RAJALAKSHMI ENGINEERING COLLEGE
(AN AUTONOMOUS INSTITUTE)

A MINI PROJECT BY :
LOHIT S (230701165)
MANU S D (230701179)

IN PARTIAL FULFILLMENT OF THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Certified that this project titled **"UG Dissertation"** is the bonafide work of **Lohit S (230701165) and Manu S D (230701179),** who carried out the project work under our supervision during the year **2024–2025**.

Signature Of Faculty-In-Charge

Submitted for the Practical Examination held on _____________

Internal Examiner

External Examiner

Abstract of the Project

The Fitness Tracking Application is a comprehensive platform designed to empower users to monitor and manage their health and wellness effectively. It provides features for tracking food intake, physical activity, and medical conditions, helping users achieve a balanced lifestyle while maintaining health records.

This application integrates a user-friendly JavaFX interface, allowing seamless navigation through modules like food tracking, activity tracking, and user profile management. The backend utilizes a MySQL database to securely store user data, ensuring efficient and reliable data handling. Key functionalities include profile creation, database connectivity for persistent storage, and interactive features for user engagement.

The Fitness Tracking Application aims to enhance the user's ability to visualize their fitness progress through an intuitive design and robust backend framework. By facilitating accurate tracking and insights, this project provides an efficient tool for personal health management, suitable for users at all fitness levels.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

The Fitness Tracking Application is an all-in-one solution designed to assist users in managing their health-related activities. By offering modules for food tracking, activity monitoring, and medical condition management, the application provides a centralized platform for users to log, review, and analyze their fitness data.

The application is built with JavaFX for its frontend, ensuring an interactive and aesthetically pleasing user interface. The backend employs a MySQL database, coupled with Java, to manage user data and enable smooth interactions between the user interface and database. Key functionalities include secure user profile management, database connectivity for persistent data storage, and dynamic navigation across various modules.

This application is tailored to promote health and wellness by simplifying the tracking process and empowering users to maintain their fitness goals effectively. By combining robust backend operations with a visually engaging interface, the system ensures an optimal experience for users seeking to enhance their lifestyle and health.

1.2 IMPLEMENTATION

The Fitness Tracking Application is implemented using JavaFX to create an interactive and visually appealing graphical user interface. This interface facilitates seamless navigation and data entry across various modules, such as food tracking, activity monitoring, and user profile management.

The backend logic, written in Java, handles core functionalities including user authentication, data processing, and interaction with the database. MySQL is utilized for database management, providing a secure and efficient platform for storing user data such as profile details, fitness logs, and medical information.

The integration of JavaFX with a MySQL database ensures a cohesive and robust system, allowing users to effortlessly track and manage their health-related activities while maintaining data accuracy and persistence.

1.3 SCOPE OF THE PROJECT

The Fitness Tracking Application offers a comprehensive platform for individuals to manage and monitor their health and wellness activities. Its scope includes:

- Providing an intuitive and user-friendly interface for users to track food intake, physical activities, and medical conditions seamlessly.
- Ensuring secure storage and management of user data through a robust MySQL database.
- Enabling user profile management with features for updating personal details and managing preferences.

- Supporting scalability to cater to a diverse range of users, from casual fitness enthusiasts to individuals with specific health goals.
- Offering potential for future enhancements, such as integrating wearable device data, personalized fitness recommendations, and real-time progress analysis.

1.4 SYSTEM FEATURES

- **Registration and Login Page:** Provides secure authentication for users to access their personal fitness data.
- **User Profile Management:** Allows users to create, view, and update their profile details, including personal information and preferences.
- **Food Tracking Module:** Enables users to log daily food intake, monitor nutritional information, and track dietary habits.
- **Activity Tracking Module:** Facilitates the logging of physical activities and exercises, helping users monitor progress toward fitness goals.
- **Medical Condition Tracking:** Allows users to record and manage medical conditions, ensuring better health monitoring.
- **Database Integration:** Utilizes a MySQL database to securely store and retrieve user profiles, fitness logs, and other health-related data.
- **Interactive Dashboard:** Provides users with a centralized view of their fitness activities and progress over time.
- **Navigation System:** Seamless transition between different modules for an efficient and engaging user experience.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

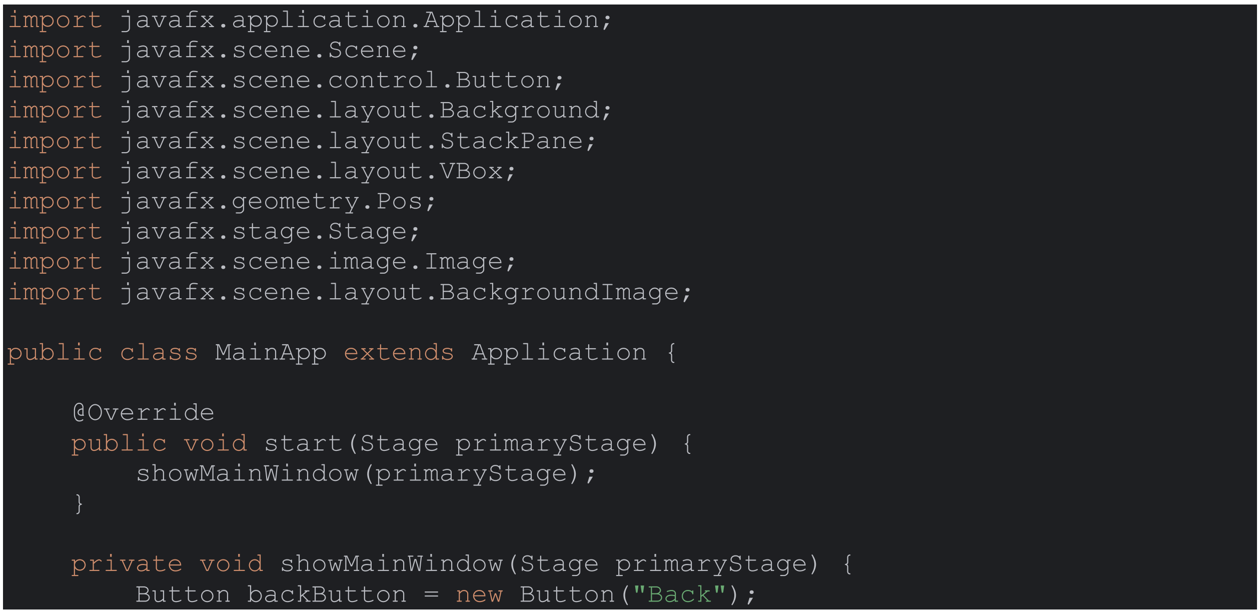
GRAPHICS	: Intel Integrated Graphics
PROCESSOR	: Intel i5 (11th GEN)
MEMORY SIZE	: 16 GB DDR4

2.2 SOFTWARE SPECIFICATIONS:

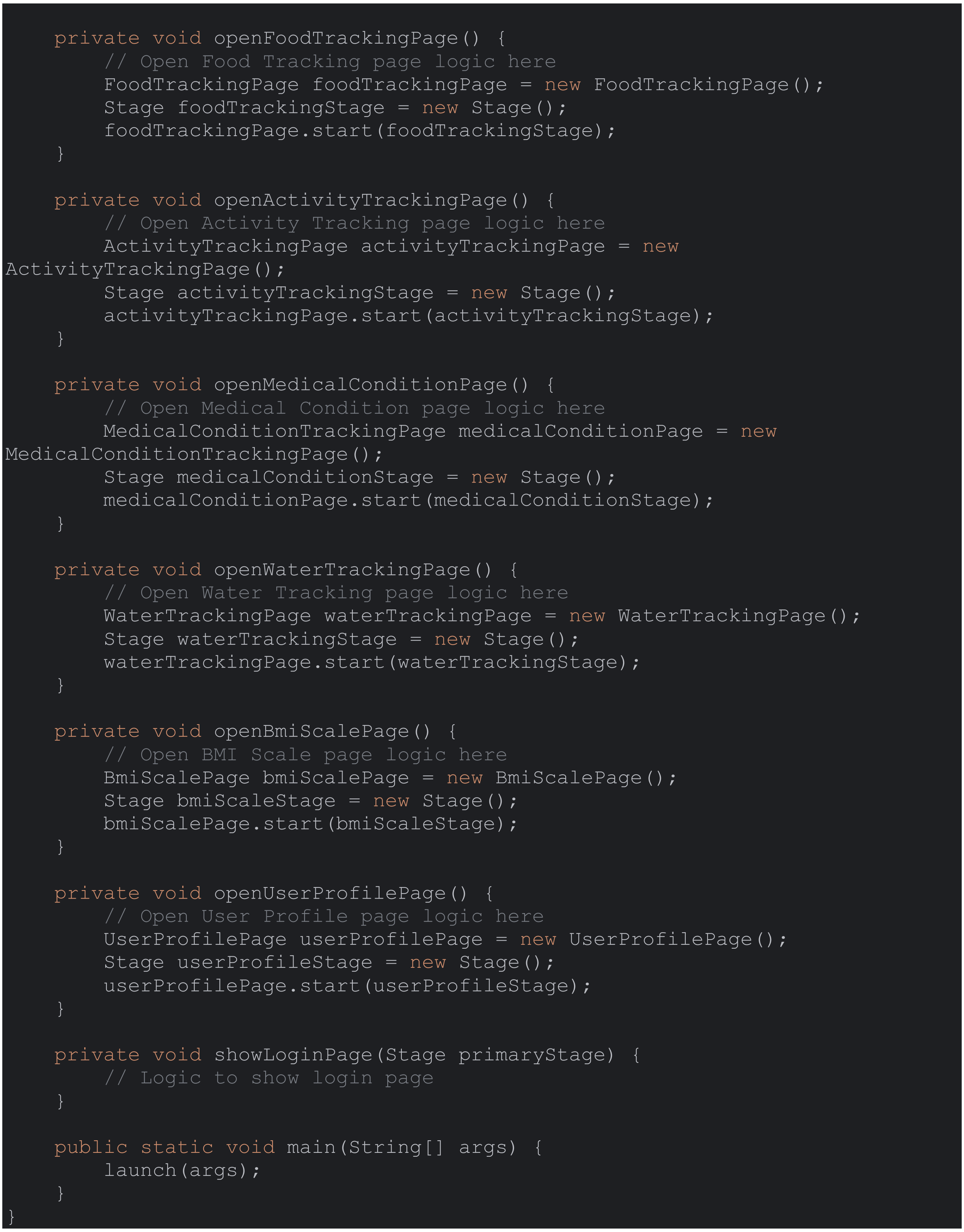
PROGRAMMING LANGUAGE	: Java, MySQL
FRONT-END	: Java
BACK-END	: MySQL
OPERATING SYSTEM	: Windows 11

SOURCE CODE

3.1 MAIN APPLICATION : [MainApp.java]



```java
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.Background;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.geometry.Pos;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.layout.BackgroundImage;

public class MainApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        showMainWindow(primaryStage);
    }

    private void showMainWindow(Stage primaryStage) {
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -
fx-font-weight: bold; -fx-text-fill: white; -fx-background-color: #333;");
        backButton.setOnAction(e -> showLoginPage(primaryStage));

        Button foodTrackingButton = new Button("Food Tracking");
        Button activityTrackingButton = new Button("Activity Tracking");
        Button medicalConditionTrackingButton = new Button("Medical Condition
Tracking");
        Button waterTrackingButton = new Button("Water Tracking");
        Button bmiScaleButton = new Button("BMI Scale");
        Button userProfileButton = new Button("User Profile");

        foodTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        activityTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-
size: 14px;");
        medicalConditionTrackingButton.setStyle("-fx-font-family: 'Arial'; -
fx-font-size: 14px;");
        waterTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-
size: 14px;");
        bmiScaleButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        userProfileButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");

        // Set up button actions to navigate to corresponding pages
        foodTrackingButton.setOnAction(e -> openFoodTrackingPage());
        activityTrackingButton.setOnAction(e -> openActivityTrackingPage());
        medicalConditionTrackingButton.setOnAction(e ->
openMedicalConditionPage());
        waterTrackingButton.setOnAction(e -> openWaterTrackingPage());
        bmiScaleButton.setOnAction(e -> openBmiScalePage());
        userProfileButton.setOnAction(e -> openUserProfilePage());

        VBox layout = new VBox(20);
        layout.getChildren().addAll(
                backButton,
                foodTrackingButton,
                activityTrackingButton,
                medicalConditionTrackingButton,
                waterTrackingButton,
                bmiScaleButton,
                userProfileButton
        );
        layout.setAlignment(Pos.CENTER);

        // Set background and scene without using additional properties
        StackPane root = new StackPane();
        Image image = new Image("file:/C:/XboxGames/image.jpg"); // Update
the file path as needed
        BackgroundImage backgroundImage = new BackgroundImage(image, null,
null, null, null);
        Background background = new Background(backgroundImage);
        root.setBackground(background);
        root.getChildren().add(layout);

        Scene scene = new Scene(root, 600, 400);
        primaryStage.setTitle("Personal Fitness Tracker");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
```



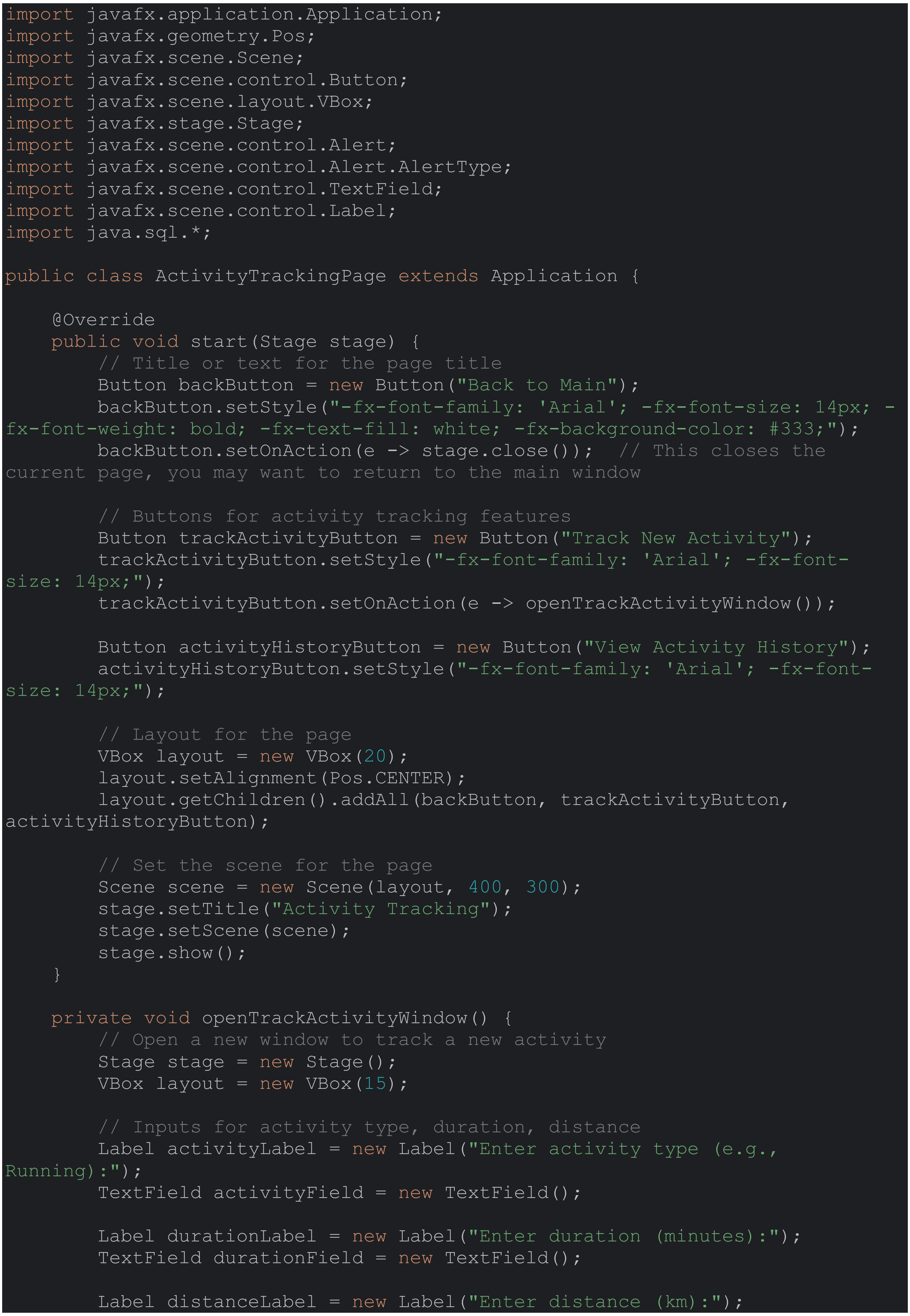
```
    private void openFoodTrackingPage() {
        // Open Food Tracking page logic here
        FoodTrackingPage foodTrackingPage = new FoodTrackingPage();
        Stage foodTrackingStage = new Stage();
        foodTrackingPage.start(foodTrackingStage);
    }

    private void openActivityTrackingPage() {
        // Open Activity Tracking page logic here
        ActivityTrackingPage activityTrackingPage = new
ActivityTrackingPage();
        Stage activityTrackingStage = new Stage();
        activityTrackingPage.start(activityTrackingStage);
    }

    private void openMedicalConditionPage() {
        // Open Medical Condition page logic here
        MedicalConditionTrackingPage medicalConditionPage = new
MedicalConditionTrackingPage();
        Stage medicalConditionStage = new Stage();
        medicalConditionPage.start(medicalConditionStage);
    }

    private void openWaterTrackingPage() {
        // Open Water Tracking page logic here
        WaterTrackingPage waterTrackingPage = new WaterTrackingPage();
        Stage waterTrackingStage = new Stage();
        waterTrackingPage.start(waterTrackingStage);
    }

    private void openBmiScalePage() {
        // Open BMI Scale page logic here
        BmiScalePage bmiScalePage = new BmiScalePage();
        Stage bmiScaleStage = new Stage();
        bmiScalePage.start(bmiScaleStage);
    }

    private void openUserProfilePage() {
        // Open User Profile page logic here
        UserProfilePage userProfilePage = new UserProfilePage();
        Stage userProfileStage = new Stage();
        userProfilePage.start(userProfileStage);
    }

    private void showLoginPage(Stage primaryStage) {
        // Logic to show login page
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.2 USER PROFILE DESIGN :

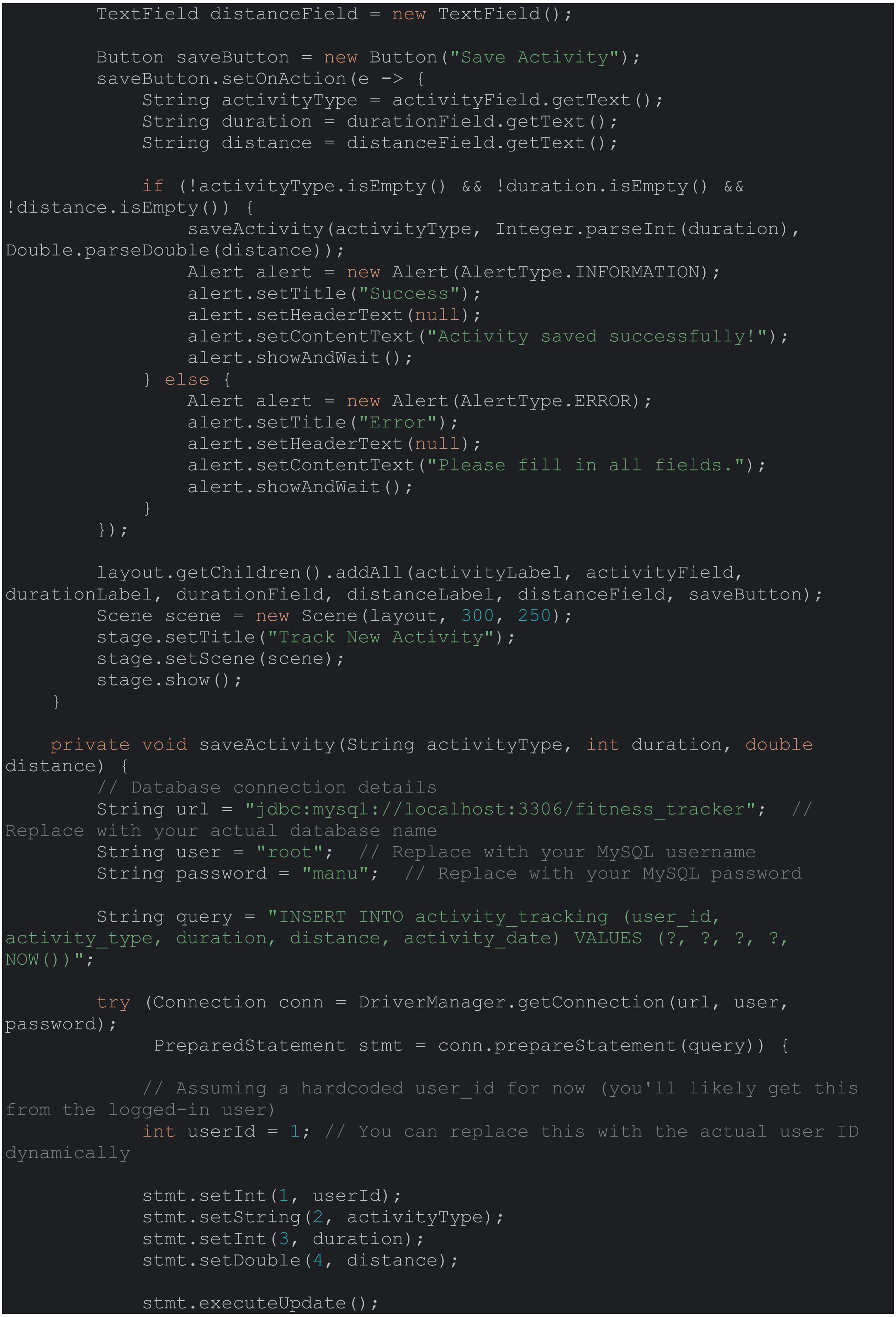
```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class UserProfilePage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("User Profile");
        titleLabel.setFont(new Font("Arial", 24));

        // User information fields
        Label nameLabel = new Label("Full Name:");
        TextField nameField = new TextField();

        Label emailLabel = new Label("Email Address:");
        TextField emailField = new TextField();

        Label passwordLabel = new Label("Password:");
        PasswordField passwordField = new PasswordField();

        // Profile picture (for simplicity, use a placeholder)
        Label pictureLabel = new Label("Profile Picture:");
        Button changePicButton = new Button("Change Picture");

        // Button to save updated profile
        Button saveButton = new Button("Save Profile");
        saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");

        // Label to show save status
        Label statusLabel = new Label();
        statusLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Action for the "Save Profile" button
        saveButton.setOnAction(e -> {
            String name = nameField.getText();
            String email = emailField.getText();
            String password = passwordField.getText();

            // Insert or update user profile in the database
            if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()) {
                saveUserProfile(name, email, password);
                statusLabel.setText("Profile updated successfully!");
            } else {
                statusLabel.setText("Please fill in all fields.");
            }
        });

        // Back button to go to the previous page
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
```

```
        backButton.setOnAction(e -> stage.close());  // Close the current page

        // Layout for the page
        VBox layout = new VBox(15);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(
            titleLabel,
            nameLabel,
            nameField,
            emailLabel,
            emailField,
            passwordLabel,
            passwordField,
            pictureLabel,
            changePicButton,
            saveButton,
            statusLabel,
            backButton
        );

        // Set up the scene
        Scene scene = new Scene(layout, 400, 350);
        stage.setTitle("User Profile");
        stage.setScene(scene);
        stage.show();
    }

    private void saveUserProfile(String name, String email, String password) {
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";
        String user = "root";
        String pass = "manu";

        try (Connection conn = DriverManager.getConnection(url, user, pass)) {
            String query = "INSERT INTO user_profiles (full_name, email, password) VALUES (?, ?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setString(1, name);
                stmt.setString(2, email);
                stmt.setString(3, password);  // You should hash the password in a real app
                stmt.executeUpdate();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.3 ACTIVITY TRACKING PAGE DESIGN :



```java
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.TextField;
import javafx.scene.control.Label;
import java.sql.*;

public class ActivityTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title or text for the page title
        Button backButton = new Button("Back to Main");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -
fx-font-weight: bold; -fx-text-fill: white; -fx-background-color: #333;");
        backButton.setOnAction(e -> stage.close());  // This closes the
current page, you may want to return to the main window

        // Buttons for activity tracking features
        Button trackActivityButton = new Button("Track New Activity");
        trackActivityButton.setStyle("-fx-font-family: 'Arial'; -fx-font-
size: 14px;");
        trackActivityButton.setOnAction(e -> openTrackActivityWindow());

        Button activityHistoryButton = new Button("View Activity History");
        activityHistoryButton.setStyle("-fx-font-family: 'Arial'; -fx-font-
size: 14px;");

        // Layout for the page
        VBox layout = new VBox(20);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(backButton, trackActivityButton,
activityHistoryButton);

        // Set the scene for the page
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("Activity Tracking");
        stage.setScene(scene);
        stage.show();
    }

    private void openTrackActivityWindow() {
        // Open a new window to track a new activity
        Stage stage = new Stage();
        VBox layout = new VBox(15);

        // Inputs for activity type, duration, distance
        Label activityLabel = new Label("Enter activity type (e.g.,
Running):");
        TextField activityField = new TextField();

        Label durationLabel = new Label("Enter duration (minutes):");
        TextField durationField = new TextField();

        Label distanceLabel = new Label("Enter distance (km):");
```



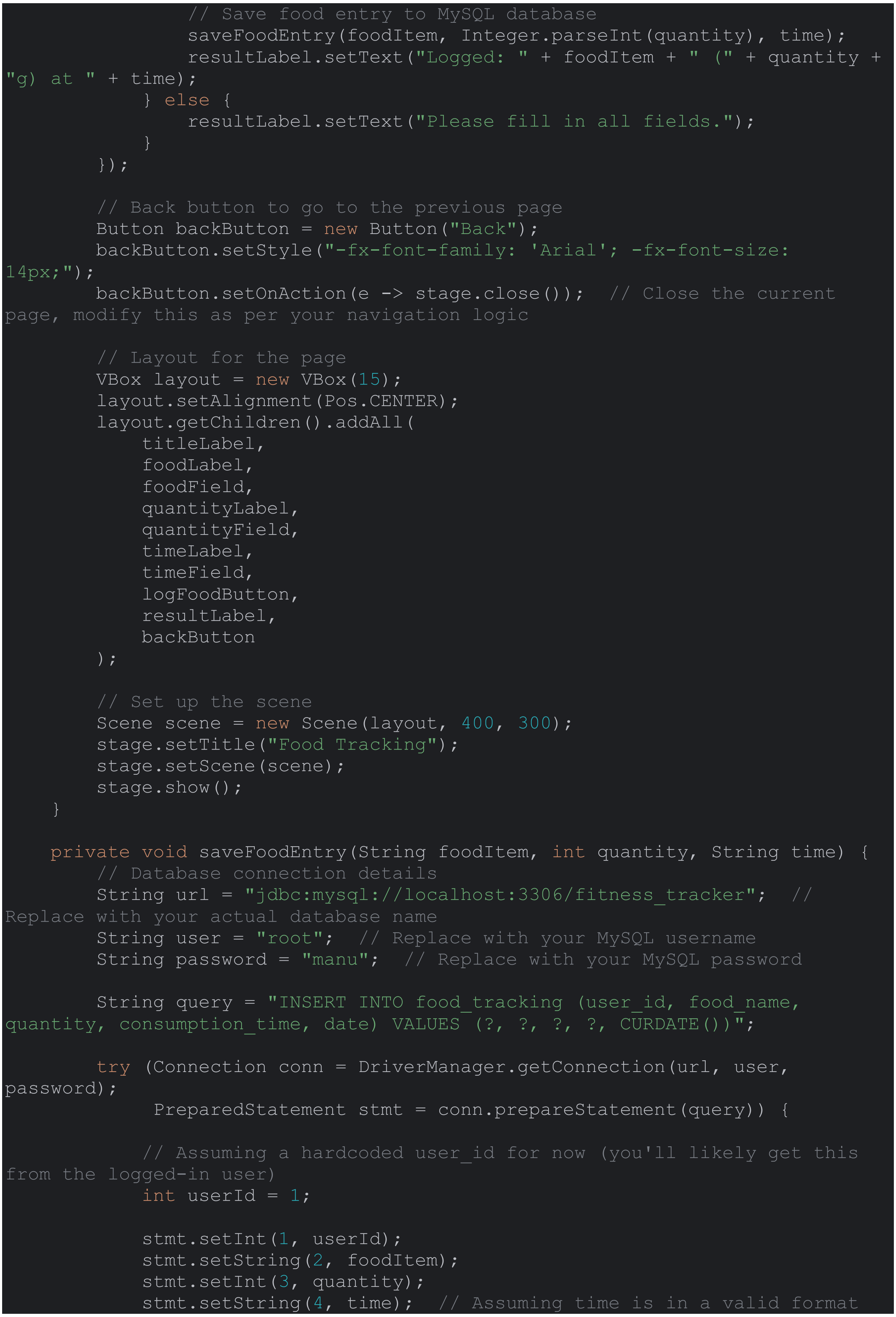
```
        TextField distanceField = new TextField();

        Button saveButton = new Button("Save Activity");
        saveButton.setOnAction(e -> {
            String activityType = activityField.getText();
            String duration = durationField.getText();
            String distance = distanceField.getText();

            if (!activityType.isEmpty() && !duration.isEmpty() &&
!distance.isEmpty()) {
                saveActivity(activityType, Integer.parseInt(duration),
Double.parseDouble(distance));
                Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Success");
                alert.setHeaderText(null);
                alert.setContentText("Activity saved successfully!");
                alert.showAndWait();
            } else {
                Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error");
                alert.setHeaderText(null);
                alert.setContentText("Please fill in all fields.");
                alert.showAndWait();
            }
        });

        layout.getChildren().addAll(activityLabel, activityField,
durationLabel, durationField, distanceLabel, distanceField, saveButton);
        Scene scene = new Scene(layout, 300, 250);
        stage.setTitle("Track New Activity");
        stage.setScene(scene);
        stage.show();
    }

    private void saveActivity(String activityType, int duration, double
distance) {
        // Database connection details
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";  //
Replace with your actual database name
        String user = "root";  // Replace with your MySQL username
        String password = "manu";  // Replace with your MySQL password

        String query = "INSERT INTO activity_tracking (user_id,
activity_type, duration, distance, activity_date) VALUES (?, ?, ?, ?,
NOW())";

        try (Connection conn = DriverManager.getConnection(url, user,
password);
             PreparedStatement stmt = conn.prepareStatement(query)) {

            // Assuming a hardcoded user_id for now (you'll likely get this
from the logged-in user)
            int userId = 1; // You can replace this with the actual user ID
dynamically

            stmt.setInt(1, userId);
            stmt.setString(2, activityType);
            stmt.setInt(3, duration);
            stmt.setDouble(4, distance);

            stmt.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.4 BMI SCALE PAGE DESIGN

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class BmiScalePage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("BMI Scale");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for height and weight
        Label heightLabel = new Label("Enter your height (in meters):");
        TextField heightField = new TextField();

        Label weightLabel = new Label("Enter your weight (in kilograms):");
        TextField weightField = new TextField();

        // Button to calculate BMI
        Button calculateButton = new Button("Calculate BMI");
        calculateButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 
14px;");

        // Label to display the result
        Label resultLabel = new Label("Your BMI will be displayed here.");
        resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Calculate BMI when the button is clicked
        calculateButton.setOnAction(e -> {
            try {
                double height = Double.parseDouble(heightField.getText());
                double weight = Double.parseDouble(weightField.getText());
                if (height > 0 && weight > 0) {
                    double bmi = weight / (height * height);
                    resultLabel.setText(String.format("Your BMI is: %.2f", 
bmi));
                    saveBmiToDatabase(height, weight, bmi);  // Save the
result to the database
                } else {
                    resultLabel.setText("Please enter valid positive
values.");
                }
            } catch (NumberFormatException ex) {
                resultLabel.setText("Please enter valid numbers.");
            }
        });

        // Back button to go to the previous page
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        backButton.setOnAction(e -> stage.close());  // Close the current
page, modify this as per your navigation logic

        // Layout for the page
        VBox layout = new VBox(15);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(
            titleLabel,
            heightLabel,
            heightField,
            weightLabel,
            weightField,
            calculateButton,
            resultLabel,
            backButton
        );

        // Set up the scene
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("BMI Scale");
        stage.setScene(scene);
        stage.show();
    }

    private void saveBmiToDatabase(double height, double weight, double bmi)
{
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";  //
Replace with your actual database name
        String user = "root";  // Replace with your MySQL username
        String password = "manu";  // Replace with your MySQL password

        String query = "INSERT INTO bmi_tracking (user_id, height, weight,
bmi, bmi_date) VALUES (?, ?, ?, ?, NOW())";

        try (Connection conn = DriverManager.getConnection(url, user,
password);
             PreparedStatement stmt = conn.prepareStatement(query)) {

            // Assuming a hardcoded user_id for now (you'll likely get this
from the logged-in user)
            int userId = 1; // Replace with actual user ID dynamically

            stmt.setInt(1, userId);
            stmt.setDouble(2, height);
            stmt.setDouble(3, weight);
            stmt.setDouble(4, bmi);
            stmt.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.5 FOOD TRACKING PAGE DESIGN

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class FoodTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Food Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for food type, quantity, and time
        Label foodLabel = new Label("Enter food item:");
        TextField foodField = new TextField();

        Label quantityLabel = new Label("Enter quantity (grams):");
        TextField quantityField = new TextField();

        Label timeLabel = new Label("Enter time of consumption:");
        TextField timeField = new TextField();

        // Button to log food
        Button logFoodButton = new Button("Log Food");
        logFoodButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");

        // Label to display the result
        Label resultLabel = new Label("Food entry will be displayed here.");
        resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Action for the "Log Food" button
        logFoodButton.setOnAction(e -> {
            String foodItem = foodField.getText();
            String quantity = quantityField.getText();
            String time = timeField.getText();

            if (!foodItem.isEmpty() && !quantity.isEmpty() &&
!time.isEmpty()) {
```

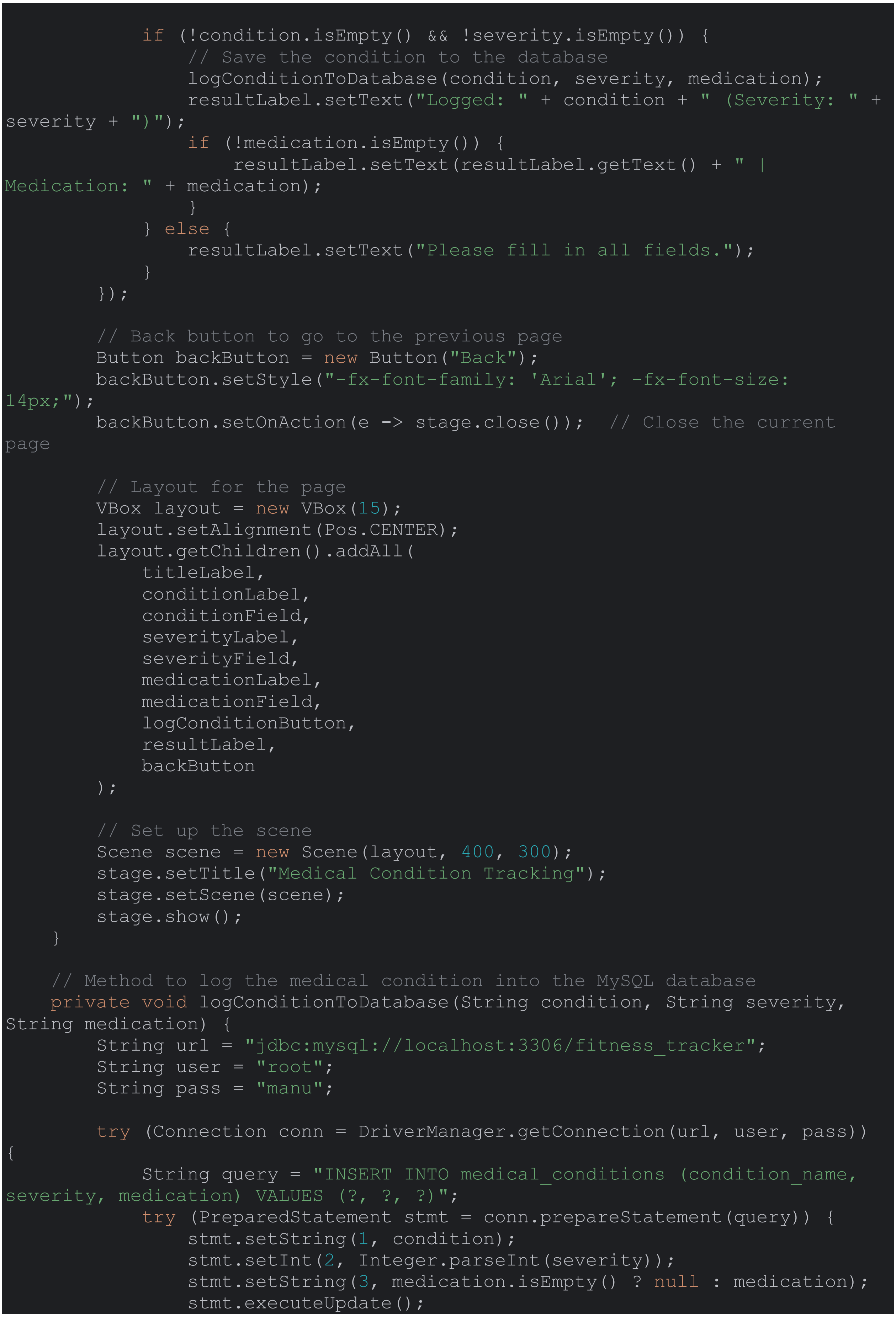


```
                // Save food entry to MySQL database
                saveFoodEntry(foodItem, Integer.parseInt(quantity), time);
                resultLabel.setText("Logged: " + foodItem + " (" + quantity +
"g) at " + time);
            } else {
                resultLabel.setText("Please fill in all fields.");
            }
        });

        // Back button to go to the previous page
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        backButton.setOnAction(e -> stage.close());  // Close the current
page, modify this as per your navigation logic

        // Layout for the page
        VBox layout = new VBox(15);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(
            titleLabel,
            foodLabel,
            foodField,
            quantityLabel,
            quantityField,
            timeLabel,
            timeField,
            logFoodButton,
            resultLabel,
            backButton
        );

        // Set up the scene
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("Food Tracking");
        stage.setScene(scene);
        stage.show();
    }

    private void saveFoodEntry(String foodItem, int quantity, String time) {
        // Database connection details
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";  //
Replace with your actual database name
        String user = "root";  // Replace with your MySQL username
        String password = "manu";  // Replace with your MySQL password

        String query = "INSERT INTO food_tracking (user_id, food_name,
quantity, consumption_time, date) VALUES (?, ?, ?, ?, CURDATE())";

        try (Connection conn = DriverManager.getConnection(url, user,
password);
             PreparedStatement stmt = conn.prepareStatement(query)) {

            // Assuming a hardcoded user_id for now (you'll likely get this
from the logged-in user)
            int userId = 1;

            stmt.setInt(1, userId);
            stmt.setString(2, foodItem);
            stmt.setInt(3, quantity);
            stmt.setString(4, time);  // Assuming time is in a valid format
```

```
(HH:mm)

            stmt.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.6 MEDICAL TRACKING PAGE DESIGN

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class MedicalConditionTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Medical Condition Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for condition name, severity, and medications
        Label conditionLabel = new Label("Enter medical condition:");
        TextField conditionField = new TextField();

        Label severityLabel = new Label("Enter severity (1-10):");
        TextField severityField = new TextField();

        Label medicationLabel = new Label("Enter medication (optional):");
        TextField medicationField = new TextField();

        // Button to log condition
        Button logConditionButton = new Button("Log Condition");
        logConditionButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 
14px;");

        // Label to display the result
        Label resultLabel = new Label("Condition entry will be displayed 
here.");
        resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Action for the "Log Condition" button
        logConditionButton.setOnAction(e -> {
            String condition = conditionField.getText();
            String severity = severityField.getText();
            String medication = medicationField.getText();
```

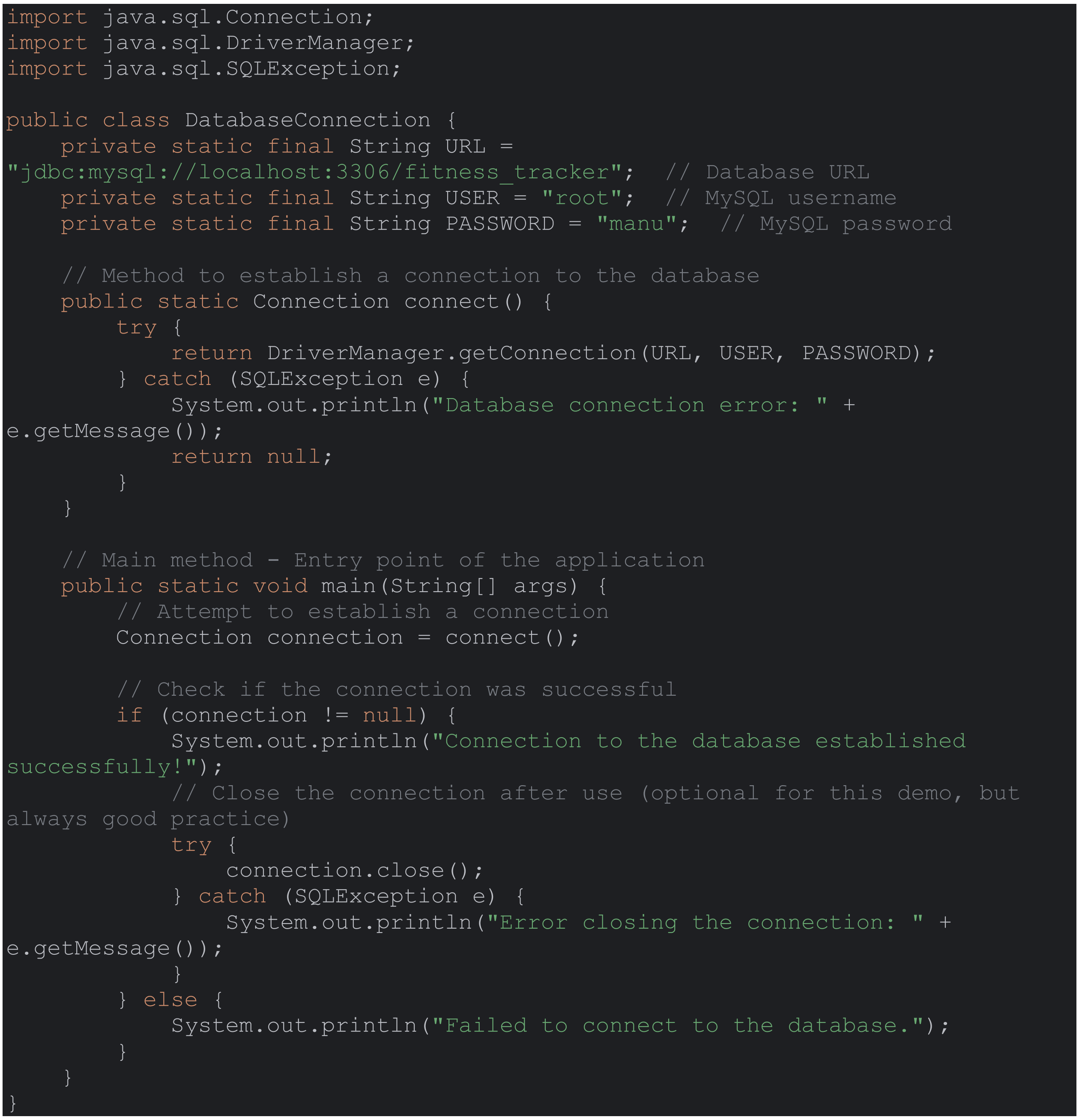


```
            if (!condition.isEmpty() && !severity.isEmpty()) {
                // Save the condition to the database
                logConditionToDatabase(condition, severity, medication);
                resultLabel.setText("Logged: " + condition + " (Severity: " +
severity + ")");
                if (!medication.isEmpty()) {
                    resultLabel.setText(resultLabel.getText() + " |
Medication: " + medication);
                }
            } else {
                resultLabel.setText("Please fill in all fields.");
            }
        });

        // Back button to go to the previous page
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        backButton.setOnAction(e -> stage.close());  // Close the current
page

        // Layout for the page
        VBox layout = new VBox(15);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(
            titleLabel,
            conditionLabel,
            conditionField,
            severityLabel,
            severityField,
            medicationLabel,
            medicationField,
            logConditionButton,
            resultLabel,
            backButton
        );

        // Set up the scene
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("Medical Condition Tracking");
        stage.setScene(scene);
        stage.show();
    }

    // Method to log the medical condition into the MySQL database
    private void logConditionToDatabase(String condition, String severity,
String medication) {
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";
        String user = "root";
        String pass = "manu";

        try (Connection conn = DriverManager.getConnection(url, user, pass))
{
            String query = "INSERT INTO medical_conditions (condition_name,
severity, medication) VALUES (?, ?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setString(1, condition);
                stmt.setInt(2, Integer.parseInt(severity));
                stmt.setString(3, medication.isEmpty() ? null : medication);
                stmt.executeUpdate();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

3.7 WATER TRACKING PAGE DESIGN

```
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class WaterTrackingPage extends Application {

    private ObservableList<WaterIntakeItem> data =
FXCollections.observableArrayList();

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Water Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Water intake entry field
        Label intakeLabel = new Label("Enter Water Intake (ml):");
        TextField intakeField = new TextField();

        // Button to log water intake
        Button logButton = new Button("Log Water");
        logButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

        // Label to show intake status
        Label statusLabel = new Label();
        statusLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Table for displaying water tracking history
        TableView<WaterIntakeItem> table = new TableView<>();
        TableColumn<WaterIntakeItem, String> dateColumn = new
TableColumn<>("Date");
        TableColumn<WaterIntakeItem, String> intakeColumn = new
TableColumn<>("Intake (ml)");

        // Adding columns to the table
        table.getColumns().add(dateColumn);
        table.getColumns().add(intakeColumn);
        // Load data from MySQL
        loadWaterData();

        // Setting items to the table
        table.setItems(data);

        // Action for the "Log Water" button
        logButton.setOnAction(e -> {
            String intake = intakeField.getText();
            if (!intake.isEmpty() && isNumeric(intake)) {
                // Save the water intake to the database
                logWaterIntake(intake);
                statusLabel.setText("Water logged: " + intake + " ml");
                loadWaterData(); // Reload data from the database
            } else {
                statusLabel.setText("Please enter a valid number.");
            }
        });

        // Back button to go to the previous page
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 
14px;");
        backButton.setOnAction(e -> stage.close()); // Close the current page

        // Layout for the page
        VBox layout = new VBox(15);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(
            titleLabel,
            intakeLabel,
            intakeField,
            logButton,
            statusLabel,
            table,
            backButton
        );

        // Set up the scene
        Scene scene = new Scene(layout, 500, 400);
        stage.setTitle("Water Tracking");
        stage.setScene(scene);
        stage.show();
    }

    // Helper method to check if the input is a number
    private boolean isNumeric(String str) {
        try {
            Integer.parseInt(str);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    }

    // Method to load water intake data from the database
    private void loadWaterData() {
        data.clear(); // Clear the existing data
        String url = "jdbc:mysql://localhost:3306/your_database";
        String user = "root";
        String pass = "your_password";
        try (Connection conn = DriverManager.getConnection(url, user, pass))
{
            String query = "SELECT date, intake FROM water_intake ORDER BY
date DESC";
            try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery(query)) {
                while (rs.next()) {
                    String date = rs.getString("date");
                    String intake = rs.getString("intake");
                    data.add(new WaterIntakeItem(date, intake));
                }
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    // Method to log water intake into the database
    private void logWaterIntake(String intake) {
        String url = "jdbc:mysql://localhost:3306/fitness_tracker";
        String user = "root";
        String pass = "manu";

        try (Connection conn = DriverManager.getConnection(url, user, pass))
{
            String query = "INSERT INTO water_intake (date, intake) VALUES
(CURDATE(), ?)";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setInt(1, Integer.parseInt(intake));
                stmt.executeUpdate();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }

    // Class for representing each water intake item
    public static class WaterIntakeItem {
        private String date;
        private String intake;

        public WaterIntakeItem(String date, String intake) {
            this.date = date;
            this.intake = intake;
        }

        public String getDate() {
            return date;
        }

        public String getIntake() {
            return intake;
        }
    }
}
```

3.10 DATABASE CONNECTIVITY

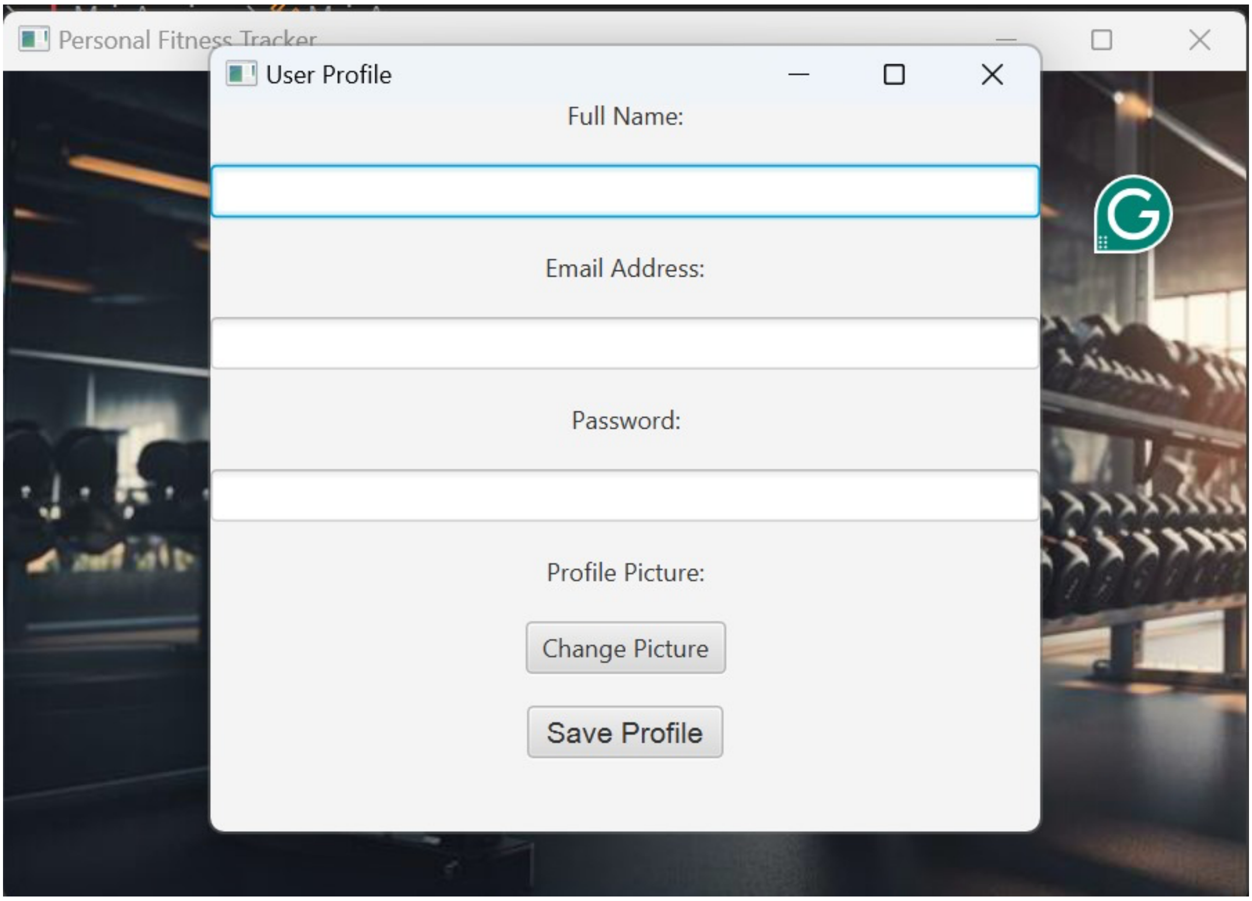


```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/fitness_tracker";  // Database URL
    private static final String USER = "root";  // MySQL username
    private static final String PASSWORD = "manu";  // MySQL password

    // Method to establish a connection to the database
    public static Connection connect() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            System.out.println("Database connection error: " +
e.getMessage());
            return null;
        }
    }

    // Main method - Entry point of the application
    public static void main(String[] args) {
        // Attempt to establish a connection
        Connection connection = connect();

        // Check if the connection was successful
        if (connection != null) {
            System.out.println("Connection to the database established
successfully!");
            // Close the connection after use (optional for this demo, but
always good practice)
            try {
                connection.close();
            } catch (SQLException e) {
                System.out.println("Error closing the connection: " +
e.getMessage());
            }
        } else {
            System.out.println("Failed to connect to the database.");
        }
    }
}
```

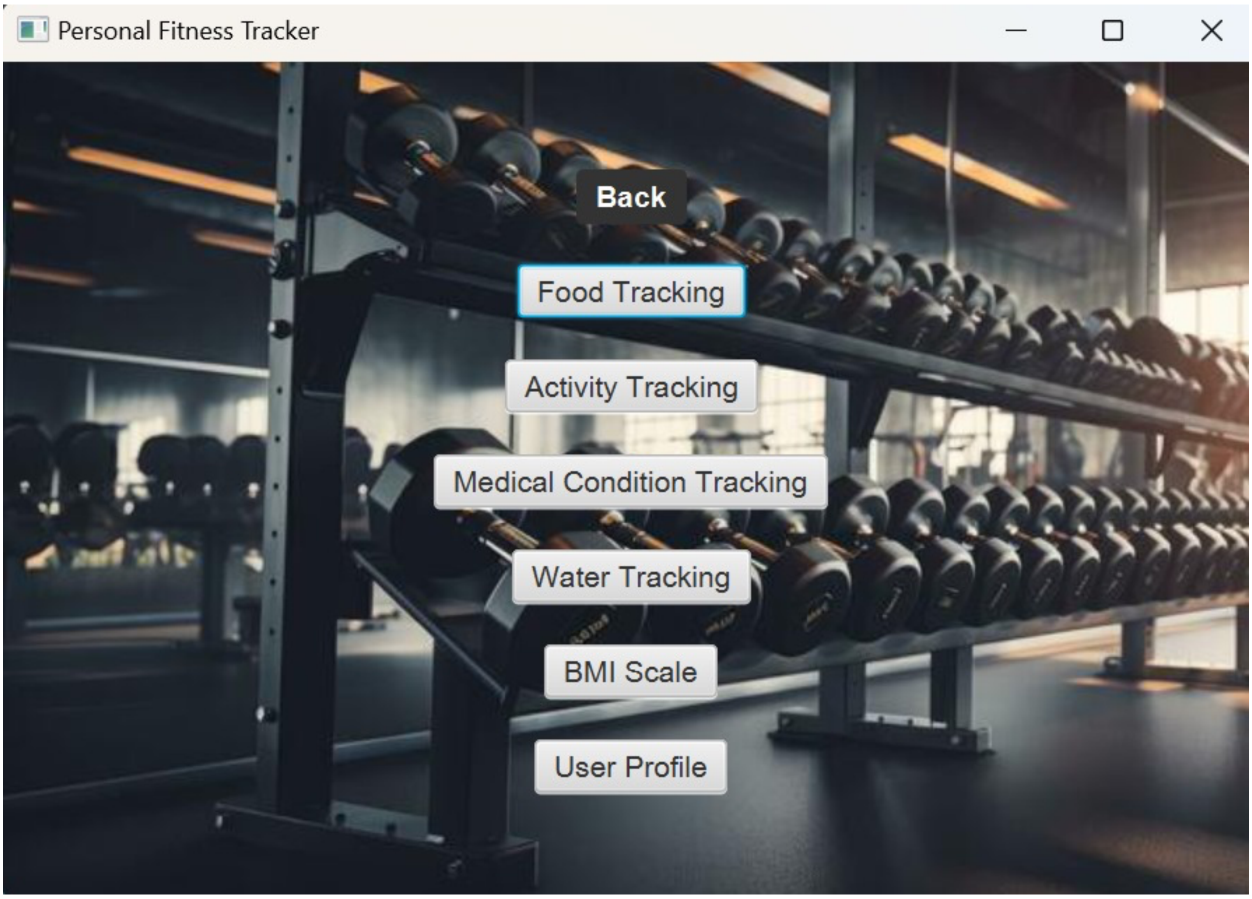
}

SNAPSHOTS

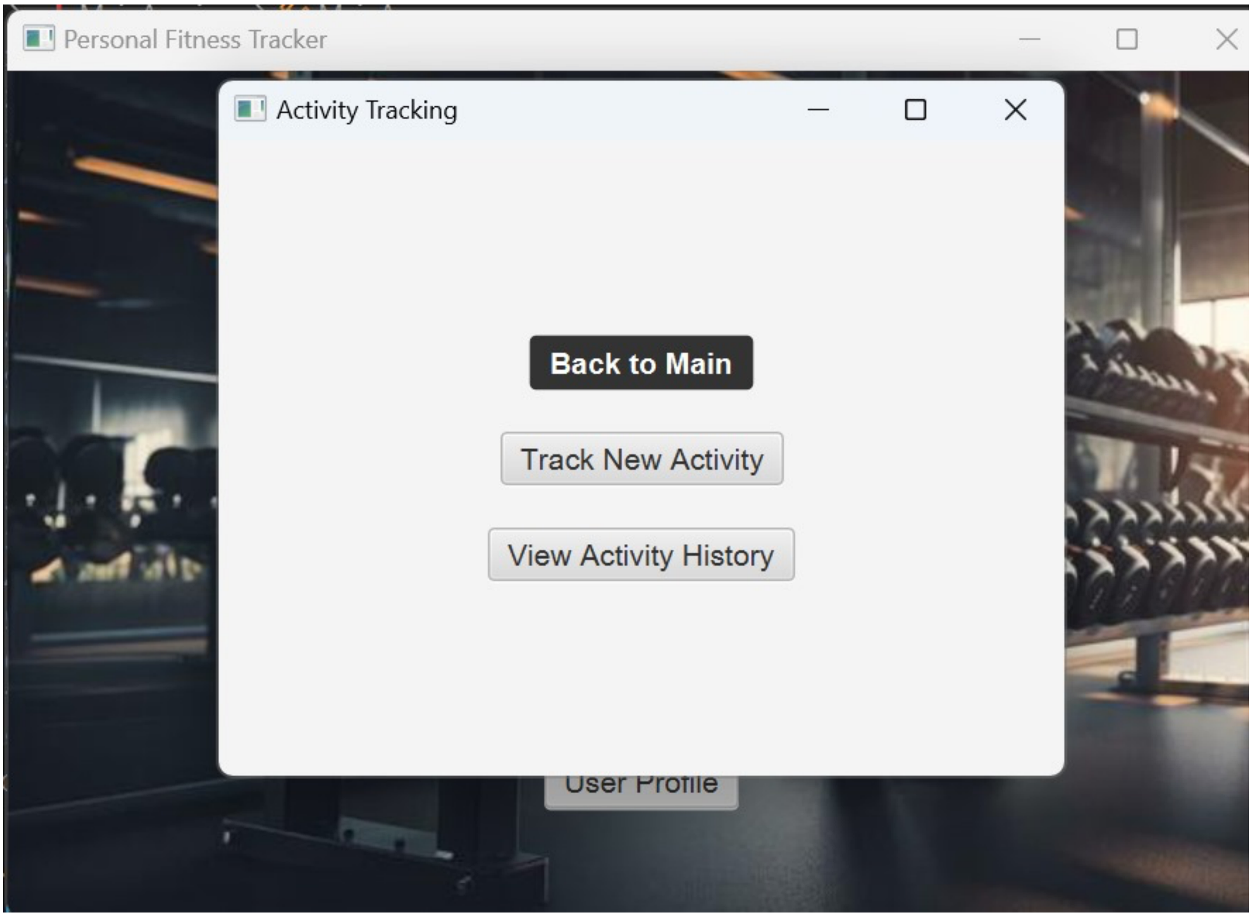
4.1 LOGIN PAGE:



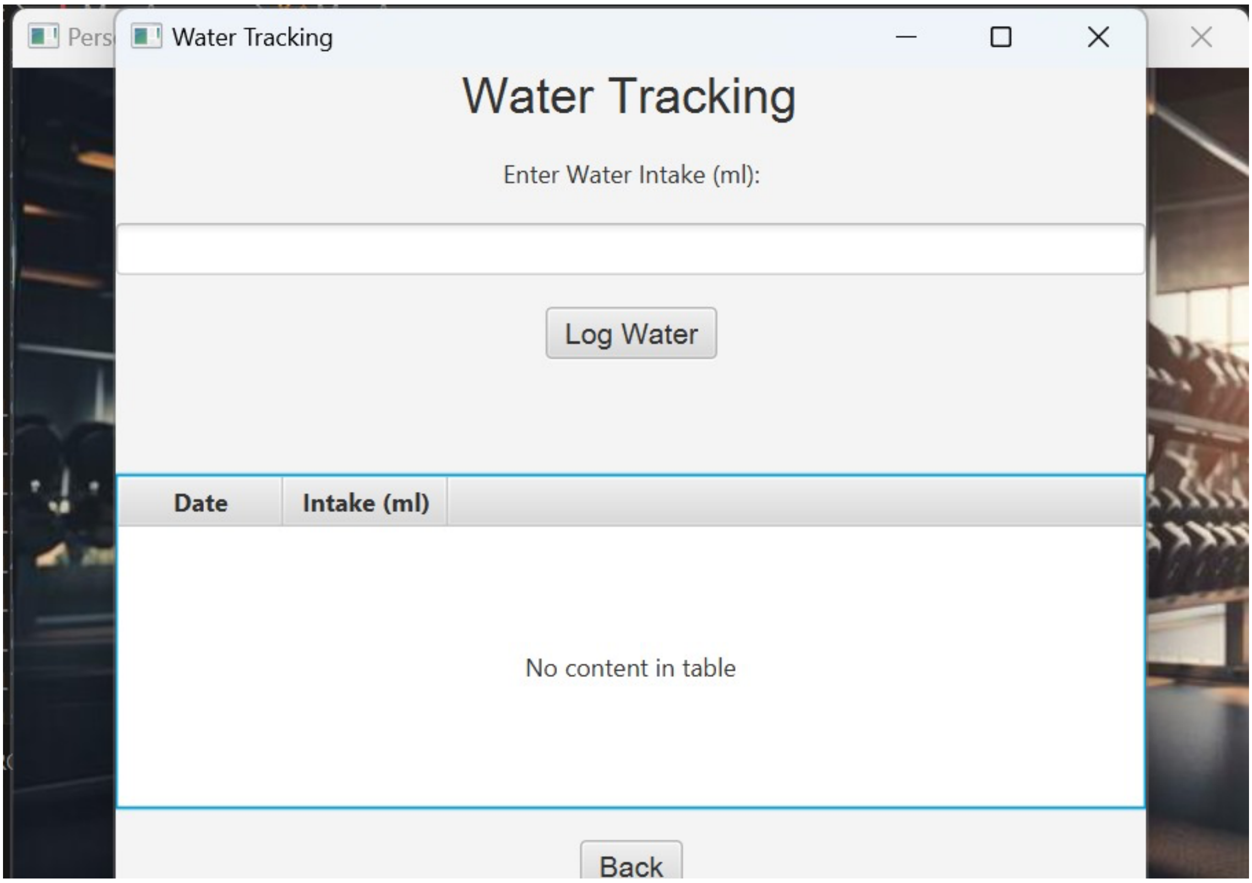
4.2 HOME PAGE:



4.3 ACTIVITY TRACKING PAGE:



4.4 WATER TRACKING PAGE:


Water Tracking

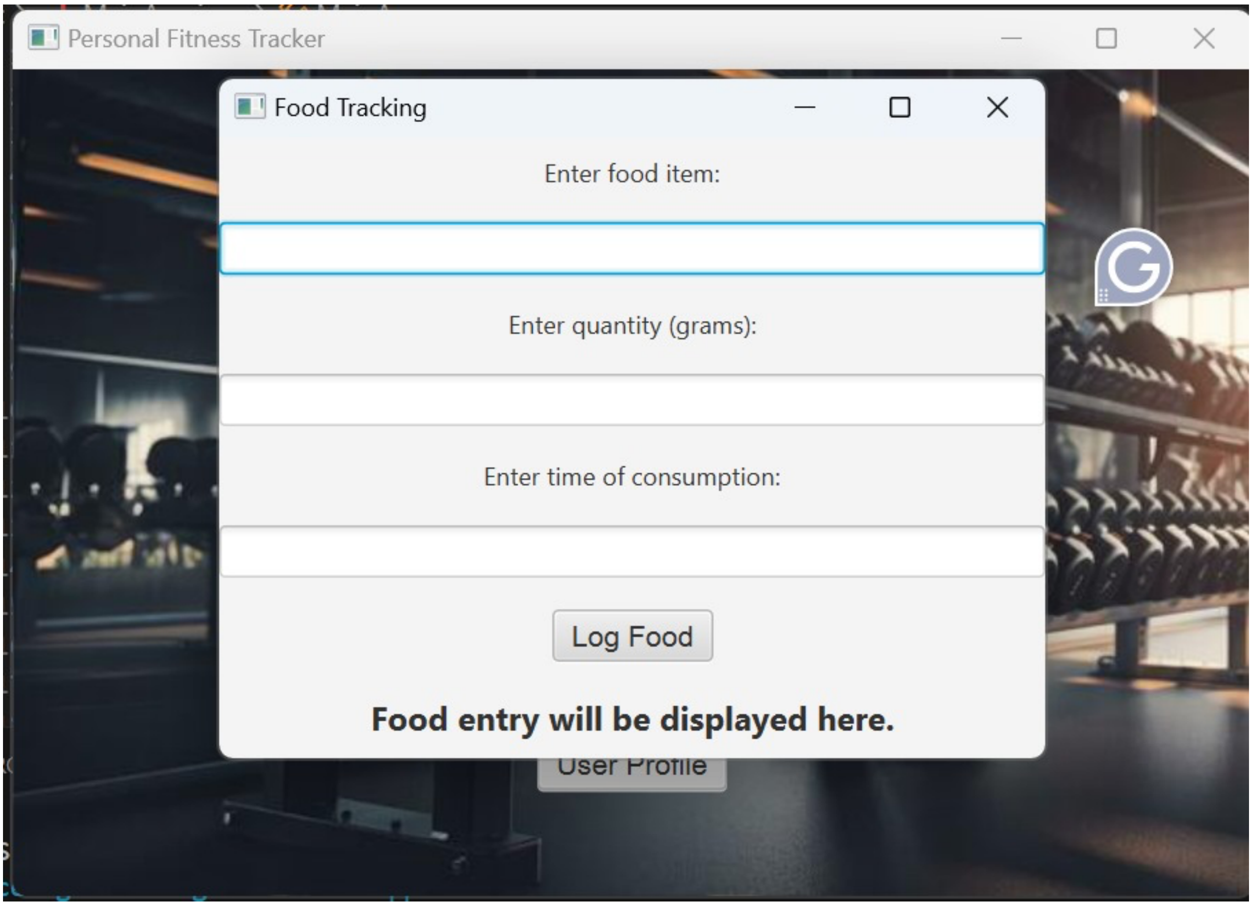
Water Tracking

Enter Water Intake (ml):

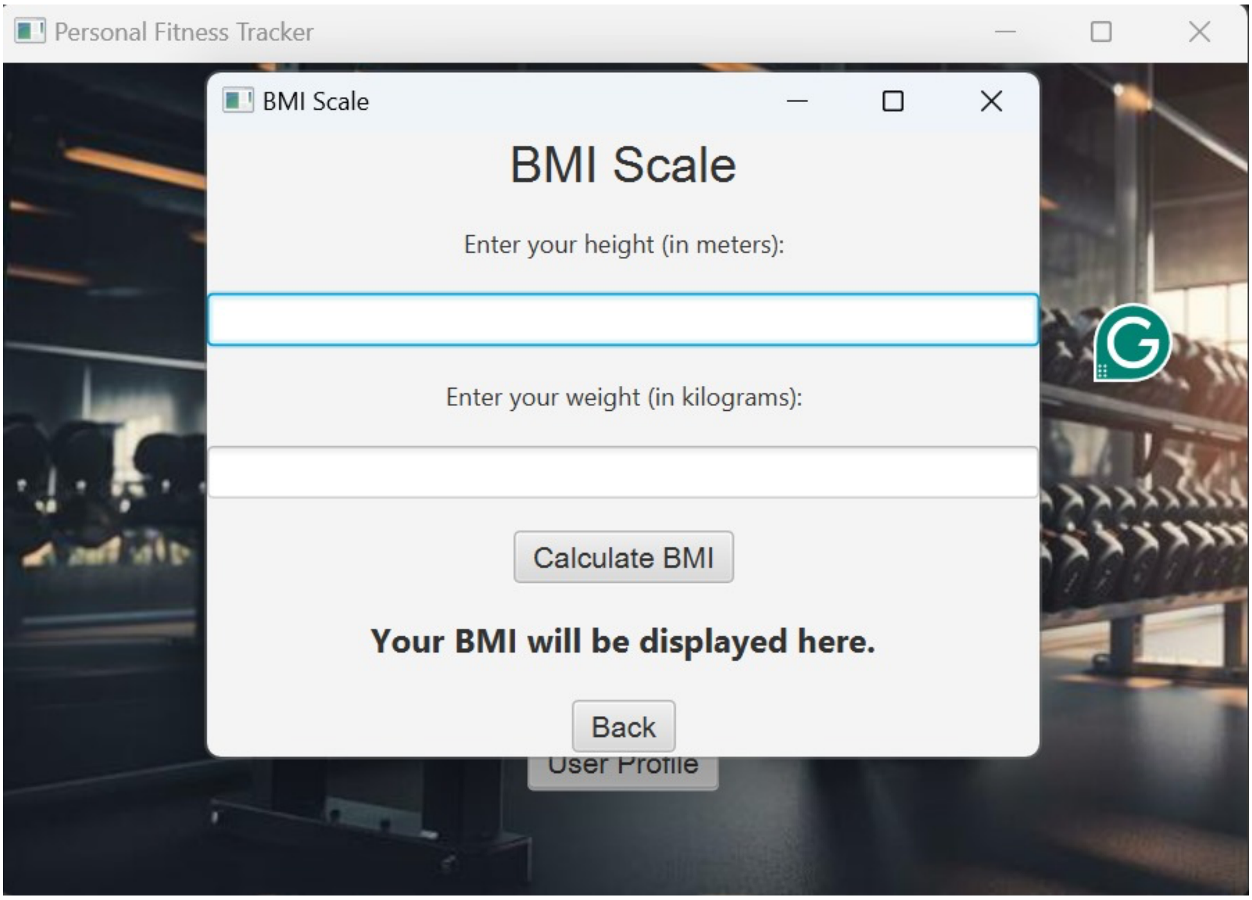
Date	Intake (ml)

No content in table

4.5 FOOD TRACKING PAGE:



4.6 BMI SCALE PAGE:



Personal Fitness Tracker

BMI Scale

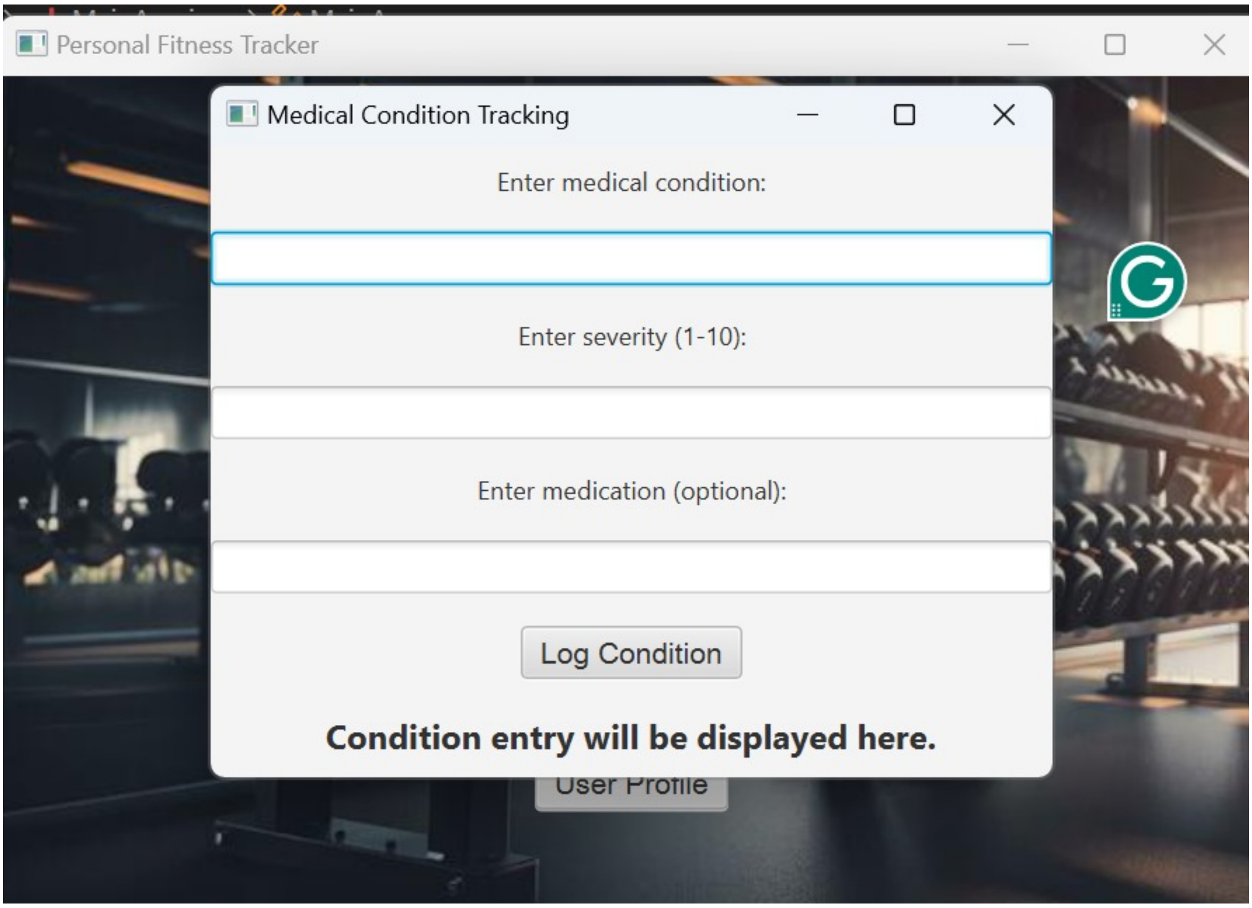
BMI Scale

Enter your height (in meters):

Enter your weight (in kilograms):

Your BMI will be displayed here.

4.7 MEDICAL CONDITION TRACKING PAGE:



Personal Fitness Tracker
Medical Condition Tracking
Enter medical condition:
Enter severity (1-10):
Enter medication (optional):
Log Condition
Condition entry will be displayed here.
User Profile

CONCLUSION:

The **Fitness Tracking Application** project was successfully developed, incorporating essential features such as food tracking, activity monitoring, and user profile management. Leveraging **JavaFX** for a visually engaging user interface and **MySQL** for secure data storage and management, the system ensures seamless, efficient, and user-friendly operations.

The project empowers users to monitor their health and fitness effectively while maintaining data integrity and offering scalability for future enhancements, such as integrating wearable device data or providing personalized fitness recommendations. Overall, this application serves as a robust solution for promoting a healthier lifestyle and simplifying fitness tracking for users.

REFERENCES:

1. https://openjfx.io
2. https://dev.mysql.com/doc/
3. [https://docs.oracle.com/javase/tutorial/jdbc/](https://docs.oracle.com/javase/ tutorial/jdbc/)
4. https://www.w3schools.com/sql/
5. https://stackoverflow.com/