SLASH

Project Evaluation Report

Difficulties faced during Project Evaluation and Setup -

1. After cloning the project “c”- slash, the installation guide mentioned to use the package manager pip and install all the package dependencies from the requirements.txt file. I used the command that was mentioned in the installation guide. Using the specified command, the installation was successful and then I tried to run the application using the further instructions. But, an error occurred while running the application saying that the required packages are not found/installed. Diving deeper into the issue, I found that I had 2 versions of python installed on my local machine(latest one for vscode and the other for jupyter notebook) and the dependencies got installed on the latter version. To resolve the issue, I kept the LTS version(jupyter notebook one) and uninstalled the other one. On top of this, I had to use an earlier version of Python (3.9 or lower). The later versions of Python does not have support with NumPy.
2. From a contributing developer’s perspective, it was very much confusing as there was no information about the design language system for the project. To be more specific, There was no particular formatter specified for the code, no frontend theme or design guidelines, no system structure/design related diagram because of which we had to manually figure out the flow of the system by using a trial and error approach.
3. On using the project, we found that the command line version was functioning as mentioned in the documentation with a few minor edge-case related technical bugs here and there. Further, on using the UI version, i.e., “Slash-Mini”, everything was functional and the application displayed multiple search results as one would expect. On a closer look, we found 2 major issues -
    a. The scraped search results were not always related to the search query. For example, we searched for “Dell Inspiron 14 Plus” and we got results related to all types of products including what we searched for like laptop skins, covers, bags and few other models related to Dell Inspiron series with the actual product ranked lower in the list. This made it difficult to find the relevant product from a user experience point of view.
    b. The results filter controls triggered a new search request on clicking the filter button instead of filtering the currently fetched results.

Things that could have been avoided -

1. The biggest issue was the version of Python. It was not clear what version of Python would work. It would be nice to have some documentation that noted the last working Python version.
2. It was hard to follow the flow of the code. Some sort of diagram to help developers understand the code would be nice. On top of this, it could be nice to understand some key design decisions in order to understand why certain implementations was used.

3. Scraping products from etsy could have been avoided as most of the products fetched from the same were almost or completely irrelevant.
4. The command line version of the tool could have been completely avoided and if the same amount of efforts had been taken for developing the web application version, the product would be able to reach a wider percentage of users. It is also hard to use without extensive instructions.
5. The use of HTML tables in the UI makes the webpage non-responsive which makes it less accessible in general.

Proposed solutions to mitigate the above problems -

Practices to perform in project 2

1. We would ensure that the documentation would be easy to understand for anyone who would try to clone and run the project, keeping in mind the following things which we would follow
    a. We would write possible errors which we came across or anyone who clones the project may come across.
    b. A proper documentation regarding the information of test cases.
    c. Information for style checkers, code formatters and syntax checkers.
    d. A note to keep in mind that not to include too many videos of the project on the Readme.MD file
    e. A separate folder for videos can be created to show working of some functions of the application.
2. On top of this, some sort of diagram to show the flow of code throughout the different classes. Some simple box diagrams should suffice.
3. We can mention versions of all of our software that was working at the time so future developers can emulate the setup used when the application was working.
4. We can include some reasoning as to why we picked certain software or made certain decisions in the software documentation.