

USDC Test Project – Shyam Patel

Overview

I implemented the function *findSearchTermInBooks(searchTerm, scannedTextObj)* so that for any given search term, all lines (for each book) would be parsed and all lines with the matching word would be included in the result. The input JSON object was pre-parsed to have a correct format where each book is a nested object inside an JSON array. To implement this, I iterated over all the books, and then for each book I iterated over its “Content” and checked if the line contained the substring search Term.

Implementation Notes

- The code is based on the assumption that *scannedTextObj* is fed in with a correctly formatted object as mentioned in the direction document.
- For words separated by lines (example dark- on line 9 and ness on line 10), the search term would not match for either of those lines as they do not contain the full word, so neither would be included
- If given more time, I would clarify that these assumptions are valid and if not, I could modify the code to account for them.
 - Example: if for dark-ness, both lines containing subsections were to be included, I could modify the code to look at the last word and see if it is a substring of the term. In the case that it is, I could conditionally check the first word of the next line (if it is the direct next line sequentially), to see if the remaining term was there.

Testing

- To test, I wanted to ensure extensive coverage of possible cases in functionality.
- I tested edge cases like there being an empty object or a search term not being found within any of the lines.
- Additionally, I tested cases involving multiple books, a search term with multiple words, and lines with varying caps lock to ensure that the exact search term was found.
- If given more time, I could write more thorough tests to generate a large book or a collection of large books which would provide a very thorough search.

Comments

- One thing that I am proud of relating to my solution is its readability. The code is self-explanatory and does group together messy code, so that another developer could easily parse it.