The artifact I chose to enhance for my ePortfolio is the inventory management app from CS210. The project was first created as a C++ console application that would read and write to a text file to count item frequencies. For my capstone project, I converted it into a web-based application using HTML, CSS, and JavaScript, making it more interactive and accessible. The program allows users to input, store, search, and sort inventory items dynamically.

One of the key improvements I implemented was the merge sort algorithm, which efficiently sorts inventory items by name A-Z and Z-A, or by frequency. Additionally, I enhanced the search functionality to support multiple search terms which allows users to look up more than one item at a time just by separating items by a coma. I noticed my program was saving multiple of the same item when processing data so in order to prevent duplicate entries I made the program update an item’s frequency instead of adding new instances, making the database more efficient. These enhancements demonstrate my ability to apply algorithmic principles to optimize data handling, a key competency in computer science.

When I first planned this enhancement in Module One, my goal was to improve the efficiency and usability of the application while aligning with course outcomes related to algorithm design and computing solutions. I believe I successfully met these objectives, as the improvements I made align with Course Outcome 3, which focuses on designing and evaluating computing solutions using algorithmic principles. While I originally planned to enhance just the sorting functionality, I expanded the scope to include search optimization and database efficiency in order to demonstrate a deeper understanding of data structures and real-world application development.

Throughout the process, I learned a lot about integrating sorting algorithms into dynamic web applications. One challenge I faced was ensuring that search results were sorted correctly without reloading the entire inventory, which initially caused unintended behavior. Debugging these issues with dev tools in windows taught me how to store and manage search results separately to avoid resetting data.

This enhancement was a rewarding experience because it reinforced the importance of designing efficient and scalable systems. It also demonstrated my ability to analyze and improve existing code, a critical skill in software development. This enhancement was not just a technical upgrade, it was an opportunity for me to grow as a developer and gain insights into how algorithms and data structures can improve real-world applications such as inventory management.