Project Description: Nutrilog is a nutrition tracker that stores and analyzes your food data based on what you ate each day over the course of a week. NutriLog also calculates your daily caloric needs based on user information and develops sample meal plans based on food group preferences.

Competitive Analysis: MyFitnessPal is the industry leader in tracking a user's food data. My project will be similar to MyFitnessPal in that I will track the macronutrients that the user consumes by having the user type in the food he/she eats. However, my application is different because it presents a direct meal plan that the user can substitute into his/her day to reach the necessary amount of calories needed for the user's goals (lose, maintain, or gain weight). MyFitnessPal contains blog links to new food recipes and occasional meal ideas, but does not present sample meal plans to users.

PlateJoy is an application that aims to create custom meal plans for a user based on his/her lifestyle, goals, and nutritional needs. It has a wide variety of meals, but does not have a tracker for the user's food. Overall, my application tries to combine the key features of a nutrition tracker and meal plan creator.

Structural Plan: The finalized project will consist of five files. main.py will include the bulk of the project, where all the graphics and user experience features are contained. It will call functions that exist in the other files. nutritionReceiver.py is a small script that contains the web driving function that uses Selenium to receive nutritional information from the user's input. nutritionCalc.py contains the functions necessary for calculating daily caloric needs, storing my custom database of meals for breakfast, snack1, lunch, snack2, and dinner, and for creating meal plans based on my database. I plan to create one more python file that reads and writes to a CSV file, allowing the user to see his/her food data over the course of the week.

Algorithmic Plan: There are three key algorithmic parts to my project. First is the algorithm for obtaining food data based on what the user ate. To do this, I used Selenium web driver to open nutrionix.com, a food database, with the user's food added to the search query in the URL. To obtain the nutritional information, I used XPATH to find the relevant information, allowing me to click on the desired food and return the nutritional element. Second, I need to analyze the user's food data. I plan on doing this by writing and appending to a CSV file after the user finishes his/her day of eating, and then by reading from that CSV file to create a bar graph for each day's nutritional content over the course of a week, labeled by day. Third, I created a recursive function to develop meal plans for the user, based on the necessary number of calories needed per day (calculated by a formula with user information). The trickiest part of this was manually creating a 2-dimensional list database of foods for each meal in the day. I needed the food, food group (poultry, grain, fruit, etc.) and its macronutrients. In the developed function, I used Python's built in random methods to select for random food in the appropriate list and if the food can be legally added (fits the caloric need without going over), append the food. However,

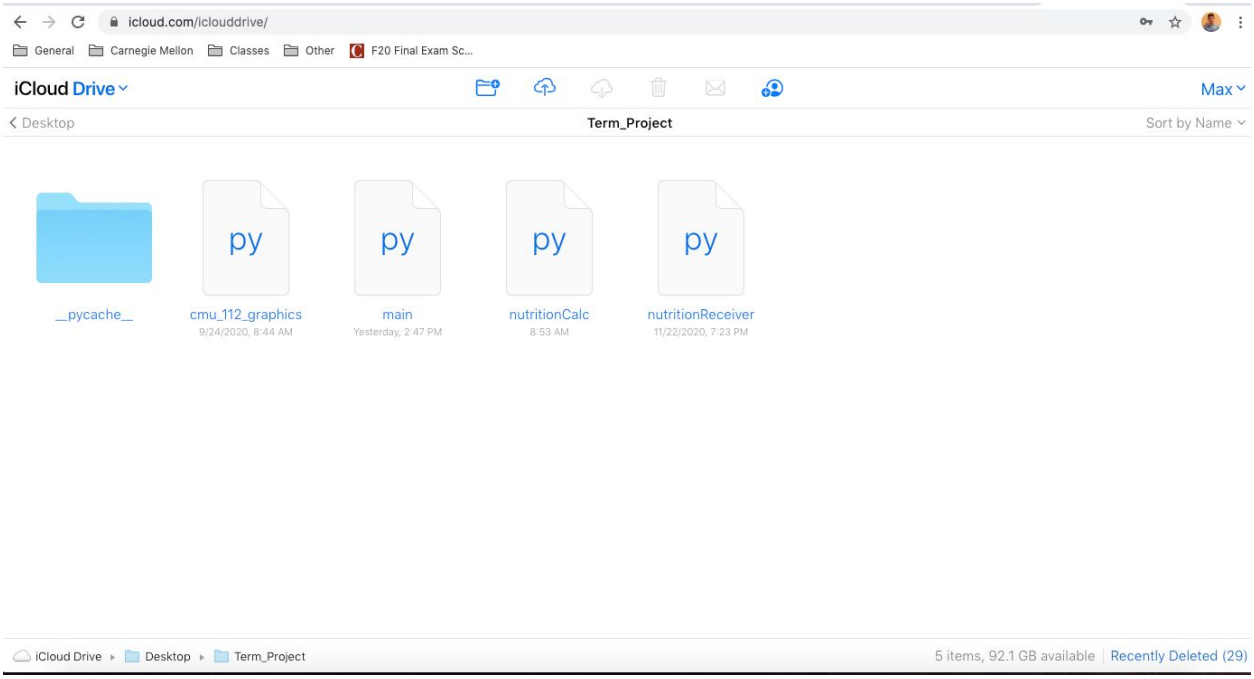
if the meal's calories are too high, I call the function again to repeat, randomizing the food selection once more. To ensure the meals are tailored to the user's preferences, if the meal plan does not possess a food of the user's preference, dairy for example, the meal plan is called again and randomized until it meets the requirements.

Timeline Plan: I've already completed the first key part of obtaining user food data, but not analyzing it. I've also developed the meal plans for the user, but the database can have more foods added to aid in the random selection process, and some bugs within it will need to be tweaked over the next 5 days. I still need to create a button/function where the user's day of nutrition data is appended to a CSV file and the ability to read that data and view it as a visual bar graph, which will analyze the data. I plan on completing this by Thursday, December 3. I plan to spend the last two days before the MVP deadline debugging the application.

TP2 Update: No major design changes made. However, some new features include saving and loading user information locally, along with being able to choose an "anti-preference" for foods. Also, the visualization of the user's weekly data does not use macronutrients, but rather the calories consumed with a percentage of how close the user was to reaching the caloric goal for that day.

TP3 Update: Some key design additions were made. First, the visualization of the user's data as bar graphs has been assigned colors depending on the percentages of the macronutrient consumed in relation to standard averages (with a key to indicate what the colors mean). This means that the user has the ability to sort his/her data by macronutrient (protein, carbohydrates, and fat), not just by calories. Additionally, a 'lock' feature was added to the meal plan creator, allowing users to lock in as many as wanted in case the user really likes certain meals but wants the others to be randomized. To help do this, my database of foods was further updated and improved by adding more foods. Lastly, when the mouse is moved over the graph of the sorted macronutrients, a number appears, displaying how many calories that macronutrient contributed towards the day's calories (disappears when the mouse moves off of the graph).

Version Control Plan: My project's work will be saved on iCloud Drive. Below is a picture that demonstrates the project constantly being uploaded to iCloud Drive and stored.



Module List: Selenium