Greg Lee (gglee2), Daniel Gleason (dlg2), Rendy Harris (rendyh2), Jeewon Koo (jeewonk2)

Group 12 Project Report: RecipEZ

1. **Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

Compared to our original proposal, our group did not make any major changes in direction of the project. Project goal, functionality we planned including data storage and functions of the website stay the same. However, our front-end user interface is slightly different from the UI mockup from the original proposal.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

Our application is successful in that it has many ways to search for recipes based on various criteria such as ingredients, ease of use, or average ratings from other users. This makes it quick and easy to find recipes which suit the needs of the user. It also provides a way to leave feedback on recipes, which can further help other users.

One element of the website that could be more useful is its “clickability”. We were unable to complete the functionality of being able to click on a search result to be brought to a recipe that contains more detailed information and instruction. That would be the first future extension we would look into.

3. **Discuss if you changed the schema or source of the data for your application**

The source of our data remained the same. We found the original dataset from Kaggle (https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions). The original dataset includes information of recipes and users. We pre-cleaned the original dataset to remove redundant information or attributes that we are not going to use in the dataset. We also utilized a pickle file in order to map ingredient IDs to ingredient names. After these steps, we divided datasets into smaller datasets so that we can modify the dataset the same as the ER diagram we designed.

4. **Discuss what you changed in your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**

We slightly changed the schema as we began implementing our database. The first change we made was moving certain attributes from the Recipe entity to other entities. The reason we made these changes was to better support the relational nature of our design and allow for more complex queries. We also added some attributes to store metadata such as the number of reviews a user has written.

5. **Discuss what functionalities you added or removed. Why?**

One of the main functionalities that we wanted to initially include was searching and filtering out recipes on multiple ingredients. But for the project we decided that we would filter based on one, due to the fact that multiple ingredients on different calls to the query would be really expensive. As new queries would need to be made and then intersected once more.

6. **Explain how you think your advanced database programs complement your application.**

Our advanced database programs (triggers and stored procedure) complement our application because they add further value to our ratings system and make our results more trustworthy. The triggers added, which keep track of inserts and deletes to the review table, give us more information about users. This information can be used to see which reviewers are more active and trustworthy, so we can treat their ratings as more reliable. Our stored procedure leverages this information to compute trustworthiness scores for reviews and display trusted average ratings to end users.

7. **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**

a. **Greg Lee:** One technical issue we faced was finding ways to improve the performance of our queries. In Stage 3, we did analysis of the effects of adding various indexes to our database using EXPLAIN ANALYZE but failed to find any significant speedup. Future extensions could include finding different ways to optimize our queries.

b. **Daniel Gleason:** Setting up the initial Flask server was difficult due to our team's limited web-dev experience, so I based our initial code off of a project I did in a previous class. Updating the dependencies and adjusting some of the boilerplate code was pretty difficult and frustrating, but after figuring it out together we were able to start making quick progress.

c. **Jeewon Koo:** When I try to connect a database server on GCP through our application, I always get a CORS policy issue. The detailed error message was "CORS policy no 'access-control-allow-origin'". This issue was resolved after I downloaded a package called "flask_cors" and implemented a few lines of CORS authentication such as "response.headers.add('Access-Control-Allow-Origin'. '*'"

d. **Rendy Harris:** One big problem was parsing the data from a csv file to edit and modify the data so that it could be easily added into the gcp, for the recipes csv file there were alot of parsing errors due to the abundance of punctuation that the data had. Another was generally linking the front end to the back end as our team was not that proficient and it was a new experience for us coding the front end.

8. **Are there other things that changed comparing the final application with the original proposal?**

As it is described above, our final application UI is changed compared to the original proposal UI mock. Specifically, we did not make a page for if the website cannot find any recipes. This is because the dataset we chose is huge so we assume that there is at least one recipe that matches with what ussr wants to search.

9. **Describe future work that you think, other than the interface, that the application can improve on**

As we stated in our original proposal, specialized filtering could be a useful feature. To be more specific, a feature that allows users to filter recipes by diet preferences such as vegan or by the culture of the cushion such as Asian food. This will be useful in terms of helping users to find a recipe exactly catered to their needs. Another feature that will make the application better could be keeping track of previously viewed recipes. This feature will allow users to return to a recipe they enjoyed or review ones they were interested in but have not tried yet.

10. Describe the final division of labor and how well you managed teamwork.

Ultimately, everyone played an equal role in designing and implementing our project. Much of the work was done concurrently on Zoom calls with all members present. In terms of offline work, Daniel and Greg focused more on backend tasks like database design and application implementation while Jeewon and Rendy focused on frontend tasks such as the UI of the website.

(Note: If GitHub commits are a factor, for some reason Rendy does not show up as a contributor on our repo but he did indeed contribute commits to our codebase)