Aryaman Singh
Dresden Friar
Jason Gao
Jackson Vondemkamp
José Camacho

YumBoard

Team 1

Project Description

YumBoard is a social media app designed to allow anyone to see recipes their friends are making, and share their own culinary creations. It features a built-in grocery list which interfaces with shared recipes, and can give you an estimate of cost for your grocery list.

YumBoard

Tools Used To Complete Project

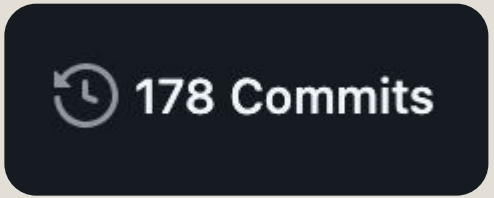
Github Project Board

- Used for project management
- Decent to work with, was a bit clunky with epics & story points

Git/Github Version Control

- Used for version control/repository hosting respectively.
- Very nice to work with

GitHub



PostgreSQL

- We used this for our database storage
- It worked without issue, and was easy to set up.

Visual Studio Code

- Used as our IDE
- Wide range of extensions and good community support made it a great choice.

HTML & CSS

- Used for frontend website design
- Classic frontend suite, works well and is basically impossible to avoid in web dev.

JavaScript

- Used as the programming language for our server
- Is relatively quick

Bootstrap

- Used for frontend
- A library for making good-looking websites quickly

Handlebars

- Used for templating with html (makes reusing code easier)
- No issues, easy to use
- Not really relevant in today's time (Replaced by React, Vue, etc)

handlebars

(Jackson Vondemkamp)

NodeJS/Express

- Used to host a server using JavaScript
- Package Manager makes life easier, things work without a hitch.

Nodemon

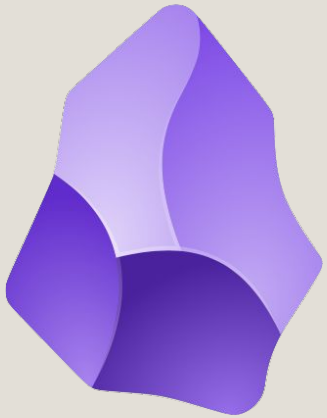
- Used in development to automatically restart the server when things change
- Worked flawlessly

Mocha & Chai

- Used for backend api testing
- Quick to set up, simple to do some basic testing

Obsidian

- Used for writing markdown for our README.md file.
- Great markdown software, makes writing a bit easier.



Kroger API

- Used for calculating the price of ingredients throughout the site
- Easy to make api requests. Difficult to parse data.

Render

- Hosting service to run our NodeJS server.
- Took some time to get going, but then it worked well.

Discord

- Used for group communication
- Had some nice features, like pins, and provided a space to meet online

Agile Development Methodology

- Development methodology (used with Github project board)
- Added a lot of headache, but helped a bit with organization

Challenges We Faced

Not using branches at first

01

At the start of the project, we just committed directly to main, so stuff kept getting overwritten.

GETTING THE KROGER API TO WORK

02

Making sure my getProductToken() logic didn't constantly re-fetch tokens and hit rate limits, while also automatically renewing them when they expired.

Parsing the list of ingredients and calling the priceFor() function from KrogerAPI asynchronously

COMMENTS

03

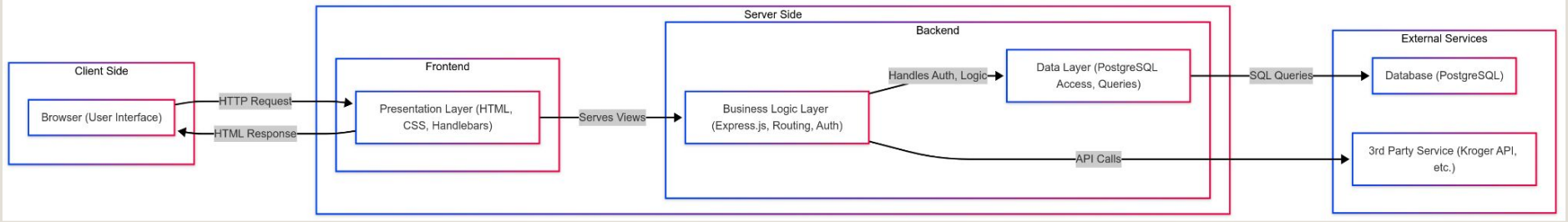
When we first added a commenting feature, comments and replies couldn't be distinguished from each other. Instead of having a flat list of comments, we made it take the form of a nested tree where replies are located under parent comments.

MERGE CONFLICTS

04

Merge conflicts were an issue because we had instances of changes being overwritten, and numerous bugs being introduced through merging.

Architecture Diagram



Future Scope/Enhancement

1 ADDING MORE GROCERY STORE API'S

That way users are able to get the total cost of their grocery cart for other supermarkets they might have nearby, not just Kroger.

2 RECIPE VIDEOS

Recipe Videos allow users to visually document and share how a dish is made. This feature enhances recipe clarity, making it easier for beginners to understand.

3 RECIPE SCALE-UP MODE

It's a feature that lets users scale up recipes and organize meals for multiple servings. Automatically adjusting ingredient quantities while preserving ratios.

Demo Video