United States Digital Corps Software Engineering Assessment

The first thing I worked on is working on the overall design of the function itself. With the inputs and outputs, I have been given, the function, *findSearchTerminBooks* (), analyzes the text in the each content and finds the string inside the Text. The first thing to do is search for the content of the book. I was pondering about the uses of which searching algorithm can be useful for this program. I concluded that searching for the content one by one is by far the most optimal method of finding the results because there could be a case where there are multiple lines with the word we are looking for in the book, despite the other searching methods being the faster than my optimal choice, one method won't be able to find all the occurrences of the word in the text and other choices might alter the overall program all together. For simplicity's sake I search for the content's text one by one to find our word in each content's text. There is another case for this assessment which is they're going to a case where there are multiple books other than *twenty thousand leagues under the Sea*. If this is the case, then for each book we are looking at the content then for each content we are looking at the line of text that has a word that matches with the word I am looking for. And if there was going to be a case where there are multiple results then the result should be an array of results.

As I previously stated, I currently have a way of going through each book and, for each book, go through each content of the book. I need to look at the text of the content. JavaScript has a built-in function called includes() where it reads a text and returns true if the word is in the array and false if there is no word that match in the sentence.

With my conclusion, for each book, I am reading each content and for each content I am searching in the content's text and if the content's text has a word that matches with the term in its sentence, then the results will return the ISBN, page, and line for each book.

I did enjoy this project; my proudest solution is solving the case of finding a term in multiple books instead of one book. This is an excellent solution for similar cases such as finding terms on multiple websites that share the same terms for research purposes. There wasn't a huge steppingstone while solving this problem but the most difficult topic was how to structure the information so I can implement my method of searching the content of the book and if so, would this affect the software engineering staff in the future. Even though this assessment specifically says modify the *findSearchTerminBooks* function, would this mean I can't alter anything outside the function so anything outside the program would be null and void? During my time working on this assessment, I was wondering about creating a structure around searching through the contents of the instead of my optimal method, but I feared this might alter the program significantly. I've also solved a few edge cases I resolved in this assessment such as the information wasn't inputted in the function, or if the term was a number instead of a word, and a case if the term wasn't found. There are a few niches I want to discuss such as the real application of this project. For example, what would happen if we had nearly 5,000,000 books or nearly 5,000,000 contents for each book? This is a glaring problem because imagine looking through a list and you are scanning a list one by one and there is nearly about 10,000 of content on the list it's very tedious and take a huge amount of time.