HEALTH TRACKER

A MINI-PROJECT REPORT

Submitted by

VIJAY. G 241901124

VIDHUSH ADITHYA 241901122

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI - 602105

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “HEALTH TRACKER APP” is the bonafide work of “VIJAY.G (241901124), VIDHUSH ADITHYA (241901122) 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

_____

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We hereby declare that the mini project report Health Tracker, 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, VIJAY. G

Chennai VIDHUSH ADITHYA

November 2025

ABSTRACT

The **Health Tracker** is a desktop-based application developed using **Java Swing** for the frontend and **MySQL** as the backend. The primary objective of this project is to help users monitor their daily nutritional intake by calculating **calories, proteins, carbohydrates, and fats** based on the foods they consume.

The system allows users to enter a food name and quantity, after which the application retrieves nutritional values from a MySQL database using **JDBC**. The application then automatically calculates the total nutrient intake for the given quantity and provides a detailed summary. The clean and modern user interface uses a card-style layout for improved readability and user experience.

This application demonstrates practical implementation of **database integration**, **Java GUI development**, and **real-time data processing**. Users can also add new food items to the database when needed, making the system adaptable and expandable.

By automating nutritional calculations, the Health Tracker minimizes manual effort, ensures accuracy, and provides relevant health insights. This project can be extended further with features like user profiles, daily logs, charts, and mobile/cloud integration.

Keywords: Java, MySQL, JDBC, Swing GUI, Nutrition Monitoring, Calorie Calculator, Database Integration.

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.

VIJAY. G

VIDHUSH ADITHYA

TABLE OF CONTENTS

LIST OF ABBREVIATION

ABBREVIATION	FULL TERM
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
JDBC	Java Database Connectivity
Java Swing	Java Framework (for GUI)
GUI	Graphical User Interface
UI	User Interface
ER	Entity-Relationship

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The **Health Tracker** is a desktop-based application developed using Java (Swing GUI) and MySQL to help users easily monitor their daily nutritional intake. The system provides a platform where users can input a **food name** and **quantity in grams**, and the application automatically fetches nutritional values — including **calories, protein, carbohydrates, and fat** — from the MySQL database. The application uses **JDBC** to establish a connection between the Java program and the database. The intuitive and modern user interface displays input fields, calculated results, and a table showing the user's cumulative daily intake. Users can also add new food items to the database directly through the application, making the system flexible and continuously expandable.

This project demonstrates real-time data processing, GUI development, and database integration while providing a practical health monitoring tool.

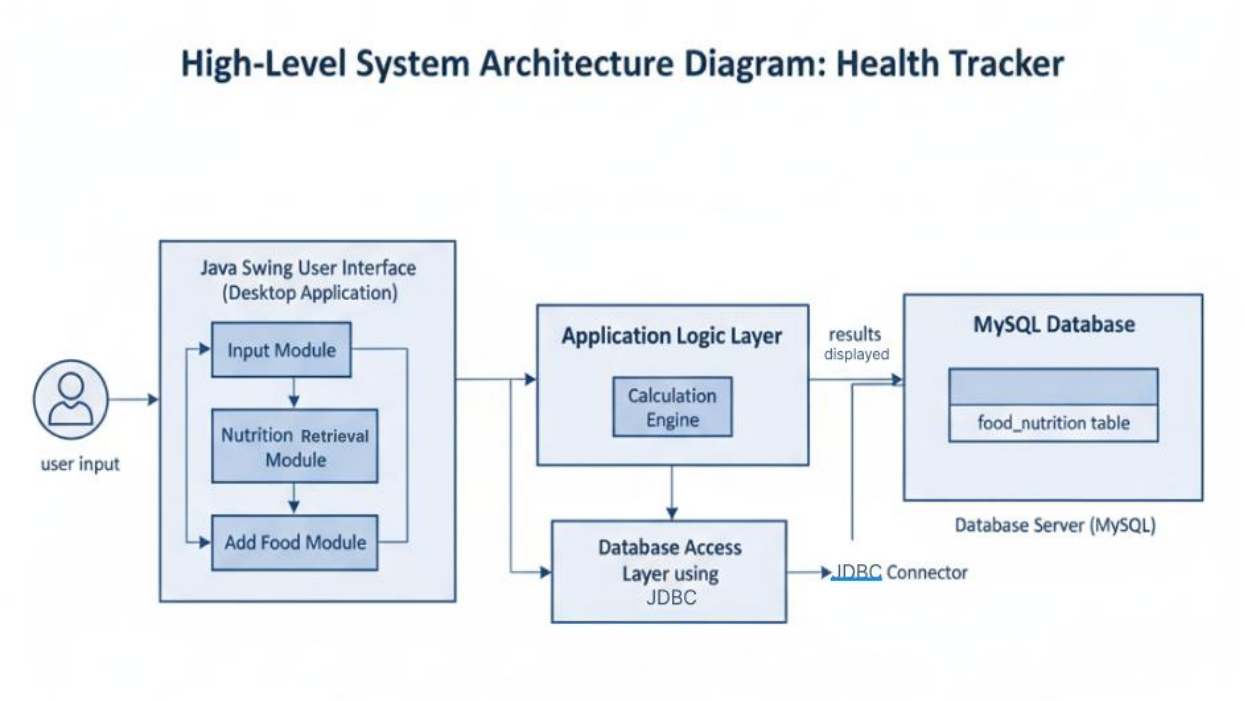

Fig.1.1 High-Level System Architecture

1.2 Scope of the Work

The scope of the **Health Tracker** project covers the development of a complete nutritional monitoring system that helps users track, analyze, and understand their daily dietary intake with ease.
The system has been designed to simplify the calculation of calories, proteins, carbohydrates, and fats based on the foods consumed by the user. The primary focus of this project is to provide users with a fast, accurate, and reliable digital tool that automatically retrieves nutritional data from a MySQL database and displays the results in a clean, modern interface.

The project includes the creation of a structured **food nutrition database** containing nutritional values for common food items per 100 grams. By connecting this database with Java using **JDBC**, the system enables seamless data retrieval and real-time calculations. Users only need to input the food name and quantity, and the system computes the total nutritional intake instantly. This removes the need for manual calculations and reduces the chances of errors.

The scope also includes features that allow users to **add new food items** to the database through the application interface. This makes the system dynamic and customizable — users are not restricted to predefined database items but can expand the database according to their dietary needs or local food availability.

Furthermore, the interface of the Health Tracker presents daily entries in a table format, enabling users to view and compare multiple food items consumed during the day. The use of Swing components and a card-style UI layout ensures smooth user interaction, readability, and simplified navigation.

Beyond the current implementation, the project has wide potential for future expansions.
The system can be easily extended to include:

- A user login system to maintain individual dietary profiles
- Daily, weekly, and monthly intake reports
- Graphical charts for calorie and macro analysis
- Integration with external nutrition APIs
- Tracking of user goals such as weight loss, muscle gain, or balanced diet
- Cloud synchronization for multi-device accessibility
- Export options to PDF or Excel for diet logs

Overall, the scope of the Health Tracker project covers developing a complete, structured, and expandable nutrition-tracking solution that can be adopted by fitness enthusiasts, dieticians, students, or anyone aiming to maintain a healthy lifestyle.

1.3 Problem Statement

In today's fast-paced lifestyle, many individuals struggle to maintain a balanced diet or keep track of the nutritional value of the food they consume. Although nutritional information is available on various websites or mobile apps, users often find it difficult to calculate accurate calorie and macronutrient intake for the actual quantities they eat. Manually calculating nutritional values using online charts or labels can be time-consuming, repetitive, and prone to errors.

Most commonly, users face the following issues:

- **Lack of a centralized system** to store, calculate, and analyze nutrition data
- **Manual errors** when calculating calories and macronutrients for different quantities
- **Difficulty accessing reliable nutritional data** for local or custom food items
- **Limited flexibility** in popular mobile apps that do not allow adding personalized items
- **No easy desktop-based application** for nutrition tracking that works offline
- **Difficulty maintaining daily logs** due to lack of simple tools

These challenges make it hard for users to maintain a consistent diet plan or achieve fitness goals. People trying to lose weight, gain muscle, or follow specific dietary recommendations often struggle because they do not have access to an accurate and user-friendly nutrition calculator.

Additionally, existing digital solutions tend to rely heavily on internet connectivity or paid subscriptions. They may also include unnecessary features, making them overwhelming for beginners who simply want a quick and accurate nutrient calculator. Users who prefer simplicity and privacy often require an offline, desktop-based solution that stores data locally.

Therefore, there is a need for an efficient, easy-to-use, and accurate nutrition tracking system that:

- Automatically calculates nutritional values from stored data
- Eliminates manual effort and human calculation errors
- Stores food items in a structured database
- Allows the addition of new foods
- Provides instant results based on user input
- Offers a clean and simple graphical user interface
- Works offline without requiring any third-party services

The Health Tracker project was developed to address all these challenges by providing a reliable and user-friendly desktop application that uses Java and MySQL to deliver fast, accurate, and customizable nutrition tracking.

1.4 Aim and Objectives of the Project

The main aim of the Pharmacy Management System is to create a simple and efficient software application that helps pharmacies manage their daily work. It is designed to replace manual record-keeping with a digital system that can store, update, and organize data about medicines, customers, and sales quickly and accurately. The goal is to make pharmacy operations faster, reduce human errors, and improve overall management through an easy-to-use computer application.

The specific objectives of the Pharmacy Management System are:

1. To build a system that stores and manages information about medicines, customers, and sales.
2. To automate billing and stock updates after every sale.
3. To reduce manual work and human errors in data entry and calculation.
4. To provide a user-friendly graphical interface for easy navigation and operation.
5. To use MySQL for storing data securely and JDBC for connecting it with the Java application.
6. To display all data in tables for better viewing and quick access.
7. To help the pharmacist easily track stock levels, expiry dates, and sales records.
8. To design the project so it can be expanded in the future, adding features like login security, barcode scanning, and automatic alerts.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Component	Minimum Specification
Processor	Dual-core 2.0 GHz or higher
Memory (RAM)	4GB (Minimum), 8GB (Recommended)
Storage	200MB free space
Display	1366x768 resolution

2.2 SOFTWARE SPECIFICATIONS

Component	Specification
Operating System	Windows 10/11, macOS 10.15+, Linux
Front-End	Java Swing
Back-End	MySQL 8.0 or above

Core Language	Java SE 17
Dependencies	MySQL JDBC Driver (connect JAVA to MYSQL), JDK 17 (Core Java Runtime)

CHAPTER 3

MODULE DESCRIPTION

The Health Tracker system is designed using a modular architecture, where each module handles a clearly defined functionality. This modular approach ensures better maintainability, reusability, and ease of understanding while developing or extending the system.

The major modules included in this project are:

3.1 Food Input Module

The Food Input Module is responsible for collecting user input and validating it before processing. It acts as the primary interaction point between the user and the application.

Functions:

- Accepts input for **food name** and **quantity (in grams)**.
- Validates the entered values to avoid empty or invalid inputs.
- Sends the input to the Nutrition Retrieval and Calculation modules for further processing.
- Displays error messages for invalid entries, missing fields, or non-numeric quantities.

3.2 Nutrition Retrieval Module

The Nutrition Retrieval Module communicates with the MySQL database to fetch the nutritional values of the entered food item.

Functions:

- Establishes a JDBC connection with the database.
- Executes SQL queries such as: SELECT calories_per_100g, carbs_per_100g, protein_per_100g, fat_per_100g FROM food_nutrition WHERE food_name = ?
- Retrieves nutritional values per 100g from the database.
- Returns the values to the calculation module.
- Prompts the user when a food item is *not found* in the database.

3.3 Nutrition Calculation Module

This module uses the data retrieved from the database and calculates the total nutritional values based on the quantity entered by the user.

Functions:

- Performs real-time calculations such as:
- total_calories = (calories_per_100g / 100) * quantity
- total_carbs = (carbs_per_100g / 100) * quantity
- total_protein = (protein_per_100g / 100) * quantity
- total_fat = (fat_per_100g / 100) * quantity
- Updates the daily totals displayed in the UI.
- Adds the calculated result to the daily intake table.

3.4 Add Food Module (Database Update Module)

This module allows users to add new foods to the database when they are not found.

Functions:

- Displays an input form for entering new food details.
- Accepts nutritional values per 100g:
    - Calories
    - Carbs
    - Protein
    - Fat
- Inserts new records into the MySQL database using SQL INSERT statements.
- Ensures the system is expandable based on the user's needs.

3.5 User Interface Module

The UI module is responsible for displaying input fields, buttons, results, and tables using Java Swing. The UI follows a **Modern Card-Based Layout** to ensure a clean and user-friendly experience.

Functions:

- Displays input fields for food and quantity.
- Shows results (calories, carbs, proteins, fats) in card-style labels.
- Maintains a table of all food items entered for the day.
- Provides "Add & Calculate", "Clear", and "Add Food" functions.

- Ensures consistent design using Swing panels, layouts, and styling.

3.6 ER Diagram

The Entity–Relationship (ER) Diagram shown above represents the logical structure of the database used in the **Health Tracker** application. The system primarily operates on the **Food_Nutrition** entity, which stores essential nutritional information for various food items, such as calories, carbohydrates, proteins, and fats per 100 grams. This entity forms the foundation of the application, enabling accurate nutritional calculations based on user input. Each food item is stored as a unique record in the Food_Nutrition table, identified by a Primary Key (PK). The attributes stored in this entity ensure that the system can efficiently retrieve nutritional values whenever the user enters a food name and quantity.

Although the current version of the Health Tracker uses a single primary entity, the design is **scalable and future-ready**. Additional entities such as **User_Details** and **Daily_Intake_Log** can be easily introduced to extend the system's functionality. In such an extended design, the Daily_Intake_Log entity acts as a bridge between the User and Food_Nutrition entities, storing details of each food item consumed by a user, along with calculated totals such as calories or macronutrient values.

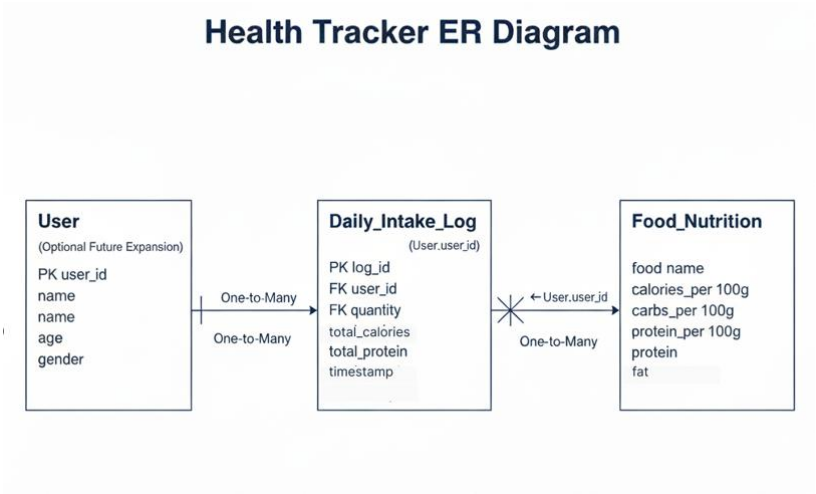

Fig.3.1 ER Diagram

3.7 Database Schema

Explanation: Database Schema of Food_Nutrition

The food_nutrition table stores the nutritional information of various food items used in the Health Tracker application. Each record in this table contains details such as the food name, calories, carbohydrates, protein, and fat per 100 grams. This table serves as the core component of the system because all nutritional calculations depend on the data stored here.

Whenever a user enters a food name and quantity, the system retrieves the corresponding nutritional values from this table and calculates the total intake. By maintaining all nutritional information in a structured and normalized format, the system ensures accurate retrieval, efficient data management, and quick calculation of nutritional values.

Column	Type	Nullable	Indexes
id	int	NO	PRIMARY
food_name	varchar(150)	NO	food_name
calories_per_100g	float	NO	
carbs_per_100g	float	NO	
protein_per_100g	float	NO	
fat_per_100g	float	NO	

Fig.3.2 Database Schema of Food_Nutrition

CHAPTER 4

CODING

4.1 Introduction

This chapter describes the implementation of the Health Tracker system, detailing the source code used to develop the application. The project is implemented using **Java Swing** for the graphical user interface and **MySQL** for the backend database, connected through **JDBC (Java Database Connectivity)**.

The coding structure is modular, with separate sections for database connectivity, nutrition retrieval, calculation logic, adding new food items, and the graphical user interface (GUI). The goal of the coding phase is to integrate all modules so that the application functions seamlessly and performs accurate nutritional calculations based on user input.

4.2 Database Connection Code

This module handles the connection between the Java application and the MySQL database.
It ensures that the database is accessible when retrieving nutritional values or inserting new food items.

```
private Connection getConnection() {

   try {

      String url = "jdbc:mysql://localhost:3306/health_tracker";

      String user = "root";

      String pass = "your_password";

      return DriverManager.getConnection(url, user, pass);

   } catch (SQLException e) {

      JOptionPane.showMessageDialog(this,

         "Database connection failed: " + e.getMessage());

      return null;

   }
```

```
}
```

4.3 Nutrition Retrieval Module

This module retrieves the nutritional values of a food item from the database based on user input. If a food item is not found, the user is prompted to add it to the database.

```
private FoodData getNutrition(String foodName) {
    Connection conn = getConnection();
    if (conn == null) return null;
    try {
        PreparedStatement ps = conn.prepareStatement(
            "SELECT * FROM food_nutrition WHERE LOWER(food_name)=LOWER(?)");
        ps.setString(1, foodName);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            return new FoodData(
                rs.getString("food_name"),
                rs.getDouble("calories_per_100g"),
                rs.getDouble("carbs_per_100g"),
                rs.getDouble("protein_per_100g"),
                rs.getDouble("fat_per_100g")
            );
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
```

```
    }

    return null;

}
```

4.4 Calculation Logic Module

This module computes the **calories, carbohydrates, protein, and fat** intake for the quantity entered by the user. It uses the nutritional values retrieved from the database and multiplies them based on the quantity in grams.

```
private NutritionResult calculate(FoodData data, double qty) {

    double cal  = data.calories  * qty / 100.0;

    double carb = data.carbs     * qty / 100.0;

    double prot = data.protein   * qty / 100.0;

    double fat  = data.fat       * qty / 100.0;

    return new NutritionResult(cal, carb, prot, fat);

}
```

4.5 Add Food Module

When a food item is not found in the database, the system provides an option to add it. This module inserts the new food item into the food_nutrition table.

```
private void addFoodToDatabase(String name, double cal, double carb, double prot, double fat) {

    try {

        Connection conn = getConnection();

        PreparedStatement ps = conn.prepareStatement(

            "INSERT INTO
food_nutrition(food_name,calories_per_100g,carbs_per_100g,protein_per_100g,fat_per_100g)
VALUES (?,?,?,?,?)"

        );
```

```
        ps.setString(1, name);
        ps.setDouble(2, cal);
        ps.setDouble(3, carb);
        ps.setDouble(4, prot);
        ps.setDouble(5, fat);
        ps.executeUpdate();
        JOptionPane.showMessageDialog(this, "Food Added Successfully!");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
    }
}
```

4.6 User Interface (UI) Module

The User Interface (UI) is designed using **Java Swing**, with a modern card-based layout for better readability and user experience.

```
public HealthTracker() {
    setTitle("Health Tracker");
    setSize(820, 560);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    JPanel main = new JPanel(new BorderLayout());
    add(main);
    JLabel header = new JLabel("Health Tracker");
    header.setFont(new Font("Arial", Font.BOLD, 26));
```

```
    main.add(header, BorderLayout.NORTH);
    setupUI(main);
    setVisible(true);
}
```

CHAPTER 5

SCREENSHOTS

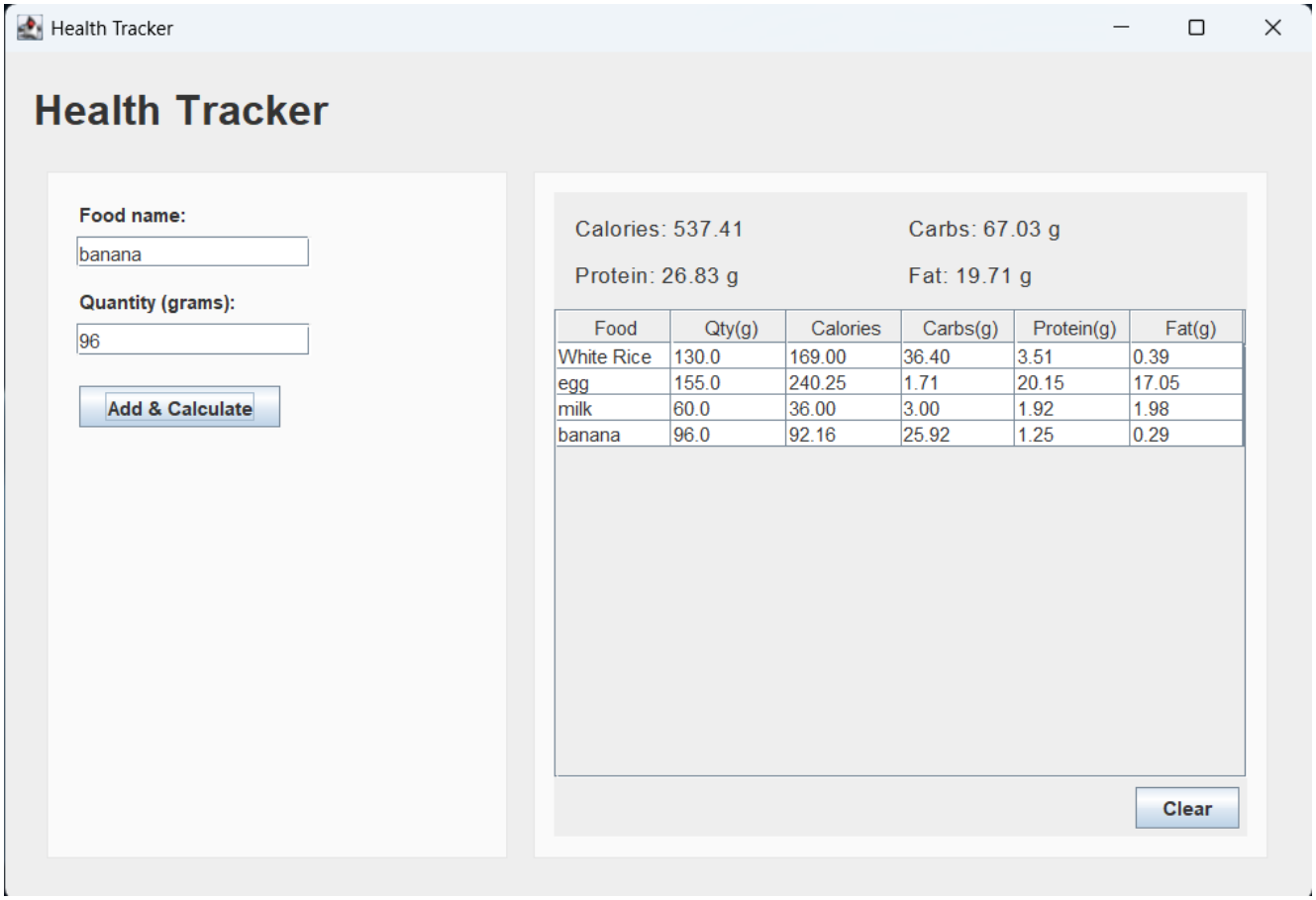

Fig.5.1 Health Tracker App UI

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The **Health Tracker** application was successfully developed and implemented using **Java Swing** for the front-end and **MySQL** for the back-end database. The system achieves its primary goal of helping users calculate and monitor their daily nutritional intake, including **calories, carbohydrates, proteins, and fats**, based on the food items they consume.

By integrating Java with MySQL through **JDBC**, the system ensures smooth, accurate, and real-time data retrieval from the food_nutrition database. The card-style modern interface makes the application visually appealing, intuitive, and easy to navigate even for non-technical users.

The application eliminates the need for manual calculations, reduces human error, and provides users with instant and reliable nutritional information. he ability to add new foods to the database further enhances the flexibility and usefulness of the system, making it adaptable for different dietary needs and regional food variations.

Overall, the project demonstrates practical knowledge of **GUI development**, **database connectivity**, **modular programming**, and **real-time data processing**. The Health Tracker stands as a functional and user-friendly desktop tool that can support individuals in maintaining a healthier lifestyle.

6.2 Future Enhancements

Although the current version of the Health Tracker meets the essential requirements, several improvements can be incorporated to increase its usability, scalability, and effectiveness.

Future versions of the system may include:

1.User Authentication and Profiles

Adding a secure login system where multiple users can store personalized daily intake records, goals, and progress.

2.Daily / Weekly / Monthly Reports

Automatically generating nutrition reports in graphical and tabular formats to help users analyze their dietary trends.

3.Graphical Charts and Data Visualization

Integrating bar charts, pie charts, and line graphs to visually display calorie and macro intake over time.

4. Cloud Synchronization

Syncing user data online, enabling access from multiple devices such as PCs, laptops, and smartphones.

5. Diet Recommendations

Providing suggestions based on user goals such as weight loss, weight gain, or balanced diet.

6. Integration with Nutritional APIs

Fetching nutritional data from trusted online sources to expand the database automatically.

7.Notifications and Alerts

Implementing reminders for daily intake logging or alerts when users exceed recommended limits.

8.Advanced Food Search

Supporting autocomplete search, filtering by food categories, and displaying common serving sizes.

REFERENCES

1. Oracle Corporation, “Java Swing Tutorial”, https://docs.oracle.com/javase/tutorial/uiswing/
2. MySQL AB, “MySQL 8.0 Reference Manual”, https://dev.mysql.com/doc/refman/8.0/en/
3. FormDev Software, “FlatLaf Look and Feel for Swing”, https://www.formdev.com/flatlaf/
4. GeeksforGeeks, “JDBC – Java Database Connectivity”, https://www.geeksforgeeks.org/jdbc-java-database-connectivity/
5. TutorialsPoint, “Java Swing Components and Layouts”, https://www.tutorialspoint.com/java_swing/
6. W3Schools, “MySQL Tutorial”, https://www.w3schools.com/mysql/
7. Oracle University, “Java Database Programming Using JDBC”, https://www.oracle.com/java/technologies/javase/jdbc.html
8. Stack Overflow Discussions, “Common JDBC and Swing Implementation Queries”, https://stackoverflow.com/
9. Official MySQL Connector/J Documentation, https://dev.mysql.com/doc/connector-j/8.0/en/
10. National Institutes of Health (NIH), **Dietary Reference Intakes (DRIs)**, https://www.nal.usda.gov/human-nutrition-and-food-safety/dri-calculators
11. World Health Organization (WHO), **Healthy Diet Guidelines**, https://www.who.int/news-room/fact-sheets/detail/healthy-diet