The original artifact that I am using for all of these enhancements is a project from April 2024 that was originally developed to read a text file of grocery items purchased, track the frequency of purchases, and export the frequency data to a text file.

I selected this item because I wanted to work on a single project for all three of the improvement areas. I think that a single project makes more sense for the purpose of showcasing my abilities in an interview context rather than multiple smaller projects. During an interview I may not have time to discuss five different projects and how my work on them proves my capabilities so I figured that a single project would help me showcase my abilities in a more concise manner. Another reason that I chose this artifact was that it was a pretty simple project to begin with, and I wanted something that would allow me to really explore and expand upon the initial development.

As they relate to the Software Design and Engineering requirements, my changes involved converting the program from C++ to Python, creating a frontend web interface for interacting with the data, and embracing OOP principles by adding an Item class to increase the modularity and data abstraction in the program.

In my original documentation, I stated that I wanted to achieve the following two course objectives: Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals; Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts. I believe that I have done a fair job at accomplishing the first by implementing well-founded and innovative techniques to deliver value. However, I haven’t done a great job at the second one. I believe that by improving my in-line comments in the program I can increase the level of my written communications in order to provide more value to other developers or interviewers who interact with the program.

I would say that this section was probably the largest chunk of development work that I needed to implement in order to achieve the desired outcome. This was due to the fact that the program did not possess any user interface originally (other than a data output file if you can really consider that any type of frontend). So, I decided to use Flask in order to implement the frontend, which proved to be a useful and simple tool for achieving what I needed. Flask in combination with the HTML template that I used provided a small learning curve that required me to spend a fair amount of time learning the different implementations. However, by using my previous projects throughout my Computer Science journey as well as some internet resources, I was able to piece together what I needed in order to provide the necessary functionality. Furthermore, once I figured out the basic layout for the HTML and Flask, the actual business logic, design, and routing was fairly simple. Since the original program wasn’t extremely complex or large, converting it from C++ to Python was relatively simple. The same goes for creating the Item class that provided some data abstraction and an easier interface for manipulating the data. After I had my basic functionality covered, I added a feature to the web interface that I was curious about and ended up learning a little more about session management in the process. It was just a script within the HTML that allowed the page to stay in the same scroll position when the page was reloaded. Since the page was reloaded when the data was altered, I found it annoying to have to scroll back to the previous location if data was being altered on one of the bottom items. So, I did a little research into how not to have that happen; in the process I was pleasantly surprised to find myself learning more about session management and how to utilize it.