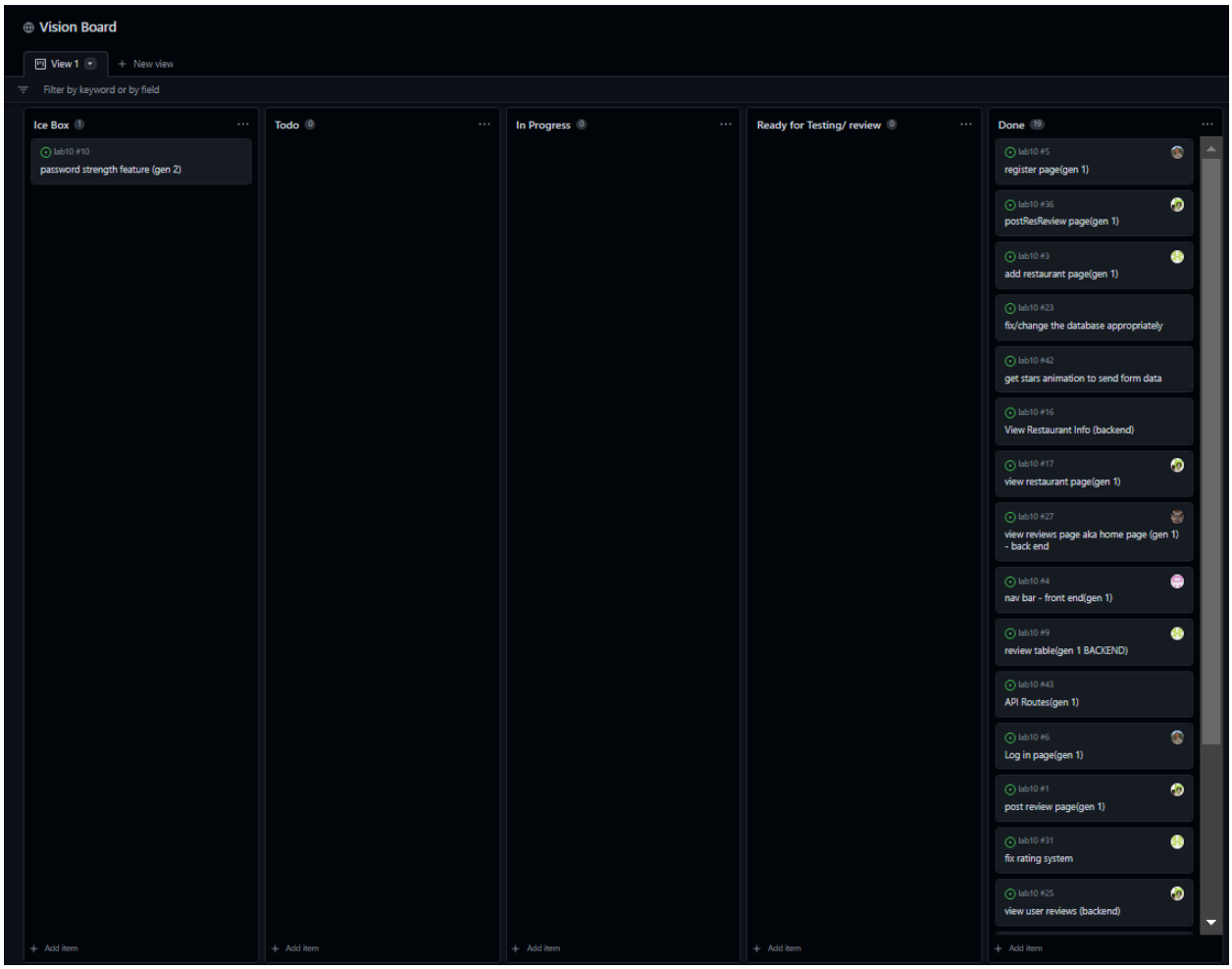
Food Finder

Team Members:
Jensen Lavering, Oscar Rhoades, Aria Barbour, Manas Gupta, Albashir Ali

Project Description:
Food Finder is a website that allows users to find restaurants, add restaurants, and leave reviews. This product uses a google maps api that allows users to see the exact location of the restaurants. Our easy user interface is pleasing to the eye and makes it easy for users to add a review or find a restaurant near them. Overtime with more users the more restaurants there will be on our website. This creates a system where growth is determined by the number of users rather than developer input.

Project Tracker:
(image of project board)



Demo Video:

VCS:
https://github.com/JensenLav/lab10

Contributions: (less than 100 words each)

Oscar: Wrote the frontend on view reviews and view restaurants that displays the restaurants and reviews on the map with leaflet. Integrated the google maps geocoding api with the rest of the webapp to convert addresses to geographic coordinates locations could be mapped.Wrote the SQL queries for routes. Used EJS and node querystrings to created informative error messages, designed reviews frontend.

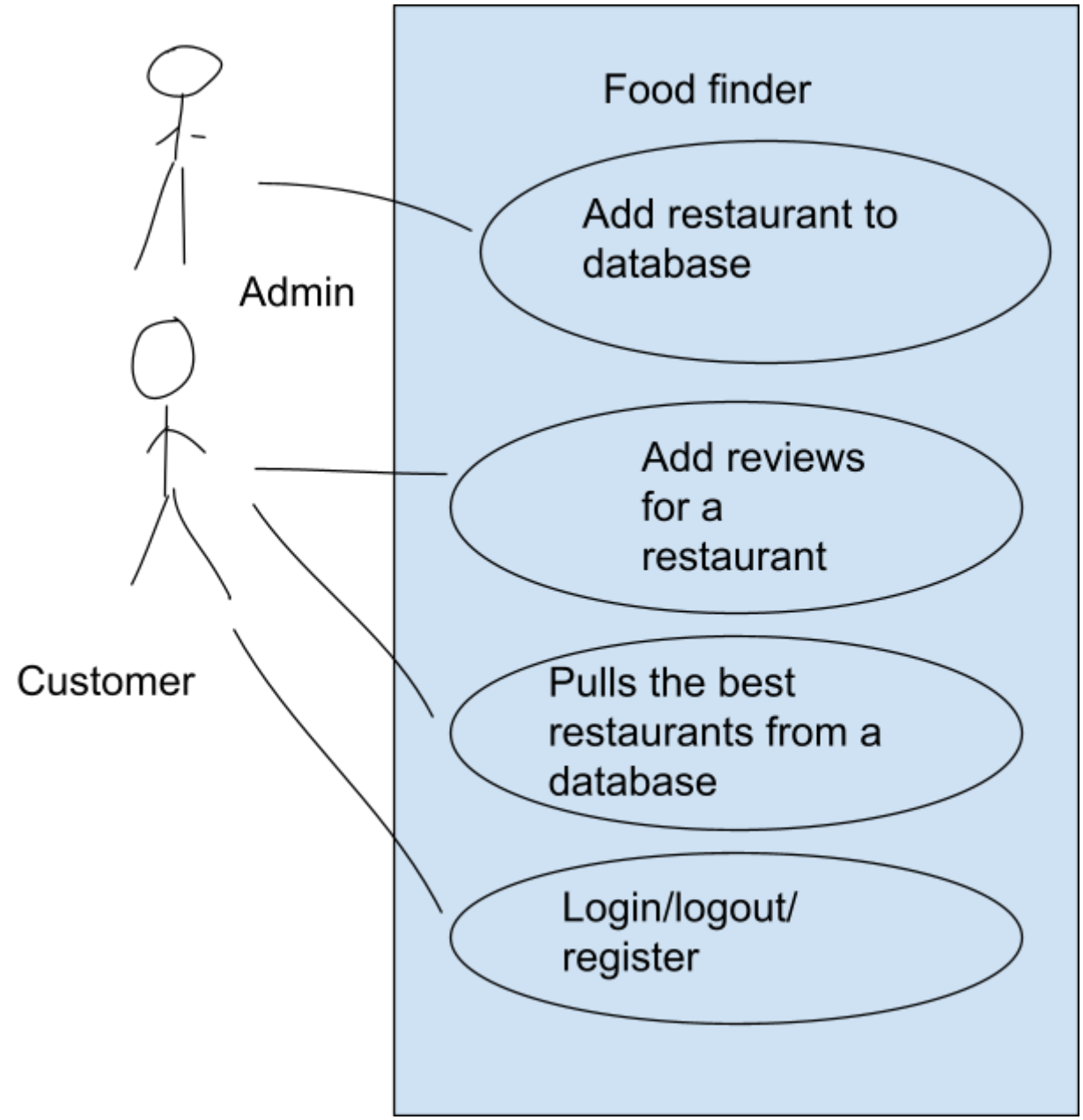
Aria: Wrote the front end for pages used in the project: postReview and viewRestaurants (not on maps). I also wrote the front end for postResReview which was like postReview but isn't used in the project; I ran out of time when trying to figure out how to send specific data to another page through a button. I also contributed to our index.js file for the corresponding pages. I also worked on the database. Some of the tools used: bootstrap, JS, HTML, CSS, JQuery.

Jensen: Wrote the original database before google api was added. I also fixed the star rating system that was not working. I also did the html for the footer.

Manas: Worked on front end for the display reviews page using HTML, CSS and bootstrap. Implemented Javascript logic into the HTML file using EJS, designed the navigation bar for our website.

Albashir: I created the full stack for the login and register page. I created the authentication for login and the rest of the website. I also worked on creating the footer and landing page. My landing page uses a script js file to help with the css and the buttons on the front end. I also used many API's to create the landing page including Google APIs and cloudflare. In addition, I stored the data from the register for the authentication and user data of the rest of the website. I also created the Header.ejs file.

Use Case Diagram:



Test results:

Testing login page:
Test case: When user enters the correct username and password
Result: User is successfully logged in
Test case: When the user enters incorrect login informatioN

Result: User is not logged in (no new session is created) and an error is thrown

Testing Register page:
Test case: User tries to login with a valid email
Test result: A new account is created and the information is pushed into the database

Testing posting reviews:
Test case: User inputs a review
Test result: Upon testing this case, we verified that this information is successfully added to the database

Observations:

- What are the users doing? - The users are logging in, creating an account, adding restaurants, posting reviews for restaurants, and viewing reviews of specific restaurants
- What is the user's reasoning for their actions? To find the best restaurant in town
- Is their behavior consistent with the use case? Yes
- If there is a deviation from the expected actions, what is the reason for that? No deviation from intended actions is found
- Did you use that to make changes to your application? If so, what changes did you make? We made no changes to our initial plan

Link to demo:
https://drive.google.com/file/d/1eUDyv3biunCm-zm8uY2ImCcEEJgaoHjS/view?usp=sharing

Deployment: http://csci3308.int.colorado.edu:49163/