# Project Documentation: Fitness Tracking Application

## Project Overview

The Fitness Tracking Application is a console-based system designed to help users track their fitness activities, participate in challenges, and monitor progress. It features separate modules for users and administrators, each with distinct functionalities. The system is powered by a MySQL database to manage user data, workouts, challenges, and feedback efficiently.

## Features

### User Module

1. Add Workout: Users can log details of their workouts, including type, duration, calories burned, and date.  
2. View Workouts: Retrieve and display all workout entries for a user.  
3. Update Workout: Modify details of an existing workout.  
4. Delete Workout: Remove a workout entry from the system.  
5. View Progress: Analyze fitness progress over time.  
6. Show Challenges: Display available fitness challenges.  
7. Join Challenges: Enroll in specific challenges.  
8. Display Challenge History: View challenges previously joined by the user.  
9. Check for Reviews: Read feedback or reviews from administrators.  
10. Exit: Safely exit the application.

### Admin Module

1. Create User: Add new users to the system with basic details like name, email, and password.  
2. Update User: Modify user information, including name, email, or password.  
3. Delete User: Remove a user and their associated data.  
4. Add Challenge: Introduce new fitness challenges for users.  
5. View Fitness Content: Display all logged workouts for administrative analysis.  
6. Add Reviews for Users: Provide feedback or motivational reviews for specific users.  
7. Update System Settings: Modify global application settings.  
8. Exit: Exit the admin dashboard.

## System Architecture

### Backend

- Programming Language: Java  
- Database: MySQL  
- Libraries: JDBC for database interaction

### Database Schema

The system uses the `fitness\_tracker` database, consisting of the following tables:  
  
1. Users Table: Manages user accounts.  
 - Columns: id, name, email, password  
2. Workouts Table: Stores details of user workouts.  
 - Columns: id, user\_id, workout\_type, duration, calories\_burned, Date  
3. Challenges Table: Holds information about fitness challenges.  
 - Columns: id, name, description  
4. UserChallenges Table: Tracks user participation in challenges.  
 - Columns: id, user\_id, challenge\_id, joined\_at  
5. SystemSettings Table: Stores application-wide settings.  
 - Columns: id, setting\_name, setting\_value  
6. Feedback Table: Records feedback provided by administrators for users.  
 - Columns: id, user\_id, feedback

## Code Modules

### User Application (User Class)

The user interface manages workout entries, challenge participation, and progress tracking.  
Key functionalities are implemented through methods provided by the UserDAO interface and its implementation class (UserDAOImpl).  
  
Exception Handling:  
- Input validation for numbers and dates.  
- Catching SQL and runtime exceptions to ensure smooth operation.

### Admin Application (AdminApp Class)

The admin dashboard allows management of users, challenges, feedback, and settings.  
Key functionalities are implemented using the AdminDAO interface and its implementation class (AdminDAOImpl).  
  
Exception Handling:  
- Validation for user inputs (e.g., email format, password length).  
- Catching database errors during CRUD operations.

## Database Creation Script

CREATE DATABASE fitness\_tracker;  
USE fitness\_tracker;  
  
CREATE TABLE Users (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(100),  
 email VARCHAR(100) UNIQUE,  
 password VARCHAR(100)  
);  
  
CREATE TABLE Workouts (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 user\_id INT,  
 workout\_type VARCHAR(100),  
 duration INT,  
 calories\_burned INT,  
 Date DATE,  
 FOREIGN KEY (user\_id) REFERENCES Users(id)  
);  
  
CREATE TABLE Challenges (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(100),  
 description TEXT  
);  
  
CREATE TABLE UserChallenges (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 user\_id INT,  
 challenge\_id INT,  
 joined\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (user\_id) REFERENCES Users(id),  
 FOREIGN KEY (challenge\_id) REFERENCES Challenges(id)  
);  
  
CREATE TABLE SystemSettings (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 setting\_name VARCHAR(255),  
 setting\_value VARCHAR(255)  
);  
  
CREATE TABLE Feedback (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 user\_id INT,  
 feedback TEXT,  
 FOREIGN KEY (user\_id) REFERENCES Users(id)  
);

## How to Run the Application

### Prerequisites

- Install Java JDK (version 8 or higher).  
- Install MySQL and create the fitness\_tracker database using the provided script.  
- Configure the DatabaseConnection class with the correct database credentials.

### Steps

1. Compile the Java code:  
 javac User.java AdminApp.java  
2. Run the user application:  
 java User  
3. Run the admin application:  
 java AdminApp

## Authors

Ayush Rawat, Harsh Vardhan Singh, Abhishek Sharma