**Personal Fitness Tracker Project Proposal**

**Name: Favour Ojobor**

**Overview**

This project is all about creating a Personal Fitness Tracker, a web app that helps people keep track of their workouts, what they eat, and how they’re improving over time. The problem I want to solve is that a lot of people, including myself, find it hard to stay on top of fitness goals without a simple tool to log everything in one place. My motivation comes from wanting to get healthier and liking the idea of building something useful for myself and others who want to exercise more or eat better. It’ll be fun to see my progress and help others do the same, on a software I worked on!

**Target Audience**

The app is for regular people like me who want to get fit but might not be pros at it. This includes students, busy adults, or anyone starting a fitness journey—maybe ages 16 to 50. It’s for people who need an easy way to track workouts and food without complicated gym equipment or expensive existing apps. They might use it on their phone or computer, so it needs to work well on both.

**Major Functions**

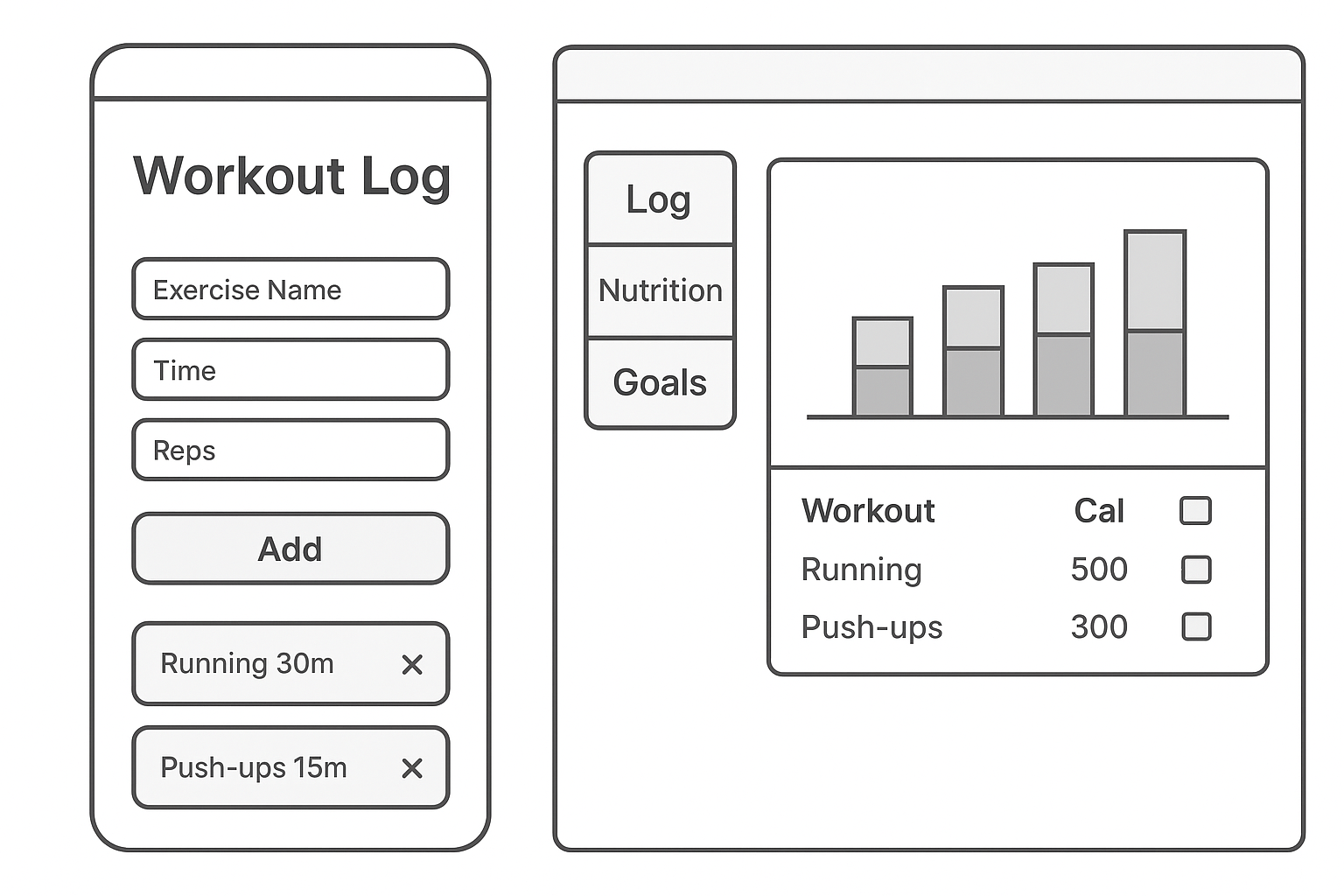
1. **Workout Log**: Let users add exercises like running or lifting weights, including how long they did it and how many reps, so they can see what they’ve done each day.
2. **Nutrition Tracker**: Allow users to enter what they eat (like breakfast or snacks) and see how many calories they’ve had, helping them stay on a healthy diet.
3. **Progress Dashboard**: Show a simple chart or list of their workout and calorie goals versus what they’ve achieved, so they can feel good about their progress.
4. **Goal Setting**: Let users set weekly goals, like exercising 3 times or eating under 2000 calories, with a checkmark when they hit them.
5. **Milestone Alerts**: Pop up a message or change a color when users reach big goals, like losing 5 pounds or doing 100 push-ups total.
6. **Data Export**: Give an option to save their logs as a text file they can download, so they can keep a record or share with a trainer.
7. **Activity Suggestions**: Suggest new exercises or recipes based on what they’ve logged, making it easier to try new things.
8. **Mobile-Friendly View**: Make sure the app adjusts nicely for phones, with big buttons and easy scrolling, so it’s usable anywhere.

**Wireframes**

* **Mobile View (Workout Log)**:
  + Top: Title "Workout Log" in big text.
  + Middle: A box with fields for exercise name, time, and reps, plus an "Add" button.
  + Bottom: A list of today’s entries, each with a delete option.
* **Desktop View (Progress Dashboard)**:
  + Left: A side menu with tabs for Log, Nutrition, and Goals.
  + Right: A big chart showing weekly progress with colors (green for goals met, red for not).
  + Bottom: A table with workout and calorie details.

|  |
| --- |
| **+-------------------+--------------------+**  **| [Log] | |**  **| [Nutrition] | +------------+ |**  **| [Goals] | | Chart | | <- Chart Area**  **| | | Green|Red | |**  **| | |------------| |**  **| | +------------+ |**  **+-------------------+ | Workout | Cal | <- Table**  **| Running 30m | 500 | |**  **| Push-ups 15m| 300 | |**  **+-------------------+--------------------+** |

**Wireframe pictorial view (Mobile/Desktop) below;**

****

**External Data**

* **Nutritionix API**: I’ll use this to get calorie info for foods users enter, like how many calories are in an apple or chicken. I’ll need to store the food name and calorie count.
* **Fitbit API**: This will pull in steps or heart rate data if users connect their Fitbit. I’ll store daily step counts and activity duration.
* **Storage**: I’ll use the browser’s local storage to save user logs (exercises, food) and API data (calories, steps) so it works offline.

**Module List**

* **workout.js**: Handles adding and deleting workout logs, with functions to save to local storage.
* **nutrition.js**: Manages food input, calculates calories using the Nutritionix API, and updates storage.
* **dashboard.js**: Builds the progress chart and checks goals, using data from other modules.
* **api.js**: Connects to Nutritionix and Fitbit APIs, fetches data, and handles errors.
* **utils.js**: Includes helper functions like getting local storage or rendering lists, shared across modules.
* **style.js**: (Not a real file, but idea) Holds CSS animation code for milestone alerts and chart updates.

**Graphic Identity**

* **Color Scheme**: Green (#4CAF50) for progress and success, gray (#B0BEC5) for backgrounds, and red (#F44336) for alerts or missed goals. It’s calm but shows status clearly.
* **Typography**: Use “Arial” for regular text (easy to read) and “Bold Arial” for headings to stand out.
* **Icon**: A simple dumbbell shape in green, with a heart outline in gray, drawn as an SVG for the app logo.
* **Other Styling**: Buttons will be big and rounded, with a hover effect that turns them slightly darker green.

**Schedule**

* **Week 5**: Plan the HTML structure for the log and dashboard, set up API keys, and start workout.js.
* **Week 6**: Build the nutrition tracker and nutrition.js, add CSS animations for alerts, and test local storage.
* **Week 7**: Finish the dashboard and dashboard.js, integrate Fitbit API, fix errors with ESLint, and deploy to GitHub Pages.

**Trello Board Link**: https://trello.com/b/lbJs3sNn/fitness-tracker

Set up HTML, Connect Nutritionix API, Add Chart, etc.

* **Details**: Cards added for each function (e.g., “Workout Log” with subtasks like “Add Form” and “Save to Storage”) and move them as I go. It’ll grow as I think of more tasks.

**Challenges**

* **Biggest Challenges**:
  + Learning to use the Fitbit API might be tricky since I’ve never done it before, especially getting the authentication right.
  + Making the chart for the dashboard could be hard if I mess up the JavaScript math or CSS layout.
  + Keeping the code clean with ESLint rules might take extra time if I forget comments or make sloppy errors.