Spotify Song Recommender

Apple Ko
Christopher Oh

CS 122: Final Project Presentation

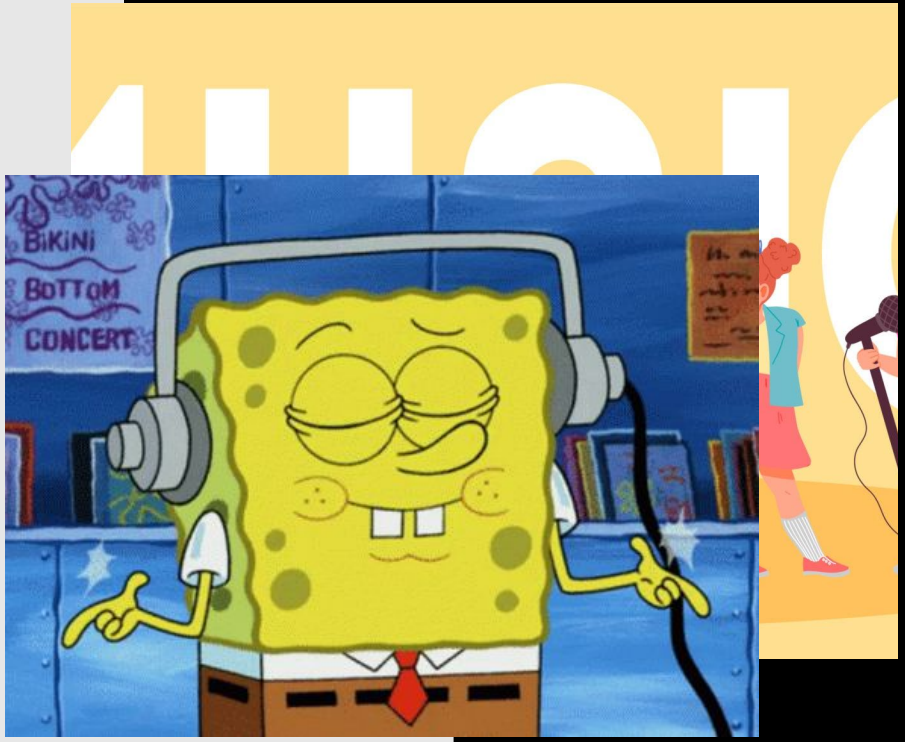
Your Spotify profile isn't just what you listen – it's who you are!

What if you could build your own custom music dashboard to showcase your personality? And also build your own recommendation system?

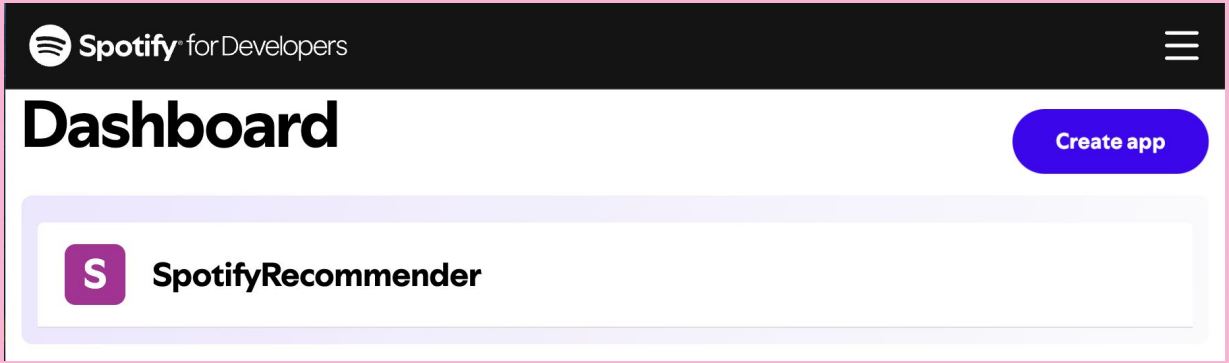
Project Goals

- Recreate your own Spotify music profile that can be customized

- Build your own personal recommendation system to discover new artists and albums



Spotify App, Libraries & API calls



Spotify API calls used:

sp.current_user()

sp.current_user_top_artists()

sp.current_user_top_tracks()

sp.search()

```
# Install spotipy library
!pip install spotipy
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter # to keep track of genre count
from IPython.display import display, Image, HTML # to make dashboard interactive w/ images and HTML
import requests # to make GET requests
```

How it works!

This project was made entirely on Google Colab Notebook, therefore does not require the user to make any extra downloads onto their device. All you need to do is 3 simple steps.

Requires a pre-existing Spotify account.

Step #1

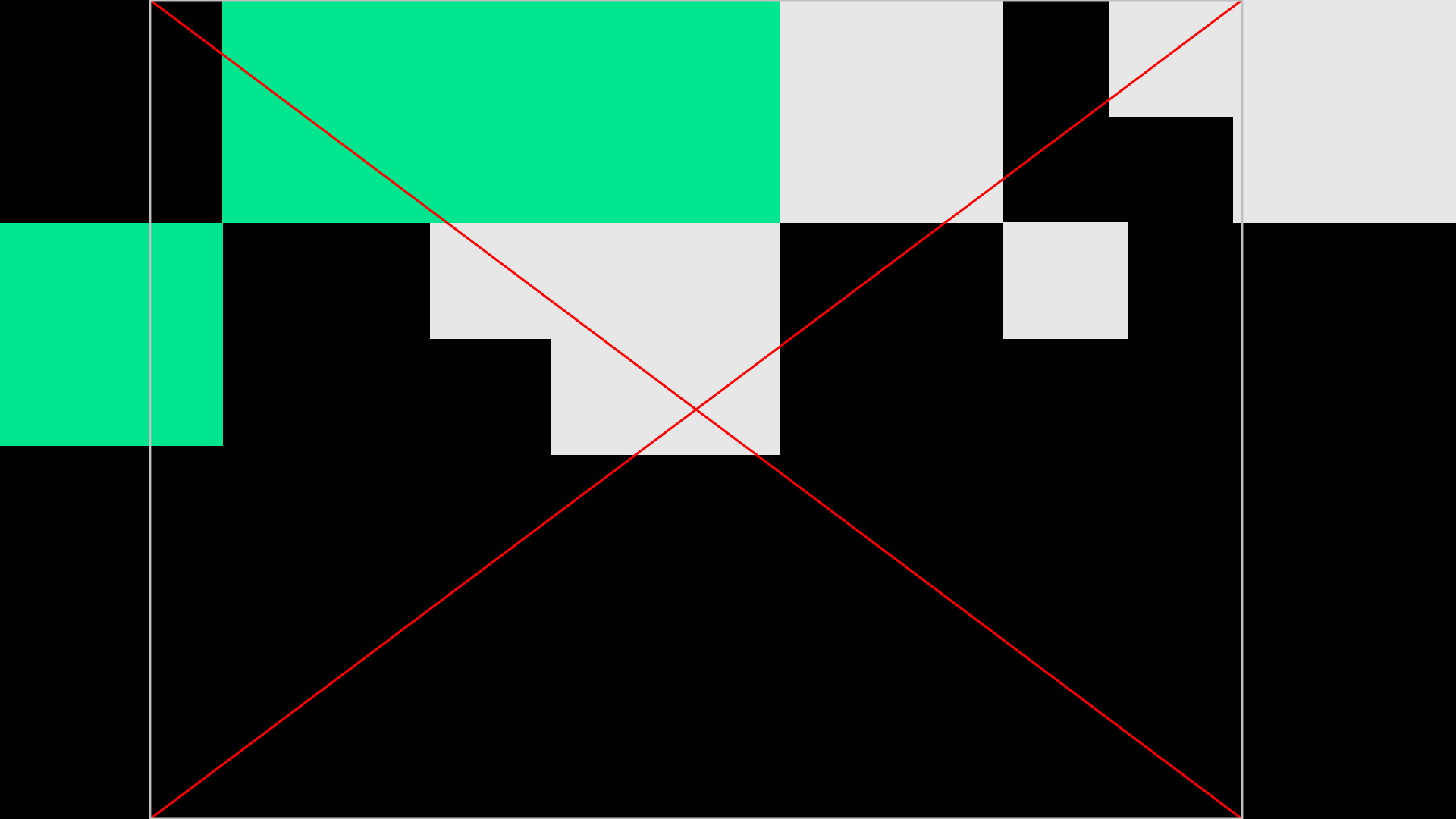
Log in to your Spotify account to authorize the app to get access to your account info.

Step #2

Customize the parameters as you wish to change your dashboard.

Step #3

Run all the cell blocks to see your own Music Profile and Recommendation list that you built!



Challenges

Spotify deprecated some critical endpoints

Problem

- Could not use endpoints to build the recommendation system , find related artists and similar songs based on audio features

Workaround

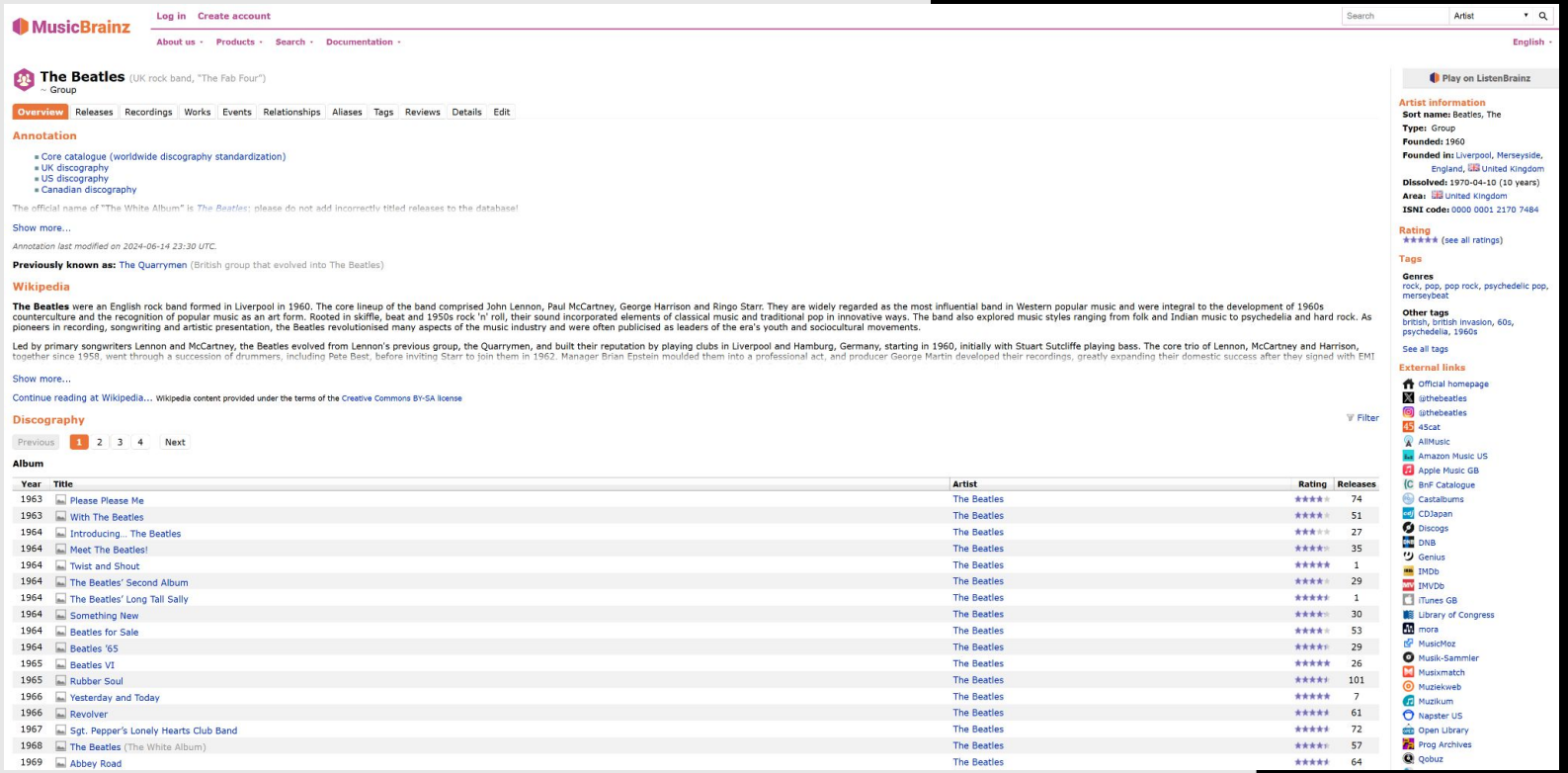
- Implement our own logic by using sp.search( ) method to look for similar artists in user's top genres
- Use another library 'musicbrainz' to implement the recommendation system

Web API endpoint integration

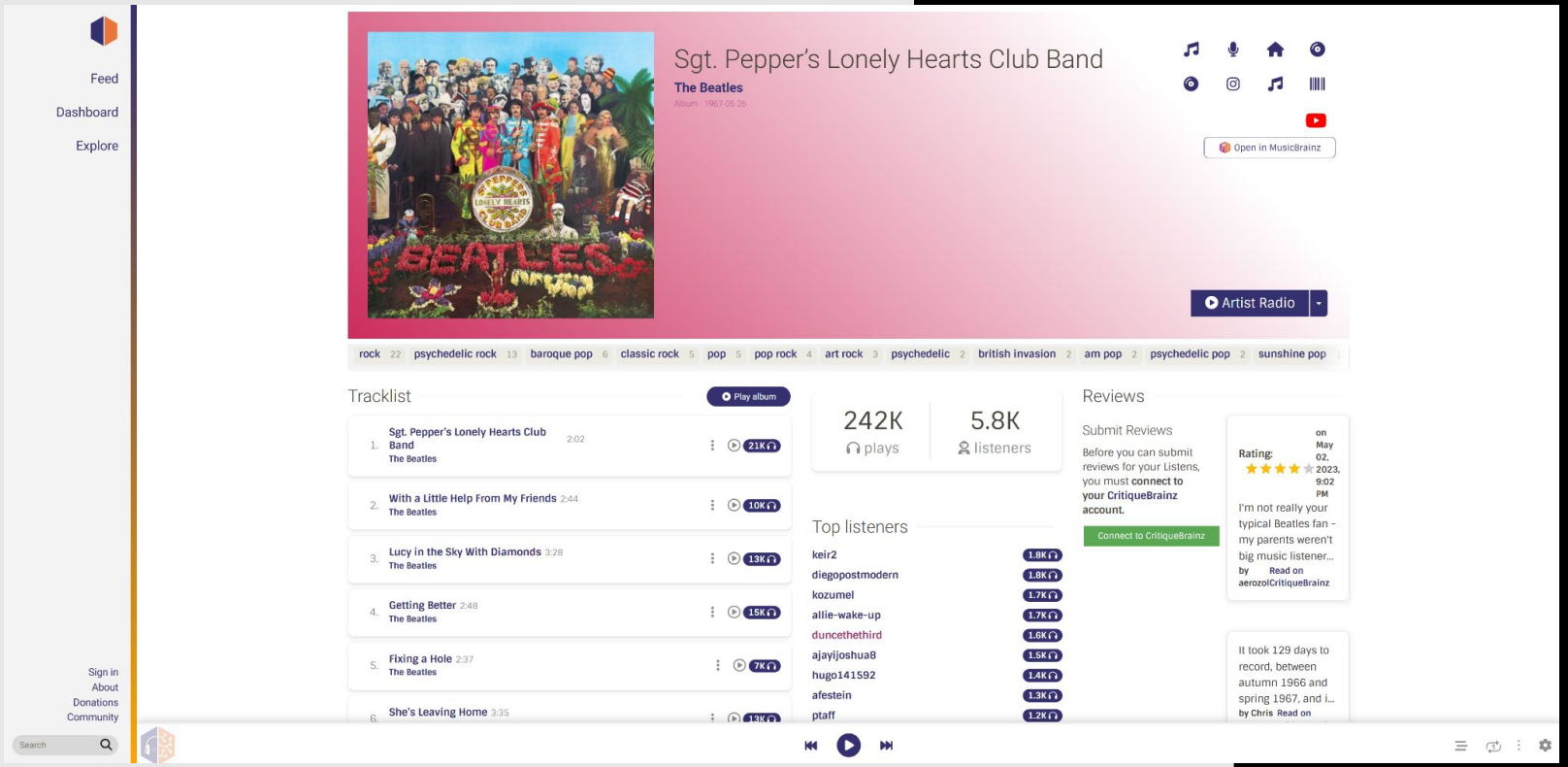
Effective today, ***new*** Web API use cases will no longer be able to access or use the following endpoints and functionality in their third-party applications. Applications with existing extended mode Web API access that were relying on these endpoints remain unaffected by this change.

1. Related Artists
2. Recommendations
3. Audio Features
4. Audio Analysis
5. Get Featured Playlists
6. Get Category's Playlists
7. 30-second preview URLs, in multi-get responses (`SimpleTrack` object)
8. Algorithmic and Spotify-owned editorial playlists

MusicBrainz



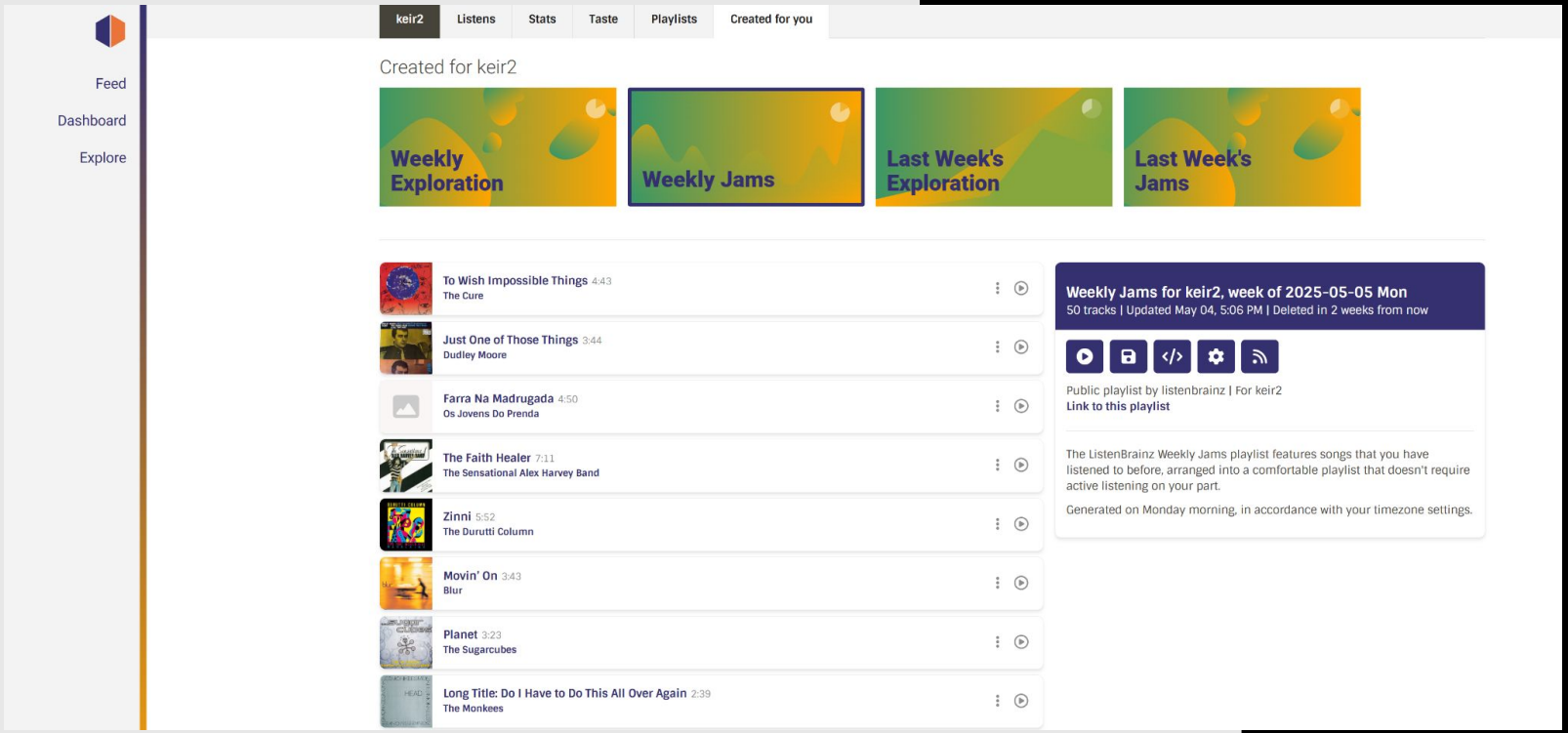
ListenBrainz



PlayWright

```python
from playwright.async_api import async_playwright

# Async function to extract all MusicBrainz recording links from a user's recommendations page
async def get_listenbrainz_recording_links(user):
    async with async_playwright() as p:
        print(user)
        # Launch the browser in headless mode
        browser = await p.chromium.launch(headless=True)
        page = await browser.new_page()

        # Go to the user's recommendations page
        url = f"https://listenbrainz.org/user/{user}/recommendations/"
        await page.goto(url)

        try:
            await page.wait_for_selector('.playlist-items')
            playlist_items = await page.query_selector_all('.playlist-items a[href^="https://musicbrainz.org/recording/"]')

            if playlist_items:
                songs = {await item.get_attribute('href') for item in playlist_items}
            else:
                print(f"No MusicBrainz links found for user {user}.")
                songs = set()

        except Exception as e:
            print(f"Error getting recommendations for user {user}: {e}")
            songs = set()
```

ListenBrainz Recommendations



Spotify Popularity Score

popularity integer

The popularity of the track. The value will be between 0 and 100, with 100 being the most popular.
The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, i
the most part, on the total number of plays the track has had and how recent those plays are.
Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate
tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from trac
popularity. ***Note**: the popularity value may lag actual popularity by a few days: the value is not updated in real time.*

Recommendations

1		Something in the Room She Moves Julia Holter	Something in the Room She Moves
2		Looking Inwardly The Chameleons	What Does Anything Mean? Basically (2009 Rema...
3		Always Time Wharp, Elizabeth Glenn-Copeland, Beverly Glenn-Copeland	TRANSA
4		Fa Ce La The Feelies	Ork Records: New York, New York
5		Eaten Alive E clipping., Jeff Parker, Ted Byrnes	Visions of Bodies Being Burned
6		Saudosismo Caetano Veloso	Caetano Veloso
▶		Evol Adrianne Lenker	Bright Future
8		City Streets Shigeru Suzuki	WHITE HEAT

NEXT STEPS

1. Deploy as a Flask App

Build this dashboard into a live website for better authentication flow and interactivity. This would also allow users to share their profile with friends.

2. Animation

Add hover effects and animations to make the dashboard more interactive and dynamic.

3. Integrate Recommendation System

Integrate the recommendation system into the app for a smoother user experience.

THANK YOU!