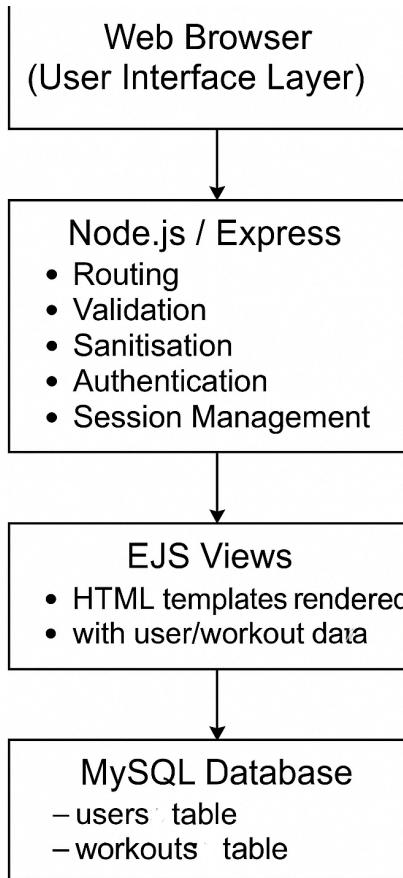


Outline

The Fitness Tracker app assists you in tracking workouts. Just sign up and log in to keep your data safe. Now you can add your workout logging what you did, how long, how hard, how many calories burnt and any notes you want to add. All activity is stored in a MySQL database so that you can track your development over time. You can search your workout history and see stats like total workout time and total calories burned. All in all, it's an easy and safe fitness logging application which is built with Node.js, Express, EJS, and MySQL.

Architecture

The application uses a two-tier architecture. The application tier uses Node.js with Express for routing, business logic, validation, sanitisation, authentication, and session management. EJS templates are used to render the user interface. The MySQL database is the data tier that captures user accounts and workout information. The server uses connection pooling through MySQL2 to access the database. Every component adopts the same structure and design as used in the lab exercises.



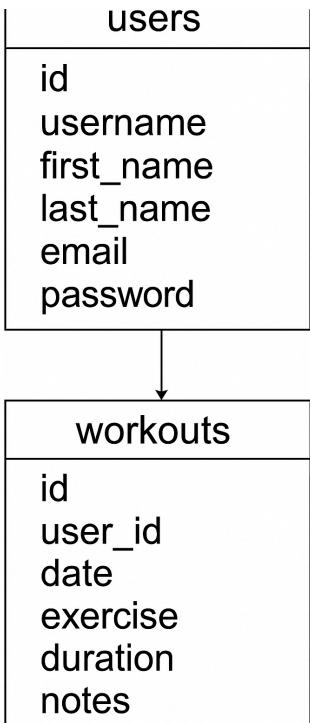
Data Model

The data model consists of two main tables.

Users secure their user profile information and hash passwords.

workouts, which keeps track of each workout entry with a link to the username that created it.

The workouts table includes the name of the workout, the duration, intensity, calories burned, notes and the time stamp. Each user can store their own workout history privately, while search queries and statistic queries are supported.



User Functionality

The user receives access to all fitness-tracking features after registration or login. On the Add Fitness Workflow page, users can enter the fitness name, duration, intensity, calories, and additional notes. The entry is approved, cleaned, kept in the data store and acknowledged to the user.

The logged-in user can see his/her own workouts on the Workout History page. The Search page helps users find workouts by name quickly using the keyword filter. The Statistics page gives a summary of user activity indicating total minutes exercised and total calories burned. These are calculated from all workouts.

Public pages include the Home and About pages. All the pages related to workouts use session-based access control and redirects the unauthenticated visitors to the Login page. Another API is available at /api/workouts that returns all the workouts in a JSON format. This is very similar to the labs' examples of API development.

Add screenshots for:

- Home page
- Add Workout
- Workout List

- Search page
- Statistics page

Advanced Techniques

I have hashed the password using bcrypt to secure the user's password. To prevent XSS attack, all free texts including username, workout name, notes and more are sanitized with express-sanitizer. Custom input validation was accomplished through express-validator, allowing us to impose minimum length and number constraints.

The workout pages were protected using a custom middleware redirectLogin to facilitate session-based Authentication using express-session. I made a JSON API endpoint which gives workout data in a structured manner. I used this technique for an extra credit not required by lab.

Below is an example code snippet illustrating sanitisation and validation when adding workouts:

```
let newRecord = [
    req.session.userId,
    req.sanitize(req.body.workout_name),
    req.body.duration,
    req.body.intensity,
    req.body.calories,
    req.sanitize(req.body.notes)
];
```

This shows the integration of session authentication, sanitisation, and database interaction in one route.

AI Declaration

I made very limited use of AI while completing this assignment. AI assistance was used only for small tasks such as clarifying error messages and checking minor syntax issues. All design decisions, implementation details, database structure, routes, validation logic, and page layouts were created and written by me using the old labs. No AI tool wrote any substantial part of the application, and all code included in the final submission was manually produced, tested, and verified by me. The project represents my own understanding and work.