Project Name: CookBook

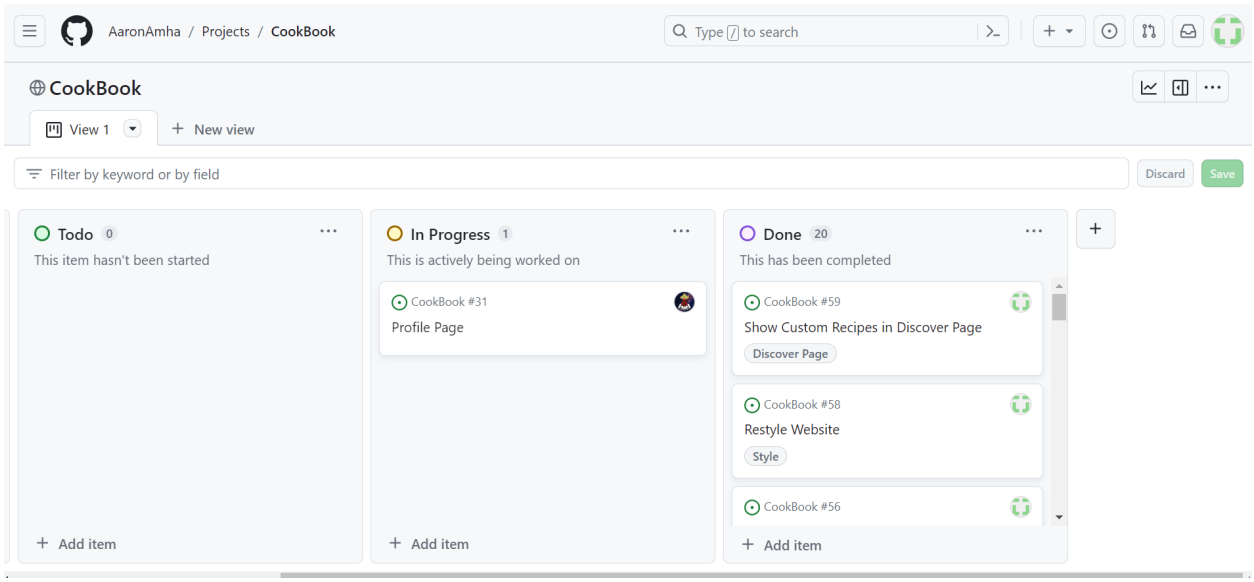
Group Members: Aaron Amha, Michael Dempsey, Sarah Sharroufna, Alex Foucher

Project Description:

The project CookBook is designed to serve as a social media platform for users interested in cooking. The website uses tools such as PostgreSQL to store, update, and help display recipes and user information. There are seven features in the CookBook project that allow for better functionality. The first feature of the project is the Login page. The Login page was designed to be a check to see if the user has made an account. If they have, they will be redirected to our Discover page. Otherwise, the user is redirected to the Registration page. The Registration page allows users to create a personal account, with a hashed password for protection. After logging in, the user is redirected to the Discover page, which shows the recipes from the Spoonacular API or custom recipes created by users. On the Discover page, the user also has the option to favorite recipes, which will then be stored in their Favorites tab. Another feature is the Add Recipe page, where users can add their own recipes to the Discover tab. The final two features are the View Recipe and Logout pages. The View Recipe page allows users to comment on, like/dislike, and view recipes. The logout button logs the user out of their account.

Project Tracker - GitHub Project Board:

- Link to your Project Tracker (for instructor & TAs):
    - https://github.com/users/AaronAmha/projects/1
- Screenshot showing your project in your project tracker



Video: 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

https://drive.google.com/file/d/1gYvjrmQv0lJ_42CQ1xEw9-zGVnNoznhg/view?usp=sharing

VCS: Link to your git Repository. Instructor/TAs will check, weekly, to ensure the following are stored in your VCS repository:

- Source Code, Test Cases, Video demo, README.md in GitHub, Project documentation, Project Board

- https://github.com/AaronAmha/CookBook

Contributions:

Alex Foucher:

Worked on the initial use case diagram, and the initial UATs. For the website: created the addRecipe page functionality which allows for both user and api recipes to be shown on the same page, and for the user to be able to add their own recipes. Created the recipes table which handles the function described. Added image handling using the ‘Multer’ package. Worked on the Project Report. Altered discover page to pull from database, not API.

Aaron Amha:

Developed the register, login, and profile page, while throwing in an extra hand on the addRecipe page and function. Figured out a way to make the registration page a bit more personable by implementing an upload profile picture option, requiring the ‘Multer’ package. Additionally, the registration inquires more personal information from the user so I contributed to creating the chef/user database. When contributing to the addRecipe page, I implemented a javascript function that adds a button where you can add and remove extra ingredients as an addRecipe interactive option. For my listed pages, I did focus largely on user interface, incorporating floats, tables, row aligns and containers. Presentation powerpoint, ReadMe and Project Report. System Architecture.

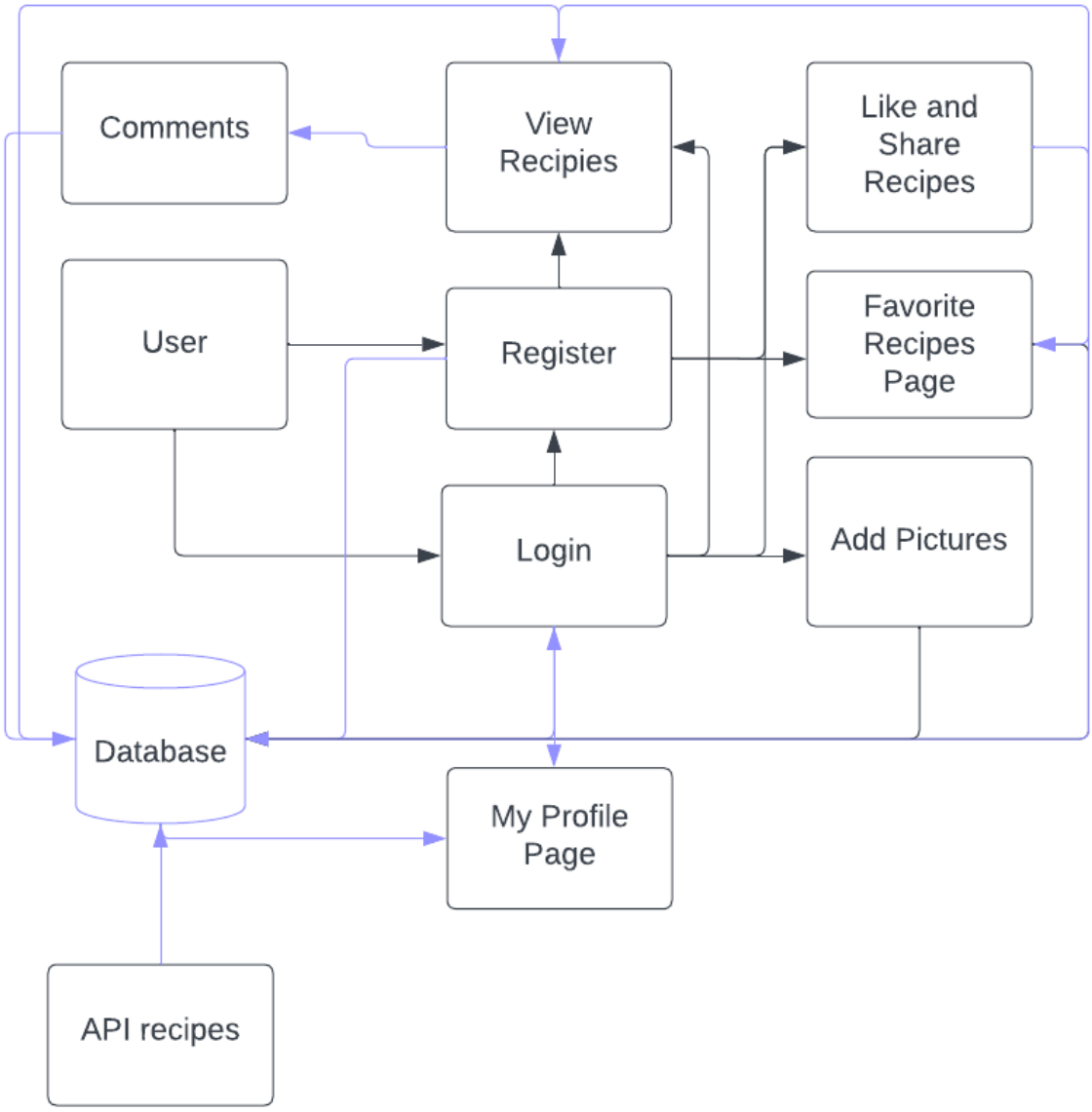
Sarah Sharroufna:

As a contributor, I had a pivotal role in a lot of back-end development, leading the creation of essential tables like recipes, reviews, and reviews-to-recipes within the create.sql database. My contributions extended to the user comments feature, covering both front-end and back-end aspects. On the front end, I developed the comments feature using comments.ejs, while on the back end, I implemented the required logic for comments and reviews, establishing a functional commenting system. To augment the user experience, I populated the database with initial reviews for existing recipes. Additionally, I integrated helpful APIs when we encountered API limits, enhancing the project's functionality.

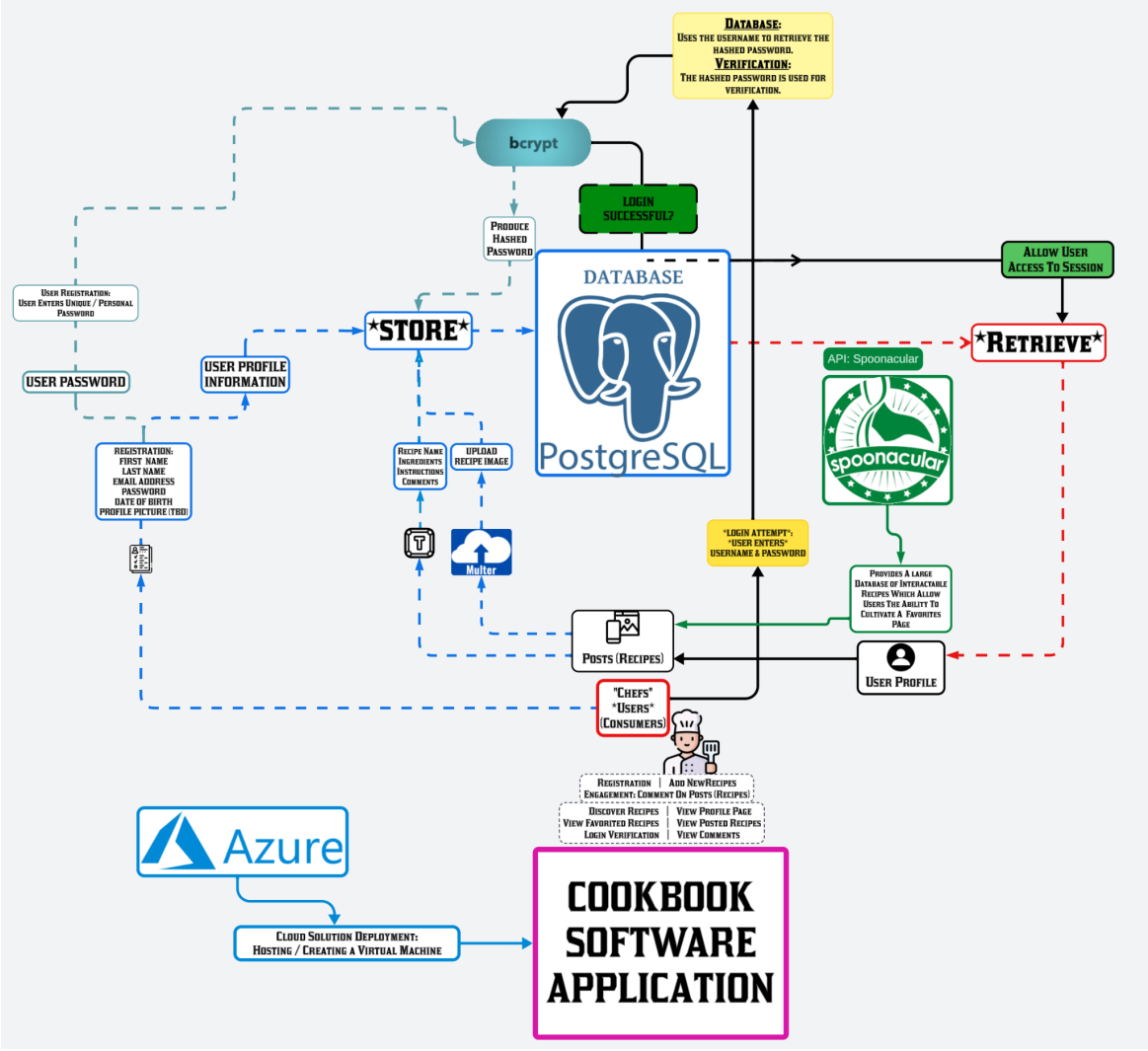
Michael Dempsey:

I implemented the "likes" functionality. Clicking “Like” increases the overall likes of the recipe while “Dislike” decreases the likes. I also worked on the Favorites page and the ability to favorite recipes. I worked on maintaining search queries after leaving the Discover page and having custom recipes show up after searching. For the Add Recipe page, I worked on helping merge two versions created by Alex and Aaron. I also fixed displaying the instructions and ingredients when viewing recipes. Additionally, I worked on improving the aesthetics of the website. Finally, I resolved a bunch of merge conflicts with other branches.

Use Case Diagram: You need to include a use case diagram for your project. You can build on the use case diagram you created in the proposal. If you built a complete use case diagram for the proposal, you can include it as is.



High System Architecture:



Test results: In Lab 11, you created a Test Plan. You need to include the test results and observations in the project report. Refer to [this] for more information

Feature 1: Register

We have a Register page that allows users to add a new account into the database. The user must input all the necessary information, including their name, email, date of birth, username, and password. After providing their information, they are redirected to the Login page.

UAT:

- Given I have provided all the necessary information such as my name, email, date of birth, username, and password
- When I click the register button, my account is added to the database
- Then I am redirected to the Login page.

Testing:

- To test this feature, we will have a user input their information on the Register page
- The user will fill out all the necessary information and then click the “Register” button
- Upon submitting, the user should be redirected to the Login page, where they will be able to input their login information
- The user will be someone who did not work on the project so we can have a fresh pair of eyes to test if our systems are working as intended

Observations:

- The user inputted all their information on the Register page
- The user’s actions were what you would expect from dealing with a Register page
- The user was skeptical about only having one input field for the password
    - Usually, there is another input field that has you confirm your password to ensure that you typed what you wanted to type
    - If we had more time, we would have a “Confirm Password” input field. We would also check if the inputted password was valid (exceeded a certain length, contained different symbols, etc)
- There should be a check on the Register page making sure that the inputted username is not in the database
- Other than that, the registration was successful

Feature 2: Login

We have a login page that requires a user to enter their credentials before accessing

the site. The login page recognizes if the user has a valid login or if the user needs to register.

UAT:

- Given I have entered a username and password that is stored in the database
- When I click the login button, I will be redirected to the Discover page.
- If I input a username that is not stored in the database, I will be redirected to the Register page.
- If I input a username that is stored in the database but the password is wrong, I will receive a message stating that I inputted the wrong information.

Testing:

- To test this feature,we will have the user, after registering, login to the page.
- The user will input their login credentials and submit
- Upon a successful login, the user will be redirected to the Discover page
- If the login is unsuccessful, the user will be either redirected to the Register page (if the inputted username was not found in the database) or given a message stating they inputted the wrong username/password (if the username was found in the database)
- The user will be someone who did not work on the project so we can have a fresh pair of eyes to test if our systems are working as intended

Observations

- After the user had registered, we had the user purposefully input the wrong password to start off with. A message came up stating that the user inputted the wrong username/password
- We then had the user input the wrong username, which brought up the Register page.
- After going back to the Login page, the user inputted their correct login credentials, which redirected them to the Discover page
- The user's action were consistent with what you would expect with a login page
- Based off the user's actions, the login page functions as it should

Feature 3: Discover

We have a Discover page that allows users to be able to find new recipes, which are either predetermined or retrieved through an API. After clicking “Sort by Likes”, the Discover page is sorted by whichever recipe has the most likes.

UAT:

- Given I have clicked on the Discover page after logging in
- Default recipes from the API will show up, which I can sort.
- Clicking “Sort by Likes” will cause the recipes to be arranged by most likes. Clicking “Unsort” will cause the cards to be supported alphabetically
- When I search using the navigation bar, recipes containing the search word would appear

Testing:

- We will have the user search for various recipes
- The user will verify that recipes relating to the search term come up
- We will have the user sort by likes and verify that the recipe with the greatest number of likes comes first
- The user will be someone who did not work on the project so we can have a fresh pair of eyes to test if our systems are working as intended

Observations

- After logging in, the user accessed the Discover page
    - All the default recipes showed up on the front page
- With dummy values with likes already added to the database, the user was able to sort by likes. The recipes with the greatest number of likes appeared first
    - It would be nice to be able to sort in other ways. For example, we could sort by least popular, most recent, or by food type
- The user searched up the term “chicken”, and all the recipes that contained “chicken” in the name appeared
- The user favorited the first recipe that came up, which caused the search to reset

    - We thought it would be better to retain the user's search after favoriting so we modified the Discover page to save the user's last search
- Based off the user's actions, the Discover page seems to be functioning correctly

Feature 4: Add Recipe

The Add Recipe page allows users to add custom recipes to the database. The page gives users the ability to give their recipe a name, an image, instructions, and ingredients. Upon submitting their custom recipes, the recipe is added to the database, and the user is redirected to the Discover page.

UAT:

- Given I have clicked on the Add Recipe tab,
- I will be able to input information regarding my personal recipe including the name, an image, instructions, and ingredients.
- After submitting my custom recipe, it will be added to the database, and I will be taken back to the Discover page. I will be able to see my custom recipe on the Discover page when I search for it
- If I do not provide all the necessary information for my custom recipe, I will be given a message stating that I need to fill out all input fields

Testing

- The user will attempt to add a new custom recipe to the database
- The user will attempt to submit their recipe without filling in the required information to see if the error message comes up
- Then, the user will input all the necessary information and submit their recipe
- The user will verify that their custom recipe showed up on the Discover page
- The user will be someone who did not work on the project so we can have a fresh pair of eyes to test if our systems are working as intended

Observations

- The user clicked on the Add Recipe tab and inputted their custom recipe information
- The user filled in all the information except for the image upload.
    - The user got a message stating that they needed to add all the required information
- The user filled out all of the required information and added their recipe
    - Their recipe was named “Cake” and used a stock photo of a slice of cake
    - The user added two ingredients, flour and sugar
    - The user gave brief instructions regarding how to bake the cake
- After submitting their recipe, the user was redirected to the discover page.
- The user searched “Cake” in the navigation bar, and their custom recipe showed up
- Based off the user’s actions, the add recipe page seems to be functioning correctly

Deployment: Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.

http://recitation-16-team-4.eastus.cloudapp.azure.com:3000/

The app was deployed using Microsoft Azure. We set up a Linux Virtual Machine and connected to it. On the virtual machine, we installed Docker and set up the SSH key for GitHub.