CS 411

SPRING '24

FINAL REPORT

Swipr

Pranav Nagarajan

Vikram Battalapalli

Raahul Rajah

Ritvik Manda

Changes from Original Proposal:

Along the course of the project, there weren't any significant changes to the core idea of the project; most of the differences were small design choices and minor adjustments. For instance, we initially had the customer choose among more clothing type options including t-shirts, button downs, polos, shorts, pants, jackets, hoodies, zip-ups, etc. However, we removed that since we wanted to focus more on the clothing color rather than the types of clothing. Thus, the customers can choose a more general clothing genre, such as tops, bottoms, and outerwear, just not the specifics. Additionally, the users can now filter the reviews of items so they can see the most popular items. This additional functionality gives the user more ways to filter through the clothing items to get the best recommendations. These tweaks helped refine the project's appearance and functionality without deviating much from our original proposal. So, while there were some modifications made along the way, they were primarily focused on enhancing the project's overall quality rather than changing its fundamental idea.

Usefulness of the Application:

Our application has succeeded in fulfilling all the essential requirements of an online clothing platform, closely aligning with what we initially outlined in our proposal. It effectively meets the needs and expectations we set. Users can seamlessly browse, select, and filter out clothes that are suitable for their skin tone. Additionally, our website offers an interactive like-dislike feature, which makes the shopping experience more dynamic and engaging. After liking and disliking, the users can view their likes and dislikes and manage their choices, which is very useful. Users can also select clothing based on filtering the reviews such as setting price range, sustainability level, brand, minimum star rating, etc. This allows users to search through the filtered reviews and select a clothing item based on their liking.

Changes in Data Schema and Source:

We made enhancements to our application's data structure by incorporating additional features into the schema. Notably, we introduced a profile picture field as a medium blob in the customer table, improving user profiles with visual elements. Additionally, to strengthen data management and accountability, we implemented an audit log feature to track additions, updates, and deletions within the Customer table. These improvements make our application work better and become stronger, ensuring that our data is well-managed and secure. We maintained all other aspects of the data schema and source, using the four original brands of clothing for our app since they are very popular amongst our target audience and using the original data for the reviews as well, since we did not find any need to change them.

Changes to ER Diagram and Table Implementations:

We kept the ER diagram largely the same as what we had before because we spent a lot of time improving it in the previous stages. Our original diagram seemed to have all the right parts for our application to work well. Sticking with the same general plan made things easier for us during development. The only change we made was creating an Audit Log table which kept track of any creations, deletions, and updates of the Customers table. This was to keep track of the users of the site in case of further future development which would need this info. Additionally, we saved the user's profile picture directly within the database in a medium blob format. These changes required the creation of some triggers and foreign keys but the functionality of the app and the data schema remained almost the same as before, leading to a very similar ER diagram.

Functionalities Changes:

While we mostly adhered to our initial plans for the app's features, we made one significant change. Initially, users had the option to select from various clothing categories such as t-shirts, button downs, polos, shorts, pants, jackets, hoodies, zip-ups, etc. However, during the development process, we decided to revise this approach. We chose to prioritize assisting users in finding the perfect colors of clothing rather than categorizing them by type. Therefore, we opted to remove these subcategories. This adjustment allows users a broader range of options when browsing for clothes, ensuring a more flexible and personalized shopping experience. Overall, the rest of the functionalities described in our original proposal along with the additional review filtering are fully integrated.

Enhanced Database Integration:

Our advanced database programs are integral to enhancing user experience and smoothing out interactions. By implementing filters for reviews based on brand, sustainability level, price, minimum star rating, and more, we ensure users access relevant information aligned with their preferences. Similarly, our database enables filtered clothing searches by college, brand, price, and other parameters, simplifying the browsing process. Additionally, users can easily modify their profiles, updating information to reflect their evolving preferences and styles. Moreover, the infrastructure we created uses procedure implementation, ensuring efficient inventory management and profile management. Additionally, we allow administrators to perform CRUD (Create, Read, Update, Delete) operations, maintaining the integrity of the platform's data and functionality. These features collectively contribute to a user-friendly platform that caters to diverse needs and preferences.

Technical Challenges Encountered:

Vikram Battalapalli: During the development of the skin tone algorithm, there were a few challenges in migrating from python to NodeJS. For one, I had to transition our logic to use

Sharp in JS and implement the weights from our model directly into mathematical equations to match skin colors to clothing colors. Additionally, I had to build the api pipeline to send the user's profile picture to the backend using Base64 format and store this in our database in mediumBlob format, which took additional data processing steps. Additionally, throughout our development process, there were issues with communication between our front and backend through our api gateway. These required debugging and coming up with efficient ways to bring data from our database to the screen so users could interact with the application in the desired manner. For example, we used a state management system using a Context API to save the user's login session and propagate their likes and dislikes of the clothing through multiple sessions. Additionally, I implemented various error handling methods into our frontend and backend to make it robust to any problems in the api gateway and have a complete functional app.

Pranav Nagarajan: Transitioning to front-end development presented a significant challenge for me. As someone with a background in backend and algorithms, understanding frontend technologies like JavaScript, HTML, and CSS was a learning curve. Integrating our backend functionalities into the front end and mastering deployment processes on platforms like Google Cloud Platform added complexity. Through self-learning and guidance from teammates, I navigated these challenges, ultimately gaining a deeper understanding of full-stack development. I made sure to create some personal documentation to make things easier for the next time I try to build a web application.

Raahul Rajah: We had some technical challenges when creating the CRUD functionality for the app, especially when creating our Admin page. On the Admin page, we could delete customers as well as delete, update, and add new clothing items. Full-stack development and using a MERN stack with My-SQL was something that I was not familiar with before creating this project, so it was challenging to figure out the routing between the front-end React and the back-end to connect the buttons on the screen to the actual SQL queries. I had to use numerous online resources to understand the routing and implement the routing required for the pages of our website. Initially, I tried to delete customers from the customers table using our front end. However, I was not able to because of a foreign key constraint. We then had to modify our original schema to add an on-delete cascade, because our reviews associated with deleted customers also needed to be deleted. Next, when updating the clothes, every time I clicked the update button the clothes table was updated with null entries for the specific clothing ID I was modifying. This was because I was not formatting the information correctly when sending it to the backend to run the update query. After fixing these issues, the CRUD functionality was updated, and I learned a lot about full-stack development through this project.

Ritvik Manda: A major technical challenge I encountered was, once we were able to connect to the GCP SQL database, how to exactly extract the information from the database through SQL statements. A simple SQL statement was not enough, and the connection itself was a lot more complex than it seemed. To implement this, we had to do numerous research on the GCP framework and with this being my first time working with React, I had to understand syntax and everything related to the issue.

Changes Beyond Proposal:

Our final project was almost mostly similar to the original proposal. The only changes have been in how the app was structured. For instance, we removed the need for super likes since it logically did not make sense to us. For instance, a person can still decide whether or not they want to buy an item from either of the likes or dislikes list, since we have the freedom to do so. As a result, it would render the super like useless. We also added an Admin page to demonstrate CRUD functionality, which was successful. However, in our original proposal, we did not mention this page. Overall, the bulk of the proposal remained constant with tweaks here and there.

Future Development Pathways:

Looking ahead, there are several areas where our clothing app can further improve beyond just enhancing the interface. One significant enhancement could be integrating the entire shopping experience directly into the app. Presently, users can browse, select, and interact with different clothing items seamlessly. However, to elevate the user experience to the next level, we envision incorporating the actual purchasing process within the application itself. By enabling users to complete transactions and track order details directly through the app, we can provide a more comprehensive and streamlined shopping journey. This would not only enhance convenience for users but also foster stronger engagement and loyalty to our platform. Additionally, integrating features such as personalized recommendations based on past purchases and preferences could further enrich the overall shopping experience, making it more tailored and enjoyable for each user. Thus, by focusing on enhancing the functionality and usability of the app beyond its interface, we can continue to evolve and provide an even more compelling solution for our users.

Team Dynamics and Division of Work:

In our team, we had a clear division of labor that allowed us to efficiently tackle each aspect of the project. Rahul took the lead on tasks related to data processing and organization. He successfully cleaned the data obtained from clothing websites and structured it into datasets categorized by color and type. His attention to detail ensured that our datasets were well-organized and ready for further analysis. Vikram focused on developing the AI model responsible for extracting skin tone and tint from images. Additionally, he worked on implementing the algorithm to determine preferred clothing colors for users based on their skin tone and finding the dominant colors in the clothing items. His expertise in artificial intelligence greatly contributed to the accuracy and effectiveness of this feature. He also oversaw all of the full-stack development and built the api gateway to communicate between the UI and the backend. Ritvik played a key role in implementing the UI functionality of our application. He ensured that users could seamlessly interact with the platform, including features such as sorting clothes based on preference and managing custom lists. His dedication to user experience resulted in a smooth and intuitive interface. Pranav was responsible for implementing CRUD functionality and handling tasks related to creating, updating, removing, and deleting items within the application. His contributions ensured that users had full control over their lists and could easily manage their preferences. Overall, our team collaborated effectively, with each member bringing their unique skills and expertise to the table. Through clear communication and coordination, we successfully integrated our contributions to deliver a cohesive and feature-rich application.