Muscle Hacker

Objective:

A website that will allow a user to record an exercise’s performance, e.g., a bench press performed on Tuesday 11/22/2022 at 8:00am that had 6 reps per set, 5 sets, and was at 200lbs. This singular exercise will be part of a day’s work out, which will be part of a weekly plan, etc.

The obvious benefit of that is to track progress over time. I plan on using streamlit.io to show charts of an exercise’s progression. This tracking will also coordinate the next segment of the user’s workout plan. It should be able to calculate the number of reps, weight, etc. the user should aim to perform, a goal system. The goal system will help allow the algorithm to setup the next work out plan.

There are a few stages of the app that allows me to work in stages to keep the goals smaller and turn around a working product.

1. Decent looking site
2. Log-in creds
3. Entries of workouts
4. Plans of workouts based on muscle recovery
5. Display of workouts and their histories.
6. Tuning of the plans based on performance of prior exercises
7. Increase user data fields
8. Tuning – adding a gym intelligence
9. Tuning - adding a randomization or gradual change of exercise plans i.e., swapping chest flys for incline bench.
10. Cross user intelligence maybe?? This part is possibly out of scope.

I found the below list of apps that I can take samples of to get better ideas.

<https://www.lifewire.com/best-weight-lifting-apps-5079430>

API:

This is where I think I’ll need some help, as my understanding of how to use an API isn’t strong. I understand that it is a structured set of data, that is easy to grab.

Take the below api-ninjas example:

API-ninjas.com has a list of exercises. <https://api-ninjas.com/api/exercises>

I can use the name, muscle, type, equipment, and instructions. Am I to use the API as library of exercises instead of self-hosting them? Or do I use it to slowly build the library out as the user/program builds it?

There are nutritional APIs I could implement but that is outside of my scope right now. I can investigate things like Fitbit APIs, but I personally do not have a Fitbit right now to test this.

<https://getstream.io/blog/fitness-api/>

I found a BMI Calculator I can add when I implement user data.

<https://rapidapi.com/navii/api/bmi-calculator/>

Framework:

Bootstrap will provide the templates for the front-end work. The theme will be modified to fit a trope-y hacker look for now. Green, lucida console text, black background. It’s simple but should allow for more time for the backend.

Django will be the primary framework for everything else and managing the Database. The admin feature provided by Django will still be used.