Student name: Christopher Lee

Student ID: 52105866

Student email: [c.lee1.21@abdn.ac.uk](mailto:c.lee1.21@abdn.ac.uk)

Git repository URL: <https://github.com/topherlee/music-db-webapp/> (private)

Total Git commits: 44 commits

Git log file attached: git-log.txt

WebApp type: Music database

Heroku deployment URL: <https://lollingstone.herokuapp.com>

Dataset used:

Taken from Million Song Dataset.

<http://millionsongdataset.com/sites/default/files/AdditionalFiles/unique_artists.txt>

<http://millionsongdataset.com/sites/default/files/AdditionalFiles/tracks_per_year.txt>

Database filename: artist\_tracklist.db

Number of tables: 2 (artists and tracklists. Both tables are linked by the artist\_id column)

Total artists in database: 26970 artists

Total tracks in database: 476352 tracks

Number of pages: 5 (Homepage, Artists list, Tracks list, Search page, Artist’s Details page)

Templates: artist\_details.html, artists.html, home.html, search.html, tracks.html, and layout.html (all HTML files extends to layout.html)

Testing used: Selenium, Behave (4 test scenarios: home.py, artists.py, tracks.py, search.py)

Why & how app developed:

The app is developed as a way for users to explore the database and discover new artist and music titles that spans from the 20s up until the year 2010. The app is developed in Python with Flask using SQLite3 database. Testing was done by using Behave and Selenium.

Homepage: Graphical user interface, text, application, email

Description automatically generated

I’m feeling lucky button will redirect user into a random artist’s details page. Meanwhile the

Tracks button will redirect user to a YouTube results page for a random track from the database. In the middle of the homepage, statistics regarding the average tracks per artist and the total count of artists and tracks recorded in the database are shown dynamically by querying the totals from the SQLite database.

Artists page:

Table

Description automatically generated

In the artists page, the entire artists’ names from the database are listed and the names are hyperlinked onto their own artist’s details page. Additionally, the MB links will redirect the user onto the respective artist’s MusicBrainz page. The third column of the table shows each artist’s count of tracks that are available in the database.

Example of Aerosmith’s details page:

Graphical user interface, text, application, email

Description automatically generated

In each artist’s details page, users can view a list of the artist’s tracks that are available in the database. Clicking on the title of the tracks will redirect the user onto the YouTube search result for the artist track’s title. Below the artist’s name, a link to their MusicBrainz page and statistics of the artist are also shown.

Tracks page:

In the tracks page, users can view the tracks that are available in the database by filtering the year they were released and the starting letter of the track title. This was done so that the browser would not load the whole tracklist data onto the page.

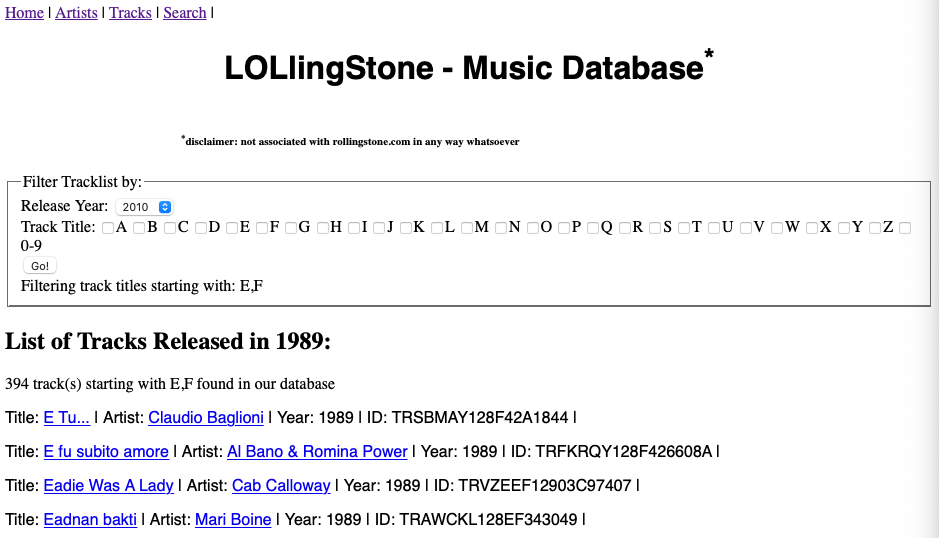


Figure Tracks page showing list of tracks released in 1989 starting with letters E and F

Users can view tracks released in a particular year by selecting the year from the dropdown list and filter the track titles shown by choosing a single or multiple letters. Clicking on the title will redirect the user onto the YouTube search result page for the respective track and clicking on the artist’s name will redirect user onto the artist’s details page.

Search page:

Graphical user interface, text, application, email

Description automatically generated

Finally, users can also search for an artist or a track title by entering the search query in the search field and to filter the search result by choosing the radio button below the search field.