**Fitness Tracker Application Project Documentation**

**Project Overview**

The Fitness Tracker Application is an online platform designed to help users monitor their workouts, track progress, participate in challenges, and engage with reviews. This project leverages Java and MySQL, following modular coding principles to ensure scalability and maintainability.

**Key Features**

**User Features:**

1. **Workout Management:**
   * Add workouts (type, duration, calories, date).
   * View all workouts.
   * Update or delete workouts.
2. **Progress Tracking:**
   * View overall fitness progress based on recorded workouts.
3. **Challenges:**
   * Show available fitness challenges.
   * Join challenges and view challenge history.
4. **Review Management:**
   * View feedback and reviews related to user performance.

**Admin Features:**

1. **User Management:**
   * Create, update, and delete user accounts.
2. **Challenge Management:**
   * Add, modify, or remove fitness challenges.
3. **View Feedback:**
   * Access and manage user reviews.

**System Architecture**

**Modules:**

1. **Database Layer:**
   * Handles interactions with the MySQL database.
   * Tables: users, workouts, challenges, reviews.
2. **DAO Layer:**
   * Provides data access operations for entities like User, Workout, and Challenge.
3. **Service Layer:**
   * Implements business logic for admin and user functionalities.
4. **UI Layer:**
   * Console-based interface for users and admins.

**Technologies Used:**

1. **Programming Language:** Java
2. **Database:** MySQL
3. **IDE:** Eclipse
4. **Build Tool:** Maven (optional for dependency management)

**Class Design**

**1. FitnessApp**

* Entry point of the application.
* Handles admin and user login/sign-up processes.

**2. AdminApp**

* Provides admin-specific functionalities such as managing users and challenges.

**3. User**

* Manages user-specific operations, including workouts, challenges, and reviews.

**4. DAO Interfaces:**

* **UserDAO:** Methods for user-related operations.
* **WorkoutDAO:** Methods for workout management.
* **ChallengeDAO:** Methods for challenges.

**5. DAO Implementations:**

* **UserDAOImpl:** Implements UserDAO with CRUD operations.
* **WorkoutDAOImpl:** Implements WorkoutDAO for workout-specific tasks.
* **ChallengeDAOImpl:** Implements ChallengeDAO for challenge-related operations.

**6. Utility Classes:**

* **DatabaseConnection:** Manages database connections.
* **ValidationUtil:** Provides utility methods for input validation.

**Database Schema**

**1. users Table:**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| Id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| Name | VARCHAR(50) | NOT NULL |
| Email | VARCHAR(50) | UNIQUE, NOT NULL |
| Password | VARCHAR(50) | NOT NULL |

**2. workouts Table:**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| user\_id | INT | FOREIGN KEY (users.id) |
| type | VARCHAR(50) | NOT NULL |
| duration | INT | NOT NULL |
| calories | INT | NOT NULL |
| date | DATE | NOT NULL |

**3. challenges Table:**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| name | VARCHAR(50) | NOT NULL |
| description | TEXT | NOT NULL |
| start\_date | DATE | NOT NULL |
| end\_date | DATE | NOT NULL |

**4. reviews Table:**

| **Column Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT |
| user\_id | INT | FOREIGN KEY (users.id) |
| review | TEXT | NOT NULL |

**Execution Steps**

**Prerequisites:**

1. Install Java (JDK 8 or higher) and set up the environment variables.
2. Install MySQL and create the required database schema.
3. Install Eclipse IDE.

**Setup:**

1. Clone the project repository or create the folder structure as described.
2. Import the project into Eclipse:
   * Open Eclipse and select **File > Import > Existing Projects into Workspace**.
3. Configure the database connection in DatabaseConnection.java:
4. private static final String URL = "jdbc:mysql://localhost:3306/fitness\_tracker";
5. private static final String USER = "root";
6. private static final String PASSWORD = "your\_password";
7. Create the database schema using the SQL script provided.

**Run the Project:**

1. Compile and run the FitnessApp class.
2. Choose between admin or user functionality based on your role.

**Future Enhancements**

1. **Web-Based Interface:** Transition from console-based to a web application using frameworks like Spring Boot.
2. **Mobile App Integration:** Develop Android/iOS applications.
3. **Enhanced Security:** Implement encryption for sensitive data like passwords.
4. **Gamification:** Add badges and rewards for achieving fitness milestones.
5. **Analytics:** Provide detailed insights and recommendations based on user data.

**Contributors**

1. **Divyansh** - Lead Developer
2. **Team Members** - Aryan Gupta, Raghav Dixit, Sudhir Gupta

**Conclusion**

The Fitness Tracker Application offers a comprehensive platform for fitness enthusiasts to manage their workouts, track progress, and participate in challenges. This documentation provides a detailed guide for setting up, executing, and understanding the system's functionality. By following the modular approach, this project is designed for easy scalability and future enhancements.