**** 

**P.S.R ENGINEERING COLLEGE**

(An Autonomous Institution, Affiliated to Anna University, Chennai)

**Sevalpatti, Sivakasi - 626140.**

**Department Of**

**Computer Science and Engineering**

**Fitness Tracker in java**

**Presented by**

S. Mohamed Faiz (95192201302)

**Table of Contents**

1. Abstract
2. Project Overview
3. Objectives
4. Features
5. Technical Architecture
6. Database Schema
7. Appendix
8. Installation and Setup
9. Usage Instructions
10. Testing
11. Future Enhancements
12. Conclusion

**1)Abstract**

The Fitness Tracker application is a Java-based console application designed to assist users in monitoring and managing their fitness activities. With the increasing awareness of health and wellness, this application provides a user-friendly platform for individuals to track their workouts, calories burned, and personal health metrics.

The application features a robust registration system that allows users to create profiles by entering their name, age, weight, height, and gender. Users can log various workouts, detailing the type of exercise, duration, calories burned, steps taken, and distance covered. This data is stored in a MySQL database, ensuring secure and efficient data management.

The application employs a Model-View-Controller (MVC) architecture, separating the data handling, user interface, and business logic for improved maintainability and scalability. The console-based interface facilitates easy interaction, while the underlying Data Access Object (DAO) pattern ensures smooth communication with the database.

By providing users with the ability to view their personal details and workout history, the Fitness Tracker application empowers individuals to take charge of their fitness journey. Future enhancements, such as a graphical user interface and integration with fitness devices, are planned to further enrich the user experience and functionality of the application.

**2)Project Overview**

The Fitness Tracker application is a comprehensive Java-based console application developed to empower users in managing and optimizing their fitness journeys. In a world increasingly focused on health and wellness, this application serves as a vital resource for individuals seeking to monitor their physical activities, track their progress, and make informed decisions about their health.

**Purpose and Goals**

The primary purpose of the Fitness Tracker application is to provide users with an intuitive platform for tracking various fitness activities, including workouts, calories burned, and personal health metrics. By offering a structured way to log and analyse fitness data, the application aims to promote a healthier lifestyle and encourage users to set and achieve their fitness goals. The application is designed to cater to a diverse audience, from fitness enthusiasts to beginners, by providing essential tools that facilitate self-monitoring and accountability.

**3)Objectives**

**User -Friendly Interface:** Create an easy-to-use platform for tracking fitness activities.

**Secure Data Storage:** Safely store user information in a database.

**Workout Logging:** Enable users to record workouts and view their fitness history.

**Health Monitoring:** Make it easy for users to access their health data.

**4) Features**

**User Registration:** Users can register by providing their name, age, weight, height, and gender.

**Workout Logging:** Users can log their workouts, including type, duration, calories burned, steps taken, and distance covered.

**User Details Retrieval:** Users can view their personal details stored in the system.

**Workout History:** Users can view their workout history to track progress over time.

**Admin Login:** A simple admin login system to access the application.

**5)Technical Architecture**

The application follows a Model-View-Controller (MVC) architecture:

**Model:** Represents the data structure through the ‘User’ and ‘Workout’ classes.

**View:** The console interface for user interaction.

**Controller:** The DAO (Data Access Object) classes (‘User DAO’ and ‘WorkoutDAO’) handle database operations.

**Technologies Used**

**Programming Language:** Java

**Database:** MySQL

**JDBC Driver:** MySQL Connector/J

**Tools are used:**

1. Eclipse IDE for java code.
2. MySQL workbench for backend.

**6)Database Schema**

The application uses a MySQL database with the following schema:

**Create database**

CREATE DATABASE FitnessTracker;

USE FitnessTracker;

**Users Table**

CREATE TABLE Users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

age INT,

weight FLOAT,

height FLOAT,

gender ENUM('Male', 'Female', 'Other')

);

**Workouts Table**

CREATE TABLE Workouts (

workout\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

workout\_type VARCHAR(50),

duration INT, -- in minutes

calories\_burned FLOAT,

steps INT,

distance FLOAT, -- in kilometers

FOREIGN KEY (user\_id) REFERENCES Users(user\_id) ON DELETE CASCADE);

**7)Appendix**

**DatabaseConnection.java**

package com.fitness.tracker;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

private static final String URL = "jdbc:mysql://localhost:3306/fitnesstracker";

private static final String USER = "root"; // Change to your MySQL username

private static final String PASSWORD = "cse23"; // Replace with your MySQL password

public static Connection getConnection() {

try {

// Load MySQL JDBC Driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Establish Connection

Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);

System.out.println("Database Connected Successfully!");

return conn;

} catch (ClassNotFoundException e) {

System.out.println("MySQL Driver not found!");

e.printStackTrace();

return null;

} catch (SQLException e) {

System.out.println("Database Connection Failed!");

e.printStackTrace();

return null;

}

}

public static void main(String[] args) {

getConnection(); // Test database connection

}

}

**User .java:**

package com.fitness.tracker;

public class User {

private int id;

private String name;

private int age;

private float weight;

private float height;

private String gender;

public User(String name, int age, float weight, float height, String gender) {

this.name = name;

this.age = age;

this.weight = weight;

this.height = height;

this.gender = gender;

}

public User(int int1, String string, int int2, float float1, float float2, String string2) {

// TODO Auto-generated constructor stub

}

public String getName() {

// TODO Auto-generated method stub

return name;

}

public int getAge() {

// TODO Auto-generated method stub

return age;

}

public float getWeight() {

// TODO Auto-generated method stub

return weight;

}

public float getHeight() {

// TODO Auto-generated method stub

return height;

}

public String getGender() {

// TODO Auto-generated method stub

return gender;

}

public String getUserId() {

// TODO Auto-generated method stub

return null;

}

}

**User DAO.java**

package com.fitness.tracker;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class UserDAO {

// Method to insert a new user

public static boolean addUser(User user) {

String sql = "INSERT INTO Users (name, age, weight, height, gender) VALUES (?, ?, ?, ?, ?)";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, user.getName());

pstmt.setInt(2, user.getAge());

pstmt.setFloat(3, user.getWeight());

pstmt.setFloat(4, user.getHeight());

pstmt.setString(5, user.getGender());

int rowsAffected = pstmt.executeUpdate();

return rowsAffected > 0; // Returns true if insertion was successful

} catch (SQLException e) {

System.out.println("Error adding user: " + e.getMessage());

e.printStackTrace();

}

return false;

}

// Method to get user details by ID

public static User getUserById(int userId) {

String sql = "SELECT \* FROM Users WHERE user\_id = ?";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setInt(1, userId);

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

System.out.println("User ID: " + rs.getInt("user\_id"));

System.out.println("Name: " + rs.getString("name"));

System.out.println("Age: " + rs.getInt("age"));

System.out.println("Weight: " + rs.getFloat("weight") + "kg");

System.out.println("Height: " + rs.getInt("height") + "cm");

System.out.println("Gender: " + rs.getString("gender")); // ✅ FIXED

System.out.println("------------------------------------");

}

} catch (SQLException e) {

System.out.println("Error retrieving user: " + e.getMessage());

e.printStackTrace();

}

return null;

}

}

**Workout.java:**

package com.fitness.tracker;

public class Workout {

private int id;

private int userId;

private String workoutType;

private int duration;

private float caloriesBurned;

private int steps;

private float distance;

public Workout(int userId, String workoutType, int duration, double caloriesBurned, int steps, float distance) {

this.userId = userId;

this.workoutType = workoutType;

this.duration = duration;

this.caloriesBurned = (float) caloriesBurned;

this.steps = steps;

this.distance = distance;

}

// Getters and setters

public int getUserId() {

// TODO Auto-generated method stub

return userId;

}

public String getWorkoutType() {

// TODO Auto-generated method stub

return workoutType;

}

public int getDuration() {

// TODO Auto-generated method stub

return duration;

}

public float getCaloriesBurned() {

// TODO Auto-generated method stub

return caloriesBurned;

}

public int getSteps() {

// TODO Auto-generated method stub

return steps;

}

public float getDistance() {

// TODO Auto-generated method stub

return distance;

}

}

**WorkoutDAO.java:**

package com.fitness.tracker;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class WorkoutDAO {

public static void addWorkout(Workout workout) {

String query = "INSERT INTO Workouts (user\_id, workout\_type, duration, calories\_burned, steps, distance) VALUES (?, ?, ?, ?, ?, ?)";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, workout.getUserId());

stmt.setString(2, workout.getWorkoutType());

stmt.setInt(3, workout.getDuration());

stmt.setFloat(4, workout.getCaloriesBurned());

stmt.setInt(5, workout.getSteps());

stmt.setFloat(6, workout.getDistance());

stmt.executeUpdate();

System.out.println("Workout recorded successfully!");

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void getWorkoutsByUserId(int userId) {

String query = "SELECT \* FROM Workouts WHERE user\_id = ?";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, userId);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

System.out.println("Workout ID: " + rs.getInt("workout\_id"));

System.out.println("Workout Type: " + rs.getString("workout\_type"));

System.out.println("Duration: " + rs.getInt("duration") + " minutes");

System.out.println("Calories Burned: " + rs.getFloat("calories\_burned"));

System.out.println("Steps: " + rs.getInt("steps"));

System.out.println("Distance: " + rs.getFloat("distance") + " km");

System.out.println("------------------------------------");

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**FitnessTrackerApp.java:**

package com.fitness.tracker;

import java.util.Scanner;

public class FitnessTrackerApp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Username: ");

String username = scanner.nextLine();

System.out.print("Enter Password: ");

String password = scanner.nextLine();

if (username.equals("admin") && password.equals("password")) { //Improved login check

for(int i=0;i<=4;i++) {

System.out.println("Welcome to Fitness Tracker!");

System.out.println("1. Register User");

System.out.println("2. Log Workout");

System.out.println("3. View User Details");

System.out.println("4. View Workouts");

System.out.println("5.Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

System.out.print("Enter name: ");

String name = scanner.nextLine(); // Use nextLine() to avoid skipping

System.out.print("Enter age: ");

int age = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter weight (kg): ");

float weight = scanner.nextFloat();

scanner.nextLine();

System.out.print("Enter height (cm): ");

float height = scanner.nextFloat();

scanner.nextLine();

System.out.print("Enter gender (Male/Female/Other): ");

String gender = scanner.nextLine(); // Use nextLine() to capture full input

User user = new User(name, age, weight, height, gender);

UserDAO.addUser(user);

break;

case 2:

System.out.print("Enter user ID: ");

int userId = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter workout type: ");

String workoutType = scanner.nextLine(); // Use nextLine() for full input

System.out.print("Enter duration (minutes): ");

int duration = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter calories burned: ");

float calories = scanner.nextFloat();

scanner.nextLine();

System.out.print("Enter steps: ");

int steps = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter distance (km): ");

float distance = scanner.nextFloat();

scanner.nextLine();

Workout workout = new Workout(userId, workoutType, duration, calories, steps, distance);

WorkoutDAO.addWorkout(workout);

break;

case 3:

System.out.print("Enter user ID: ");

int uid = scanner.nextInt();

scanner.nextLine();

UserDAO.getUserById(uid);

break;

case 4:

System.out.print("Enter user ID: ");

int wid = scanner.nextInt();

scanner.nextLine();

WorkoutDAO.getWorkoutsByUserId(wid);

break;

case 5:

System.out.println("Exiting the application. Goodbye!");

scanner.close();

System.exit(0);

break;

default:

System.out.println("Invalid choice! Please enter a valid option.");

break;

}

}

} else {

System.out.println("Invalid Username or Password");

}

scanner.close();

}

}

**8)Installation and Setup**

**Prerequisites**

* Java Development Kit (JDK) installed (version 8 or higher).
* MySQL Server installed and running.
* MySQL Connector/J (JDBC driver) added to the project classpath.

**Database Setup**

1. Create a database named ‘FitnessTracker’.
2. Execute the provided SQL schema to create the necessary tables.

**Application Setup**

1. Clone or download the project files from the repository.
2. Update the ‘DatabaseConnection.java’ file with your MySQL username and password.
3. Compile and run the ‘FitnessTrackerApp.java’ file.

**9)Usage Instructions**

1. Run the application using a Java IDE or command line.
2. Log in using the admin credentials (username: ‘admin’, password: ‘password’).
3. Choose an option from the menu:

* Register a new user.
* Log a workout for an existing user.
* View user details by entering the user ID.
* View workout history for a specific user.
* Exit the application.

**10)Testing**

**Test Cases**

* **User Registration:** Verify that users can register successfully and that their data is stored in the database.
* **Workout Logging:** Ensure that workouts can be logged and that the data is accurately reflected in the database.
* **Data Retrieval:** Test the retrieval of user details and workout history to ensure accuracy.

**Testing Tools**

* Manual testing through the console interface.
* SQL queries to verify data integrity in the database.

**11) Future Enhancements**

* Implement user authentication and session management for enhanced security.
* Develop a graphical user interface (GUI) for improved user experience.
* Add features such as goal setting, reminders, and progress tracking.
* Integrate with fitness devices or APIs for automatic data logging.

**12)Conclusion**

The Fitness Tracker application serves as a foundational tool for individuals looking to monitor their fitness activities. With its simple design and functionality, it provides users with the ability to track their health metrics effectively. Future enhancements can further improve its usability and features, making it a comprehensive fitness tracking solution.