**Final project**

Build a multi-step “food tracker app” from scratch using JavaScript.

Throughout this multi-step project, we're going to build a **Food tracker app** from scratch. Both Margit and my topics will be part of this project. You'll not be provided with any html, css or js files, however, you will be provided an **API endpoint/documentation**.

You should follow a step-by-step approach that mimics how a developer would build a project! If possible, try to refactor your code from time to time as well as deal with some edge cases before final submission.

This project is different than the other **to do list app** project as in the course as it tries to replicate real-life projects. It may feel a bit more challenging so take your time!

**Apply course knowledge**

Another goal of the final project is to help you apply the knowledge that you acquired in Programming JS course. Here's a list of some of the topics that it'll help you review:

* Data structures (numbers, strings, arrays, objects, arrays of objects) and their associated methods
* Basic functions
* if conditions
* Iterations and Reduce
* Optional chaining & nullish coalescing
* Fetch API and the FetchWrapper
* DOM (selection, events, etc.)
* JSON
* Async await

**Food API**

We'll be using Food API as a backend which is hosted on my Firebase account. So you do not have to create any new Firebase account. Since anyone can read/add food entries to this API, I have made it possible for you to have your own space where you can add/read food entries. For that you may replace **{namespace}** with any random string.

**Base URL** is: [https://firestore.googleapis.com/v1/projects/programmingjs-90a13/databases/(default)/documents/{namespace}](https://firestore.googleapis.com/v1/projects/programmingjs-90a13/databases/(default)/documents/%7bnamespace%7d)

For example, you can replace **{namespace}** by **helloworld** and the API base URL becomes <https://firestore.googleapis.com/v1/projects/programmingjs-90a13/databases/(default)/documents/helloworld>

Now, assuming nobody else used this **{namespace},** you should see **{ }** if you open the base URL in a new tab.

If you see some data, you need to choose another **{namespace}.** For example, you can add a random number after it   
<https://firestore.googleapis.com/v1/projects/programmingjs-90a13/databases/(default)/documents/helloworld54321>

At any point in the final project, you can change the **namespace** if you'd like to have a fresh start. Please note that the API is public meaning **anyone** can read/add food entries.

**Fetch Wrapper**

As usual, I will provide you with the FetchWrapper class that you have also used in your classroom. While you are not required to use it, I recommend that you do as it simplifies your code. There is a file called “fetch-wrapper.js” attached, feel free to use it.

In case you will need more info, feel free to send MS Teams chat message / email

**Any packages:**

Npm.js

https://www.npmjs.com/

Npm start – start of Local.host

So important (in .json file):!!!

mkdir parcelExplore (create of directory)

code parcelExplore / open in VS code

npn i @kalwar/simple\_number\_formatter install extensions

<https://www.postman.com/product/tools/>

<https://www.chartjs.org/docs/latest/>

"dependencies": {