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).

Our final project's direction is the same as the one in our original proposal as based on our stage 1 proposal submission.

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

It is easy for the user to view perfumes and recommendations based on price in two ways- the recommended perfumes after the quiz inform you of their price and organize themselves according to how much they match you, and the keyword search page where it shows the average price of the dupes of each search result. It also has a robust authentication system and the ability to retake the quiz to get new recommendations for perfumes in case mood or life situations change. However, we don't have a review system in the website, which means users must go off of perfume price and scents.

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

We did not change the data source explicitly, but we had to auto-generate some missing chunks of data, such as user data and some dupe data, since we had to identify dupes manually and create those relations.

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?

We added a new table called UserRecommendedPerfumes which acts as a storage space for all the perfumes that are recommended for a user as per their quiz. This table has a primary key of UserId and a foreign key of PerfumeID that references PerfumeID from Perfumes. We added it to be able to access user-tailored perfumes more easily and display them simpler, since storing it in a table means the query to display it would be less complex, requiring less compute. We believe this to be a more suitable design because it's less intensive on the database and the user as they wouldn't have to retake the quiz multiple times in order to receive their matches - especially if there wasn't any changes in their choices.

Discuss what functionalities you added or removed. Why?

We stayed pretty close to the original proposal when it came to the design and implementation of our final project. We kept the design quite similar and the key difference is that we did not implement the graph of user recommended perfumes based on price. We did not implement the graph of user recommended perfumes based on price because we believed that it already is easy for the user to view perfumes and recommendations. We also did not implement the reviews page because our review dataset wasn't extensive and the reviews were unsubstantiated.

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

Our advanced database programs complement our application by providing a framework for us to do the computation required to display perfume results for the user. Our stored procedures allow us to efficiently display the dupes of a perfume in the recommended perfumes section, as well as display average dupe price in the search results. These are important as the user can easily find the dupes for the perfume they are interested in, and are able to gauge the dupes average cost easily to make a determination whether the dupe or the actual product is better suited towards their needs. Our trigger update the UserRecommendedPerfumes with top matches (determined by MatchScore) based on the user's preferred notes and feelings each time those values are updated, which happens every time they retake the quiz and alter their choices. This allows for easy feasibility in automating the process of updating the new recommended perfumes when the values change. For the transaction, where the user is able to get a suggestion of what popular notes to search for based on the most seen in the reviews from the top reviewer which is proving validity. We think this complements our application because it allows the user to combat being unsure of what exactly they want, a common predicament, attempted to be solved by both the quiz as well as this feature. Another advanced database program we made use of was constraints. We implemented constraints not only in the form of primary/foreign key relationships. We also implemented a constraint that allows us to delete users safer. More specifically, on deletion of a user, we also delete any row in the UserRecommendedPerfumes table that references this user. With this, users who wish to delete their account can remove their information from every part of the application.

.

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.

Rithvik: One technical challenge we encountered was issues with authentication and permissions on GCP. For example, when attempting to write triggers into our mySQL database, we encountered a permissions error that said: ERROR 1419 (HY000): You do not have the SUPER privilege and binary logging is enabled. We fixed this issue by searching the error code and finding out that we needed to enable a flag in order to give ourselves permission, despite the fact that we were all administrators on the mySQL instance. We solved the issue by editing our mySQL instance, and under flags, enabling the log_bin_trust_function_creators flag which gave us the permission to write triggers and other procedures in our application.

Richie: Another technical challenge we encountered was that of writing clean code that was easy to navigate. We decided to jump right into the coding of this project in React.js immediately. An important aspect of React is the use of modularity and components that enhances code navigation and readability. Since we jumped right into coding, we encountered issues in making sure the right files were being updated, finding the appropriate file to modify in the frontend and backend, and debugging the code for issues whether it was on the database side or not. Looking back, planning the structure of our components definitely would help, especially with passing React States from one component to another to ensure all pages and components were appropriately accessible.

Lathika: Another technical challenge was retrieving the data from multiple sources as well as generating the data ourselves. We found that a couple of our sources had misaligned categories that wouldn’t match up, which initially made it difficult to gauge all the values in relation to each other. We also didn’t have a couple of necessary datasets like users, and didn’t have sufficient data for a couple of our dupes, which required us to make our own python scripts to do so. This posed a challenge as to how to maintain diversity within the data itself. However, we were able to mostly combat this by pulling certain notes and feelings from a bank that we came up with.

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

We did not have any other major changes other than the omission of the graph or the shelving of the reviews.The main thing we did was add a UserRecommendedPerfumes table as we found that it made it made more sense to hold the previous results of the quiz if the user wasn’t interested in retaking the quiz each time.

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

One of the main improvements that could be made to the application is to add live price tracking so that users can see the ebbs and flows of the perfume market and make more informed purchases. On the same vein, another improvement that could be made is to include links to vendors for perfumes so that users can purchase the perfumes after being recommended them instead of needing to go and search the web for them. Another possible improvement is to have an AI model that asks new questions each time the user takes the quiz so that they can still update their preferences while also having the feeling of novelty, as well as being able to provide a more accurate match score. We can also include a direct purchasing feature so that users can purchase from our website itself, leading to more ease of use for the user, and potential partnerships with companies in the future.

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

Our final division of labor was fairly equal, with everyone working on a bit of everything- Rithvik worked a lot with GCP hosting and React frontend and backend linkage, Richie worked a lot on React frontend, backend, and testing, and Lathika worked a lot on SQL advanced database programs and backend. We handled teamwork very well as we are all friends and were able to hold each other accountable.