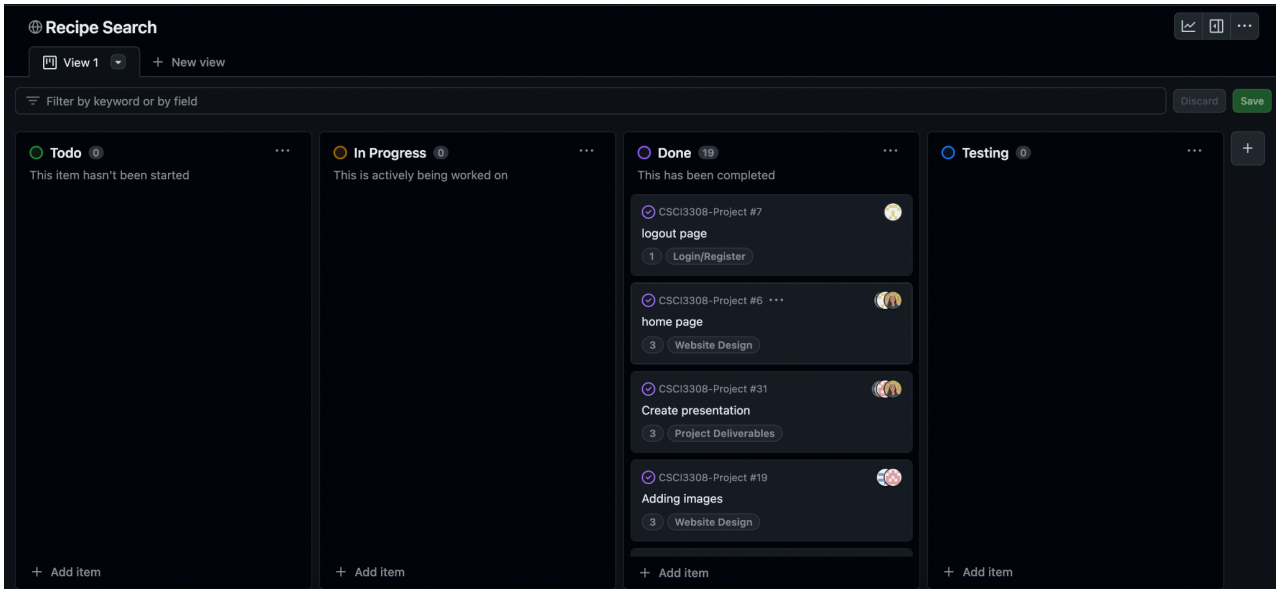
Title: Chef It Up

Who: Kaile Suoo, Charan Sai Kakula, Bianca Gautam, Zach Erdley

Project Description

Our application, Chef It Up, is a culinary companion that helps users sift through a multitude of recipes to find the perfect match for their culinary skills, busy schedules, and more. By offering a user-friendly interface, the application streamlines the process of finding dining recipes. On our page, users are able to search for a specific food they are looking for, cuisine, and meal type. Any allergies or diets can also be selected to find a perfect match for the user's needs. To try something new, check out the recommendations page which will show the highest rated recipes based on past reviews. The value of this application is that it will save users time and encourage users to explore while having fun in the kitchen. Furthermore, the application aims to be a versatile tool in meal planning and preparation. The intent is not only to cater to those who cook all the time, but also beginners who want to start cooking. With an emphasis on personalization, the app ensures that each user gets a unique experience suited to their individual taste, making cooking at home a more accessible and enjoyable endeavor.

Project Tracker: https://github.com/users/ksuoo/projects/1



Video:
https://cuboulder.zoom.us/rec/share/NtO0Wf_wNgD3rF64tOi7Som1kMJ6lxNs7serCifTh9oLd_-AqgxDJP4XY9qVn8kD.PKoE73ctzubsvBzE?startTime=1702010535000
Passcode: FnB&0Y33

VCS: https://github.com/ksuoo/CSCI3308-Project

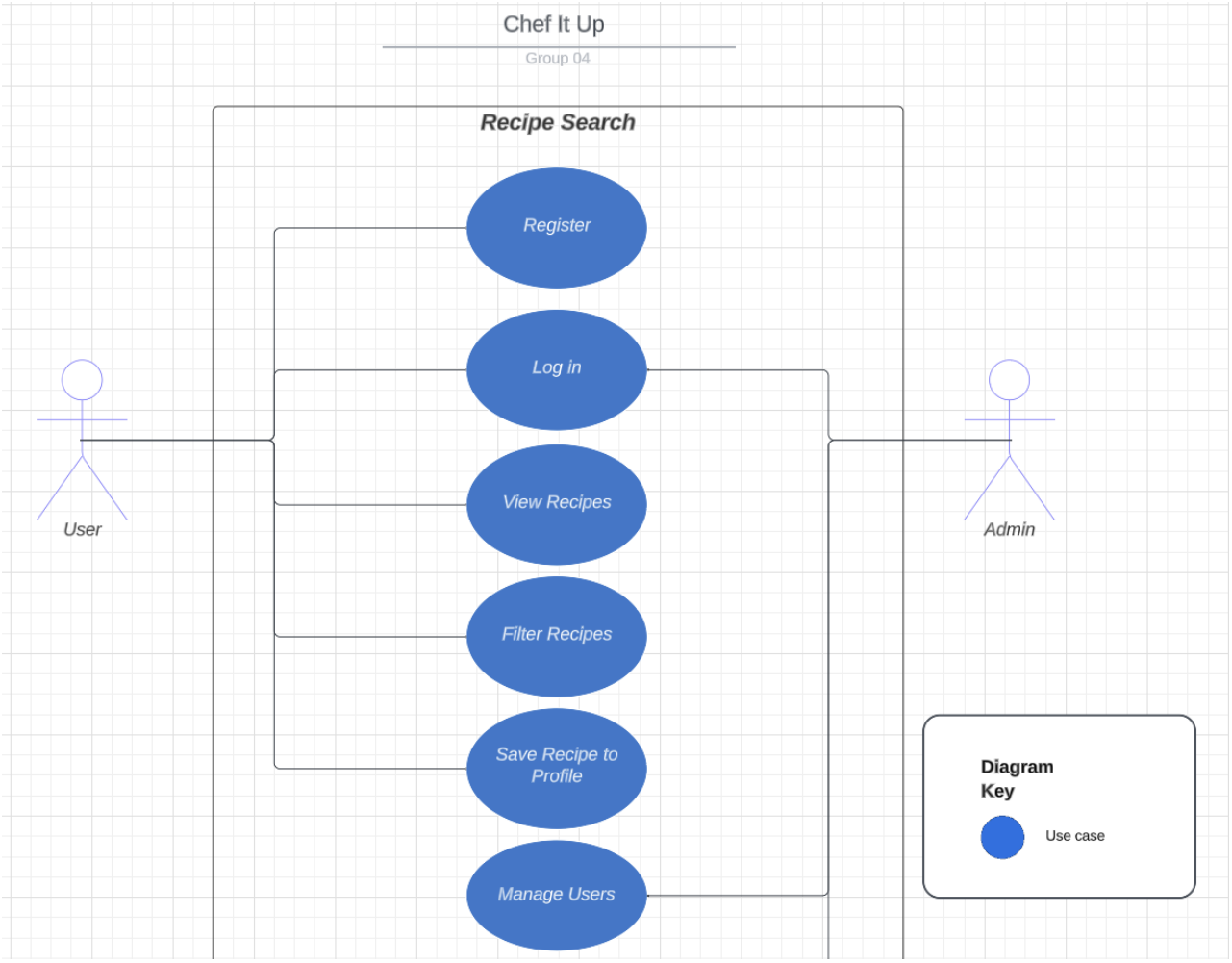
Contributions:
Kaile: I worked on the functionality of the pages on our website and wrote the routes for login, register, and the filter. I also worked with the API and the pages that used it. The first page using the API was the recommendations page which would sort the recipes highest rating and display them to the user. Another feature I worked on was allowing the user to select filters on the home page and find recipes based on their selections. Once the website was properly functioning, I worked on error messages and how the website looked.

Charan: First, I worked on putting together a bunch of recipes into a SQL file which could be used as our recipe database. However, our group scrapped that because we realized it was more effective and efficient to use a recipe API. I also worked on the CSS part of our website including making sure the text fit in well with the rest of the page and adjusting the navbar and footer to make sure the pages were visually appealing. I also helped Zach with inserting our icon into the navbar as there were issues with the path and the order of the files in the directory.

Bianca: The very first thing that I did was that I was getting the wireframe of the website made as well as figuring out the other logistics of the website. Throughout the project I was the scribe during team meetings with our TA Bhoomika. Later, I coded the recipe filtering using the SQL table that later got scrapped but used as a template as to how to do the API call. And at the end of the project I helped the student from the Cyber Security Class get the website working.

Zach: I worked mostly on the style of the pages, helping to design a graphic and make the pages flow feel better for the user. I worked a lot on a bug we had with images not being able to be loaded to the page, and was able to resolve it by changing the routing on the Docker file. I also had to totally restructure the file structure for the images to get them to containerize correctly within the Docker environment. Finally, I also worked on the menu bar at the top and made sure the pages it existed on made sense.

Use Case Diagram:



Test Results:

Below is a screenshot of the testing output when docker-compose up is run. The written out test cases can be found in the server.spec.js file.

```
  Server!
Database connection successful
    ✓ Returns the default welcome message
    ✓ positive : /register (78ms)
    ✓ Negative : /register. Checking invalid
    ✓ positive : /login (66ms)
    ✓ Negative : /login. Checking invalid ac
    ✓ Postive : /discover. Checking filters
    ✓ Negative : /discover. No recipes retur
```

Our written test cases were created for register, login, and filter (/discover).

- The positive register test case checks that the username and password are correctly input into the users table. The negative register case ensures that the user has a valid input and does not violate primary key constraints for username.
- The login test cases make sure that the user is able to login to the account they created, and if it does not exist, they will be redirected to register to make an account.
- The filter tests check that the recipes being returned by the query are displayed properly to the user and if there are no results, an error message will be displayed.
- Other tests we planned for in our UAT Plan were to make the website inaccessible unless logged in which we verified by typing http://localhost:3000/home in our browser which would automatically redirect us to login.
- At the time of creating our plan, we aimed to implement a feature where the user could save recipes, but we have changed it to a recommendation page which we validated to work properly as well.

Deployment: http://recitation-11-team-04.eastus.cloudapp.azure.com:3000/

To run locally, download the repository from the GitHub repository and have docker installed. Then run the following commands into your terminal:

1. docker-compose build
2. docker-compose up

Once docker is up, navigate to http://localhost:3000/ in your browser to view the page