**Prakruti Dholiya**

**Nashr El Auliya**

**Nourhan El Sherif**

**Yidi Wu**

**Statement of Work**

All team members worked together to choose and complete the final project, and multiple tasks were shared and jointly worked on.

The initial project idea agreed upon involved designing a game similar to PA3/PA4. Project idea was changed as we realized that rather than building another game, we would prefer to build something that can generate a Bill of Materials for engineers. This would also involve learning Python and web scraping, neither of which team members were familiar with, and thus this project would build our skill sets and teamwork skills.

**Task 1: Revising New Project Details (all)**

Nashr suggested the new idea of building an automated BOM. All members agreed this would be preferred as our final project, and in a team meeting, all members discussed what aspects would be required to complete this project, as well as how to team up on the aspects.

Item 1: Web scrape using Python – Yidi and Prakruti

Item 2: Building ranking algorithm in C++ – Noura and Nashr

Item 3: Link Python to C++ algorithm – Noura

Item 4: Build GUI with Python – Yidi

Item 5: Output to csv file from C++ – Noura and Nashr

Item 6: Documentation and Misc Requirements – Prakruti

**Task 2: Build Initial Timeline (prakruti)**

Prakruti worked on building a timeline for the project based on initial team meeting discussion. Prakruti also reread through project requirements and checked in with the team to make sure misc. requirements such as the documentation, “minimum 1 branch on GitHub” and “each member must commit once to GitHub” were being met.

**Task 3: Learning and Practicing Web Scraping (yidi, prakruti, noura)**

The three aforementioned members took time to learn basics of web scraping with Python and familiarizing themselves with Python syntax/structure, Selenium, and Beautiful Soup. Team members also researched which websites would be useful to web scrape from. Prakruti suggested Amazon, Newark Electronics, McMaster, AdaFruit. Noura also suggested Home Depot. Yidi and Noura looked through the HTML structure for each of these webpages, and Newark Electronics was chosen based on both its relevance to engineering project items and its straightforward HTML structure. Yidi attempted to web scrape Amazon, but realized there are security blocks in place.

**Task 4: Web scraping Newark Electronics (yidi, prakruti)**

Yidi worked largely on setting up code to web scrape the Newark page. Prakruti worked with Yidi to troubleshoot the code as the team ran into errors with class types and scraping from Newark. Once this was fixed, Yidi further developed code to take in all necessary search result values. Prakruti worked on creating a function that looked through the scraped data and chose the best option for quantity and its price for each search result option in C++.

**Task 5: Building Algorithm in C++ (noura, nash)**

Noura and Nashr worked on developing an algorithm in C++ that would take quantity and price values from python per item and rank which of the search results would be optimal. Initially, the team had decided to utilize a “best seller” flag as a check for quality, but as there were discrepancies in the Newark webpage with this flag, we were unable to utilize a quality check. All the data structures were set up, but functions for the algorithm were not yet created or implemented.

**Task 6: Linking C++ and Python (noura)**

Noura attempted to find ways to link C++ and Python. File I/O would not be optimal in our case as the web scraped data would have to be output to Excel in an ineffective format. Noura tried embedding python in C++ to call the python script, but the python.h library failed to be found. Multiple efforts were made to amend the path of the libraries to include python, but this still did not work. Due to this, our team shifted from working between C++ and Python to working in just Python.

**Task 7: Simplifying Scraped Data (prakruti)**

Prakruti worked on creating a function that looked through the scraped data and chose the best option for quantity and its price for each search result option. Initially, there had been some progress in C++, where sample lists of quantities and prices would be optimized. However, after our team realized that we could no longer work in C++, Prakruti redesigned this as a function in python and simplified the function as well. With python, since there is no need to declare variables, work between static and dynamic values, etc. this code was significantly simplified. This also rendered previous C++ progress unusable.

**Task 8: Rebuilding Algorithm in Python (noura, nash)**

Noura and Nash worked on completing the algorithm in Python, as well as translating the existing C++ algorithm components into Python. Once again, working with Python rather than C++ also significantly simplified the code for this task, and rendered previous C++ progress unusable. Yidi then connected the data structures she stored the scraped data in with the algorithm being built.

**Task 9: Building GUI (yidi)**

Yidi built a GUI for the program with tkinter. This interface will take in the user’s budget, up to 10 items to search and their respective quantities. The GUI will output an error message if the item search does not work, as well as generate an output message with the name of the BOM file, the total cost, and if the budget requirement was met or not by the cheapest BOM possible.

**Task 10: Output to CSV (nash, yidi)**

Nashr and Yidi worked on Python code that would output the final item’s name, lowest cost, optimal quantity to buy, and the URL link for the product. Nashr initially built this by utilizing a class, but Yidi directly included this in her file, which is faster (similar to how ‘inlining’ would be faster in C/C++ but less modular). Both used the ‘openpyxl’ library. Our team opted to use the faster method rather than the more modular method.

**Task 11: Documentation (prakruti, noura)**

Prakruti and Noura organized all documentation for the project, including the statement of work, project documentation, real project timeline, readme file, and project architecture. Both also checked that the misc. project requirements were so far met. Once Task 12 was done, all files were organized on one person’s laptop and pushed in a zip file labeled Project.zip. This file contains the src folder and the doc folder.

**Task 12: Test Cases and Video (prakruti, noura, nash)**

The three members worked on running test cases for the final project, making sure that outputs were as expected, the GUI worked correctly, and the error-checking worked correctly as well. Members worked together to create the demo video as well. All files were organized on one person’s laptop and pushed in a zip file labeled Project.zip. This file contains