**Matchme:**

**Course Project Report Phase 1**

Jennifer Mendez

Student Number: 200454895

Department of Technology & Visual Arts, Georgian University

COMP 1006: Intro to Web Programming Using PHP,

Professors Jessica Gilfillan & Shaun McKinnon

February 27, 2021

Matchme is an interactive web application meant to work as a digital version of the user’s closet. The user will upload the main characteristics (name, image, type, season, etc.) of each item in its closet. Once the information uploaded, the website will serve as an interface where the user can log how many times each item is used, add comments, and ultimately, have an application that will easily display and classify all of the items that the user has. This application eliminates the need to rummage through the user’s closet which in turn, means less time spent dressing, a more organized room, and the possibility for more innovative clothing combinations.

The main challenge when starting the application was to decide which were the main characteristics of a clothing item. For the application to be useful it had to detail the most distinctive elements of a clothing item while still being general enough to allow for a variety of items. Several iterations of the table in the database were created until I reached the one that best fulfilled the requirements.

The entries on the table are the following:

1. Item ID: Used for database maintenance purposes, this is hidden from the user. The item id gives the ability to edit and delete specific rows from the table while having a name distinctive enough to make it easy to recognize while coding the different id parameters. Originally this column was named “ID” but it made it hard to distinguish which id I was referring to when adding the functionality to delete and edit the table entries. Because getting the right id is vital to ensuring the table has full functionality, the column was changed to a have more identifiable name.
2. Item: Name of the clothing item, makes it easy for the user to distinguish if the item is a skirt, a dress, etc. This was left as a text input to give the user more liberty when choosing item names.
3. Season: Winter, Fall, Spring, or Summer. Makes it easier to classify items of clothing so the virtual closet is easier to navigate. It also makes it easier to create clothes combinations because it provides a guideline for the application. This was set as a select input for ease of use.
4. Type: Provides broad categories for the individual clothing items. Allows the application to identify the general purpose of an item (top, bottom, outerwear, etc.), makes the items easier to classify and combine. Also set as a select input for ease of use.
5. Color: One of the main characteristics of clothing items, makes it easy for the user to distinguish between items of the same type (for example a black skirt and a pink skirt). Used as a text field to provide flexibility for the user (set a particular shade of a color, indicate if the item has a pattern, etc.).
6. Occasion: Sets the level of formality fit for the clothing item and categorizes clothes into the appropriate dressing code. While there are more dressing codes than the ones listed in the application, I decided to consolidate them into less categories. I only used the most common dressing codes (casual, formal, etc.) because for example, the possibility of every user having a white tie ballgown in their closet is minimal. Less options make it so the user does not get overwhelmed or confused while using the web application.
7. Times worn: Takes a numeric value. This input is meant to help the user determine if they really should keep the item in question. When sorted by number of times worn, it makes it easy to see that whether a particular piece of clothing is really being used to its maximum potential. This allows the user to either try to incorporate said item into more outfits or get rid of it to make space for more wearable clothing pieces.
8. Comments: This is an open text input so the user can make extra annotations about particular pieces of clothing. For example, to note that certain shoes are uncomfortable, of that a pair of jeans has stretched too much. Paired with the number of times worn, it allows the user to assign a real value to the item of clothing and determine whether it is worth keeping.
9. Email: utilized for now, to distinguish which entries belong to which user. When the ability to log in with an email and password is incorporated, this column will not be available to the user.

Another challenge while working on the project, was to find a way for the select items to echo the corresponding data while editing. While text inputs can echo the user’s previous choices through the value tag inside the input, the same cannot be said for the select tag. To overcome this challenge, I created a first option with a blank value that would be hidden once the user clicked on it. This option had logic built into it that would attempt to retrieve the pertinent value from the database. If said value was empty (if the user was adding an item for the first time), then the first option would echo “Please select an option”. If the value was not empty, then the first option would show “Previous choice: $choice”.

The following steps are focused towards adding more functionality to the application and enhancing the user experience. For example, the ability to add an image and associate it with a particular row from the database, which will make the webpage more visual. And, of course, the creation of a log in page, and the ability to safely store and retrieve user information is another vital function. When this function is incorporated, and the user is logged in, only the information pertaining to that particular user will be available.