The artifact I chose for my ePortfolio is the CS210 inventory app that was originally developed in C++. This program was designed to read words from a text file, count their occurrences, and display the results both numerically and visually in the form of a histogram. The initial implementation of the program used file input and output for storing and retrieving data, with all interactions taking place in the console itself. While functional, this version lacked user interactivity and accessibility beyond a local machine.

To improve the application, I ported it to JavaScript, HTML, and CSS in order to turn it into a web-based tool. Instead of reading words from a file, users can now enter text dynamically in a web form, process it in real-time, and view the word frequencies instantly. I included this artifact in my ePortfolio because it demonstrates my ability to apply programming concepts across different languages and environments. The transition from C++ to JavaScript required me to rethink how the program processes data, particularly because JavaScript handles data storage and manipulation differently than C++. Instead of using map<string, int> as in C++, I implemented JavaScript’s Map object to efficiently store and retrieve word frequencies. Additionally, since JavaScript does not support file input/output in a browser, I redesigned the program to work with user input fields instead. These changes highlight my ability to adapt code to different platforms while preserving core functionality.

This enhancement helped me meet my learning objectives by reinforcing my knowledge of data structures, algorithms, and web development. I successfully maintained the program’s efficiency while transitioning it to a new technology stack. The experience also allowed me to improve my understanding of things like event-driven programming, as JavaScript applications function differently from sequential C++ programs. Instead of a menu-driven console system, I had to implement interactive buttons and event listeners to trigger actions like processing text and displaying results.

During the enhancement process, I encountered several challenges. One of the biggest difficulties was adjusting from a file-based system to handling text dynamically in the browser. Unlike C++, which can easily read and write files, JavaScript requires alternative approaches, such as text inputs and real-time DOM manipulation. Another challenge was debugging JavaScript code in a browser environment because it requires different tools and techniques compared to debugging in a traditional C++ compiler. I did improve my ability to troubleshoot and test web applications effectively, however, I found it very difficult to adjust to these changes.

Overall, this enhancement strengthened my problem-solving skills and adaptability as a programmer. By successfully porting the application, I demonstrated my ability to convert command-line programs into interactive web applications, an essential skill in modern software development. Looking ahead, I plan to further improve this project by adding sorting functionality so users can organize frequency data in different ways. Additionally, I would like to enhance the histogram visualization with a more graphical representation using Chart.js. Learning about live server was also fascinating, it was great being able to see my changes in real time.