Jenice Mario

Professor Brooke Goggin

CS 499 Computer Science Capstone

July 21, 2024

CS 499 Milestone Two Narrative

The artifact was originally a C++ application created for Project 3 of CS 210 Programming Languages class taken in 2024 and was, essentially, a grocery tracker that had three options: look up item frequency, print all frequencies, and print histogram frequencies (displayed by asterisks [Ex. Beets: \*\*\*]). These three features were all outputted to the console from a .txt file holding the data. The enhancement now has been ported to be a web-based application called Inventory, designed to help users manage and analyze grocery item data. It allows users to upload grocery data from .txt or .xlsx files, look up the frequency of specific items, print all item frequencies with sorting options, highlights low-frequency items, and generates histograms to visualize item frequencies.

The reason for the inclusion of this artifact is that this is a basic project that can be built upon. This is in regard to both the features that can be refined and added, as well as it being a very apt program to port to a web-based application as there is front-end and back-end functionality. It showcases my knowledge of the importance of UI, this is specifically seen by the red highlighting of low frequency items. The artifact was improved because it now additionally also allows users to upload grocery data from .txt as well as .xlsx files, looks up the frequency of specific items, print all item frequencies with sorting options, highlights low-frequency items, and generates histograms to visualize item frequencies instead of asterisks. The enhancement also features the improvements of a header, an off-white background, the “Print All Item Frequencies” feature displaying multiple columns so that the user does not have to scroll, as well as an auto-scroll to fit for the “Print Histogram of Item Frequencies” feature so the user can immediately view the full histogram.

I did meet the course objectives I planned to meet with this enhancement in Module One. The course outcomes I have achieved with this enhancement are Course Outcome 2: Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts (Southern New Hampshire University (n.d.)) and Course Outcome 4: 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 (Southern New Hampshire University (n.d.). Course Outcome 2 is met by transforming the original C++ program into a web-based application using JavaScript, HTML, and CSS. I designed and developed a user-friendly interface that is more visually appealing and functional than the original console output format. I ensured the layout, text, and color scheme were clear and easy to use. Interactive elements, such as buttons and dropdowns, enhance the user experience. The inclusion of a dynamic histogram using Chart.js provides a graphical representation of data, making the output more engaging and accessible to users. The comprehensive comments in the code and the README file I created ensure the application is understandable and maintainable by other developers. These efforts demonstrate my ability to produce technically sound and coherent communications, both in the code and documentation. Course Outcome 4 is met by transitioning from a console-based C++ application to a web-based application using modern web development techniques and tools. This required using JavaScript, HTML, and CSS to create an interactive and dynamic application. The integration of Chart.js for data visualization is an innovative approach that enhances the application's value by providing professional-quality visual representations of data. The enhanced functionality of the application, including support for both .txt and .xlsx file uploads, dynamic sorting, and integrated search functionality, significantly improves its usability and value. Performance optimizations in the JavaScript code ensure that the application remains responsive and performant. These improvements demonstrate my ability to deliver innovative computing solutions that meet industry-specific goals. The course outcomes I still have to achieve are Course Outcome 1: Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science, Course Outcome 3: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices, and Course Outcome 5: Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources (Southern New Hampshire University (n.d.). As of right now, I have no updates to my original outcome-coverage plans.

One key area of learning was user experience design. This focus led to replacing the browser prompt with an integrated search input for item frequency lookup, making the application more user-friendly. Providing real-time feedback to users, such as highlighting low-frequency items and dynamically updating the histogram, was essential for a smooth user experience. Another significant learning area was data visualization. Integrating Chart.js into the application allowed me to present data interactively and appealingly. This integration enhanced my ability to create visual representations of data. Error handling and user feedback were other critical aspects of the enhancement process. Ensuring that users were informed about actions being taken, such as unsupported file types or successful data processing, was important. One of the main challenges was ensuring that each section of the interface was cleared when a different section was clicked. Managing the application's state to achieve this was complex. Additionally, the histogram feature posed a challenge as it took a long time to load after being displayed once. Optimizing this feature to improve performance was necessary. Another challenge was with the sort feature also not clearing properly when switching between different functionalities. Ensuring the sorting dropdown only appeared during the "Print All Item Frequencies" feature required careful implementation. Changing the search lookup from a browser pop-up to a built-in feature was initially challenging but ultimately improved the application's usability.

Below are screenshots of the flow of the original program.

**A screenshot of a computer

Description automatically generated**

Start Menu/First screen/Main Menu

A screenshot of a computer

Description automatically generated

Menu Option One: Look up Item Frequency

* Prompts the user to enter an item name.
* Displays the frequency of the specified item.
* Returns to Main Menu

A screenshot of a computer program

Description automatically generated  
Menu Option Two: Print All Item Frequencies

* Displays all items and their frequencies.
* Returns to Main Menu

A screenshot of a computer program

Description automatically generated

Menu Option Three: Print Histogram

* Displays a histogram of item frequencies.
* Returns to Main Menu

A screen shot of a computer program

Description automatically generated

Menu Option Four: Exit

* Exits the program.

Below are the screenshots of the flow of the new, enhanced web-based application.

A close up of vegetables

Description automatically generated

Main page upon startup.

A screenshot of a computer

Description automatically generated

Pop-up after “Choose File” and “Load data” have been clicked.

A close-up of vegetables

Description automatically generated

“Look Up Item Frequency” Feature.

A close-up of vegetables

Description automatically generated

Error Handling for “Look Up Item Frequency” Feature.

A close-up of vegetables

Description automatically generated

“Print All Item Frequencies”.

A close-up of a computer screen

Description automatically generated

Sort options with the “Print All Item Frequencies”.

A graph of a graph

Description automatically generated with medium confidence

“Print Histogram of Item Frequencies” feature, the page auto-scrolls when clicked on to view the whole histogram.

A screenshot of a computer

Description automatically generated

The ability to re-load a new file and use all of the features.

References  
Southern New Hampshire University. (n.d.). *CS 499 module one assignment template*.