The Faculty of Information and Communication Technology

Mahidol University

Project 2: Prototype IR System

Search engine for Spotify chart top 120 at Dec 24 2023 - Jan 4 2024 in Thailand

Benjaphol Kositanon	6588051
Vichayuth Ngamsittipong	6588061
Punnathorn Laohachote	6588179

ITCS 414 Information Retrieval and Storage

Assoc. Prof. Dr. Suppawong Tuarob

November 25, 2024

Table of Contents

Introduction

In this project, an extensive search engine for Spotify's Top 120 chart in Thailand will be used between December 24, 2023, and January 4, 2024. Spotify, a well-known music streaming service worldwide, provides information on the most streamed and popular songs during a given period through its Top Charts. This search engine's main objective is to give users deep details about these top-charting songs in an easily accessible way so they can learn more about the popular music. It is also advantageous for those who are struggling with Thailand's music industry and want to learn about the well-known songs from this period. All 120 songs and metadata appear in a search engine, which enables users to do searches using a variety of keywords, including song name, artist, genre, description, and lyrics. The goal is to develop a user-friendly and informative resource that enables both casual users and passionate supporters to discover Thailand's music industry at the current time.

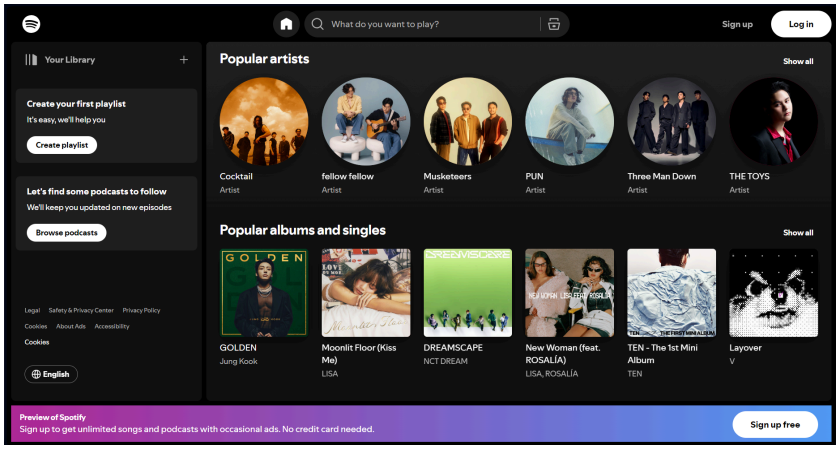
Problems we are trying to solve

People's daily routines, activities, and quiet times often come with music, which plays a vital role in their life. Even though a lot of people like to listen to music, some people need help remembering the names of the songs they hear. Instead, they might recall the name of a producer or musician they like, or a little part of the lyrics that spoke to them. Others may have a connection to music because they are passionate fans of a certain musician or producer and want to learn more about their work.

Since we want to satisfy these needs, we have created an individual search engine that relies on Spotify's Top 120 Charts in Thailand to guide users through music trends from December 24, 2024, to January 4, 2025. Users may find and connect with their favorite music even if they are unaware of its title according to this search engine, which allows users to find songs by lyrics, artist names, producers, and more. Our website improves how people experience and explore music by filling the gap between curiosity and connection by allowing a variety of music discovery interests.

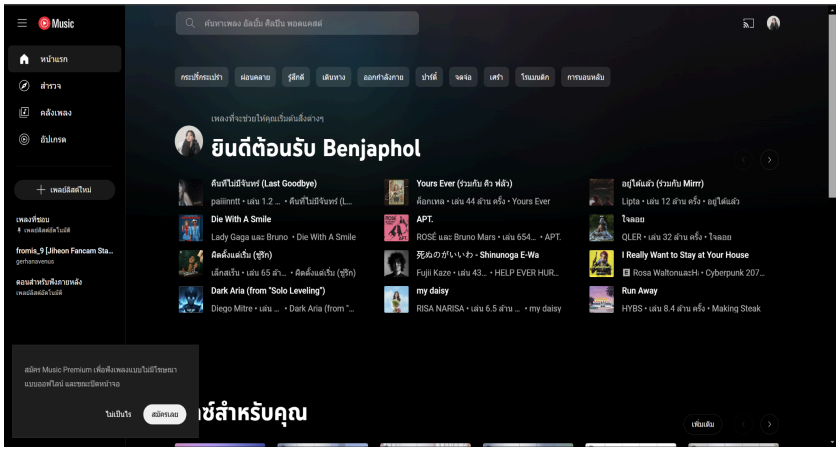
Existing relevant system

Spotify



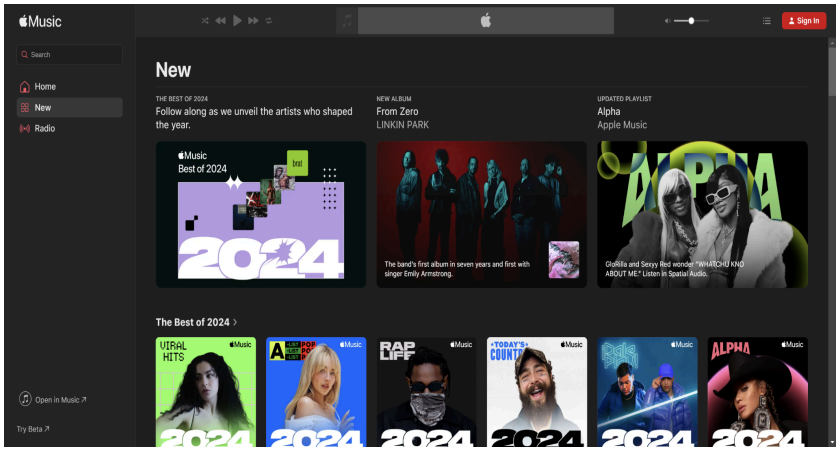
Spotify is a popular digital music streaming service that gives users access to millions of songs and podcasts from artists all over the world. Users can listen to music on the computer, phone, tablet, or other devices. Spotify offers both a free and a premium subscription plan. With a free plan, users can listen to music with ads, and users can't skip songs on shuffle mode. With a premium plan, users can download songs to listen to offline.

Youtube Music



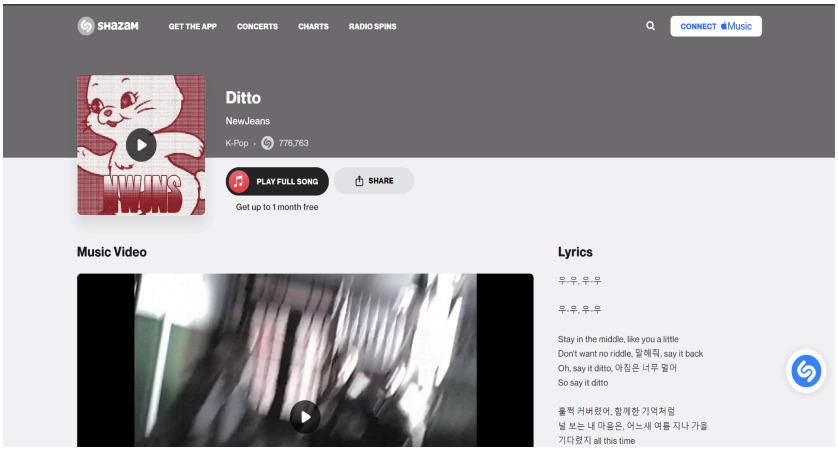
Youtube Music is a music streaming service developed by Google. It allows users to listen to millions of songs and podcasts, similar to Spotify. However, it also offers music videos, which sets it apart from other music streaming services. Youtube Music offers both a free and a premium subscription plan. The free plan allows users to listen to music with ads and limited features, while the premium plan removes ads, enables background play, and offers offline listening.

Apple Music



Apple Inc. created Apple Music, a music streaming service that requires a subscription. Users can access a huge collection of songs, albums, playlists, and unique content through it. Since its June 2015 launch, Apple Music has grown to become one of the most popular music streaming services globally.

Shazam



Shazam is a music identification service and mobile app that lets users listen to a brief audio sample to find and identify songs, TV series, and ads. Shazam Entertainment created the app, which recognizes audio content using cutting-edge audio fingerprinting technology.

Implementation

i.) Data collection example document, and Data statistics:

Our data collection method uses a manual procedure to assure accuracy and relevancy. The data is mainly taken from Spotify's music charts, only the Top 120 Charts in Thailand. The collected data include the music's name, artist, genre, release date, chart position, producer, songwriter, streams, lyrics, image, and link. First, all collected information is saved and arranged in Google Sheets for convenient access, collaboration, and modifications. However, in order to improve working with Elasticsearch, the data has been converted from CSV to JSON format. This transformation ensures connection with Elasticsearch's document-based structure, allowing easy indexing and querying of the data. The JSON files serve as the foundation for building a flexible and efficient search system that can perform complex analytics and provide current information on music trends.

Example of CSV file:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Artist	Genre	Release Date	Position in chart	Producer	Songwriter	Streams	Lyrics	Description	Song Image Link	Spotify Link
2	Standing Next to	Jung Kook	K-Pop, Funk, Dis	3 November 202	1	Andrew Watt, Ci	Ali Tamposi, Anc	2,964,873	Standing next to	This song is a m	https://i.scdn.co/	https://open.spot
3	Seven (feat. Latt	Jung Kook, Latto	K-Pop, UK garag	3 November 202	2	Andrew Watt, Ci	Andrew Watt, He	2,899,289	Weight of the wo	Seven' is a song	https://i.scdn.co/	https://open.spot
4	wherever u r (ft.	UMI, V	R&B, Soul	30 December 20	3	V-Ron, Sagun Kl	Anoop D'Souza,	2,728,711	Ah	Wherever u r by	https://i.scdn.co/	https://open.spot
5	Love Me Again	V	K-Pop, R&B	11 August 2023	4	freekind., Jinsu F	Donghyun Kim, f	1,875,529	Ooh, let me love	"Love Me Again"	https://i.scdn.co/	https://open.spot
6	Closer Than This	Jimin	K-Pop, R&B, Po	22 December 20	5	GHSTLOOP, Ay	August Rigo, Ay	1,638,791	너도 기억하니? .	Closer Than This	https://i.scdn.co/	https://open.spot
7	3D (feat. Jack H	Jung Kook, Jack	K-Pop, R&B	2 October 2023	6	BloodPop®, Dav	BloodPop®, Dav	1,240,800	One, two, 3D	'3D' is an R&B p	https://i.scdn.co/	https://open.spot
8	Like Crazy	Jimin	K-Pop, Synth po	24 March 2023	7	Pdogg, GHSTLC	BLVSH, Chris Ja	1,161,329	(I think we could	Like Crazy is a s	https://i.scdn.co/	https://open.spot
9	ดาวหางฮัลเลย์ (H	fellow fellow	T-Pop, Pop	29 August 2023	8	Tan Liptapallop,	Panithi Lertudom	942,407	ระหว่างที่มีดาวดว	"ดาวหางฮัลเลย์ (H	https://i.scdn.co/	https://open.spot
10	Money	LISA	K-Pop, Hip-hop	10 September 2	9	TEDDY	Bekuh Boom, 24	921,754	It's the end of the	"Money" is descr	https://i.scdn.co/	https://open.spot
11	LALISA	LISA	K-Pop, Hip-hop,	10 September 2	10	TEDDY	Bekuh Boom, 24	819,099	내 뒷모습만 봐도	"Lalisa" is the so	https://i.scdn.co/	https://open.spot
12	ฟ้ารักพ่อ (DILF)	BADMIXY, ยุ้ย ญ	T-Pop, Pop	2 December 202	11	BADMIXY	benlussboy, วันเ	788,523	He missed me a	ฟ้ารักพ่อ (DILF) '	https://i.scdn.co/	https://open.spot
13	อิจฉา	MEYOU	T-Pop, Pop	1 December 202	12	TimeTime	MEYOU & PURE	688,595	ที่น้ำตามันไหลเพร	เพลง อิจฉา ได้พูด	https://i.scdn.co/	https://open.spot
14	วันนี้ปีที่แล้ว	MEYOU	T-Pop, Pop	10 August 2023	13	Peerapon lamjar	MEYOU & Peera	671,435	ในวันนี้เมื่อปีที่แล้ว	เพลง "วันนี้ปีที่แล้ว	https://i.scdn.co/	https://open.spot
15	One Of The Girls	The Weekend, ,	R&B, Synth pop,	23 June 2023	14	The Weeknd, MI	MIKE DEAN, Ab	630,355	Lock me up and	"One of The Girls	https://i.scdn.co/	https://open.spot
16	ร้องไปกับฟ้า (Orig	D Gerrand, LHA	T-Pop	4 December 202	15	WITTAWIN ARM	D Gerrard, SOM	596,677	บทชีวิตขีดทางให้	ตอนแรกที่แต่ง ' ร้	https://i.scdn.co/	https://open.spot
17	ก่ำแพงหัวใจ	Mirrr	T-Pop, Pop	17 August 2023	16	Letach Ketsook,	Letach Ketsook	590,016	ใต้โปรดเถอะขอร้อ	คอนเสปมันคือเรา	https://i.scdn.co/	https://open.spot
18	FLOWER	JISOO	K-Pop, Dance-P	31 March 2023	17	24	24, KUSH, TEDI	587,707	Eh-eh-eh-eh	Jisoo's first singl	https://i.scdn.co/	https://open.spot
19	ปล่อยดาว	YEW	Alternative, Rock	31 August 2023	18	YEW	Prat Parnploy	584,605	หากฉันได้ขอสิ่งใด	เพลง ปล่อยดาว -	https://i.scdn.co/	https://open.spot
20	ยินดี - เพลงประกอ	sarah salola	T-Pop	12 June 2023	19	Guntapich	อัจฉริยา คลยไพบ	535,333	ไม่ต้องอิดถึงวันเ	เพลงของสายชัพร์	https://i.scdn.co/	https://open.spot

Example of JSON file:

ii.) Tools and software:

1. **Visual Studio Code (VS):**

   The Python and HTML code for the project was written and edited using Visual Studio Code. To increase efficiency and preserve code quality, the team made use of its extensions, which included Python-specific tools and linters. Because of its integrated Git integration, version control was smooth, facilitating team member collaboration and code management. Its live server extension also made it simpler to test and preview the web interface while it was being developed.

2. **Elasticsearch:**

   The dataset, which contained multilingual Spotify rankings in Thai, English, and Korean, was indexed with Elasticsearch. Its features, which included tools such as analysis tools and query builders, enabled speedy and exact search queries. The team was able to maximize search accuracy by modifying scoring algorithms and addressing language-specific search needs, by using Elasticsearch.

3. **Kibana:**

   Kibana was used to track and evaluate Elasticsearch's performance. It enabled the visualization of search query trends and results and offered insights into the indexed data. Throughout the project, this tool was crucial for debugging and confirming the accuracy of the indexing and search. It also aided in monitoring problems with query handling and data processing.

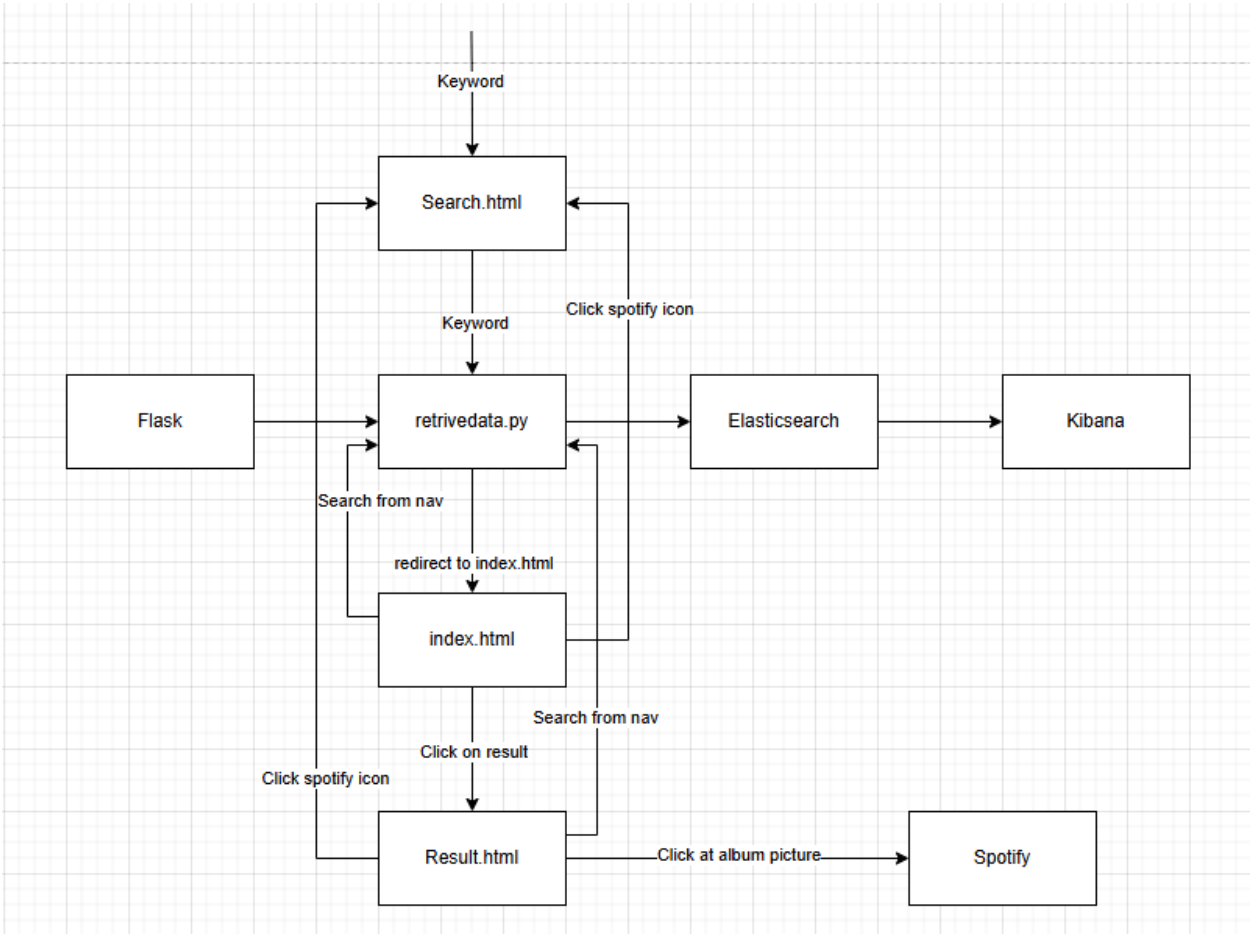
4. **Flask:**

   The backend was supported by Flask, which oversaw the APIs that linked Elasticsearch and the frontend interface. It managed query handling, data processing, and routing, guaranteeing smooth interaction between the search engine and the user interface. Despite their lack of prior Python experience, the team was able to swiftly prototype and implement backend functionality thanks to its simplicity and modularity.

5. **Google Sheets:**

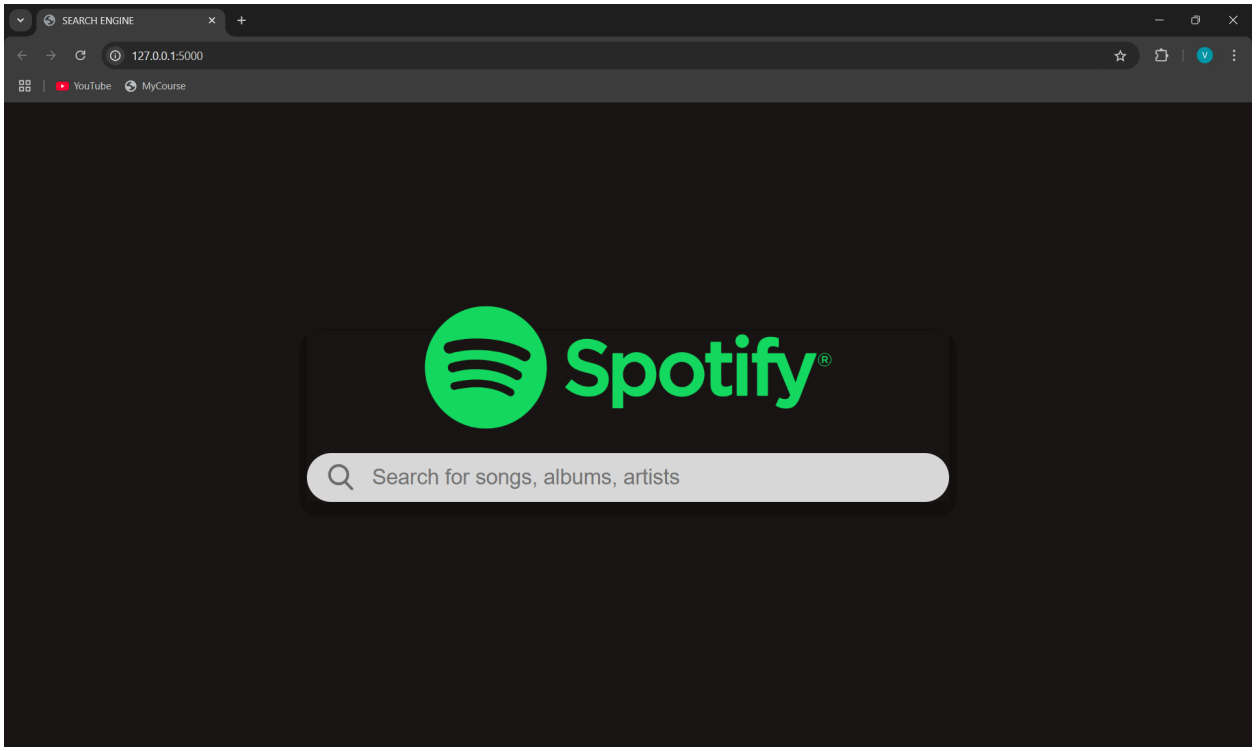
   To gather, clean, and arrange the dataset of 120 Spotify rankings, Google Sheets was utilized. To make sure the data was correct and organized before indexing it in Elasticsearch, it gave the team a platform to work together in real-time. It also provided a rapid and convenient means of monitoring modifications and validating the dataset among several team members.

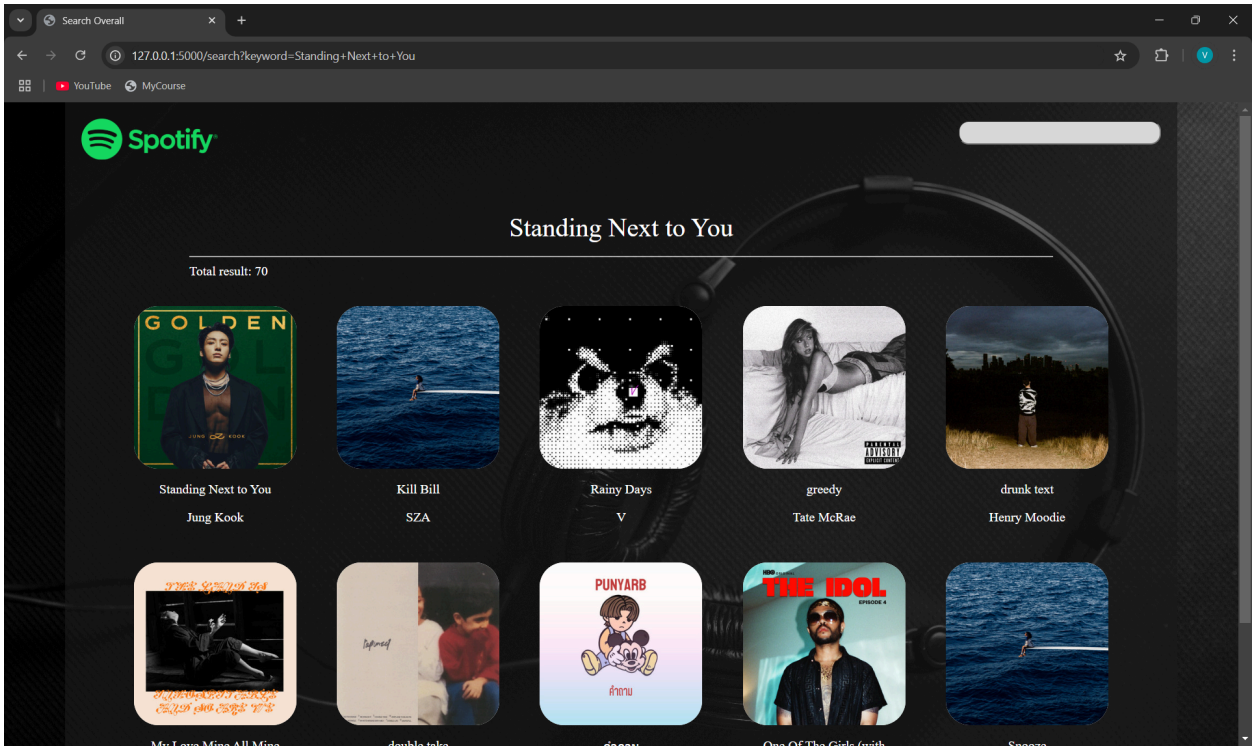
iii.) System diagram:

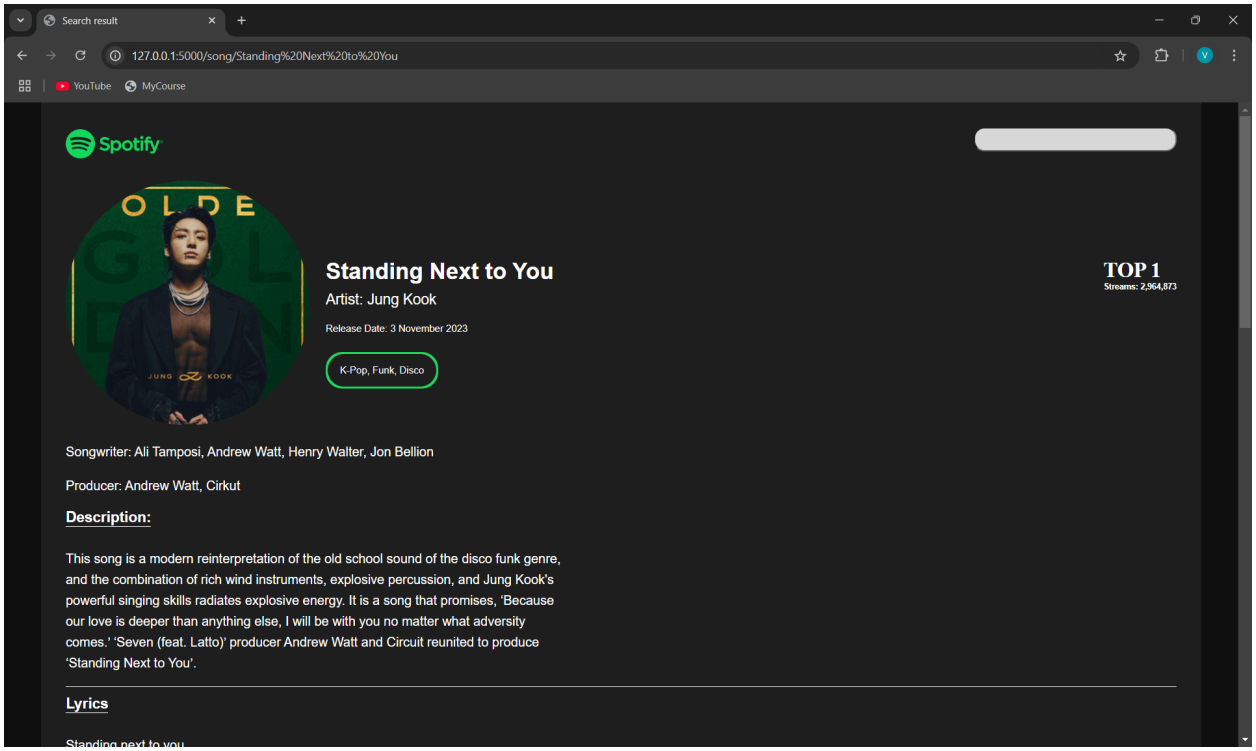


Initially, the song's information, which is stored in a JSON file, is indexed in Elasticsearch via the interface of Kibana, and then Elasticsearch's powerful query DSL is utilized via Flask to search and retrieve data through Python programming language. Additionally, the final web page is styled by using HTML, JS, and CSS to attract the users who search for the song's information by inserting a query in the search box on the web page.

iv.) Snapshot of the system:

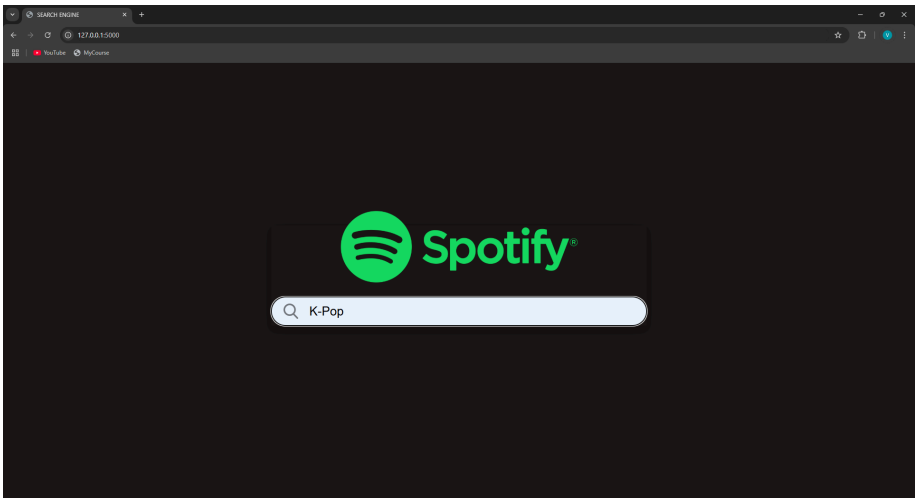


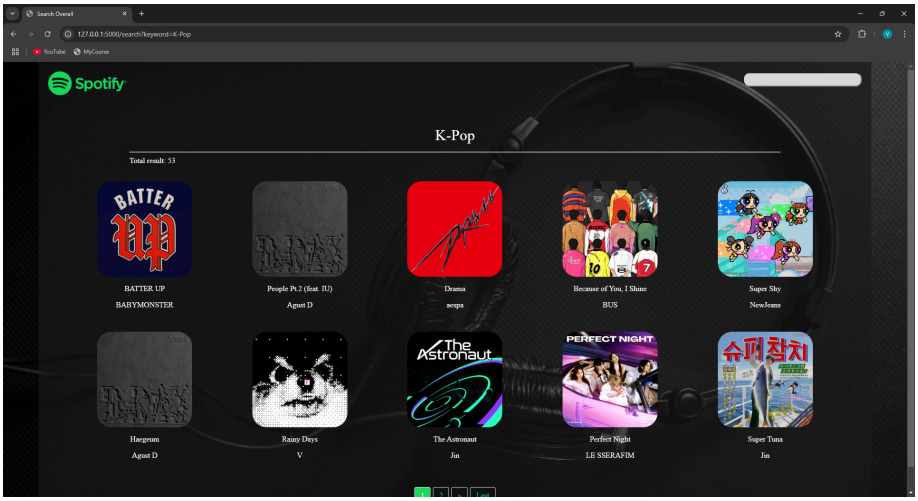

Search Overall
127.0.0.1:5000/search?keyword=Standing+Next+to+You
YouTube
MyCourse
Spotify
Standing Next to You
Total result: 70
Standing Next to You
Jung Kook
Kill Bill
SZA
Rainy Days
V
greedy
Tate McRae
drunk text
Henry Moodie
PUNYARB
THE IDOL


Search result
127.0.0.1:5000/song/Standing%20Next%20to%20You
YouTube
MyCourse
Spotify
Standing Next to You
Artist: Jung Kook
Release Date: 3 November 2023
K-Pop, Funk, Disco
TOP 1
Streams: 2,964,873
Songwriter: Ali Tamposi, Andrew Watt, Henry Walter, Jon Bellion
Producer: Andrew Watt, Cirkut
Description:
This song is a modern reinterpretation of the old school sound of the disco funk genre, and the combination of rich wind instruments, explosive percussion, and Jung Kook's powerful singing skills radiates explosive energy. It is a song that promises, 'Because our love is deeper than anything else, I will be with you no matter what adversity comes.' 'Seven (feat. Latto)' producer Andrew Watt and Circuit reunited to produce 'Standing Next to You'.
Lyrics

v.) Example step-by-step search sessions that highlight the following functionality:

1. One word query - “K-Pop”



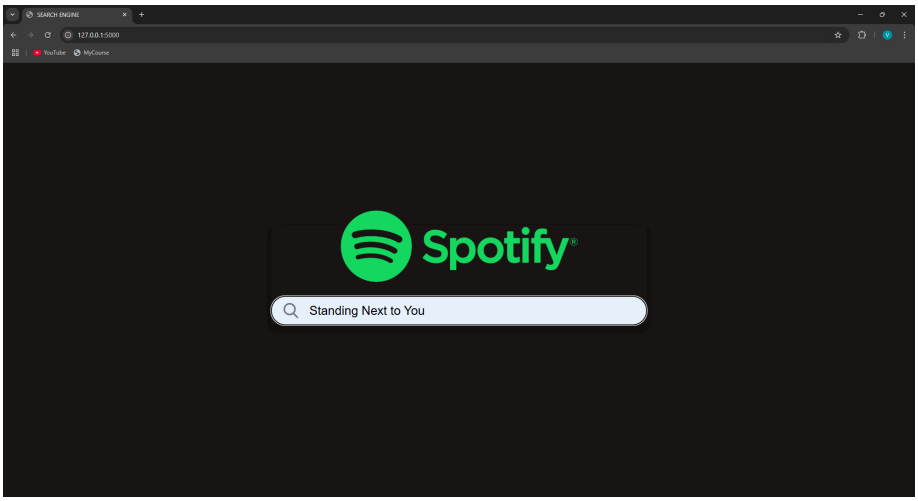


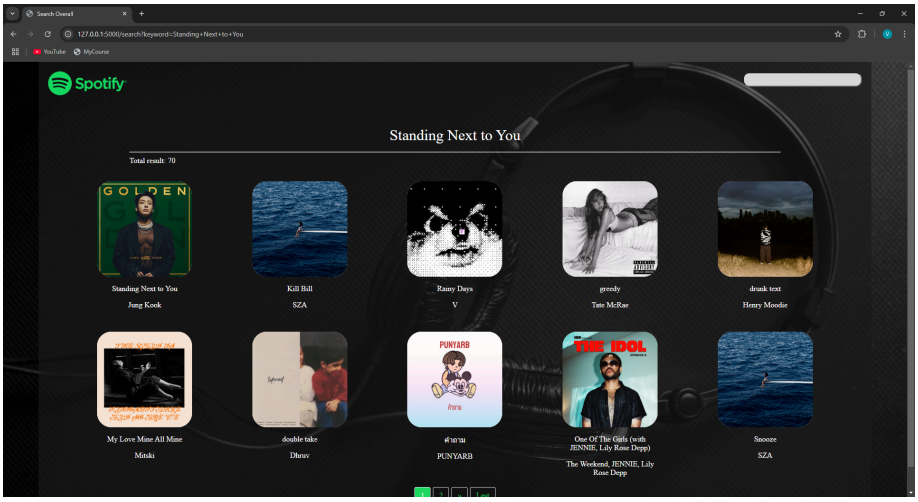
When users input the query “K-Pop” into a search box, the system will respond by directing the user to the result page

```
body = {
    'size' : page_size,
    'from' : page_size * (page_no-1),
    "query": {
        "function_score": {
            "query": {
                "bool": {
                    "should": [
                        {
                            "match_phrase_prefix": {
                                "Name": keyword
                            }
                        },
                        {
                            "multi_match": {
                                "query": keyword,
                                "fields": ["Name", "Description", "Genre", "Artist", "Lyrics", "Producer", "Songwriter", "Release Date"],
                                "fuzziness": "AUTO"
                            }
                        },
                        {
                            "match": {
                                "Genre": {
                                    "query": keyword,
                                    "boost": 2
                                }
                            }
                        }
                    ],
                    "minimum_should_match" : 1,
                    "boost" : 2.0
                }
            },
            "functions": [
                {
                    "filter": { "exists": { "field": "Name" } },
                    "weight": 3
                }
            ],
            "score_mode": "sum"
        }
    }
}
```

For the back end, the Flask app receives the query “K-Pop” and calls Elasticsearch search API with the query body. Then, Elasticsearch search API lists the documents that have the highest to lowest score calculated by the “function_score” function and returns to the Flask app. Finally, it parses the Elasticsearch response and extracts the following fields from the above pictures.

2. Multiple word query - “Standing Next to You”



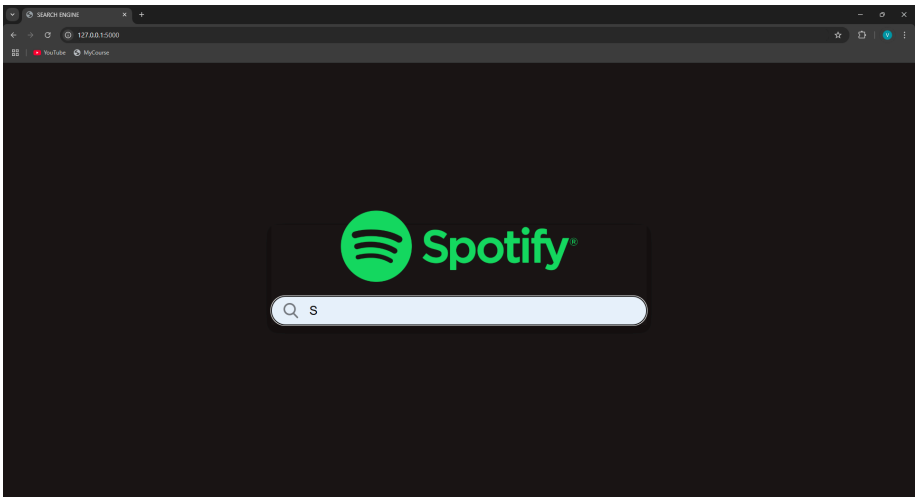


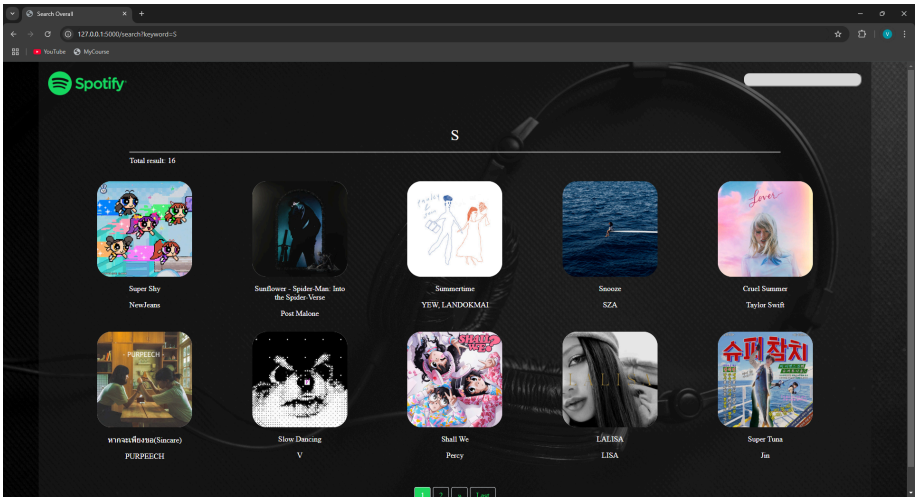
When users input the query “Standing Next to You” into a search box, the system will respond by directing the user to the result page

```
body = {
    'size' : page_size,
    'from' : page_size * (page_no-1),
    "query": {
        "function_score": {
            "query": {
                "bool": {
                    "should": [
                        {
                            "match_phrase_prefix": {
                                "Name": keyword
                            }
                        },
                        {
                            "multi_match": {
                                "query": keyword,
                                "fields": ["Name", "Description", "Genre", "Artist", "Lyrics", "Producer", "Songwriter", "Release Date"],
                                "fuzziness": "AUTO"
                            }
                        },
                        {
                            "match": {
                                "Genre": {
                                    "query": keyword,
                                    "boost": 2
                                }
                            }
                        }
                    ],
                    "minimum_should_match" : 1,
                    "boost" : 2.0
                }
            },
            "functions": [
                {
                    "filter": { "exists": { "field": "Name" } },
                    "weight": 3
                }
            ],
            "score_mode": "sum"
        }
    }
}
```

For the back end, the Flask app receives the query “Standing Next to You” and calls Elasticsearch search API with the query body. Then, Elasticsearch search API lists the documents that have the highest to lowest score calculated by the “function_score” function and returns to the Flask app. Finally, it parses the Elasticsearch response and extracts the following fields from the above pictures.

3. Partial match - “S”



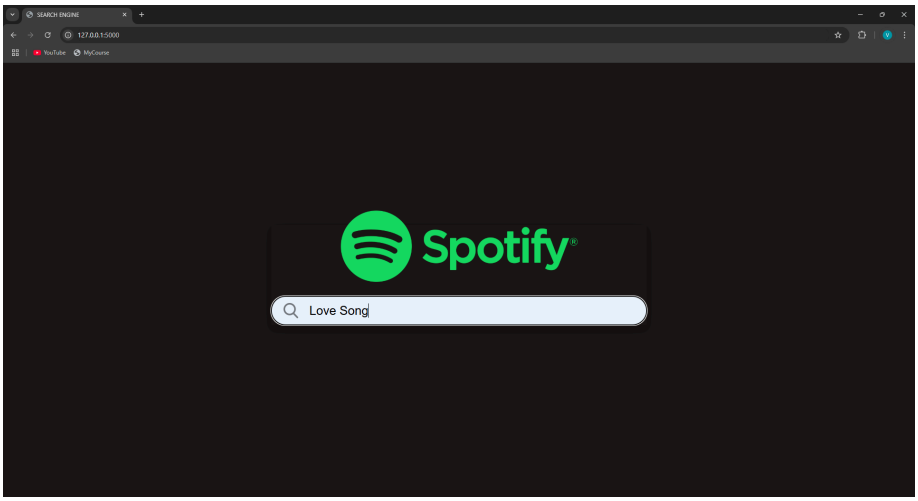


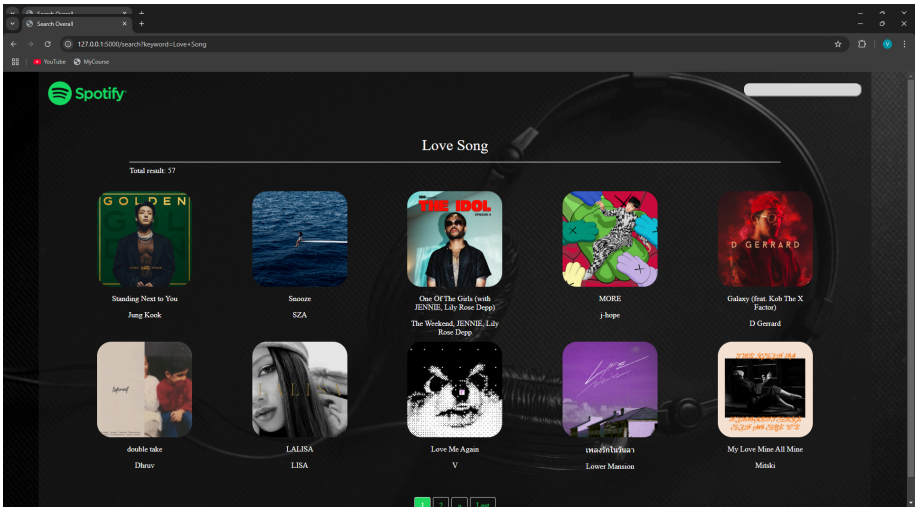
When users input the query “S” into a search box, the system will respond by directing the user to the result page

```
body = {
    'size' : page_size,
    'from' : page_size * (page_no-1),
    "query": {
        "function_score": {
            "query": {
                "bool": {
                    "should": [
                        {
                            "match_phrase_prefix": {
                                "Name": keyword
                            }
                        },
                        {
                            "multi_match": {
                                "query": keyword,
                                "fields": ["Name", "Description", "Genre", "Artist", "Lyrics", "Producer", "Songwriter", "Release Date"],
                                "fuzziness": "AUTO"
                            }
                        },
                        {
                            "match": {
                                "Genre": {
                                    "query": keyword,
                                    "boost": 2
                                }
                            }
                        }
                    ],
                    "minimum_should_match" : 1,
                    "boost" : 2.0
                }
            },
            "functions": [
                {
                    "filter": { "exists": { "field": "Name" } },
                    "weight": 3
                }
            ],
            "score_mode": "sum"
        }
    }
}
```

For the back end, the Flask app receives the query “S” and calls Elasticsearch search API with the query body. Then, Elasticsearch search API lists the documents that have the highest to lowest score calculated by the “function_score” function and returns to the Flask app. Finally, it parses the Elasticsearch response and extracts the following fields from the above pictures.

4. Ranking - “Love Song”





When users input the query “Love Song” into a search box, the system will respond by directing the user to the result page

```
body = {
    'size' : page_size,
    'from' : page_size * (page_no-1),
    "query": {
        "function_score": {
            "query": {
                "bool": {
                    "should": [
                        {
                            "match_phrase_prefix": {
                                "Name": keyword
                            }
                        },
                        {
                            "multi_match": {
                                "query": keyword,
                                "fields": ["Name", "Description", "Genre", "Artist", "Lyrics", "Producer", "Songwriter", "Release Date"],
                                "fuzziness": "AUTO"
                            }
                        },
                        {
                            "match": {
                                "Genre": {
                                    "query": keyword,
                                    "boost": 2
                                }
                            }
                        }
                    ],
                    "minimum_should_match" : 1,
                    "boost" : 2.0
                }
            },
            "functions": [
                {
                    "filter": { "exists": { "field": "Name" } },
                    "weight": 3
                }
            ],
            "score_mode": "sum"
        }
    }
}
```

For the back end, the Flask app receives the query “Love Song” and calls Elasticsearch search API with the query body. Then, Elasticsearch search API lists the documents that have the highest to lowest score calculated by the “function_score” function and returns to the Flask app. Finally, it parses the Elasticsearch response and extracts the following fields from the above pictures.

Discussion

i.) Limitations of your system:

- The description of the music might not be true since Spotify doesn't have the description of the song we have to look at throughout the internet.
- We can install Elasticsearch on only one computer. When it comes to the result it takes several minutes to see the result between the front and back end.
- Some of our ranks might not be satisfied because when we rank we include Name, Artist, and Lyrics. Some of the lyrics use the same word and the ranking will be high.

ii.) Technical difficulties and Challenges, and how you handled them:

1. Collaboration:
- Due to the limitation of Elasticsearch access slowed team work.
2. Data Preparation:
- Manual collection and multilingual formatting of Spotify rankings.
3. Enhanced Search engine:
- Improve the query to make the search engine more accurate.

iii.) Lessons learned

1. Tool Proficiency:
    - Gained hands-on experience in Elasticsearch, understanding its indexing, querying, and optimization features.
    - Enhanced proficiency in Python for data processing, scripting, and API integrations.
    - Developed skills in handling multilingual search, addressing challenges like tokenization, stemming, and relevance ranking for multiple languages.

2. Version Control:
    - Improved GitHub practices for more effective team collaboration, including the use of branches, pull requests, and code reviews.
    - Learned how to resolve merge conflicts efficiently and maintain a clean commit history.
3. Search Engine Design:
    - Acquired an end-to-end understanding of search system workflows, from data preparation to query optimization.
    - Learned to design scalable search architectures tailored to specific datasets and user needs.
    - Understood the importance of ranking algorithms to enhance search result relevance.
4. Data Handling:
    - Gained experience in data collection and preparation, including manual processes and automation techniques.
    - Learned how to normalize and clean data to ensure consistency and accuracy.
    - Managed large datasets in Google Sheets and explored better solutions for scalability.

iv.) Opportunities for future improvement

Opportunities for future improvement include automating the data collection process to reduce manual effort and ensure scalability, particularly for handling larger or more dynamic datasets. Enhancing multilingual search capabilities by incorporating advanced natural language processing (NLP) techniques, such as named entity recognition or semantic search, can improve relevance and accuracy. Implementing a more robust backend system, like a dedicated database alongside Elasticsearch, would optimize data storage and retrieval. Additionally, fostering stronger collaboration through regular code reviews and adopting DevOps practices could streamline workflows and ensure higher code quality. Finally, exploring user feedback mechanisms can guide iterative enhancements to the search system, aligning it more closely with user needs.

Conclusion

The Spotify Top 120 Chart Search Engine project showcases a simple yet effective way to help people find music from Thailand's charts during a specific time. By focusing on the most popular songs from December 24, 2023, to January 4, 2024, the project solves common problems like finding a song when you only know part of the lyrics or an artist's name. Using tools like Elasticsearch, Flask, and Google Sheets, the team created a search system that is easy to use and provides helpful results.

Although the system has some limits, such as relying on manual data collection and handling multilingual challenges, the project has been a good learning experience. The team improved their skills in tools, teamwork, and building search systems. They also found areas for improvement, such as making the system faster, easier to use, and better at handling large amounts of data.

In conclusion, this project achieves its goal of helping people discover music while offering a strong base for future improvements. It highlights how simple tools and teamwork can create solutions that are both useful and easy to use.