

Monica Um
Raquel Zuelke
Jeremey Jun
Flavio Cervantes

Final Project Plan: Fitness Tracker

1. Title of the Project

Fitness Tracker

2. Project Description

Fitness Tracker is a dashboard application that allows users to track the fitness journey by maintaining a historical diary of the workout the user completed on certain days and their personal stats such as weight, height, BMI and muscle parts diameters such as hips, legs, calves, chest, biceps, etc. The user can also give more details, such as the weight they used, number of sets, and reps completed. From this data, a “routine workout” can be generated so the user can select it, have it loaded for the day, but they are able to change inputs such as the weight, number of sets, and reps for recording. This will ease the process of creating a workout diary or give the user reference to workouts they have done and wish to repeat. In order to do this, users will be required to create an account in order to have this information stored.

This project will be built using [Node.js](#), **MySQL**, **Express**, **EJS templates**, and **external JavaScript and CSS** files. It will handle routing from user’s information, their workout history, depending on the day they select. Pictures and different types of workout the user can choose to log will be pulled from an external API and returns JSON responses back to the frontend. This project will use **Web Storage** along with **external Web APIs**. This structure ensures clean integration and consistency, while remaining organized for clarity and performance.

3. Target Audience

The target audience for this application is anyone who is interested in tracking their fitness journey, maintaining their progress or one who is new to exercising. The idea behind this is to track one's fitness journey by analyzing one’s workout histories and see data that tells one if they have either improved, or if they have become complacent and need to make adjustments. One can also learn new workouts.

4. Proposed User Interface

There will be a home page that will first inquire the user to either sign in or create an account. Once signed in, users will be given an option to either view their workout diary or begin logging some workouts for the current day. If viewing their workout diary, they can search either by day or by the workout done. When a user achieves a new personal goal/one set by the system, a

congratulatory message will appear. There will be a dropdown menu that one can select a certain body part such as upper body, then related workouts tutorials and info will appear.

5. Proposed Workload Distributions

While each of our workload will be a mixture of backend/frontend, we chose to pick more specialized roles to focus on. All team members will contribute to writing and testing code. Everyone will participate in building at least one core function (e.g., adding or editing functions, helpers, validation checks), collaborate on debugging and final report preparation. We will remain in constant communication with one another through Discord and utilize GitHub repositories to merge our work.

- **Jeremy (Database Lead)** – designs and creates the SQL database, writes table creation and insert statements, exports the final .sql file, and assists with backend query logic.
- **Flavio (UI/Frontend)** – HTML Layout, modern/responsive design, form elements, CSS file to styling, built-in validation to ensure only valid data is ingressed, and consistent styling.
- **Monica (Backend/Routes)** – develops Express routes to handle database interaction, implements API calls, and supports full CRUD operations.
- **Raquel (App Integration & JavaScript)** – Connect backend routes to EJS views, ensure the design works well on each screen size, integrate backend data.

6. Brief Description of Proposed Database Tables

- userInfo - name, email, password, userId, height, weight, goals,
- userProgress - chestDiameter, armDiameter, shouldDiameter, legDiameter, hipDiameter
- exercises - nameOfExercise, exerciseId, muscle group, equipment, instructions, imageURL, apiSource
- workouts - date, duration, userId
- workoutGroups - to categorize which workout for specific muscle groups
- workoutLogs - logId, sets, reps, weight, date, notes (**historical data to be saved)
- workoutRoutine - routineId, userId, routineName, exerciseJSON (array of exercises), createdAt
- achievements - milestones

7. How We Will Meet the Rubric Requirements:

- **Final Report Elements**

We will submit a complete final report that includes:

1. Project title and description
2. Task distribution
3. Description of AI use (e.g., ChatGPT for help with SQL, JS logic, and layout planning)

4. A screenshot of the completed app
 5. An ERD or database schema diagram
- **Form Elements (3+ Types)**
The app will use:
 1. **Text inputs**- inserting personal account information, body parts diameter.
 2. **Number inputs** - number depending on weight, reps, sets count
 3. **Select menu** - select muscle group, equipment, date range, and exercises
 4. **Button** - add workout, search for day, search for workout
 - **Web Storage**
We will use localStorage to remember the user's previous workouts, the weights, reps, and sets done on specific days.
 - **Update Records with Pre-filled Forms**
The form will be pre-filled with the user's information, like username, email, name, weight and height. The user can update the email, password, goal, weight and height.
 - **Add Records**
Users will be able to add new workouts by searching either by name, muscle group, or scrolling through a drop down list to retrieve from an API. Users can also update routine workouts so they can refer to it and update the weights, reps, and sets. Workout routine can be created and referred to for the users. They will also be able to track their progress over time and log that data.
 - **Client-side JavaScript (50+ lines)**
We will include JavaScript to:
 1. Validate form input
 2. Use `fetch()` to call APIs
 3. Handle localStorage
 4. Display fetched workouts
 5. Interact with the DOM to output 1 -3 locations simultaneously
 6. Embedded workout videos depending on the body part selected.
 - **Web APIs**
 1. **ExerciseDB API** - has workout plans such as GetExercisessByEquipment, which has instructions information on how to do an exercise
 2. **Wger API** - will be used to find a workout by type, such as pull, push, legs, upperbody, etc.
 3. **Inspirational Quotes API** - displays an inspirational quote for different days
 - **Professional and Consistent Design**
The application will have a modern layout that is easy to navigate on mobile devices and desktops. The UI will stay consistent regardless of the webpage. All styling will be stored via external .CSS file.

8. Current Team Status

Code busters is a great team, we have agreed upon doing a Fitness Tracker app and we have divided responsibilities clearly. If any issues arise (e.g., someone becomes unresponsive), we will contact the instructor as needed.