Grocery Meal Prep Application - Kale Yeah!

Daniel Portnoy, Brandon Chapple, Eddy T. Garcia, Matt Belz, Dale Madison

June 2022

Description

This application aims to solve the never ending problem of creating grocery lists from scratch. This app can generate a grocery list for you by being able to search for ingredients, display an ingredient page and create grocery lists.

Features

- Able to register a new user to the app
- Able to log in to the application
- Search functionality to search for ingredients
- Display the ingredients
- Create grocery lists

Planning - User Stories

Highlight some of your projects' User Stories. Focus on explaining what this project can do from the user's perspective.

Back End:

- A user will need access to a database of ingredients to create grocery lists
- A user will need access to a database of ingredients to create recipes.
- A user will have access to an external API of food information that gets generated for them

Front End:

- A user can register an account and login to the application.
- A user can search for ingredients/recipes and add them to their respective lists.
- A user can edit their profile including the contents of their pantry and grocery lists.

Planning - Database

Describe what tables are necessary in your DB, and how they relate to one another. Your goal is to show how you designed your database to allow for the user stories listed above.

- Users table is created to store user information such as username, email, full name, password*
- Ingredients table is created to store name, category, price , calories
- Recipes table is created to store the name of the recipe, a list of ingredients, yield, category, instructions, cooking method, and calories
- Grocery list table contains a list of ingredient items, a user , and date
- Grocery Class —> A *One to One* relationship between a grocery list and a user , One to Many where one grocery list has many ingredients
- Ingredient Class —> Many to One where a user can have several ingredients per grocery list, Many to One where a user can have many ingredients per recipe.
- Recipes Class —> One to Many where one recipe can have many ingredients , Many to one where many recipes are related to one user

Technology Stack

- Language - JavaScript, Java, JSX, CSS
- Framework - React + Spring Boot
- Template engine - React Template engine embedded in framework
- Database engine - MySQL , MySQL Workbench
- Tools & Libraries —>
- Axios

Demo

What we Learned

Back End:

- OpenAPI design with Swagger - a YAML specification file to establish endpoints and the types of HTTP responses that are desired
- Created RESTful web services to expose a set of resources (GET, POST)
- Manipulated JSON API responses to achieve desired output
- Relational database organization (Java Classes, ORM relationships)
- Testing APIs with Postman
- Aggregation and encapsulation of data using DTO's.

Front End:

- Axios implementation to fetch data from back end URL request mapping paths
- Implementation of React hooks to utilize state variables in functional components.
- React Router to navigate among views across various React components.
- JSX to create virtual DOM using XML syntax.

What's Next

Back end:

- API aggregator implementation – need more time to implement that well and understand it.
- Potentially an alternate external API that lets us consume more endpoints per day.
- Implementation of Spring Security using JWT authentication.