Final Project Report

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

For direction, we initially wanted to just have the garden aspect for our project, and kept it to just having a user be able to make their own garden and compare it to other users who also used our app. But as we thought more about it, we wanted to make more features for the individual user to play around with. Due to this idea, we decided to add a feature that would allow users to be able to see how much they save by growing their own produce in comparison to buying it from popular vendors.

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

I think our application succeeded in allowing users to create their own mock gardens, given the statistics of where they grow. We also incorporated a profit section that allows users to know how much they would save regarding which vendor they could purchase their produce from, compared to growing it themselves.

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

Because our application required quite specific data, including items like soil element percentages, there were few data sources online that fulfilled the scope that we had envisioned. Because of this, we chose to create our own data with random data generation.

Discuss what you change to 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?

The main changes to the ER diagram and table implementations were the plants table and user table. Due to time constraints, we were unable to implement the user login and account features, and decided to just have one table called garden_instance in the backend that kept track of the plants. We also had to change the garden table to accommodate this change, and added additional columns such as having a unique ID in order to allow for easier deletes. The original design would be much more ideal as it would be able to accommodate many users.

Discuss what functionalities you added or removed. Why?

Some of the features that we removed were the interactive map and the additional search functionality that would allow users to look up a location or crop and return the best crops or location to grow. While we did have a function in our backend that would return crop recommendations based on state, we were not able to fully implement this feature due to time constraints.

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

We added a database constraint on the garden_instance table to ensure that no duplicate crop/city pair is inserted into a user's garden. If a user tries to add a duplicate pair into their garden, the app will display an error message reminding them that they have already inserted that specific crop/city pair in their garden. We also implemented a stored procedure to calculate the garden profits automatically in GCP instead of sending a manually entered query to GCP whenever we wanted to get the profits, which simplifies and improves the performance of our backend. We were unable to implement other advanced program features due to time constraints.

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.

Kenny: One of the technical challenges that I faced when working on this project was ensuring the synchronization of crop selections across multiple components. For example, when a user selects a crop, it needs to be reflected in both the GardenManager component and the CropSelector after a page reload. I was able to circumvent this by syncing the selected crops and garden crop states in the parent component, which ensured consistent data flow across components.

Alyson: One of the biggest issues that I faced during this project was dealing with react states. As someone who has not done much web programming before this semester, I found it hard to predict when states would cause rerenders and when or what I should use dependencies for the useEffect hooks. At one point of time I crashed the frontend because it was stuck in an infinite refresh loop. What helped me the most was breaking my react components into smaller ones, and having minimal hooks at the top level elements. For example, the search bar was originally part of one of the higher level elements called CropSelector. The search term that the user types was a react state, so every time the user typed, It would re render the whole page. I moved this state and the search bar into their own component and passed in the necessary hooks via parameters.

Sam: One of the challenges I had a hard time figuring out at first was creating some of the advanced program features in the cloud and connecting them to our existing backend implementation. Before starting this project, I didn't have much experience with SQL integration in a React application. And since we were using GCP for our database, it only added another layer of complexity to the problem. After logging in to Google Cloud Shell to interact with the database, I tried to add a stored procedure but faced some issues. It was mainly an error that said the procedure couldn't be added because some weird conditions weren't satisfied. After some debugging, I figured out that it was due to the garden instance not being empty. I also faced a similar issue when adding database-level constraints in GCP and solved it by emptying my garden instance again. So, for some reason, adding advanced program features to the app works best if you empty/reset your garden_instance table from crops first.

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

One part of the project that we were unable to complete in time is the display on the map as to where a certain crop would grow best. We deemed fulfilling the foundations of the user experience of creating their own garden to be of higher priority, and were unable to complete this section in time.

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

Future work would include finishing some of our unimplemented features, such as the map, dual search recommendation, and user account information and authentication. Something that wasn't included in our original project report but would be a nice feature would be allowing users to collaborate on gardens.

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

Group Dynamics: We mainly communicated through discord and gave each other updates and suggestions about how to move the project forward. We also meet up during workshops and vacant class periods.

Division of Labor:

Kenny: created initial frontend setup, added insertion and deletion of crops in garden, created crop search feature, connected user interaction with sql queries for database updates

Alyson: created initial backend setup, created city search feature and vendor selection, created selected crop information preview and submit, created edit crop vendor feature, created Profit Feature

Sam: worked on frontend/backend optimizations, created a stored procedure in GCP for profit calculations and connected it to the backend, added database-level constraints in GCP for adding crops to the garden, enhanced error handling to improve UI/UX