Appendix - Client Interaction

First Interaction
Date: December 23rd, 2022
S - Student
C - Client

S: Congrats on your podcast CCA. I hope it is running smoothly.
C: Yes, thank you.
S: Have you or the students in the CCA encountered any problems while making a podcast?
C: While this is not a problem, it is still quite a hassle. As the president of this CCA, I spend quite a bit of my time doing research for my students. This includes picking sample podcasts for each student or their groups to listen to to give them an idea of what they should be aiming for in terms of the content that they want to cover. So, I guess my problem is how inefficient the process of scouting for relevant podcasts that resonate with the students is.
S: So what you would like is a website that is able to provide a better recommendation of podcast selections that will be able to provide content that resonates with the content that the students are producing? … But don't platforms like Spotify and Google Podcasts already provide a good search engine?
C: Not really. The problem is that students when users have more than one topic with which they want a specific podcast to address, searching for relevant podcasts that merge various topics become difficult. So, it would probably be better if the website is able to return a podcast most relevant to all the search inputs.
S: I've had an epiphany! From what I hear, rather than creating a better search engine, I can build on top of it by generating the most relevant keyword that encapsulates all the user inputs. Then, I can use this new keyword as the search query to get podcast results from connecting to these podcast websites.
C: Pretty much!
S: Could a web application be most suitable for this solution?
C: Yes! In fact, I was imagining a website with a search bar where you could enter keywords like "robbery", "US", and "True Crime", which returns a title, description and relevant podcast URL.
S: Alright, this means that building upon just using a platform like Spotify's search engine to find the most relevant result, I also have to analyse each podcast recommended which would require Natural Language processing. I got that. However, I worry that there will be a problem accessing the podcasts from platforms like Spotify; smaller podcast search systems like Google Podcast are probably easy to web scrape. Would that be fine?
C: That sounds good to me.
S: Okay, I will plan an outline for my solution. This will take a few days.
C: That's fine. Thank you for all your help.

Second Interaction
Date: January 5th, 2023

S: Hi, long time no see! So basically, the product will be boiled down to a web application with a very intuitive GUI written in CSS, and HTML. Furthermore, I may decide to utilise Flask to route between the different functions and documents in the interface design, making it as smooth as possible to navigate. The backend with data input processing will be written in Python because of its accessibility of important NLP modules such as Gensim, Keybert and Word2Vec as well as convenient libraries such as Numpy and Pandas for dealing with large multi-dimensional arrays.
C: Cool, I myself am interested in machine learning tools and algorithms and am keen to see how powerful these language processing modules are.
S: Great. By the way, I have made a list of the success criteria for this project. *Hands printed criteria*
C: Looks thoroughly planned except for one thing: what if I inputted three completely random words? Wouldn't the recommended podcast thus be also random? How can I be sure that the recommended podcast from my search inputs is relevant?
S: Yes, if your inputs are not relevant to one another, it would make sense that the recommended output has nothing to do with one another either.
C: Another thing, actually. Wouldn't there be a problem with the ambiguity of words? For example, "duck" could mean the animal and could also mean the verb.
S: Hmm, good point. I believe there are NLP tools that are able to find patterns between words to infer the true meaning of the input. I will make sure to implement that. Thank you! Anything else?
C: Yes, actually. Would you be able to recommend inputs for the user to type when assisting them with their search? My students struggle to come up with the holistic topic covering all of the content they want to express.
S: That could definitely be implemented, like an autofill function. Also, this reminded me that there are some inefficiencies in my code, for example, the backend processing may take a long time to load.
C: I see. Then a database would be required, no?
S: Yes, that would be needed, so that the backend processing of previously inputted data won't have to be done redundantly.
C: Great.
S: Any more suggestions?
C: Nope, I'm expecting a masterclass.

Final Interaction:
Date: January 4th, 2024

S: Hi, it has been a while since we last interacted. By now, the final product is complete, hitting all of the success criteria.
C: Great!
S: Let's review and evaluate some of the criteria of this project.
C: Okay.
S: The first one is that minimal client input is required for the main functionality of the application.

C: Yes, all the user needs to do is enter three inputs.
S: Thank you - secondly, the website uses a simple, user-friendly bootstrap template that allows users to intuitively navigate to the input form.
C: Yes, the form template chosen and simple website layout makes it easy to navigate.
S: Next, you can see from the previous inputs page that details of input records are stored in the database, correct?
C: Yes.
S: When you type into the form, do auto-suggested words appear?
C: Yes.
S: Moving onto input validation, when you enter plural words into the form as an input, you should see the processed input in the database, but you don't. This is because they are converted to their singular form before processing using NLTK lemmatization.
C: Yes, upon entering 'governments' and 'government' into the form, I don't see the word 'governments' in the database, but the input 'government' does appear.
S: If no search results appear for your input after requesting data from the Google Podcast API, then an error message is displayed, correct? And you return back to the homepage.
C: Yes.
S: Great, moving onto criterion 2, mainly covering the backend functionalities. Confirm whether the logical flow of operations was as discussed in our previous interactions: when the user submits the homepage form, the program uses those inputs to request data from Google Podcast API. The requested data includes relevant information such as URL of podcast episodes. From these episodes, the URL of the podcast homepage and hence its RSS feed is scraped. Thus, podcast descriptions can be used to create keywords.
C: That seems quite logical.
S: Great, for criterion 3, a hypothesis test was conducted. Please confirm the conclusion: similar inputs show a high relevance score and vice versa.
C: That seems to be the case.
S: Cool. Next, can you confirm that a centroid is found based on the word2vec model used for processing all the extracted keywords?
C: Yes.
S: Lastly, in the podcast recommendation page, you see that the most relevant keyword is recommended based on euler distance to the centroid.
C: Yep.
S: Is a URL displayed, directing you to Google podcast that searches for podcast episodes related to the recommended keyword?
C: Yes.
S: For criterion 4, can the datatable in the previous inputs page be sorted alphabetically by clicking on the columns?
C: The functionality works well.
S: When the user enters a query in the search box, a word cloud appears in a new webpage, yes?
C: That's correct.
S: Okay, lastly, for criterion 5, the podcast recommendation page contains user inputs, relevancy of recommended result, connection to API and word cloud of relevant keywords, yes?

C: Yes, it does.
S: A datatable and a word cloud of keywords are displayed.
C: They are.
S: Finally, does it appear that a uniform template is used across all the webpages?
C: The navigation bar is preserved and the background stays the same, so I would agree.
S: Thank you for your time!