Bellarmine University

Group Programming Report: Fitness Tracker

Mason Anderson

Kadin McWilliams

Computer Science 150

Professor Sayani Sakar

November 14, 2023

Abstract

For this project, we were tasked to code one of five different applications. These included the fitness tracker, to do list, recipe organizer, and budget manager. For this, we chose to do the fitness tracker as it was an applicable app that both of us found usefulness in. To code this, we used the Python modules: while loop, for loop, if statements, functions, classes, list, and dictionaries that we learned in class. Through struggle and help from a friend, we were able to produce an app that satisfied our goals. We created a fitness app that was able to set a caloric goal, log any number of desired workouts, how many calories were burned and if the user achieved their caloric goal. All of this was summarized in an easy to view and pleasant GUI.

Introduction

The purpose of this project was to test our Python abilities and utilize the skills that we have learned so far. When deciding what topic we wanted to pick, we discussed which one would be of the most practical use to us. We narrowed it down to the fitness tracker and the budget manager. When deciding between the two, we decided the fitness tracker was the better option because we had a more definitive vision. We also would have better use out of a fitness tracker than a budget manager, so creating a fitness tracker would be more fulfilling. The fitness tracker fulfills our need to quickly log our workouts and do an approximate calculation of the calories burned. Combined with setting our goal, we created an output to track our progress and let us know if the set goal was achieved.

Methodology

The technology and tools that we used for this project are our computers, discord, and Jupiter notebook. Using our computers, we were able to use discord to communicate during our project. Jupiter notebook was our only tool used for coding the project. The initial plan that was set out was to have a mental map set of what we wanted to include in our code. This included planning what types of functions used and where they would be applied. We also verbally ran through how we would want our code to work. When brainstorming we would start with an initial idea and then tweak this idea based on the available functions and our coding abilities. After doing research on how calories burned are calculated, we started adding in each function we needed under a class. After we set up our class and added all the functions, we created our while loop for the user input to tie up the coding process. After we had a rough draft of our code we went to work with testing. We both tried to put in extreme numbers, take non-intuitive paths, and disregard instructions. After we both tested our code, we allowed our friend to run through the code to test for additional errors. After going through the code and watching our friend we refined the interface to make going through the app more intuitive.

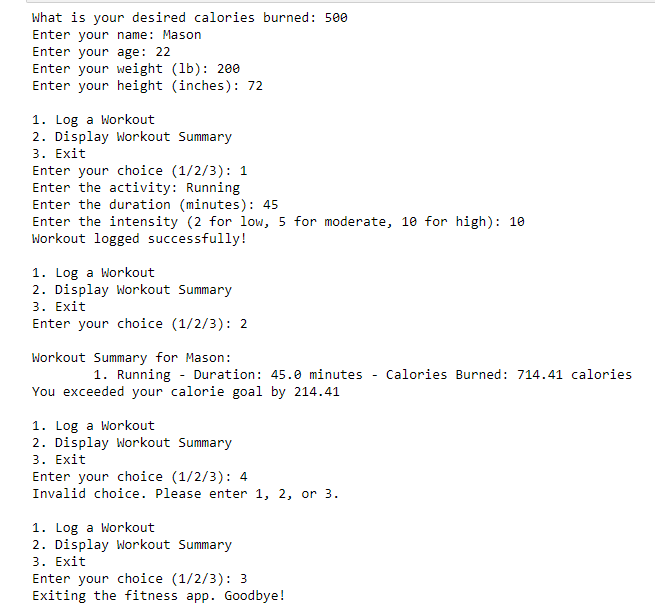
Design and Implementation

For our project, we wanted to structure it to allow for the input and calculations as well as cleanly display our objective. To do this, we utilized a class and functions to not clutter our code with excessive print statements and inputs. We also wanted to have our display be as neat as possible, so we experimented with using dictionaries, lists and spacing to be visually appealing. We used dictionaries so that when we were coding in our functions, we could easily call the keys we needed. We did this for clarity when trying to understand what each key value was inputting. When designing the architecture of our code we had to keep in mind that other people would be working with our code. This meant that we could not put variable names that only an individual person would understand. We mainly used the list's ability to easily add new information to them to store additional workouts. The program's design was to calculate and package up the workouts to be easily stored into a list. Being able to easily store information into a list significantly reduced the amount of work needed by us. We also wanted to use for and while loops for efficiency in our code. To cut down on the clutter we used plenty of for and while loops to automate tasks that would increase the amount of hard coded parts required.

Challenges and Solutions

The major problem we ran into was our summary and how to properly get it to loop. We wanted to create a for loop for unpacking each workout. Unpacking the for loop was not the challenging part but adding a counter for each work out gave us trouble. We wanted to add a counter to let the user process information faster and easier. However, to do so we would have to hard code a lengthy loop that would break down some of our architecture plans. So, we solved our problem by learning the enumerate function. With the function we could easily add a number to each work out. After having to solve this problem we focused more on making our code as easy to follow as possible. We did this to not slow down our productivity due to a lack of understanding code as we could not be in a discord call. Discord allowed us to solve not having a suitable time to meet and work on the project. Discord solved this issue by allowing us to share our work with each other effortlessly.

Results



Referencing the screenshot above, we were able to accomplish all our goals. We made sure to eliminate any potential input errors using if statements. The user can input information regarding themselves, input a caloric goal, choose to log a workout, display a summary of their workouts or exit. The summary is easy to read and an appealing GUI.

Discussion

After concluding this project, this reinforces the possibilities that we can achieve using the skills that we have already learned. After learning coding for less than 3 months, we were able to make a practical application that we could use in our daily lives. As we learn more and become more skillful in our coding, we can make more complex applications and code them to our specifications. However, we also have learned that there is much room for improvement. We only experienced one major coding block, only able to overcome it with the help of someone with much more coding experience. Another thing we learned is that it is hard to work on the project online. To share code, we shared our screens, however there were many technical difficulties. This resulted in us sending our code through text files whenever we had changes and wanted the other person to add them. Having to work with other people helped us with our organizational skills as these are key for working with other people. The overall experience taught us that improving on coding is mainly about optimizing the amount of coding that is required from the coder.

Conclusions

We decided on a practical application for our project topic. After brainstorming and working on what our goal was for the project, we started to code it together. We were able to make a functional program that utilized all the Python modules: while loop, for loop, if statements, functions, classes, list, and dictionaries that we learned in class. After going through coding and interpersonal challenges, we were able to deliver on an app that achieved our desired goal. Our project was relevant to our education by allowing us to get an experience more applicable to real world coding. We are going to be constantly working with other coders in the real world, so learning the basics of working on a team now is an important skill that we were able to hone through this project.

References

Kaminski, Jacqueline. “Metabolic Equivalents: What Are They & How to Calculate Them.” *NASM*, blog.nasm.org/metabolic-equivalents-for-weight-loss#:~:text=To%20determine%20calories%20expended%20by,%2F%20200%20%3D%20KCAL%2FMIN.&text=So%20in%2045%20minutes%2C%20this,700%20calories%20running%20at%207mph. Accessed 14 Nov. 2023.