Fitness Tracking Application Documentation

Outline

I built a fitness tracking web app that lets users record and monitor their running achievements. The app uses Node.js with Express on the backend and MySQL for the database. Users can sign up, log in, and track things like how far they ran and how long it took. I made sure to implement proper security with bcrypt for password hashing and used express-session to keep users logged in. The frontend is built with EJS templates and some basic CSS styling. The app supports multiple users, so everyone can keep track of their own runs while also being able to search through and compare achievements with other users in the database.

Architecture

The app follows a basic three-tier structure. The frontend uses EJS views with CSS for styling. The backend runs on Express.js which handles all the routing and middleware. MySQL stores all the data. When a user makes a request, it goes through Express middleware like body-parser and session handling, then hits the appropriate route handler which talks to MySQL and sends back the rendered page.

Technologies Used

For the backend, I used Node.js as the runtime with Express 5.2.1 for handling routes and middleware. Express-session (version 1.18.2) manages user sessions and bcrypt 6.0.0 handles password encryption. The database is MySQL, accessed through the mysql2 driver (version 3.16.0) which supports connection pooling.

Data Model

I set up two tables in MySQL. The 'register' table holds user account info like id, username, firstName, lastName, email, and hashedPassword, with id as the primary key. The 'achievements' table stores the fitness data with id, username, distrun (distance run), and speed (time taken). The id is the primary key and username links back to the register table.

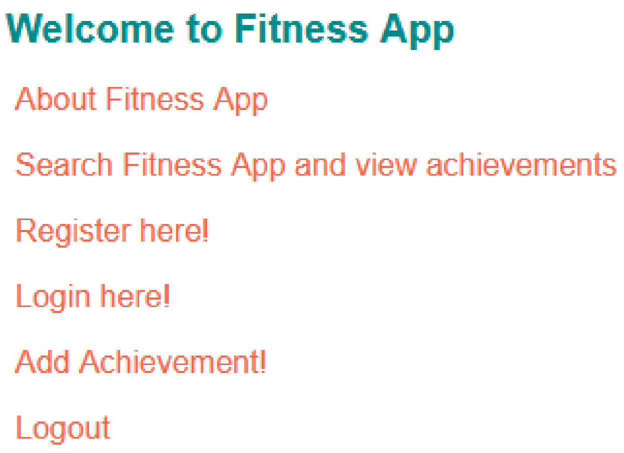
Description

The database design keeps user login info separate from their achievement data. The register table handles all the authentication stuff with hashed passwords for security. The achievements table records running stats - distance is stored as an integer (in meters) and time uses the TIME format. Username acts like a foreign key linking the tables together, so each user can have loads of achievements stored while keeping everything organized.

User Functionality

1. Home Page

The home page is pretty straightforward - it shows the app name and has links to all the main features. Users can navigate to registration, login, add achievements, search through the database, and check out the about page. I styled it with darkcyan headings and orangered links that turn white when you hover over them.



Welcome to Fitness App

About Fitness App

Search Fitness App and view achievements

Register here!

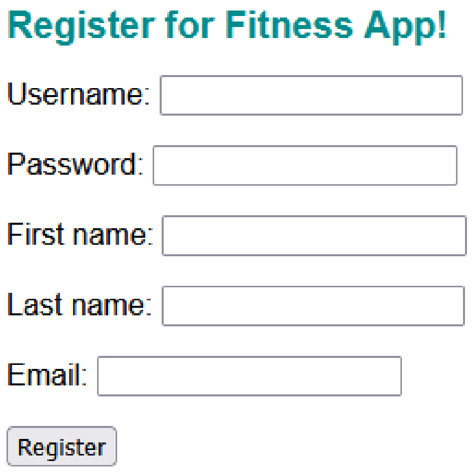
Login here!

Add Achievement!

Logout

2. User Registration

New users fill out a form with their username, first and last name, email, and password. When they submit it, the password gets hashed with bcrypt (using 10 salt rounds) before being stored in the database. After successfully registering, they get a confirmation message and a link to log in.



Register for Fitness App!

Username:

Password:

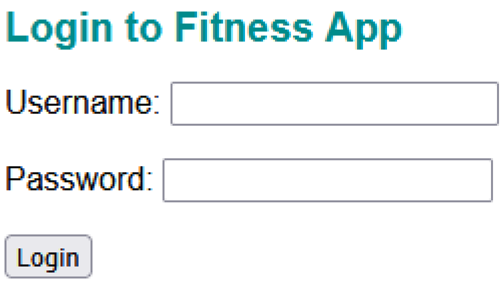
First name:

Last name:

Email:

3. User Login

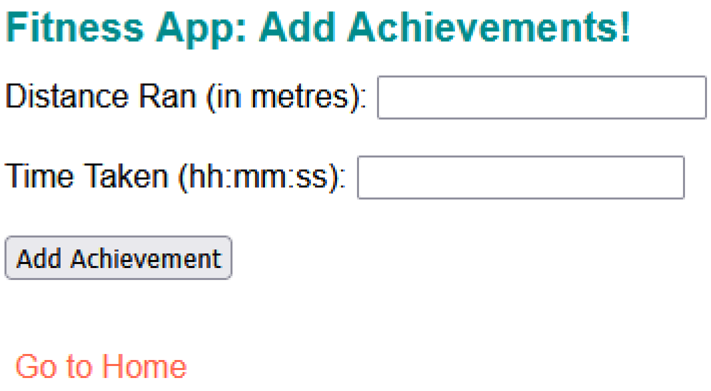
For login, users enter their username and password. The system checks the password against the hashed version in the database using bcrypt's compare function. If it matches, a session gets created that lasts 10 minutes and they're redirected to the add achievement page. If not, they see an error message.



Login to Fitness App

4. Add Achievement (Protected Route)

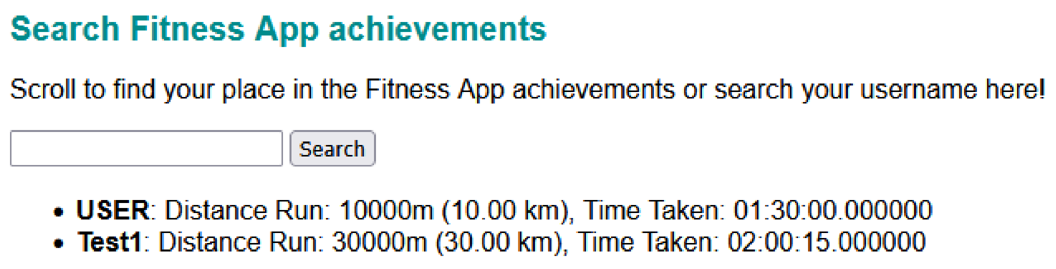
Once logged in, users can add their running achievements. They input the distance they ran in meters and the time in hh:mm:ss format. This page is protected by the redirectLogin middleware, so you have to be logged in to access it. The achievement gets saved to the database with the user's session ID.



Fitness App: Add Achievements!

5. Search and View Achievements

The search page shows all achievements in the database. You can scroll through the list which displays each user's username, distance run (in both meters and kilometers), and their time. There's a search box where you can type in a username to filter the results. The search uses SQL's LIKE operator for partial matches, so you don't need to type the exact username.



Search Fitness App achievements

Scroll to find your place in the Fitness App achievements or search your username here!

- **USER**: Distance Run: 10000m (10.00 km), Time Taken: 01:30:00.000000
- **Test1**: Distance Run: 30000m (30.00 km), Time Taken: 02:00:15.000000

6. Logout

When users logout, the session gets destroyed and the session cookie is cleared. This stops anyone from accessing their account after they've logged out. After logging out, they see a confirmation message with a link back to the home page.

AI Declaration

I used Claude AI to help write this documentation. I gave it my code, database schema, and screenshots, and it analyzed everything to create the documentation following the required format and word limits. All the technical explanations are based on my actual code.