Reflections:

Summary of the project and instructions on how to test and use it:

This project is a calorie counter app. The goal is to ask question to the user about their health information and provide a recommendation on the required calorie intake to reach their weight goals. The app uses the users age, sex, height, and weight to calculate the Body Mass Index and the Basal Metabolic Rate. It uses the frequency and intensity of exercise to calculate an exercise level and provide a calorie intake recommendation based on the BMR and the exercise level. Each use of the app creates a survey object where all the information is stored. Additional features include ability to view completed surveys and ability to delete surveys. The formulas used for the calculation are available online - https://en.wikipedia.org/wiki/Harris%E2%80%93Benedict_equation

A main menu provides all the options available to the user:

'A' – option to add a survey.

1- The app first asks the user to enter a label for the survey. If a survey with the same label is already existing, the app asks for a different label.
2- The app first asks for a goal from the user with options as maintain, gain, or lose weight.
3- The app asks for the users age, sex, height, and weight. All of these have error checking to make sure that a correct input is entered. The user will be repeated asked to enter a correct input. I have used a custom InputError class to differentiate inputs where the survey does not apply. Examples are age (only 18–65 are accepted), height (4–7 feet) and weight (50- 450 lbs). For values outside this range, the app will stop the new survey and print a message saying that it is not applicable to the user.
4- The app asks for exercise frequency and intensity.
5- The app then prints a recommended calorie intake for the user to reach their goal. It may also change the goal and suggest the opposite goal (in case the user is overweight but still enters a goal as gain weight)

'V' – option to view a survey.

1- If no survey exists, the app says so.
2- If surveys exist, the app print a list of the labels for each survey and asks which on the user would like to view based on index. It then prints the information that was entered in that survey.

'D' – option to delete a survey.

1- If no survey exists, the app says so.
2- If surveys exist, the app print a list of the labels for each survey and asks which on the user would like to delete based on index. It deletes the survey from the list.

'Q' – option to quit.

1- Prints a thank you message and quits the program.

Reflections and Challenges:

I was able to complete all the features that I intended to include in the project. I did, however, reduce the scope of my project substantially from the proposal. My proposal included numerous more variables to be asked from the user and recommendation for each of them. I realized that this would take a long time to code and go way over the lines limit, just for user input and error checking. This would also have been repeated code many times just for different variables. I decided to reduce the scope to less variables and instead add added functionality of viewing and deleting surveys. This also made the implementation more fun. Deciding the scope of the project was the biggest challenge that I faced. Another challenge I faced was to decide where and how to ask the questions to the user. It seemed like there could be many options and it was not obvious what would be best. In the end, I did data input in each of the classes where it is stored, so that I would not have to scroll around while looking for bugs. After changing the scope, the most important step I took was to map out the program in detail in a notebook. I then used a top-down approach to code, starting with class names, methods, and attributes without any code inside. I also wrote extensive doc string before writing any code, which helped me anticipate problems and streamlined the coding process and decide which classes will be responsible for which feature.

This was a very useful project to understand the how powerful objects can be. Being able to use complete objects as arguments for functions made it very easy to manage the data in this project. It also helped me realize how important a detailed plan is before any implementation.