

I started with an idea for this project that I came up with over the summer and wanted to try to make at some point. I worked at a store that sold Korean beauty, so I wanted to make a program that would help find appropriate products for customers. The program would sift through all of the products that the store had and present those that would match the skin issue that the user wanted to address and the type of product that they were looking for. Ideally in the end the program would only show the products that were listed under both categories, so both of the user inputs would be taken into account and matched up.

The first step to approaching this project was to create two databases to work from. I started by making two spreadsheets: one relating to skin issues and the other regarding types of products. Then, as tedious of a process that it was, using the store's website I went through each product that was listed and entered the data into each of the spreadsheets where their appropriate place was. For example, I entered products that were good for people with sensitive skin into a column entitled "Sensitive" and similarly products that were moisturizers to be placed in a "Moisturizer" column. Once these were completely filled out after having gone through the entire website twice for each spreadsheet's data, the next step with the databases was to convert and save them as csv files. By converting them to csv files, I could read them into PyCharm to then work on them and extracting data from them to be used in the rest of the coding project.

The next step in the project process was to start working in PyCharm with code. I first made sure that both of my csv databases were successfully within the PyCharm directory that I made for the final project. Then, I began the coding by importing the csv reader into my py file for the project, so that the csv files could be utilized. Next, I needed to extract the columns from the first database on skin issues. To do this, I used dictionary methods because from the research that I did to figure this portion out, dictionaries seemed to be the best way to extract columns from csv files. Within this dictionary section, I used the dictReader method on my csv file that I opened. Then, a for loop was used to loop through the columns and rows and appended them together. This created a means of working with the data in each column.

Then, I figured that the following step would be to create variables for each of the columns in the database, so that they could be easily retrieved later on. So, I created a variable that corresponded to each of the columns.

Next, I needed to actually create the space for the user to input what skin issue they would like to address. I simply created a variable and set it as an input function followed by my question. I also tacked on the options that were available to the user, so they would just type in exactly what I needed them to. Beneath this input variable, I created a decision tree of if-elif-else statements. Thus, if the user typed in an option, the column associated with it would print. As an aside, I added a couple of lines of code within each decision statement just for stylistic purposes so that the results would print in a clean vertical line and without any empty items.

After this, I repeated the same process for the second database. The csv file was imported, a dictionary method used, user input variable created, and lastly the if-elif-else statements were included for the corresponding data and columns for types of products.

The last major section of the project was what really challenged me, more than I initially thought it would. The final step was to figure out how to combine the results of both user inputs and print out only the results that were in both of the lists of products in each database. I researched the best ways to try to match outputs. I tried using the regex match method, but only got character matches. Then, I tried to find out how to set each of the input result lists equal to a variable in order to then just match the contents of those variables. During the process of figuring this out, I tried implementing a defined function around each set of user inputs and their corresponding decision tree statements. However, this only returned the entire chunk of user inputs and decision statements. I only wanted the results of these processes. Finally, I tried creating a new variable and set it equal to the list method that I performed within each if-elif-else statement. I had one variable for one database information and another for the second database. Then, at the end of my code, I put in a for loop for the first database variable. Within that, I nested another for loop for the second database variable. Inside the second for loop, I placed an if statement for the two variables equalling each other and then printing the result. This for loop method finally worked for me.

This project turned out to be more challenging than I initially thought it would be, which I think is great. I was also able to utilize different things that we learned throughout the semester, so it was nicely comprehensive. I am very pleased that I had the opportunity to put this program into action after first thinking of the idea even before class started and I had no idea where to begin. I am happy with the finished product and I look forward to trying to recreate it in SQL as a challenge for myself during winter break.