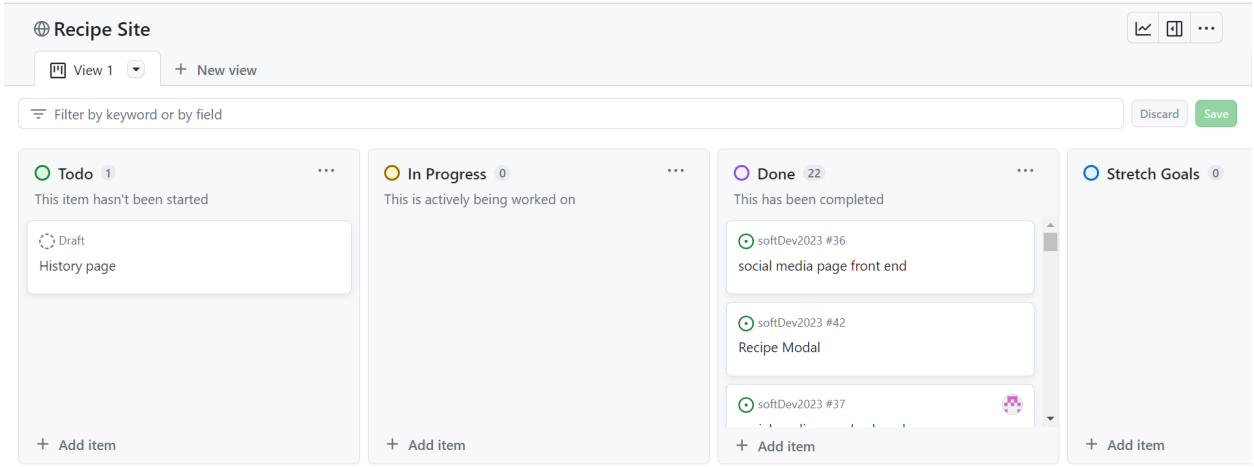
GenerAte

Collaborators: Grayson Richard - Tyler Kivelson - Johnny Wilcox - Suhana Zeutzius - Alex McDonald - Aniket Chauhan

Project Description: GenerAte is an AI-powered recipe generation website. The pages of the site include the home page, the kitchen page, the pantry page, the favorites page, and the settings page. The settings page is where you can change your username and dietary preferences, the pantry page is where you can select what items you currently have in your pantry and the favorites page is where you can see the recipes you've favorited. The kitchen page is where you can generate any recipe you want. You have the option to choose between having the recipe include any ingredient or just the ones you have selected for your pantry list. Once you have picked one of those two options, AWS Bedrock Claude V2 will generate a recipe based on that selection as well as meeting your dietary restrictions. You can then either favorite, edit, post, or delete that recipe. If you choose to post the recipe it will then show up on the homepage so that others can see and like your recipe. If you select to use only your pantry items and you don't have the proper ingredients for the recipe you want, then you will be told that you can't make the recipe with the ingredients you have and you will be given a list of the ingredients that you do need. Overall, the site is supposed to help home cooks find new unique recipes as well as reduce food waste by helping people use all of their leftover pantry items.

Project Board:
https://github.com/users/teeck111/projects/1/views/1



Video Demo:
https://youtu.be/_TewRFznp0g

Repository:
https://github.com/teeck111/softDev2023

Contributions:

- Tyler Kivelson: I worked on the front end of the home page, creating the layout for the page, adding in the EJS, text, and photo, as well as adding in the front-end functionality for the social media aspect. I also completed the front and back end of the settings page. For these features I used EJS, CSS, Node.js, docker, and postgeSQL. I also helped set up the front-end EJS of the favorites page.

- Grayson Richard: Did styling for the pantry, home, kitchen, settings and favorites page. I set up the AWS Bedrock call, this included IAM user role setup, sagemaker role setup, writing the Bedrock call in JS, prompt tailoring, ensuring the new recipe was displayed correctly, ensuring the new recipe was stored in DB and ensuring all user specifications were met. Made settings page editable. Made the recent recipes modal editable and added the backend for the post button also made the list of recent recipes display properly. Added a unique ingredient option to the pantry page.

- Alex McDonald: I worked on the database create.sql and insert.sql, and worked on part of the backend to retrieve posts, log in, and the supporting regex & SQL. I did the styling for the login and register pages. I also assisted in many parts of the site, such as in getting AWS Bedrock to be functional, and deployed the site on Azure.

- Suhana Zeutzius: I worked on the pantry page, allowing users to add and remove ingredients from their pantry and making a search bar to allow them to find ingredients quickly. I also make the like and unlike button for the recipes that were posted.

- Aniket Chauhan: I worked on the backend of the register and favorites page, with the favorites page we made an accordion using bootstrap and worked on the backend with the api callings. I also worked on the backend of the register page making sure that the user can register their email and password and worked on redirecting back to the login page. I also worked on making sure that login credentials were saved so that way no matter what email was given it was saved to allow the user to login.

- Johnny Wilcox: Worked on the front and back end of the kitchen and favorites page alongside some styling for the menu and general website. For the kitchen page I created the front end and implemented a functional modal to both edit and post past recipes. This involved the creation / modification of multiple backend routes to both update, delete, create, and post recipes. For the favorites page I worked on adding posting and un-favoriting functionality. I was also the team scribe - taking diligent notes during both scrum and team meetings as well as creating meeting agendas when necessary.

Use Case Diagram:

Tests:

Feature 1: Update User Pantry

Feature description: When a user updates the amount of ingredients they have in their pantry, this data is successfully stored in our database and displayed on the pantry page. If the user tries to add an invalid ingredient, they see an error message and the database is not updated.

1. Test Cases:
    1. When the pantry is empty, the page has a specific “empty pantry” display
    2. Adding a valid ingredient displays the ingredient on the pantry page.
    3. Removing an ingredient removes the ingredient from the displayed list.
    4. Trying to add an invalid ingredient (need to figure out functionality, this may be impossible) causes an error message to display and no change to the displayed list.

2. Description of Test Data: The user can choose an ingredient to add to their pantry from a drop-down list. The user can also delete an ingredient from their pantry with a delete button.
3. Description of Test Environment: Development environment
4. Description of Test Results: HTTP Response, Database correctly updated
5. User Acceptance Testers: Team Member's Classmates
6. Observations: Users seemed to find the UI intuitive and had no issues with the workflow. They were able to add ingredients from the quick add with no problems and liked that they could add in ingredients themselves if they couldn't find it in the quick add. No changes need to be made.

Feature 2: Recipe Counter

Feature Description: The total number of recipes the website has generated is displayed on the home page.

1. Test cases:
    1. When the website has generated no recipes, this number displays 0, not an error
    2. When a recipe is created, the counter increases by one
2. Description of test data: The counter increases for as many recipes as the user makes. Test data is the creation of a recipe
3. Description of test environment: Development environment
4. Description of test results: HTTP Response, counter variable updated, database updated
5. User Acceptance Testers: Team Member's Classmates
6. Observations: Users enjoyed the counter and liked where it was on the page. The only negative comment was about how low the counter was but that will change over time and they recognized that. No changes need to be made.

Feature 3: Favorite Recipe

Feature Description: The user can mark recipes as favorites, which are stored in the database as favorites and displayed.

6. Test Cases:
    1. When there are no favorite recipes, nothing is displayed
    2. Adding a recipe to favorites
    3. Removing a recipe from favorites
7. Description of Test Data: The user can see the recipes they have tried in the past, and have the option to click a star to mark recipes as favorites
8. Description of Test Environment: Development environment
9. Description of Test Results: HTTP Response, Database correctly updated
10. User Acceptance Testers: Team Member's Classmates
11. Observations: Users seemed to find the UI intuitive and had no issues with the workflow. They had no issues favoriting or unfavoriting recipes and found the Favorites page with ease. No changes need to be made.

Feature 4: Post on Social Media

Feature Description: The user can post their favorite recipes to the “social media” part of the home page.

12. Test Cases:
    1. They can post from the Kitchen page
    2. They can post from the Favorites page
    3. They can like or unlike posts
13. Description of Test Data: The user sees a “post” button on recipes both on the Kitchen page and Favorites page. There is a like button on the posts that the user can click on to like or unlike.
14. Description of Test Environment: Development environment
15. Description of Test Results: HTTP Response, Database correctly updated
16. User Acceptance Testers: Team Member’s Classmates
17. Observations: Users seemed to find the UI intuitive and had no issues with the workflow. They posted from the Kitchen and Favorites page with no issues. No changes need to be made.

Deployment Link:

http://recitation-013-team-02.eastus.cloudapp.azure.com:3000/