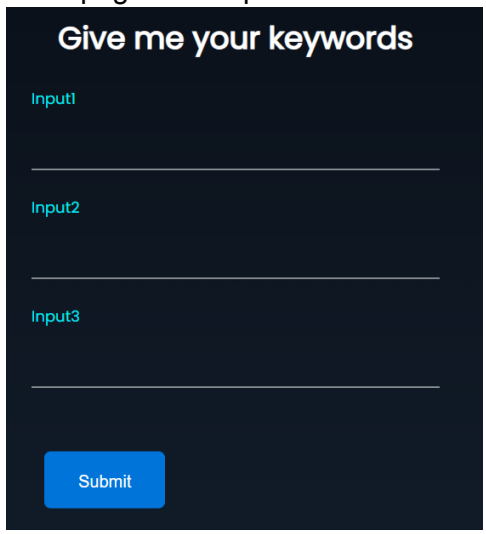
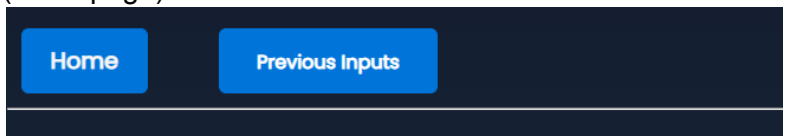
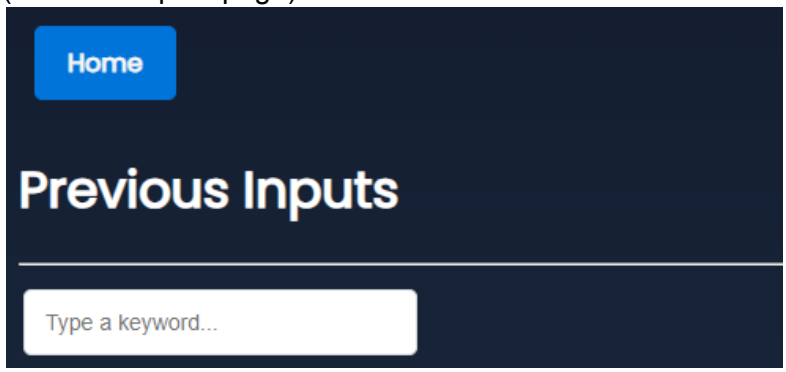
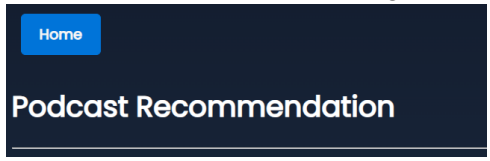
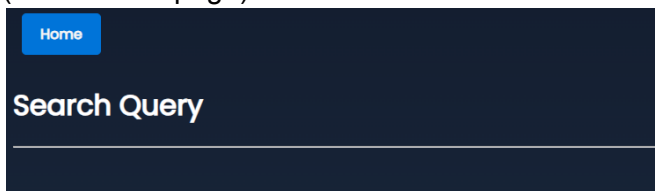
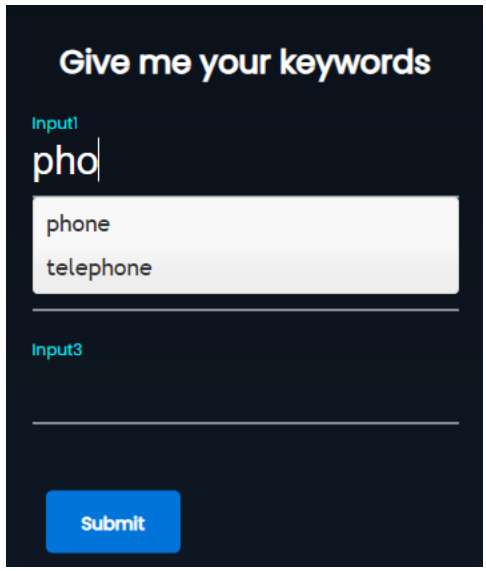
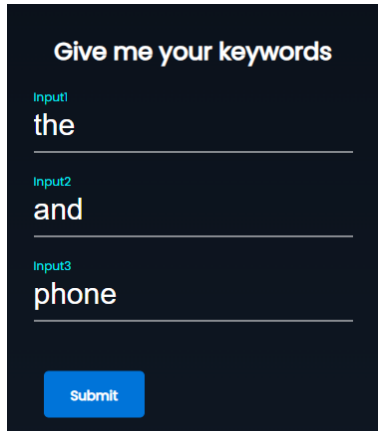
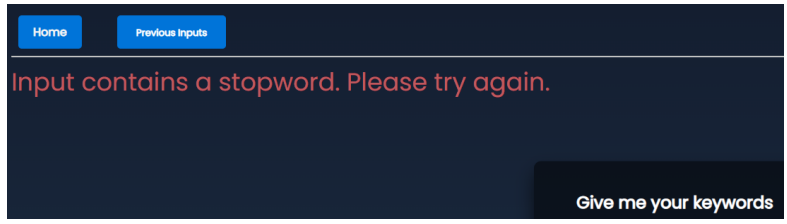
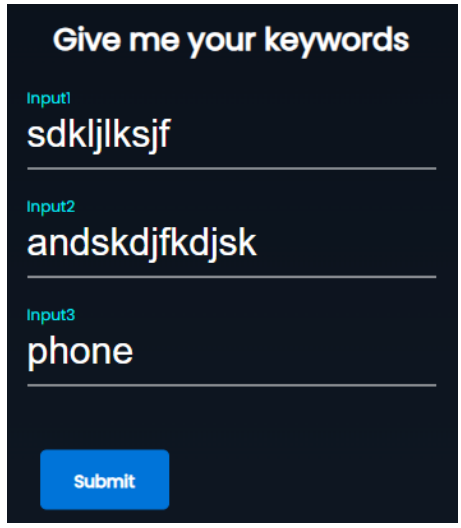
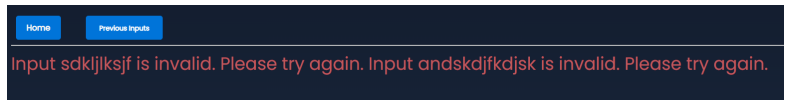
Criterion E: Evaluation

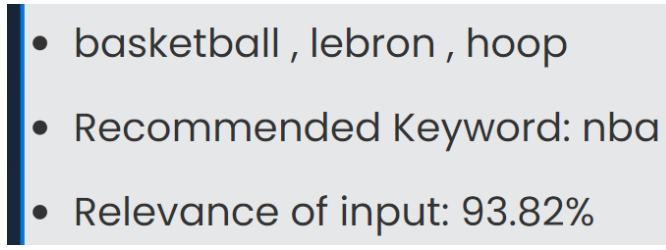
In the final client interaction, we verified the completion of all success criteria. Moreover, I completed the various tests (as outlined in criterion B) and demonstrated the product's functionality in criterion D.

Criterion	Met?	Evidence
1) a) Minimal client input	Yes	Homepage form with input limited to 3 words demonstrated in the video Client: "Yes, I only have to enter 3 words to receive an output from the system." Test: Homepage form input GUI 
1) b) Use a simple bootstrap template to create a user-friendly interface where the user can easily navigate to the three input boxes for inputting keywords.	Yes	A uniform template with a clear and intuitive layout is used including a navigation bar with links to other sections of the website. Client: "Yes." Test:

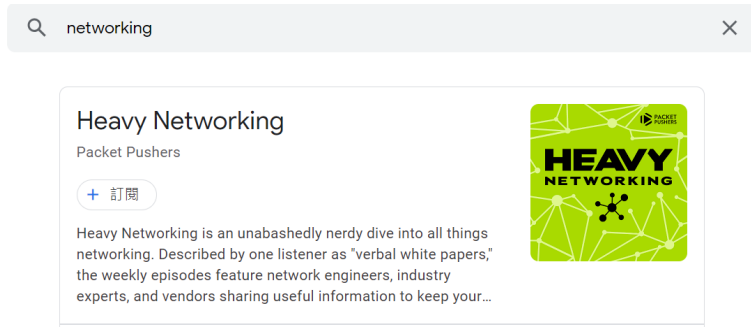
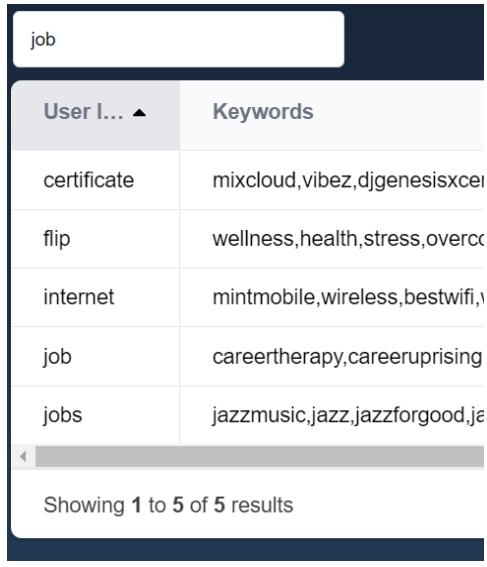
		(Homepage)  (Previous Inputs page)  (Podcast recommendation page)  (Word Cloud page) 
1) c) Input processing using Flask to define routes that can handle the inputs	Yes	User directed to podcast recommendation page using Flask after input is processed as demonstrated in the video. Client: “Yes, after submitting, the website directs me to a new page with the relevant results.”
1 d) i) Increase efficiency by storing previous input in	Yes	Based on previous inputs in the database, auto-suggested inputs can be entered and efficiently processed, leading user directly to the podcast recommendation page as

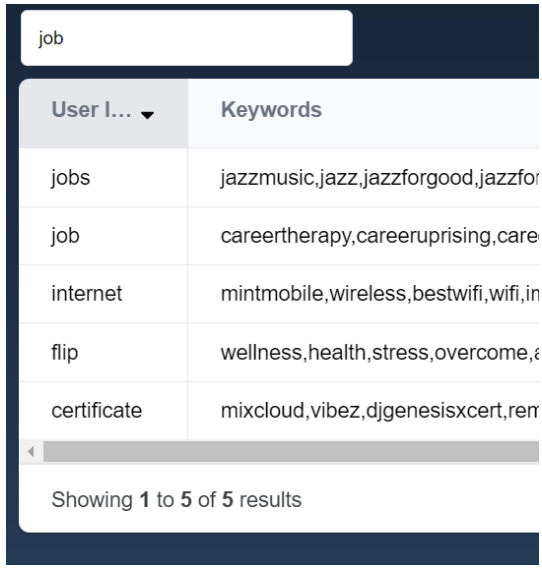
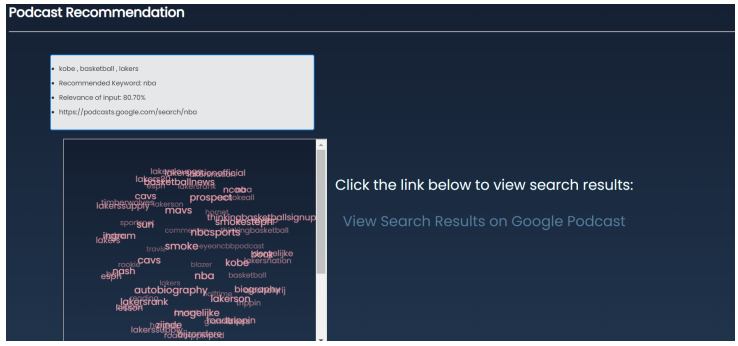
the database as cache		demonstrated in the video. Client: “Yes.”
1) d) ii) Implement auto-suggestions to assist users in inputting meaningful keywords	Yes	Auto-suggestion is demonstrated in the video. Client: “Yes.” Test: 
1) d) iii) Data cleaning techniques include converting plural words to their singular form and removing stopwords, etc.	Yes	Data cleaning techniques are demonstrated in the video for all the extreme cases. Client: “Indeed, it converts plural to singular words and returns errors when a stopword or random letter scrambles are entered.” Test:

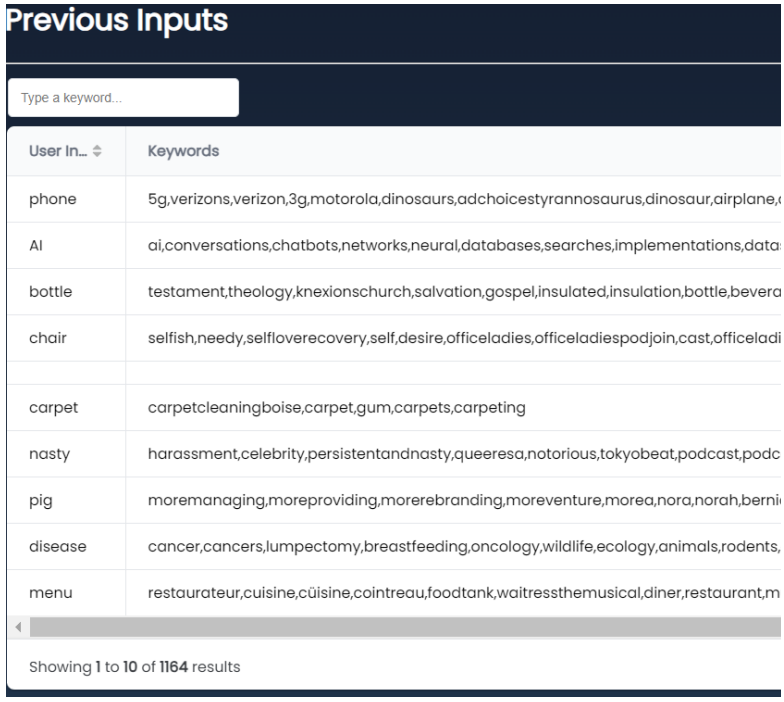
		   
2) b) Use client input to connect to Google	Yes	If no error is returned upon submitting inputs, a request to the API was successful as shown in the video.

Podcast API		Client: “Yes, this works as expected.”
3) a) i) Create keywords from the description of each podcast related to the user inputs using keyBERT	Yes	Keywords are displayed in the word cloud of the podcast recommendation page. Client: “Yes.” Test: 
3) b) i) A high confidence score is given for similar inputs and vice versa	Yes	For similar words like “basketball, lebron, hoop”, the website returns a high confidence score (>75%) and vice versa (<50%). Client: “Yes.” Test: 

		• christmas , lens , menu • Recommended Keyword: restaurant • Relevance of input: 45.64%
3) b) ii) Use the Word2Vector model to find a centroid from all of the extracted keywords processed	Yes	All relevant information relating to the centroid is demonstrated in the video. Client: “Indeed, the centroid distance is shown in the terminal. And the recommended keyword is shown on the page.”
3) b) iii) Find the most relevant recommended keyword based on Euler distance to the centroid	Yes	The recommended keyword based on Euler distance to the centroid vector is calculated and given as shown in the video. Client: “Yep.” Test: • phone , school , create • Recommended Keyword: networking
3) c) Feed centroid word back to API to find relevant podcast episodes to be displayed	Yes	The podcast recommendation page would display a link, directing user to the Google Podcast API with the search query for the recommended keyword generated. Client: “Yes.” Test: Click the link below to view search results: View Search Results on Google Podcast

		
4 a) Sort inputs and keywords alphabetically	Yes	Can the datatable in the previous inputs page be sorted alphabetically by clicking on the columns? Client: “The functionality works well. ” Test:  Upon clicking,

		
4) b) Fetch corresponding keywords when the user enters input in datatable search box of the previous inputs page	Yes	A word cloud should be displayed as demonstrated in the video. Client: “That’s correct.”
5 a) Podcast recommendation page with user inputs, relevancy of recommended result, connection to API and word cloud of relevant keywords	Yes	The podcast recommendation page contains all the elements described and are fully functional. Client: “Yes, it does.” Test: 

5 b) Database of previous inputs and relevant keywords	Yes	The data table in the previous inputs page is shown. Client: “Yes.” Test: 
5) c) Word cloud of keywords when the user enters input in the previous inputs page	Yes	Corresponds to 4) b)
5) d) A uniform simple bootstrap template across all pages with button that can redirect user back to homepage	Yes	Uniform UI demonstrated in video. Client: “The navigation bar is preserved and the background stays the same, so I would agree.”

Recommendations for Future Development

In the final interaction, my client suggested improving keyword extraction by exploring other advanced NLP models like GPT for better semantic and contextual understanding.

We agreed that scraping podcasts from multiple platforms like spotify, Apple Podcast, etc provides more accurate recommendations and decreases dependency on a singular external platform, increasing reliability.

My client also proposed personalised user accounts, storing preferences and tracking podcast recommendation history. I realised measures like encryption and access limits must be implemented for securing account data.

Additionally, I recommended autonomous updating of the database with new inputs, increasing the probability that they correspond to user inputs, as fetching data from cache is much more efficient than processing new user inputs.

Lastly, I mentioned whether machine learning models can be trained to personalise recommendations based on user preferences.

Word Count: 520