Prin Pulkes, Jessica Li, Chris Barber

Computation Project Reflection

Video: CS178_FINAL_VID.mov
Github: https://github.com/jessicazli/cs178-computation

A. Problem Statement and Solution

Make My Menu (mmm…) was inspired by our own issue of not knowing what food to cook when we live alone. With no Harvard Dhall nearby and not being able to afford to eat out, we wanted to develop something that would keep both our stomachs and bank accounts happy. We thought a recipe generator would tackle this issue very well, giving us access to an arsenal of recipes depending on specific preferences we had. However, as hard-working college students and future employees, we knew most days, we would not have the energy to go grocery shopping for ingredients on top of cooking, so we came up with the concept of a virtual 'pantry' that would allow us to input the ingredients we currently had in our kitchen and generate a recipe from only those ingredients. All in all, Make My Menu allows users to generate a variety of customizable recipes based on the ingredients they already have in their home!

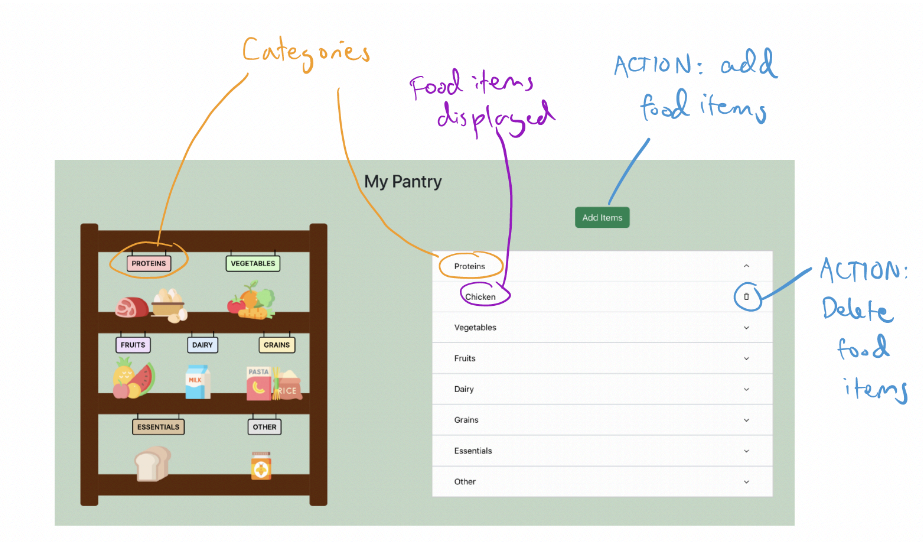
B. Concepts Breakdown

Our implementation was driven by three core concepts: a pantry, a recipe, and a cookbook.

I. *Pantry*

Purpose	To allow users to keep an online record of their food items by category

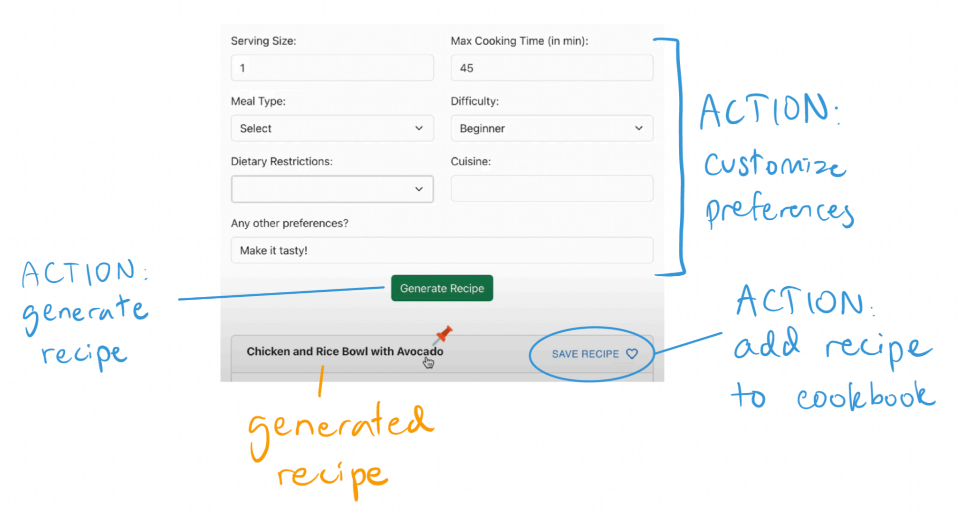
State	Empty, Non-empty
Actions	Add - add food items to your pantry Delete - delete food items from your pantry
Operational Principle	After the addition of a food item, it is accessible and can be viewed in the designated category. After the deletion of a food item, it is no longer accessible and can not be viewed in the pantry.
Pointers to Code	**foodList.js** - accordion that displays data from database and allows deletion of food items **addGroceries.js** - dialog that allows users to add food items to pantry **cabinet.js** - page that includes visuals, accordion, dialog, and essentials dialog for first-time users



II. *Recipe*

Purpose	To allow users to generate a recipe based on items in their pantry and specified preferences
State	Favorited, Un-favorited
Actions	Customize - specify preferences (serving size, cooking time, meal type, difficulty, dietary restrictions, and cuisine) for recipe generated Generate - generate a new recipe Favorite - add this recipe to your cookbook
Operational Principle	After preferences are inputted, a recipe based on those preferences and the food in the pantry is created. After a recipe is favorited, it is accessible and viewable in the cookbook.

Pointers to Code	**recipe.js** - page that allows users to input their recipe preferences, generate a new recipe, and favorite a recipe after it has been generated



Serving Size: 1

Max Cooking Time (in min): 45

Meal Type: Select

Difficulty: Beginner

Dietary Restrictions:

Cuisine:

Any other preferences? Make it tasty!

Chicken and Rice Bowl with Avocado

ACTION: customize preferences

ACTION: generate recipe

ACTION: add recipe to cookbook

generated recipe

III. Cookbook

Purpose	To allow users to build a personalized collection of favorite recipes
State	Empty, Non-empty
Actions	N/A
Operational Principle	After a recipe has been favorited, it can be viewed in the cookbook.
Pointers to Code	**saved.js** - page that displays all the recipes that a user has favorited

C. Implementation of Concepts using React

As React is a widely used and well-documented framework, there are many well-established React libraries we used to implement our concepts, which we will talk about more specifically in each concept section. There are also React libraries that we used for all our concepts though, including react-router-dom. All our concepts are on separate web pages, so we used it to handle the routing between pages and in our navigation bar component.

Using React also made it much easier to integrate our database, Firebase's Firestore, to our web application as Firestore has clear documentation for integrating React web applications specifically. This was important to us as Firestore is a noSQL database, so it allowed us to store our data in collections and documents that kept the structure of our data. For instance, with the saved recipes (for the cookbook concept), we could save the recipes in a 'saved_recipes' collection and each saved recipe was its own document in that collection. The fields of each document would contain the data we would need in our processed format, including the recipe name, cooking time, ingredients and instructions. Firestore is also very intuitive and beginner-friendly with detailed documentation, so it was our preferred noSQL database.

Additionally, inherent React affordances, such as Hooks, which allow for reactivity, and Flex, which helps with styling, were very useful for implementing our concepts which we explain more in detail below.

IV. Pantry Concept

The pantry concept is implemented in the UI with an accordion that lists a user's food items organized by food category next to a visual of a pantry that can be used to open the previously stated accordion. This implementation utilizes many of the affordances of React including its component-based nature as well as its reactivity.

In order to build the accordion and the dialog that appears when users add new food items, we created reusable components from Radix UI, a UI component library built on top of React. Further, the interactivity of the SVGs and the accordion relies on React's useState to control the accordion programmatically. React's useEffect allows us to pull a user's food item data from the database and effectively display it in the accordion. Lastly, deleting and adding food items to the database and interface rely on React's state management with asynchronous database operations that allow for immutability, conditional rendering, and dynamic data handling.

V. Recipe Concept

The recipe concept follows commonly used parameters used in both cookbooks and online recipe sites such as a dish name, serving size, cooking time, and dietary restrictions. These specifications follow the mental model of users when searching for recipes and so, we implemented this into our interface by allowing users to specify these parameters using components like Select and Button from Material UI, an open-source React component library. React's useState also helped us keep track of these user inputs in state variables which we were able to use in conjunction with OpenAI's API. Further, the generation of a recipe involved React's useEffect to pull a user's food data from Firebase and ensure that the recipe only incorporated items present in a user's pantry.

VI. Cookbook Concept

For the cookbook concept, the most important React affordance was the useEffect Hook, as it allowed us to pull all the user's favorite recipes from the database as soon as the user navigated to that page, and automatically update the page to show the recipes. Specifically, useEffect allowed us to call a function, 'startFetching', as soon as the 'My Cookbook' page was navigated to. The 'startFetching' function would fetch the saved recipes from the database and update the 'savedRecipes' state variable to the data fetched.

The useEffect Hook also lets us add dependencies as a parameter, where useEffect reruns if any of the dependencies is updated. Since we wanted to pull from the database as little as possible (for efficiency and pricing reasons) and we only needed to pull from it once to get the initial list of saved recipes, this parameter allowed us to ensure useEffect (and thus, the 'startFetching' function) would only be run once every time the 'My Cookbook' page is navigated to by adding an empty dependencies list.

We decided to style the cookbook page to look like a tack board of recipes, which took advantage of React styling, React components and the React library 'react-responsive-masonry'. One of the initial playtest feedback that we got was that the website was a bit boring, so we wanted to make it look nicer with a cohesive aesthetic. In addition, this styling made it easy to view a lot of recipes at once, so you can easily find a recipe that you want to try to cook again. It was partly inspired by the favorites/saved page on instagram where you can see a feed of your saved posts.

D. Addition of Computation & How it Affected Design

The computation that we focused on was generating the recipes from parameters that the user selects as well as the ingredients they have in their virtual pantry. We decided that LLMs would be the best way to do this given that they can take in complex parameters such as the ingredients on hand and come up with a creative dish. The OpenAI API is one of the simplest ways to integrate an LLM, and we used GPT 3.5 in the final product (due to the cost constraints of GPT 4).

In terms of affecting the design, one of the most useful parts of the OpenAI API is that there is a setting to have it output JSON instead of just text. This feature made it possible for us to style the output by putting the different variables into different parts of the JSON dictionary. For example, it might output {"ingredients": [chicken, pasta, cheese], "dish_name": "Chicken Alfredo", "recipe": "Step 1. (etc.)"}. This made it easy to style the output by inputting those variables into the JSX, and having React automatically update the page with the results. The downside of this approach is that you cannot stream the results, so it appears quite slow to the user. We decided to combat this slowness with a satisfying loading animation, which worked very well as many people commented that they appreciated the animation during the playtests.

Generating the prompt also influenced the design because we needed to take inputs from the user in order to inform the model of all of the preferences. We tested different prompts and found that this one gave the most accurate JSON and the best recipes, although I believe that we could have tweaked it even further given more time. We used the different variables from the recipe form to make the prompt match the specifications of the user.

E. Playtest Analysis

The playtest was very valuable in determining the concept instantiations that weren't intuitive to users. For instance, our user flow after creating an account and logging in was not intuitive. Once users had created an account, it redirected them to the login page, which worked but forced users to re-type their email and password, so we decided to login them in straight away once they had signed up instead. Also, after logging in, the website redirected the user to their 'profile' page. However, the pages that users are likely to use often are the pantry and recipe generator page, so it did not make sense to redirect users to the 'profile' page first. We decided to change the user flow to redirect them to the 'pantry' page instead as we wanted users to check that the list of ingredients in their virtual pantry was accurate before generating a recipe.

Another key feature that confused users was the profile page as we had included checkbox items of ingredients we determined to likely always be in stock in a household, which we called 'essential ingredients'. We recognized that for new users, adding all the ingredients that they had to our website would feel tedious, so the 'essential ingredients' list also served as a way for new users to easily add ingredients they most likely had. All the 'essential ingredients' were checked as default and users had to uncheck a box if they did not have that ingredient.

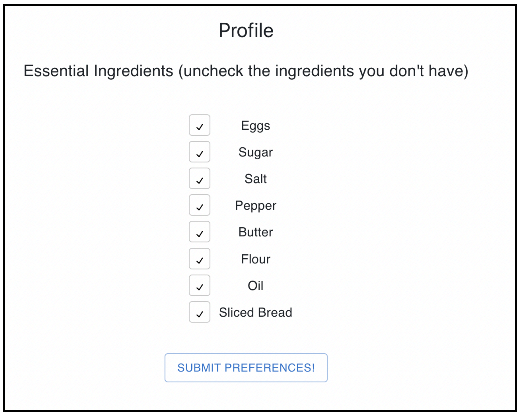
We had put the 'essential ingredients' on the profile page instead of the pantry page (where other ingredients could be added or removed) as we wanted the profile page to include long-term preferences/items that the user would be unlikely to change so that they wouldn't have to frequently re-enter these preferences/items. We had planned to include the 'essential ingredients' list, favorite cuisines, dietary restrictions, and allergies in the profile page.

However, we found that this did not map to a user's typical mental model of a 'profile' page, so the 'essential ingredients' list simply confused them. We thought of two solutions:

1. Changing the name of the profile page to 'long-term preferences' or something similar
2. Move our 'essential ingredients' list to our pantry page where all our other ingredients were

We eventually decided on option 2 as we thought it made more sense that the feature of adding and removing ingredients would be all on the same 'pantry' page. However, as we wanted to keep our list of 'essential ingredients' for new users, as we wanted to make the onboarding process of adding ingredients easier. Thus, we decided to add a user flow where if the user had created a new account (meaning they were a new user), it would direct them to their 'pantry' page and they'd get a pop-up with the 'essential ingredients' list that would only show up once. After that, they would be able to add or remove 'essential ingredients' in the 'pantry' page the same way as any other ingredient.

Before

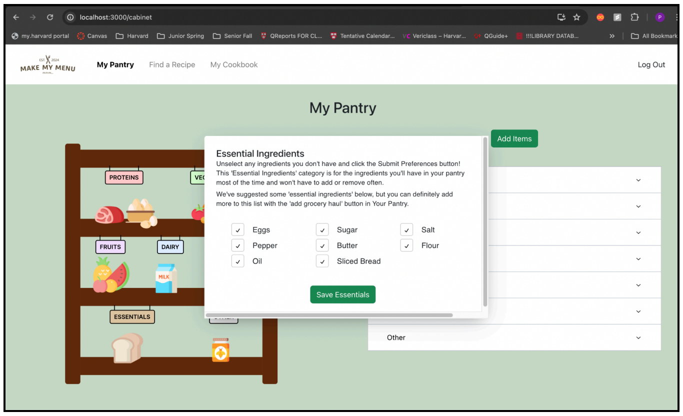


Profile

Essential Ingredients (uncheck the ingredients you don't have)

☑ Eggs
☑ Sugar
☑ Salt
☑ Pepper
☑ Butter
☑ Flour
☑ Oil
☑ Sliced Bread

After



My Pantry

Essential Ingredients

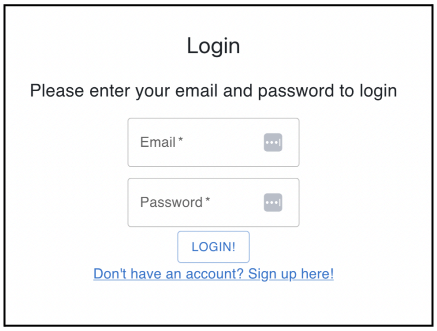
Unselect any ingredients you don't have and click the Submit Preferences button! This 'Essential Ingredients' category is for the ingredients you'll have in your pantry most of the time and won't have to add or remove often.

We've suggested some 'essential ingredients' below, but you can definitely add more to this list with the 'add grocery haul' button in Your Pantry.

☑ Eggs ☑ Sugar ☑ Salt
☑ Pepper ☑ Butter ☑ Flour
☑ Oil ☑ Sliced Bread

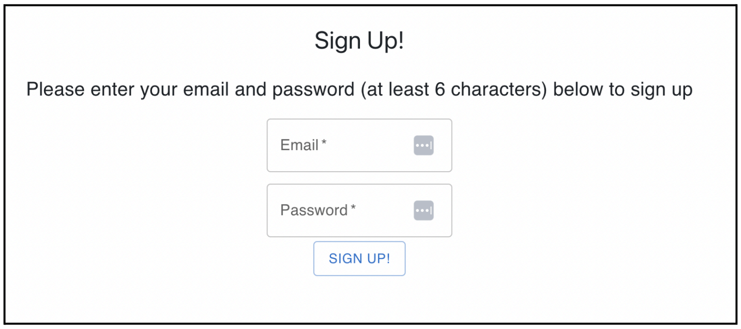
Another insight we received from playtest was that some users did not want to create an account without knowing what our website would offer. Originally, the home page of our website simply consisted of a login page that had a link to our sign up page. Thus, we decided to add a name, logo and description of our website and its purpose so that users understood what they were signing up for.

Before



Login

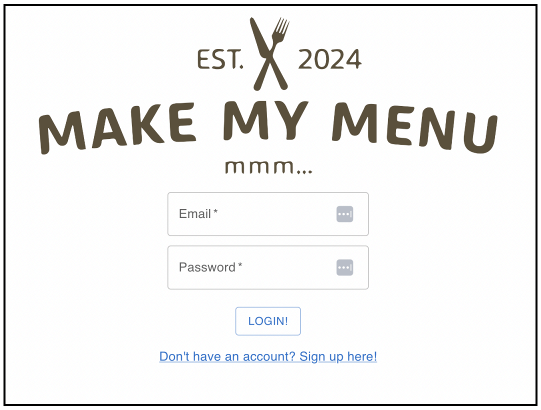
Please enter your email and password to login

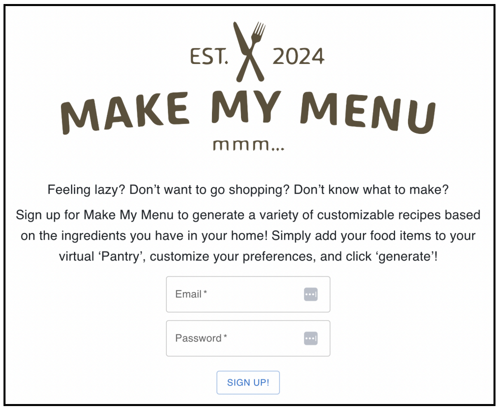


Sign Up!

Please enter your email and password (at least 6 characters) below to sign up

After





Feeling lazy? Don't want to go shopping? Don't know what to make?

Sign up for Make My Menu to generate a variety of customizable recipes based on the ingredients you have in your home! Simply add your food items to your virtual 'Pantry', customize your preferences, and click 'generate'!

Some additional requests we received from users was to make the image of the ‘pantry’ in our pantry page dynamic by graying out or not showing ingredient categories that users did not have, or even adding new ingredients onto the pantry image if users added it to their list. However, as this did not directly impact the purpose of our website, we decided to make this a less-prioritized change that we would have implemented with more time since we focused on the above changes and completing the functionality of our key concepts instead.

Another request we got was to make it more of a social website by allowing users to share recipes with others, or view other users’, such as their ‘friends’ or ‘followers’, favorite recipes. We also thought this would be a great addition to our website, but also made it a secondary change that we would implement with more time.

F. Human-AI Interaction Design Guidelines

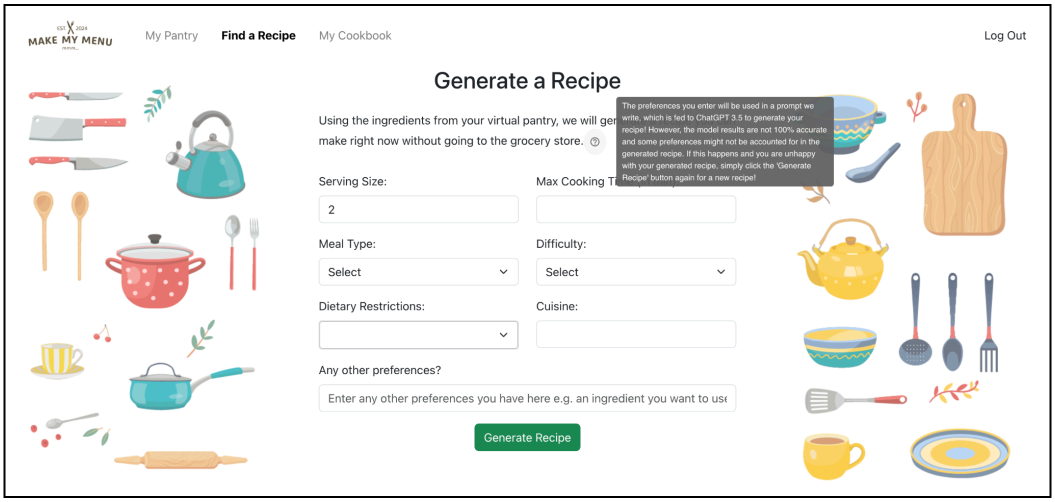
I. G1: Make clear what the system can do.

With our design, we believe it is clear that one of the only things that people can do with our website is to generate recipes. There is a form with preferences and a generate recipe button and that’s all you can do to interface with the LLM. We even had some playtesters try to break the app by inputting non-ingredients / asking it to make jokes and it still stayed true to the recipe format (even though it made some jokes within the recipe).

II. G2: Make clear how well the system can do what it can.

We let users input their preferences for a recipe through dropdowns and text input boxes, and then write a prompt based on these preferences that is fed to ChatGPT 3.5 to generate a recipe. This means that sometimes, ChatGPT does not generate a recipe that takes into account all the preferences they input. Thus, we use the term ‘preferences’ to describe these inputs, to suggest to the user that they’ll be things that we’ll take into account when generating a recipe but aren’t necessities that will always be factored in.

Additionally, we have a tooltip item that explains this to the user explicitly so that they are aware, specifically telling them: “The preferences you enter will be used in a prompt we write, which is fed to ChatGPT 3.5 to generate your recipe! However, the model results are not 100% accurate and some preferences might not be accounted for in the generated recipe. If this happens and you are unhappy with your generated recipe, simply click the 'Generate Recipe' button again for a new recipe!”



Generate a Recipe

Using the ingredients from your virtual pantry, we will ger
make right now without going to the grocery store. ⓘ

The preferences you enter will be used in a prompt we write, which is fed to ChatGPT 3.5 to generate your recipe! However, the model results are not 100% accurate and some preferences might not be accounted for in the generated recipe. If this happens and you are unhappy with your generated recipe, simply click the 'Generate Recipe' button again for a new recipe!

Serving Size: 2

Max Cooking Ti

Meal Type: Select

Difficulty: Select

Dietary Restrictions:

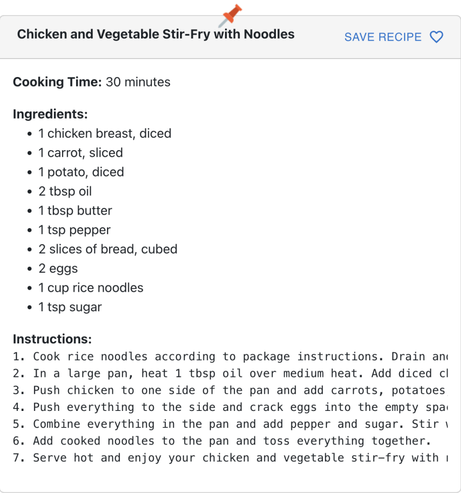
Cuisine:

Any other preferences?

Enter any other preferences you have here e.g. an ingredient you want to us

Generate Recipe

III. G4: Show contextually relevant information. & G5: Match relevant social norms.



Chicken and Vegetable Stir-Fry with Noodles SAVE RECIPE ♡

Cooking Time: 30 minutes

Ingredients:

- 1 chicken breast, diced
- 1 carrot, sliced
- 1 potato, diced
- 2 tbsp oil
- 1 tbsp butter
- 1 tsp pepper
- 2 slices of bread, cubed
- 2 eggs
- 1 cup rice noodles
- 1 tsp sugar

Instructions:

```
1. Cook rice noodles according to package instructions. Drain an
2. In a large pan, heat 1 tbsp oil over medium heat. Add diced c
3. Push chicken to one side of the pan and add carrots, potatoes
4. Push everything to the side and crack eggs into the empty spa
5. Combine everything in the pan and add pepper and sugar. Stir
6. Add cooked noodles to the pan and toss everything together.
7. Serve hot and enjoy your chicken and vegetable stir-fry with
```

As we are generating recipes for the user, we show all the information they will need to prepare the recipe, including the cooking time, ingredients (and amounts) they'll need, and step-by-step instructions. The format we output also matches the user's mental model or expectation for the format of a recipe, with the ingredients and the amounts for each ingredient listed at the top, and step-by-step instructions below.

IV. G7: Support efficient invocation. & G9: Support efficient correction.

Users can very easily request a recipe generation in our 'Find a Recipe' page by simply clicking the 'Generate Recipe' button. Additionally, we attempt to support efficient correction by allowing users to generate as many recipes as they want based on their previously inputted preferences if the first generated recipe does not match their expectations. In the future, including a place for users to give feedback or rate the quality of the generated recipe/adherence to their preferences would help us fully support efficient correction.

G. Ethical Considerations

One consideration we had involved the types of recipes being generated. As we were using ChatGPT and our prompts to the ChatGPT API are in English, the dishes being generated were more likely to be Western dishes, which is what we found when we tested our web application. This meant our web application kind of had an implicit bias towards Western dishes, which we didn't want since we were aware that users may want to try new recipes or flavors typical in other cuisines. We also didn't want to encourage people to only stick with recipes they were used to or had heard of. To combat this, we decided to add a 'cuisine' and 'preferences' input box to give the user more control over the types of dishes generated and allow them to explicitly choose a cuisine or flavor they wanted to try.

Furthermore, our project could promote unhealthy lifestyles as the purpose of our project is to allow users to quickly generate recipes based on the ingredients currently in their pantry. Thus, this could reinforce bad habits with the food the users eat. For instance, if users don't have many healthy ingredients, such as vegetables, those users are not encouraged to make healthier choices. Although we have not implemented this in our current website, a method of combating this is to ensure the algorithm factors in the nutritional balance of the recipes it suggests and gives warnings if it can not generate a nutritionally balanced recipe from the ingredients in the pantry. This would involve incorporating guidelines that promote a variety of food groups and limit the frequency of suggesting recipes high in processed ingredients, sugars, fats, or sodium.

H. Teammate Contribution

Everyone worked together on brainstorming and ideating, including coming up with the idea and concepts, designing the web application and logo, and choosing the features we wanted to prioritize for implementation.

For programming, we decided to split up the tasks into three subcategories: backend, recipe generation, and pantry.

- Prin worked mainly on the backend, including setting up and designing the structure of the database (Firestore), and integrating authentication. Prin also worked on the frontend, specifically on the login and sign up pages, the navigation bar, creating the essentials popup on the pantry page, and writing all the functions to read to and write from the database, including saving recipes and reading and writing ingredients to the database.

- Chris worked mainly on the recipe page, and integrating OpenAI for generating the recipes. He created the prompts that generated the recipes and made sure OpenAI responded with JSON so we could use the result data and make it look nice for the user. Chris also did the "my cookbook" page, which takes the saved recipes from the database and presents it on a tack board inspired page.

- Jessica worked mainly on the pantry page and the frontend–specifically creating the pantry graphic visuals on Figma, building out reusable components like the accordion and dialog, pulling data from the database for the pantry accordion, enabling deletion of food items, and enabling users to add food items to the database.

For the writeup, we also split up the work evenly. Jessica worked on section B (concepts breakdown), Chris worked on section D (addition of computation), and Prin worked on section A (problem statement) and E (playtest analysis). For the rest of the sections, everyone contributed and wrote part of the section.