

ESSAY

During the process of setting up and running the five GitHub repositories that we were provided with, we encountered several challenges that tested our troubleshooting skills and problem-solving abilities. These repositories varied in their versions, requirements, backend tools, and programming languages. We faced even more issues due to variations in the operating system that we had on our machines.

One prominent hurdle we faced was the reliance on 'requirements.txt' files, which, unfortunately, did not always work seamlessly. Instead of straightforwardly resolving dependencies, we had to resort to manually installing each requirement individually, when faced with conflicting and deprecated versions. This process was not only time-consuming but also prone to human error, as tracking and managing numerous dependencies can be overwhelming.

Furthermore, compatibility issues between different Python versions added another layer of complexity. Some repositories necessitated Python 3, while others required Python 2, forcing us to toggle between these versions. To address this, we temporarily disabled Python 3 and made adjustments in the settings by navigating to 'Apps -> Advanced app settings -> App execution aliases.' This provided a temporary solution but underscored the need to have a more streamlined approach to managing multiple Python versions.

The 'Auto_anki' repository presented a particularly challenging scenario. During the installation of 'requirements.txt,' we encountered an error related to the 'workcloud' library. Specifically, we received a runtime error stating, "Directory 'static/' does not exist from import fitz." To enable 'Workcloud' to function correctly, we had to make the difficult decision to uninstall the deprecated 'fitz' library and replace it with 'pymupdf.' However, this change created a ripple effect, as many functionalities within the codebase relied heavily on 'fitz.' Consequently, we faced numerous errors, such as 'AttributeError: module 'fitz' has no attribute 'open,'" when attempting to open and read selected PDF files. The only viable resolution in this situation was an extensive code modification effort, transforming it to work seamlessly with 'pymupdf' instead of 'fitz.'

Operating system diversity within the team added another layer of complexity. In the case of the 'Feature Hunt' project, we needed to install JavaScript, MongoDB, Flask requirements, and various other tools. Unfortunately, the documentation lacked detailed installation instructions tailored to each operating system, leading to installation challenges and inconsistencies across different team members' setups. Coordinating and addressing these discrepancies consumed valuable time and resources.

Another unexpected roadblock arose when some libraries within the repositories depended on the 'collections' library. However, 'collections' had undergone significant changes, necessitating a Python version lower than 2.0, while other repositories required Python versions 2.7 and above. This dilemma prompted us to explore various library version combinations and experiment with different Python versions to achieve a conflict-free codebase. The process

involved meticulous testing and revision, often leading to unforeseen compatibility issues that required creative problem-solving.

The repository-wise documentation issues are as follows:

The documentation for the MyDollar Bot GitHub repository provides some useful information for setting up and using the Telegram bot. It outlines the purpose of the bot, installation instructions, and how to run it. However, there are several drawbacks and areas of concern in this documentation that need attention. While installation instructions are provided, there are missing details, such as the need for setting up a Python environment and any potential dependencies or issues users might encounter, thus needing us to do it manually and facing many issues in this process. The documentation does not include a troubleshooting section to address common issues or errors users might encounter during setup or usage.

The Video mentioned in the Feature Hunt repository is too long and it fails to mention the actual purpose of the application, rather it focuses on the code contributions done by the members and their enhancements. Even the markup documentation lacks a clear and concise introduction that explains the purpose and significance of the Feature Hunt project. It briefly describes the project but does not provide a more engaging introduction. Even this repo lacked a troubleshooting section to address common issues or errors users might encounter during setup or usage, leading us to face issues while setting up docker files and other APIs. This repo uses third-party hosting, the information about which is not mentioned in detail. This made us put in a lot of effort to actually set it up as the link wasn't working.

For GIT Simplified, on the other hand, the documentation is quite detailed and does indeed mention steps for various operating systems along with a detailed description for every feature added. However, this required us to move from one page to another, through quite nested linking and the long paragraphs mentioning the details about future work were overwhelming to some extent.

The ScheduleBot's documentation suffers from a lack of proper organization, making it challenging for users to navigate. Furthermore, it fails to provide adequate information about the purposes and dependencies of installed libraries. While the README mentions connecting to Google Cloud and storing an API key, it lacks step-by-step guidance for users unfamiliar with Google Cloud, making the configuration process potentially confusing. Additionally, the README does not clarify the necessary permissions the Discord bot requires for seamless operation, leaving users uncertain about the access rights they are granting to the bot when adding it to their server. To improve the user experience, there is a pressing need for more explicit instructions, comprehensive explanations, and enhanced troubleshooting support.

The documentation for auto_anki lacks operating system-specific installation instructions, which can potentially cause confusion for users. Furthermore, it fails to specify the required versions of dependencies, resulting in compatibility issues. Additionally, it provides insufficient guidance on utilizing the Pycco library for code documentation. Moreover, the installation steps provided in

the repository could benefit from more comprehensive and detailed explanations to facilitate a smoother setup process for users.

Based on our experiences and the challenges we faced in ensuring smooth project execution across different devices, it became evident that comprehensive documentation is a critical need. As our next initiative, we are actively engaged in creating comprehensive documentation for our project. We plan to include further details regarding the specific version utilized for each project, highlight essential prerequisites, and outline dependencies. Furthermore, we are dedicated to creating a dedicated section addressing common challenges encountered during project implementation, along with immediate solutions for enhanced accessibility and problem-solving.

Member 1: Akshada Girish Malpure (agmalpur@ncsu.edu)

Member 2: Sakshi Shankar Basapure (sbasapu@ncsu.edu)

Member 3: Shonil Sateesh Bhide (sbhide@ncsu.edu)

Member 4: Rutuja Mahesh Rashinkar (rrashin@ncsu.edu)