button click can trigger an event that calls a method to perform a task.
- `ActionListener` is commonly used for button clicks, and it reacts when the user performs a specific action.

3.5 JOPTIONPANE

- `JOptionPane` is a class in Java that simplifies the creation of dialog boxes for user interaction.
- It provides standard dialog boxes for tasks like showing messages (`showMessageDialog`), getting user input (`showInputDialog`), and confirming actions (`showConfirmDialog`).
- For example, it can pop up a message saying "Task completed!" or prompt the user to enter their name or age.

3.6 CONTROL STRUCTURES

- Control structures are constructs that allow for decision-making (conditional logic) and repetition (loops) in programming.
- Conditional logic: This includes `if`, `else if`, and `switch` statements, which allow the program to make decisions based on certain conditions (e.g., if a student's score is above a threshold, award a grade).
- Loops: These include `for`, `while`, and `do-while` loops, which repeat a block of code multiple times based on a condition (e.g., iterating through a list of students to print their names).

CHAPTER 4

MODULE DESCRIPTION

4.1 STUDENT ID

This module allows students to view their attendance details securely by entering their unique Student ID. The system verifies the ID and displays the corresponding attendance record. If the ID is not found, an error message is shown, ensuring data security and accuracy.

4.2 ADVISOR MODULE

Advisor module provides complete access to attendance records. Advisors can:

- View all student attendance records.
- Add new student details, including ID, name, and attendance percentage.
- Modify existing attendance records as needed.

4.3 ADD STUDENTS MODULE

This module enables advisors to add multiple student records at once. It prompts the user for the number of students to be added and collects their details (ID, name, attendance) through input dialogs. The records are stored dynamically in an ArrayList.

4.4 MODIFY ATTENDANCE MODULE

This module allows advisors to update attendance percentages for specific students. By entering the Student ID and the new attendance percentage, advisors can keep records accurate and up-to-date.

4.5 SEND NOTIFICATIONS MODULE

This module identifies students whose attendance falls below a predefined threshold set by the advisor. If no students fall below the threshold, the system displays a message stating, "No students are below the threshold." Notifications are logged in the output area for the advisor's reference.

4.6 EXIT MODULE

The exit module closes the application safely, ensuring all operations are completed before termination.

CHAPTER 5

CONCLUSION

The Attendance Monitoring and Notification System offers an efficient and user-friendly approach to managing attendance in educational institutions. By incorporating Java Swing and AWT, the system ensures a robust and scalable design that simplifies attendance tracking and notification processes. The system's modular architecture allows for future enhancements such as biometric integration, database connectivity, and advanced reporting. Overall, this project addresses the challenges of traditional attendance systems by providing a modern, automated solution.

REFERENCES:

1. Oracle Documentation on Java Swing and AWT: Comprehensive guide for building GUIs in Java.
2. Java Collections Framework: Official documentation on using collections like ArrayList.
3. Event Handling in Java: Tutorials on managing user interactions with GUI components.
4. Stack Overflow: Community-driven solutions and clarifications on Java programming.
5. W3Schools Java Tutorials: Easy-to-follow Java concepts and examples.

APPENDICES

APPENDIX A – SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;

class Student {
    String id;
    String name;
    double attendance;

    Student(String id, String name, double attendance)
    {
        this.id = id;
        this.name = name;
        this.attendance = attendance;
    }
}

public class AttendanceMonitoringSystemSwing
{
    private JFrame frame;
    private JTextArea outputArea;
    private ArrayList<Student> students;
    private static final double THRESHOLD = 75.0; // Predefined attendance
    threshold

    public AttendanceMonitoringSystemSwing()
    {
        students = new ArrayList<>();
    }
}
```

```

        setupUI();
    }

    private void setupUI() {
        frame = new JFrame("Attendance Monitoring System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(600, 400);

        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout());

        outputArea = new JTextArea();
        outputArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(outputArea);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(2, 3));

        JButton studentViewButton = new JButton("Student View");
        studentViewButton.addActionListener(e -> studentView());

        JButton advisorViewButton = new JButton("Advisor View");
        advisorViewButton.addActionListener(e -> advisorView());

        JButton addStudentButton = new JButton("Add Student (Advisor)");
        addStudentButton.addActionListener(e -> addNewStudents());

        JButton modifyAttendanceButton = new JButton("Modify Attendance
(Advisor)");

```

```

        modifyAttendanceButton.addActionListener(e -> modifyAttendance());

        JButton sendNotificationButton = new JButton("Send Notification");
        sendNotificationButton.addActionListener(e -> sendNotifications());

        JButton exitButton = new JButton("Exit");
        exitButton.addActionListener(e -> System.exit(0));

        buttonPanel.add(studentViewButton);
        buttonPanel.add(advisorViewButton);
        buttonPanel.add(addStudentButton);
        buttonPanel.add(modifyAttendanceButton);
        buttonPanel.add(sendNotificationButton);
        buttonPanel.add(exitButton);

        panel.add(scrollPane, BorderLayout.CENTER);
        panel.add(buttonPanel, BorderLayout.SOUTH);

        frame.add(panel);
        frame.setVisible(true);
    }

    private void studentView() {
        String studentId = JOptionPane.showInputDialog(frame, "Enter your
Student ID:");
        for (Student student : students) {
            if (student.id.equals(studentId))
                { outputArea.append("Student ID: " + student.id +
"\n");outputArea.append("Name: " + student.name +
"\n");

```

```

        outputArea.append("Attendance: " + student.attendance + "%\n");
        return;
    }
}

JOptionPane.showMessageDialog(frame, "Student ID not found!");
}

private void advisorView() {
    outputArea.setText("All Student Attendance Records:\n");
    for (Student student : students) {
        outputArea.append("ID: " + student.id + ", Name: " + student.name +
            ", Attendance: " + student.attendance + "%\n");
    }
}

private void addNewStudents() {
    String countStr = JOptionPane.showInputDialog(frame, "Enter the number
of students to add:");
    try {
        int count = Integer.parseInt(countStr);
        for (int i = 0; i < count; i++) {
            String id = JOptionPane.showInputDialog(frame, "Enter Student
ID:");
            String name = JOptionPane.showInputDialog(frame, "Enter
Student Name:");
            double attendance =
Double.parseDouble(JOptionPane.showInputDialog(frame, "Enter Attendance
Percentage:"));
            students.add(new Student(id, name, attendance));

```

```

    }
    JOptionPane.showMessageDialog(frame, "Student details stored
successfully.");
    } catch (Exception ex) {
        outputArea.append("Invalid input. Please try again.\n");
    }
}

private void modifyAttendance() {
    String studentId = JOptionPane.showInputDialog(frame, "Enter Student ID
to modify:");
    for (Student student : students) {
        if (student.id.equals(studentId))
            {double newAttendance =
Double.parseDouble(JOptionPane.showInputDialog(frame, "Enter new Attendance
Percentage:"));
            student.attendance = newAttendance;
            JOptionPane.showMessageDialog(frame, "Attendance updated
successfully.");
            return;
        }
    }
    JOptionPane.showMessageDialog(frame, "Student ID not found!");
}

private void sendNotifications()
{boolean found = false;
    outputArea.append("Sending notifications for students below " +
THRESHOLD + "% attendance:\n");

```

```

        for (Student student : students) {
            if (student.attendance < THRESHOLD)
                { outputArea.append("Notification sent to Student ID: " +
student.id + " (Name: " + student.name + ") - Attendance: " + student.attendance +
"%\n");

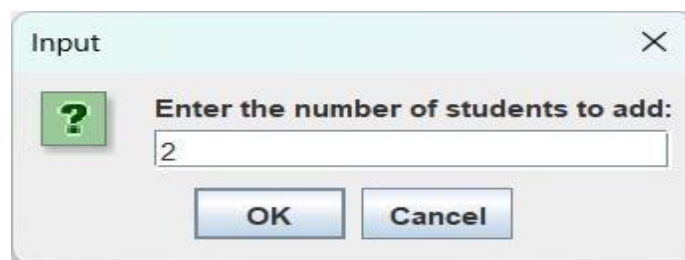
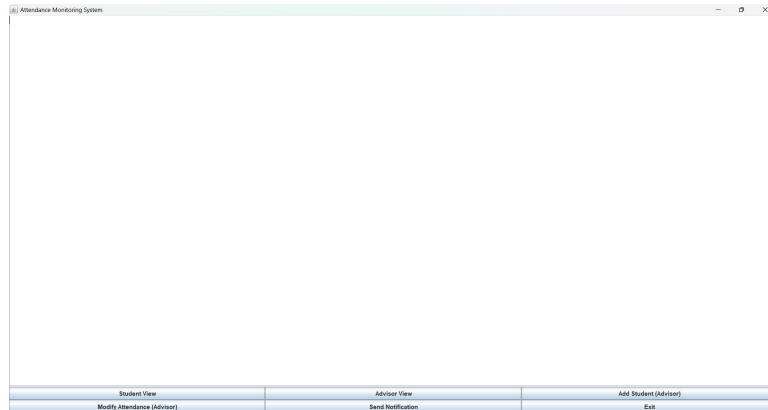
                    found = true;
                }
        }

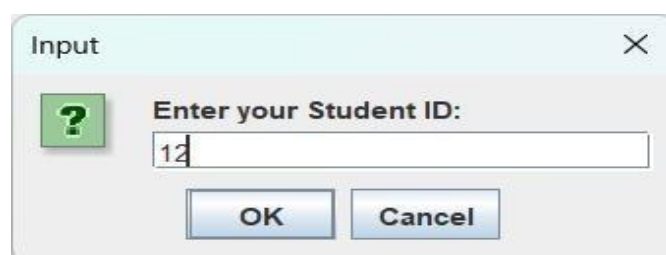
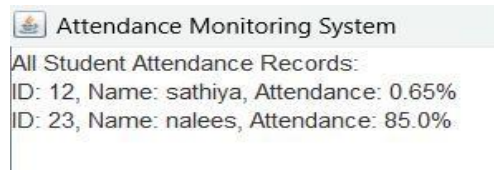
        if (found) {
            JOptionPane.showMessageDialog(frame, "Notifications successfully
sent to students and advisors!");
        } else {
            outputArea.append("No students are below the threshold.\n");
            JOptionPane.showMessageDialog(frame, "All students have
attendance above the threshold.");
        }
    }

    public static void main(String[] args)
    { SwingUtilities.invokeLater(AttendanceMonitoringSystemSwing::new);
    }
}

```

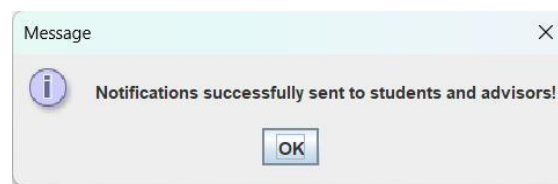

APPENDIX B - SCREENSHOTS





Student ID: 12
Name: sathiya
Attendance: 0.65%

Send Notification



Sending notifications for students below 75.0% attendance:
Notification sent to Student ID: 12 (Name: sathiya) - Attendance: 0.65%

Exit

