Alexander Younis

Workout Tracker – Project Plan

The Workout Tracker is a small full-stack project I built to practice creating a working CRUD application using Node.js, Express, MongoDB, and EJS. The main idea behind the app is to give users a simple way to log and manage their workouts. Each workout includes a title, category, duration, date, and some optional notes. The project helped me understand how the backend, database, and front-end templating system all connect together in a real web application.

Project Goals

The main goals of the project were to let users add new workouts, view all existing workouts in a clean table, edit workouts with pre-filled values, and delete workouts with a confirmation pop-up. I wanted the design to stay simple and easy to navigate, so I used Bootstrap and a shared EJS layout to keep everything consistent. Another major part of the project was deploying it online, so I hosted the app on Render and connected it to a MongoDB Atlas cluster using environment variables.

Technologies Used

Backend: Node.js + Express
Frontend: EJS templates + Bootstrap
Database: MongoDB (Atlas) using Mongoose
Other: dotenv, method-override, express-ejs-layouts
Hosting: Render
Version Control: Git + GitHub

Deployment

Push project to GitHub.
Create a Render web service.
Add MONGO_URL in Render's environment variables.
Make sure npm start runs the app.
Test CRUD features on the live deployed site.

Future Improvements

Add user accounts/login.
Add charts to track workout progress.
Add filters/search options.
Improve the mobile layout or turn it into a PWA.

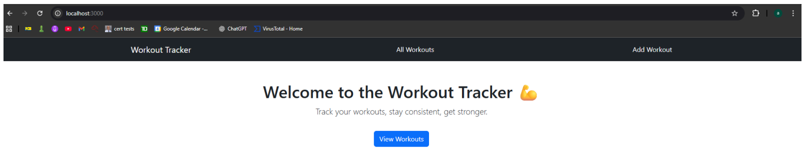
Summary

Overall, this project is a straightforward CRUD application that shows how to connect Express, MongoDB, and EJS into a full working website. It highlights core skills like routing, templating, database operations, and deployment, and it serves as a good foundation for more advanced fitness-tracking features in the future.

Site Details:

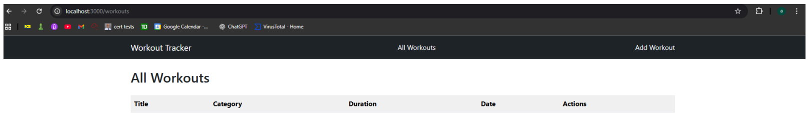
Link: https://assignment-03-9p8f.onrender.com/

Github Repo: https://github.com/mav-21/Assignment-03



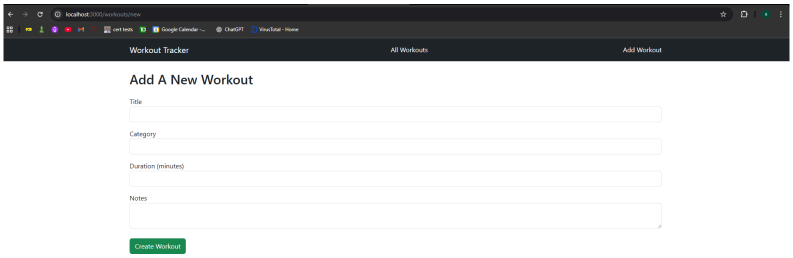
Welcome to the Workout Tracker 💪

Track your workouts, stay consistent, get stronger.



All Workouts

Title	Category	Duration	Date	Actions



Add A New Workout

Title

Category

Duration (minutes)

Notes

Citations

1. Bootstrap Team. Bootstrap 5 Documentation. https://getbootstrap.com/docs/5.3/getting-started/introduction/
2. MongoDB, Inc. MongoDB Atlas: Connection String Instructions. https://www.mongodb.com/docs/atlas/
3. Express.js Team. Getting Started with Express. https://expressjs.com/en/starter/installing.html
4. Mongoose Team. Mongoose Quick Start Guide. https://mongoosejs.com/docs/
5. MDN Web Docs. HTML Forms Overview. https://developer.mozilla.org/en-US/docs/Learn/Forms
6. MDN Web Docs. HTTP Request Methods Explained. https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods
7. Node.js Foundation. Node.js File Structure Best Practices. https://nodejs.org/en/learn
8. EJS Templating Team. EJS Guide. https://ejs.co/#docs
9. Render.com. Deploying a Node.js Service. https://render.com/docs/deploy-node-express-app
10. GitHub Docs. Using .gitignore to Protect Sensitive Files. https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files
11. dotenv Team. Environment Variable Management. https://github.com/motdotla/dotenv
12. W3C. CSS Basics and Selectors. https://www.w3.org/TR/selectors-4/
13. Stack Overflow User: various posts referenced for debugging Mongoose connection errors (e.g., ENOTFOUND and SRV lookup issues).
14. DigitalOcean Tutorials. How To Structure Node.js Apps with MVC. https://www.digitalocean.com/community/tutorials
15. Heroku Dev Center. Understanding HTTP Routing in Node. https://devcenter.heroku.com/articles/nodejs-support