Songs Recommendation System

The "Song Recommendation System " project aims to revolutionize music streaming experiences by harnessing the power of Spotify's API and AWS services. With the exponential growth of digital music consumption, personalized recommendations have become essential for engaging users and enhancing their enjoyment. By leveraging real-time data from Spotify, the system dynamically generates tailored song suggestions based on user preferences and trends. Through a combination of data collection, feature extraction, ETL processing, and recommendation generation, the system provides an intuitive and seamless user experience. With a focus on scalability, data integrity, and automation, this project sets out to redefine how users discover and enjoy music in the digital age.

Data Sources:

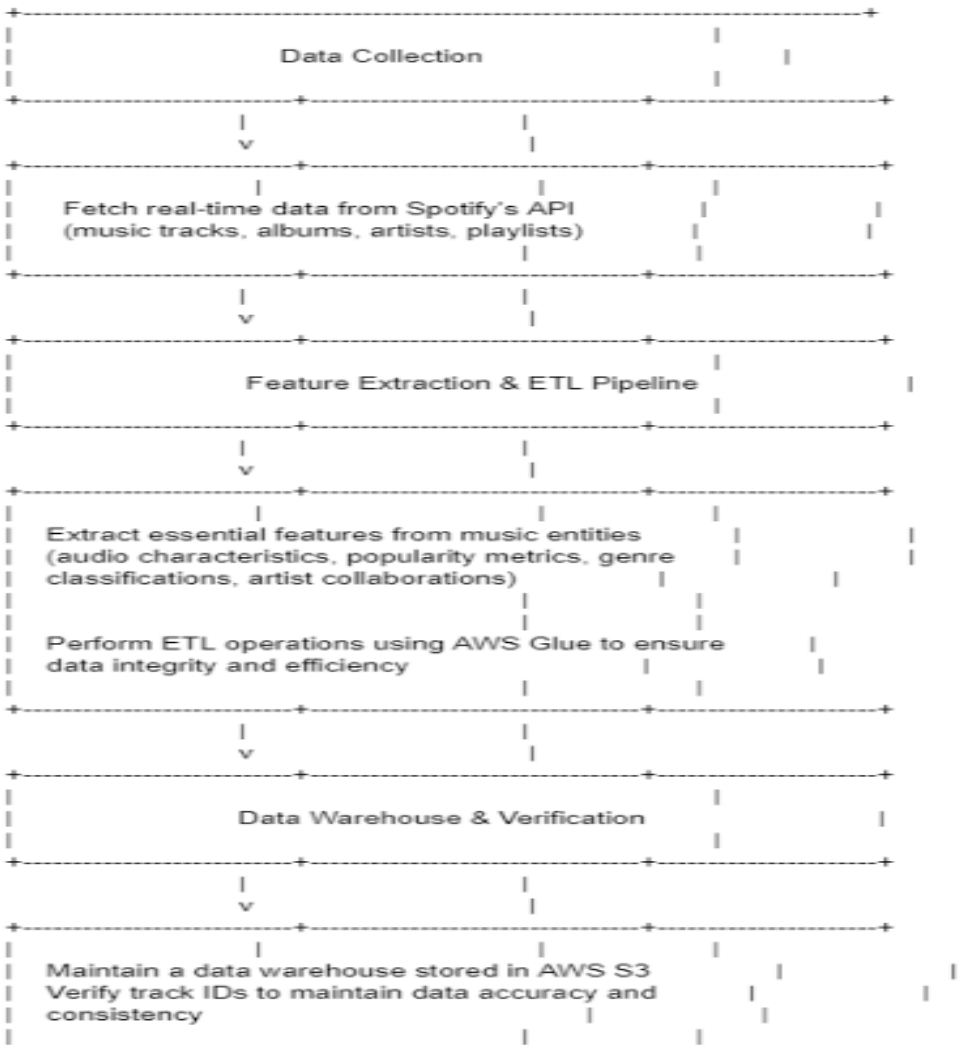
Live streaming Data:
Spotify API Data: to create a dynamic recommendation engine by continuously gathering real-time data from Spotify's API. Leveraging live streaming data enables the system to provide up-to-date and personalized song recommendations to users, enhancing their music streaming experience.
(https://developer.spotify.com/documentation/web-api),

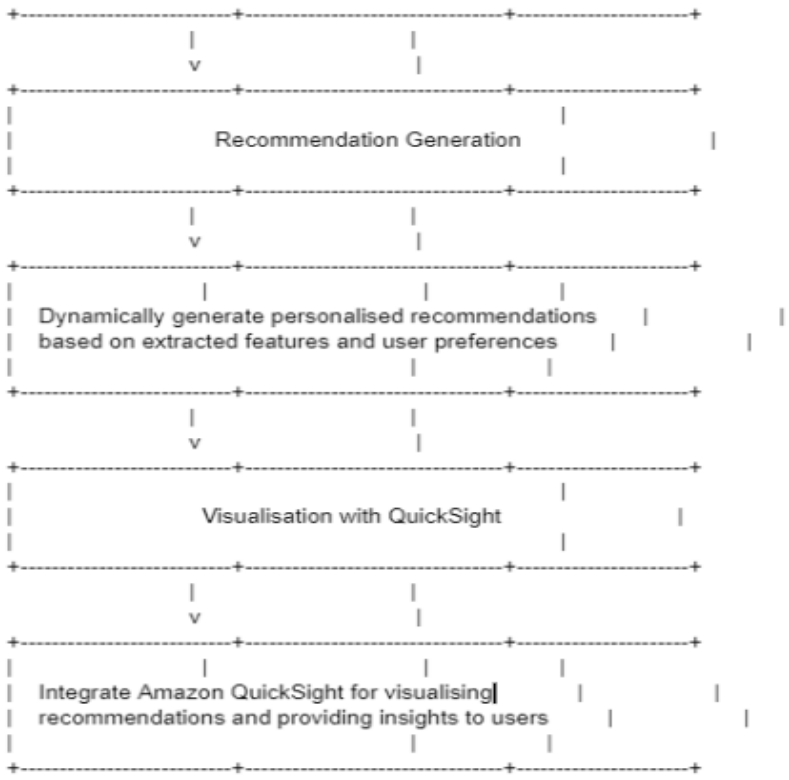
Dataset: Top Spotify Songs 2023](https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023)

The tools used:

- Spotify API: Real-time data on music tracks, albums, artists, and playlists. Primary source for gathering song and user preference information.
- AWS S3 Buckets: Store collected data, ensuring scalability, accessibility, and efficient management.
- AWS Glue: Orchestrate ETL operations, ensuring seamless data processing and transformation.
- Amazon Athena: Query and analyse structured data directly in S3, enabling efficient data retrieval and manipulation.
- Amazon QuickSight: Visualise recommendations and insights, providing an intuitive interface for exploration and discovery.
- Python Libraries:
  Spotipy: Interact with Spotify's API to fetch real-time data.

Pandas: Manipulate and analyse data retrieved from Spotify's API.





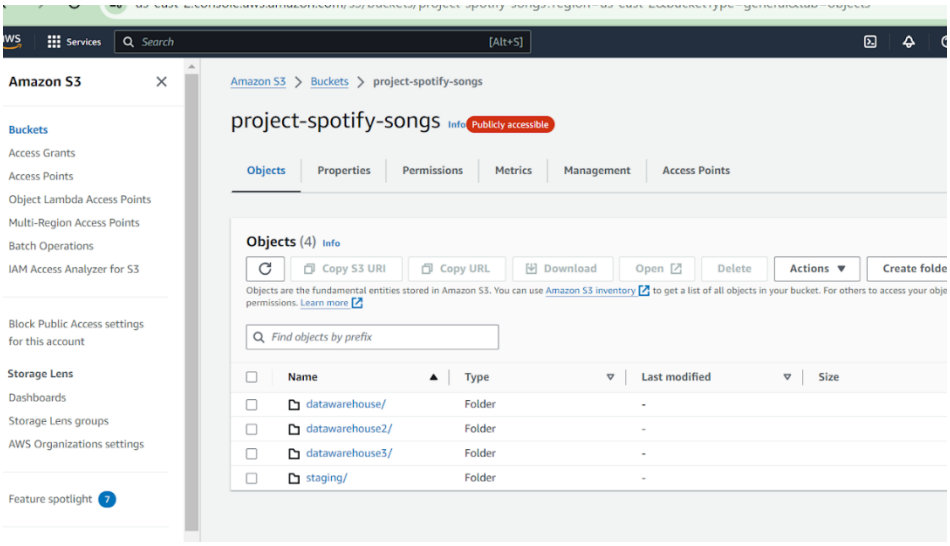
IMPLEMENTATION

Spotify Api:

- We head over to the Spotify Developer Dashboard and sign in or create an account.
- Once logged in, we create a new application by giving it a name (Song Recommender) and brief description.
- After creating our app, we'll get a Client ID and Client Secret, which are like special keys which we used to connect our app to Spotify.

AWS S3 Buckets:

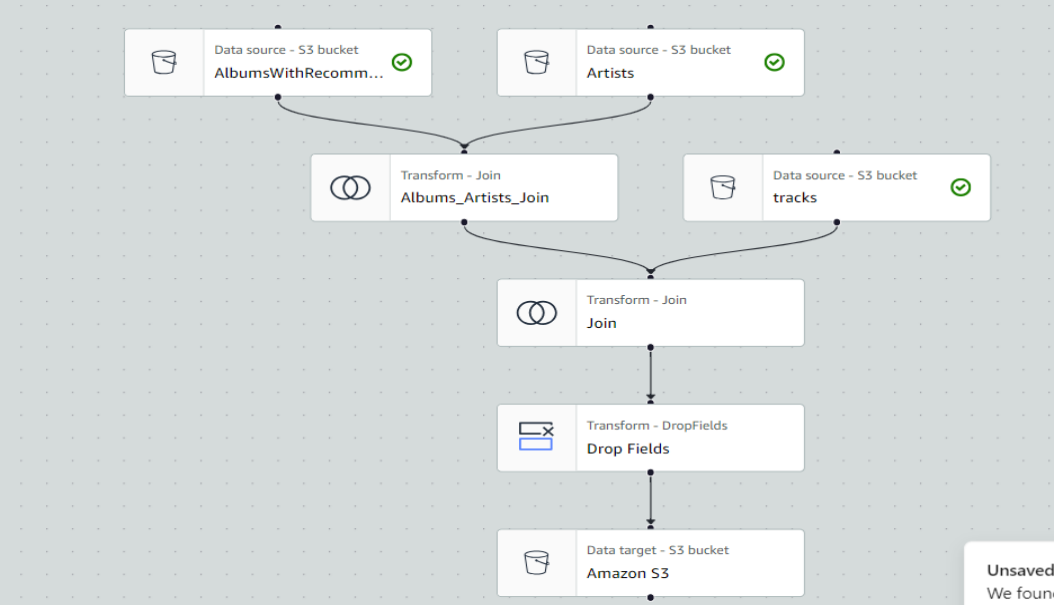
1. Bucket Information:
    - Bucket Name: project-spotify-songs
    - Access: Not publicly accessible
2. Objects in the Bucket:
    - There are the objects listed within the bucket:
        - data warehouse/ - for storing the output recommendations.
        - staging/ - for storing the input datasets for giving the output.



AWS Glue:

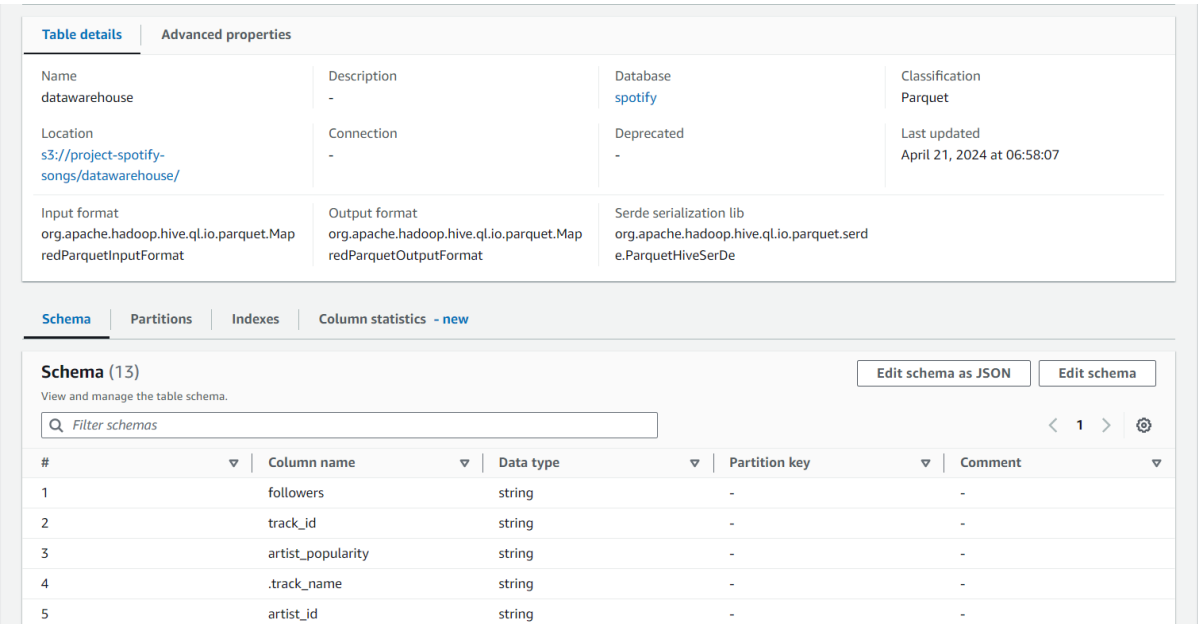
1. Data Sources:
    - There are three data sources represented as S3 buckets:
        - Albums With Recommendations
        - Artists
        - tracks
2. Transformation Steps:
    - Two join transformations are performed:
        - Albums_Artists_Join: Combines data from the “Albums With Recommendations” and “Artists” sources.
        - The output of the first join is then joined with the “tracks” data source.
    - Following the joins, there is a transformation step labelled “Drop Fields,” which likely removes unnecessary fields or columns.
3. Data Target:

    - The final output is stored in another S3 bucket named “Amazon S3.”



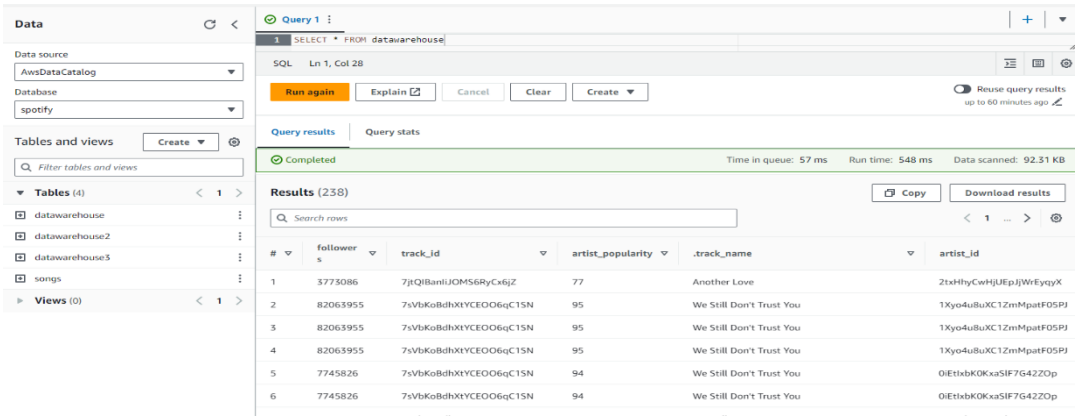
Crawlers:

we employed crawlers to automatically discover and catalogue the schema of the data stored in our S3 buckets.



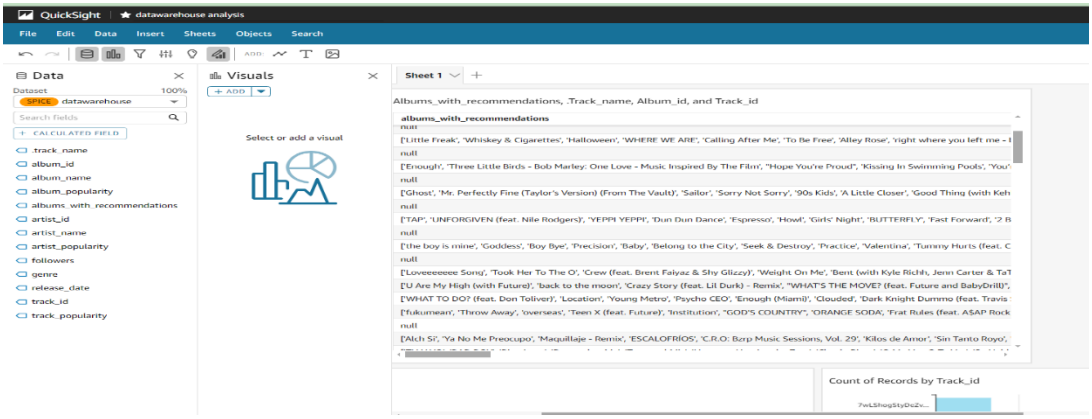
Athena:

We leveraged Amazon Athena for querying and analyzing this structured data directly in S3.



QuickSight:

This enabled us to transform the insights gained from Athena into interactive visualisations and dashboards.



In conclusion, our Song Recommendation System with Spotify API project has successfully demonstrated the power of leveraging real-time music data to enhance the user experience in music streaming platforms. Through meticulous planning and execution, we were able to design and implement a robust infrastructure utilising AWS services such as S3 buckets, Glue, Athena, and QuickSight, along with Python libraries like Spotipy and Pandas.