



FITNESS TRACKER APPLICATION



A PROJECT REPORT

Submitted by

SAKTHI VISHAL C (2302811724321095)

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 “**FITNESS TRACKER APPLICATION**” is the bonafide work of **SAKTHI VISHAL C(2302811724321095)** 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 Engineering,
Samayapuram, Trichy -621 112.



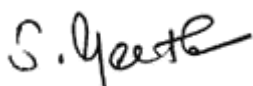
Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,
K. Ramakrishnan College of Engineering,
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 “ **FITNESS TRACKER APPLICATION**” 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 – JAVA PROGRAMMING**.



Signature

SAKTHI VISHAL C

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 acknowledgment and appreciation to the esteemed and honorable 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 Fitness Tracker Application is an intuitive mobile-based solution designed to monitor and enhance users' health and fitness goals effectively. The application features interactive dashboards for activity tracking, goal setting, and progress monitoring. Leveraging advanced algorithms and a user-friendly interface, the app provides personalized insights and seamless integration with wearable devices, ensuring an engaging and efficient fitness experience.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	1
2	PROJECT METHODOLOGY	2
	2.1 PROPOSED WORK	2
	2.2 BLOCK DIAGRAM	3
3	JAVA PROGRAMMING CONCEPTS	4
	3.1 SWING FRAMEWORK	4
	3.2 EVENT HANDLING	4
	3.3 ARRAYLIST	4
	3.4 JAVA AWT	4
	3.5 OBJECT-ORIENTED PROGRAMMING	4
4	MODULE DESCRIPTION	5
	4.1 NAVIGATION MODULE	5
	4.2 USER PROFILE MODULE	5
	4.3 ACTIVITY TRACKING MODULE	5
	4.4 GOAL MANAGEMENT MODULE	5
5	CONCLUSION	6
	REFERENCES	7
	APPENDICES	8
	Appendix A – Source code	8
	Appendix B – Screen shots	15

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

To develop a fitness tracker application that allows users to monitor their physical activities, set fitness goals, and track progress using an interactive and intuitive interface. The application aims to promote healthier lifestyles and provide real-time insights into users' fitness journeys.

1.2 OBJECTIVE

- **Activity Tracking:** Monitor physical activities such as steps, distance, calories burned, and workout sessions in real-time.
- **Goal Setting and Progress Monitoring:** Allow users to set fitness goals, track progress, and receive personalized insights to stay motivated.
- **User Profile Management:** Collect and store user details such as age, weight, and fitness preferences to provide tailored recommendations.

CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

The Fitness Tracker Application adopts a modular and scalable design to effectively monitor fitness activities and manage user data. The key components of the proposed work are as follows:

Enhanced User Interface:

Develop an intuitive GUI using Java Swing, ensuring smooth navigation for activity tracking, goal setting, and progress monitoring.

Real-Time Activity Tracking:

Utilize multithreading for seamless real-time updates of tracked data, including steps, calories, and workout duration, without performance delays.

Personalized Goal Management:

Provide features to set, edit, and track fitness goals.

Enable progress tracking with visual feedback through charts and graphs.

User Profile Management:

Maintain centralized user profiles that store details such as age, weight, and fitness preferences.

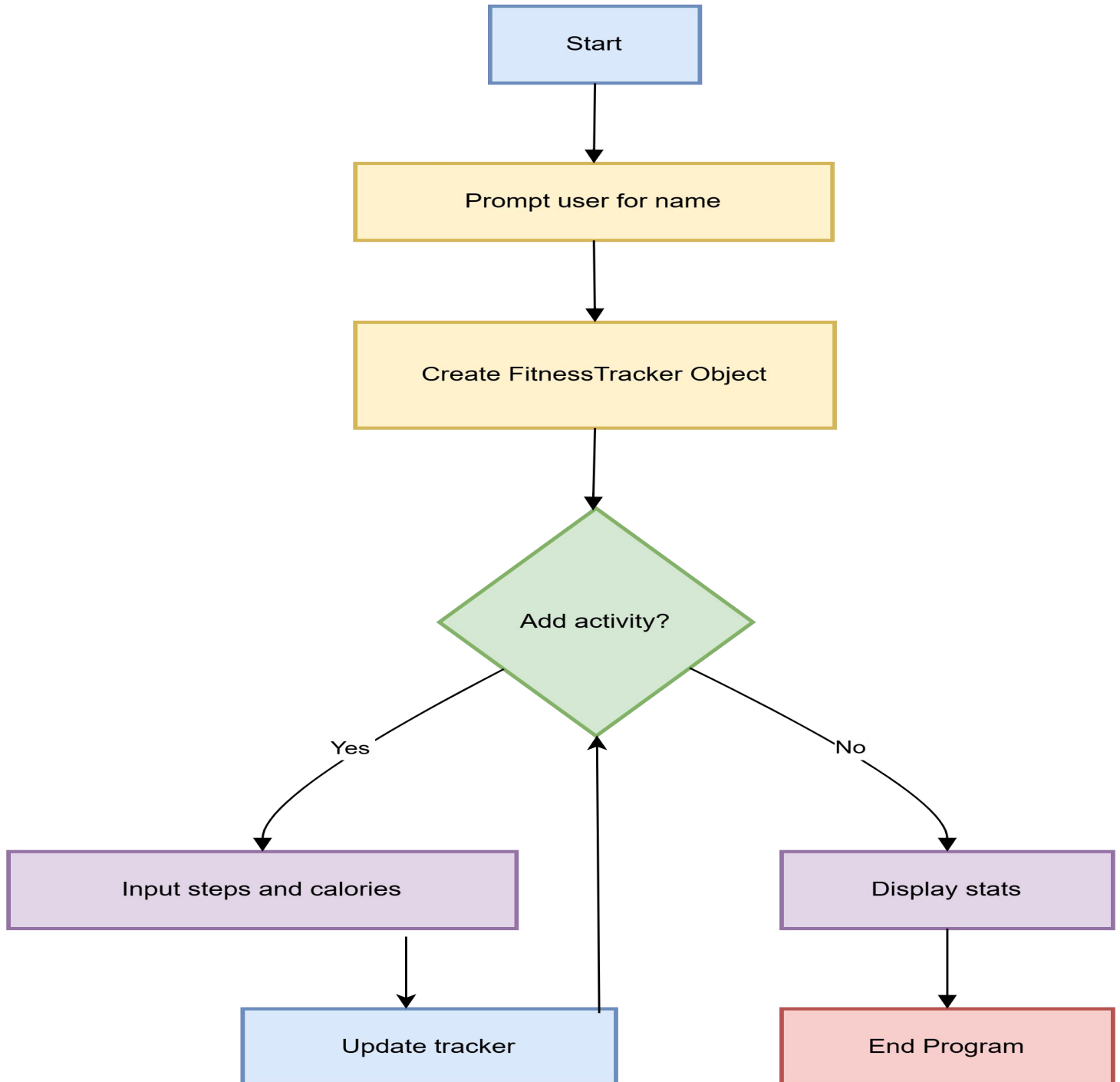
Ensure accurate data updates and personalized recommendations.

Scalability for Future Enhancements:

Plan for integration with external APIs for syncing data from wearable devices.

Enable cloud-based data storage and multi-user support for broader functionality.

2.2 BLOCKDIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 SWING FRAMEWORK

For designing the graphical user interface (GUI) of the fitness tracker.

3.2 EVENT HANDLING

To process user actions such as starting/stopping activity tracking or setting goals.

3.3 ARRAYLIST

To store and manage dynamic data like user activities, workout history, and goals.

3.4 JAVA AWT

For layout management and proper alignment of GUI components.

3.5 OBJECT-ORIENTED PROGRAMMING (OOP)

To structure the application into modular, reusable classes such as UserProfile, ActivityTracker, and GoalManager

CHAPTER 4

MODULE DESCRIPTION

4.1 NAVIGATION MODULE

Provides seamless switching between panels.

Includes buttons like "Activity Tracking," "Set Goals," and "Progress Reports."

4.2 USER PROFILE MODULE

Stores user data such as name, age, weight, and fitness preferences. Validates inputs before saving.

4.3 ACTIVITY TRACKING MODULE

Tracks activities like steps, distance, calories burned, and workout duration. Displays real-time updates with start/stop functionality.

4.4 GOAL MANAGEMENT MODULE

Allows users to set fitness goals with visual progress indicators. Dynamically updates goal status based on tracked activities.

CHAPTER 5

CONCLUSION

The Fitness Tracker Application demonstrates the efficient use of Java for building a fully functional and user-friendly application. It provides a scalable solution for activity tracking and goal management, highlighting the real-world applicability of core programming concepts. Future enhancements could include wearable device integration, cloud-based data storage, and multi-user support.

REFERENCES

Books:

- Herbert Schildt, *"Java: The Complete Reference,"* McGraw-Hill.
- Cay S. Horstmann, *"Core Java Volume I - Fundamentals,"* Prentice Hall.

Websites:

- Oracle Java Documentation
- GeeksforGeeks - Java Tutorials

YouTube Channels:

- Telusko (Java Tutorials): [YouTube Link](#)
- CodeWithHarry: [YouTube Link](#)

APPENDICES

APPENDIX A – SOURCE CODE

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

public class FitnessTrackerApp {

    private int steps;
    private double caloriesBurned;
    private double distanceCovered;

    // GUI Components
    private JFrame frame;
    private JTextField stepsInput;
    private JLabel stepsLabel, distanceLabel, caloriesLabel;
    private JProgressBar stepProgressBar;
    private static final int GOAL_STEPS = 10000; // Default step goal

    // Constructor
    public FitnessTrackerApp() {
        steps = 0;
        caloriesBurned = 0.0;
        distanceCovered = 0.0;

        // Initialize the GUI
        initializeGUI();
    }
}
```

```

private void initializeGUI() {
    // Frame setup
    frame = new JFrame("Fitness Tracker");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 400);
    frame.setLayout(new GridLayout(8, 1, 10, 10));

    // Input Panel
    JPanel inputPanel = new JPanel();
    inputPanel.setLayout(new FlowLayout());
    JLabel inputLabel = new JLabel("Enter Steps: ");
    stepsInput = new JTextField(10);
    JButton logStepsButton = new JButton("Log Steps");
    inputPanel.add(inputLabel);
    inputPanel.add(stepsInput);
    inputPanel.add(logStepsButton);

    // Stats Panel
    stepsLabel = new JLabel("Total Steps: 0");
    distanceLabel = new JLabel("Distance Covered: 0.00 km");
    caloriesLabel = new JLabel("Calories Burned: 0.00 kcal");
    stepProgressBar = new JProgressBar(0, GOAL_STEPS);
    stepProgressBar.setStringPainted(true);

    // Buttons Panel
    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new FlowLayout());
    JButton viewStatsButton = new JButton("View Stats");

```

```

        JButton resetButton = new JButton("Reset Stats");
        JButton exercisePlanButton = new JButton("View Exercise Plan");
        JButton dietPlanButton = new JButton("View Diet Plan");
        buttonPanel.add(viewStatsButton);
        buttonPanel.add(resetButton);
        buttonPanel.add(exercisePlanButton);
        buttonPanel.add(dietPlanButton);

        // Adding components to frame
        frame.add(inputPanel);
        frame.add(stepsLabel);
        frame.add(distanceLabel);
        frame.add(caloriesLabel);
        frame.add(stepProgressBar);
        frame.add(buttonPanel);

        // Add Action Listeners
        logStepsButton.addActionListener(e -> logSteps());
        viewStatsButton.addActionListener(e -> viewStats());
        resetButton.addActionListener(e -> resetStats());
        exercisePlanButton.addActionListener(e -> viewExercisePlan());
        dietPlanButton.addActionListener(e -> viewDietPlan());

        // Show frame
        frame.setVisible(true);
    }

    // Log steps
    private void logSteps() {

```

```

try {
    int stepsToAdd = Integer.parseInt(stepsInput.getText());
    if (stepsToAdd > 0) {
        steps += stepsToAdd;
        distanceCovered += stepsToAdd * 0.0008; // Average step length
        caloriesBurned += stepsToAdd * 0.04;    // Calories burned per
        step
        stepProgressBar.setValue(Math.min(steps, GOAL_STEPS)); //
        Update progress bar

        JOptionPane.showMessageDialog(frame, "✓ Steps logged
        successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
        stepsInput.setText("");
        updateStatsLabels();
    } else {
        JOptionPane.showMessageDialog(frame, "✗ Steps must be
        positive.", "Error", JOptionPane.ERROR_MESSAGE);
    }
} catch (NumberFormatException ex) {
    JOptionPane.showMessageDialog(frame, "✗ Invalid input. Please enter
    a number.", "Error", JOptionPane.ERROR_MESSAGE);
}

}

// View stats
private void viewStats() {
    JOptionPane.showMessageDialog(frame,
        "Steps: " + steps + "\n" +
        "Distance: " + String.format("%.2f", distanceCovered) + " km\n" +
        "Calories: " + String.format("%.2f", caloriesBurned) + " kcal\n",

```

```

        "Fitness Stats", JOptionPane.INFORMATION_MESSAGE);
    }

    // Reset stats
    private void resetStats() {
        int confirm = JOptionPane.showConfirmDialog(frame, "Are you sure you
        want to reset your stats?",
            "Confirm Reset", JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {
            steps = 0;
            caloriesBurned = 0.0;
            distanceCovered = 0.0;
            stepProgressBar.setValue(0);
            updateStatsLabels();
            JOptionPane.showMessageDialog(frame, "✔ Stats have been reset.",
            "Success", JOptionPane.INFORMATION_MESSAGE);
        }
    }

    // View Exercise Plan
    private void viewExercisePlan() {
        String plan;
        if (steps < 3000) {
            plan = "Beginner Plan:\n- 10 min light stretching\n- 15 min brisk
            walking\n- 5 min cooldown";
        } else if (steps < 7000) {
            plan = "Intermediate Plan:\n- 15 min jogging\n- 20 min brisk walking\n-
            10 min yoga";
        } else {
            plan = "Advanced Plan:\n- 30 min running\n- 15 min strength
            training\n- 10 min stretching";
        }
    }

```

```

    }

    JOptionPane.showMessageDialog(frame, plan, "Exercise Plan",
JOptionPane.INFORMATION_MESSAGE);
}

// View Diet Plan
private void viewDietPlan() {
    String plan;
    if (steps < 3000) {
        plan = "Diet Plan:\n- Breakfast: Oatmeal with fruits\n- Lunch: Grilled
chicken salad\n- Dinner: Steamed veggies and rice";
    } else if (steps < 7000) {
        plan = "Diet Plan:\n- Breakfast: Scrambled eggs with toast\n- Lunch:
Quinoa salad with chickpeas\n- Dinner: Baked salmon with asparagus";
    } else {
        plan = "Diet Plan:\n- Breakfast: Protein smoothie with nuts\n- Lunch:
Grilled turkey sandwich\n- Dinner: Lean steak with sweet potatoes";
    }

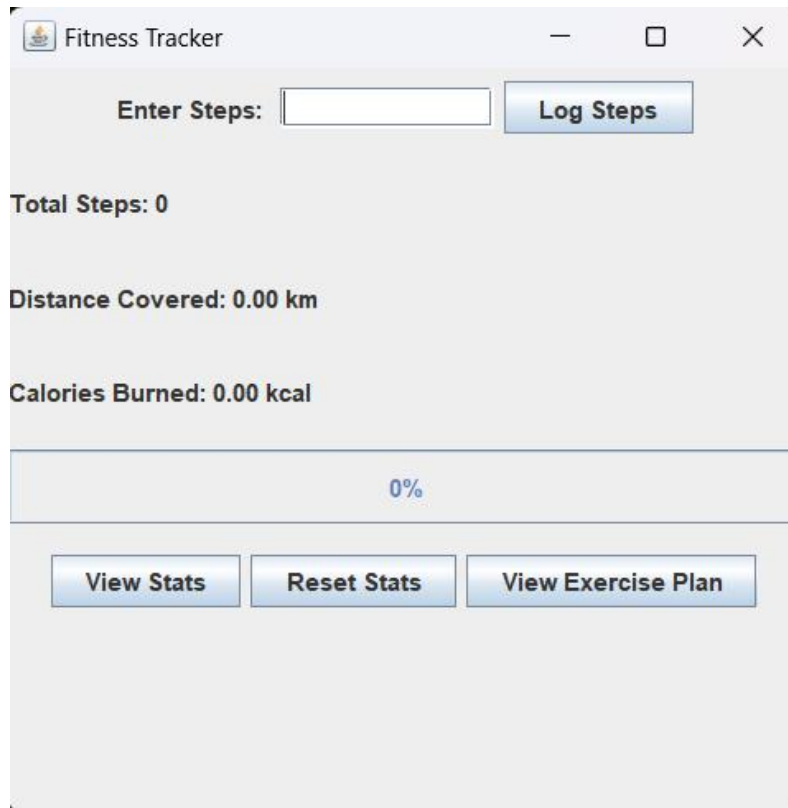
    JOptionPane.showMessageDialog(frame, plan, "Diet Plan",
JOptionPane.INFORMATION_MESSAGE);
}

// Update stats labels
private void updateStatsLabels() {
    stepsLabel.setText("Total Steps: " + steps);
    distanceLabel.setText("Distance Covered: " + String.format("%.2f",
distanceCovered) + " km");
    caloriesLabel.setText("Calories Burned: " + String.format("%.2f",
caloriesBurned) + " kcal");
}

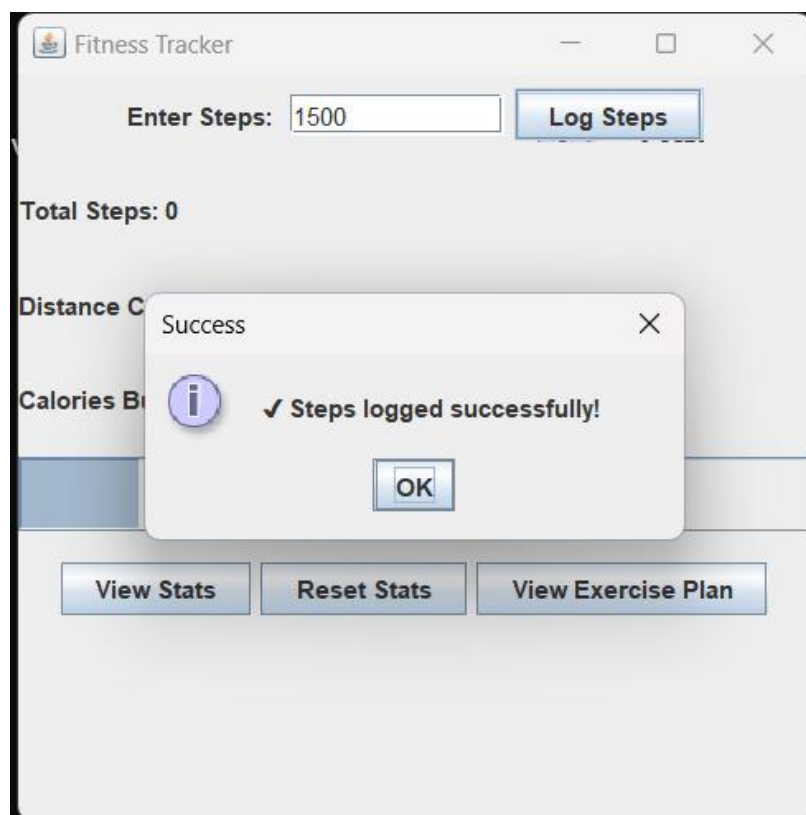
```

```
// Main Method  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(FitnessTrackerApp::new);  
}  
}
```

APPENDIX B– SCREEN SHOT



The screenshot shows a window titled "Fitness Tracker" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface includes an "Enter Steps:" label followed by a text input field and a "Log Steps" button. Below this, the current statistics are displayed: "Total Steps: 0", "Distance Covered: 0.00 km", and "Calories Burned: 0.00 kcal". A progress bar is shown with a blue fill and the text "0%". At the bottom, there are three buttons: "View Stats", "Reset Stats", and "View Exercise Plan".



This screenshot shows the same "Fitness Tracker" window, but with the "Enter Steps:" input field containing the value "1500". A modal dialog box titled "Success" is overlaid on the window. The dialog box contains a blue information icon, a checkmark, and the text "Steps logged successfully!". There is an "OK" button at the bottom of the dialog box. The background window is slightly dimmed.

