Criterion E: Evaluation

After finishing the product, I showed it to my client and we had another interview. This interview can be found in Appendix A. The success criteria were met:

1. The user interacts with a web application: **Fully Met**
2. The user fills out a form to input information: **Fully Met**
3. The user can input new pieces of clothing into the database: **Fully Met**
4. The user can delete pieces of clothing from the database: **Fully Met**
5. The user can update pieces of clothing from the database: **Fully Met**
6. All the clothing can be displayed in a table on the web application: **Fully Met**
7. Outfits can be generated from clothing in the database with specific color schemes and/or weather conditions: **Fully Met**
8. Outfits can be saved in a database and edited or deleted later: **Fully Met**
9. The client can make her own outfits manually: **Fully Met**
10. Outfits can be displayed in a user-friendly manner that allows the user to visualize it: **Fully Met**

While developing the program, the biggest change I would make is to use classes rather than parallel arrays, because they are more descriptive than the indices in parallel arrays. Additional features that I would add in the future include sorting the list of clothing with different parameters like color and changing the dropdowns to user input and looking for a match to the user input. The client additionally gave me feedback for the program. She valued the user interface and how she could add clothes to the outfit by an add button, but noted a few details that could change.

Firstly, the rating system was awkward. I included it to have a way to sort generated outfits, but the client thought that it was “unnecessary” and did not like having to fill in that field. There were also a few bugs with the program. When entering an outfit, each time one adds an item, the name field resets. My client suggested moving it “to the bottom”, but I think I can get it to not reset. A few buttons also did not fit on smaller screens, as I had designed the interface on a large monitor. My client would also like more information fields, like the clothing’s print. Overall, my client seems pleased with the program, which meets my client’s needs and provides her with a tool to plan and create outfits faster.

Improvements:

- Replacing the rating system with other ways to sort outfits, and having more ways to sort outfits. For example, on the outfits table, the user could choose to sort the outfits

by name alphabetically, by the color from red to purple, by the weather suitability from hot to cold, or by clothes that contain a particular accessory.

- Adding more fields to describe clothing. The client particularly felt that information about the pattern is important as it helps coordinate outfits. A change I would make is to allow the user to add information to show up in dropdowns, like additional colors that I did not include.
- Making the interface more friendly for smaller screens. The tables on the websites have a lot of information to display on a single row, which does not look good on screens smaller than a laptop. One way to fix this would be to use one column for color, style, fabric, and weather when screens are smaller, and creating scrollable cells.
- Creating an account system. Each user has an account with the user's own data, allowing multiple users to use the same application at the same time.
- Replacing parallel arrays with classes. Instead of having parallel arrays, which can be confusing to work with as their indices are not descriptive, object-oriented-programming will improve the readability of the code and making it easier to edit in the future.

Word Count: 496