Author

Ashwin Narayanan S
21f3001600
21f3001600@ds.study.iitm.ac.in

About me

B Tech CSE 3rd year student, Amrita Vishwa Vidyapeetham, Coimbatore.
Proud to also be part of my dream college IIT Madras.
More about me: https://ashrockzzz2003.github.io/portfolio/

Description

Create a dynamic music app where creators craft diverse songs and albums while users enjoy seamless streaming and playlist creation. Admin ensures policy compliance, flagging any violations. An appealing yet intuitive UI is crucial for an enjoyable user experience, enhancing accessibility and engagement. This is my understanding on what needs to be done!

Technologies Used

Architecture: Media View Controller (MVC)
FrontEnd: HTML, CSS, JavaScript, jinja2
BackEnd: Flask, Crypto for cryptographic security (AES algorithm) and token Based Authentication
Database Management: SQLite

Why didn't I use bootstrap?
Bootstrap is a third-party framework, but I wanted to learn more about CSS styling and master the art, as I really loved the way it works. Tried to create custom design components.

Why Token based authentication?

- Sending userEmail and userPassword every single time in the request parameters is dangerous and can be hacked into easily as it'll be there in our browser history.
- Whereas with Bearer tokens, once we send userEmail and userPassword it'll be generated and we can validate with ease with expiry date and exactly know the user's role as that will be encrypted and be a part of the token.

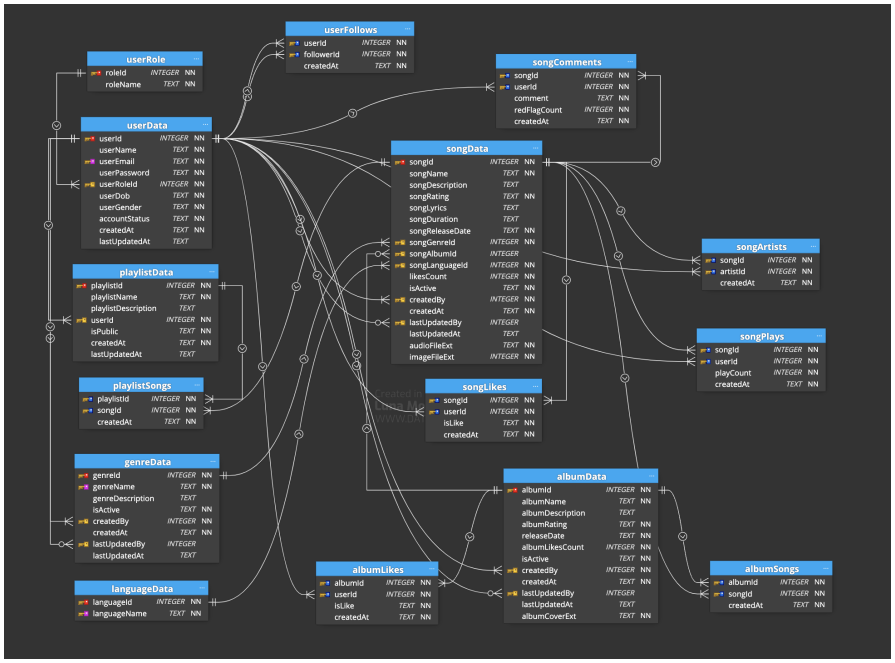
Why SQLite?

- The sole reason being it is part of the rubrics.
- From what I researched and found out, using MySQL is far more optimal than SQLite as it's not a file based database and concurrency control can be established very easily.

Encryption and Decryption algorithm Choice

- I could've easily taken RSA but then, RSA is a fixed size based cipher which will only work when we have 16 byte data. Anything more, it'll crash.
- Our use case needs a more dynamic and at the same time, a secure algorithm. The apt choice seemed to be AES after several case studies.
- Algorithms like Shift Ciphers and Substitution Ciphers seemed very easy to break with limited computing. But this will take 100 years or so to break!

DB Schema Design



*The purpose of **isActive** attribute in almost all tables*
Considering the structure of the database, it is not a good idea to remove rows from tables that have foreign key constraints. Hence, the ***isActive*** attribute will be able to virtually delete the data. The same attribute is also used to flag songs as inappropriate by the admin. They will not be shown to the user!

Tracking who made what changes in the database was also important, therefore engineered 4 attributes, ***createdBy*** and ***createdAt***, ***lastupdatedBy*** and ***lastUpdatedAt.***

API Design

Implemented API's for ***userLogin, getAllSongs, getSongById, getAllPlaylists, getPlaylistById***

- Accepts JSON body requests, returns JSON body responses.

Architecture and Features

Project Organization *(Inside the code folder)*
logs *(errorLogs),* ***middleware*** *(tokenGeneration and tokenValidation),* ***schema*** *(databaseFile, reinitialiseScript),* ***static*** *(css, js, file uploads),* ***templates*** *(role-based HTML files),* ***app.py*** *(main App),* ***requirements.txt***

Features Implemented
Authentication: *register as user, admin login, user login*
User: *stream songs, albums, view song lyrics, like/dislike songs, CRU on playlists, Public/Private playlists, (rating), register as creator, search songs by name, language, genre*
Creator: *stream creator's songs, view lyrics, see likes/dislikes, upload songs, CRU on albums, CRUD on songs, CRU on genres, CRU on languages, posters of songs, albums and mp3 files can be uploaded.*
Admin: *stream songs, albums, view song lyrics, see likes/dislikes, listen to public/private playlists, search songs by name, language, genre, CRU on genres, CRU on languages, flag songs as inappropriate, overall count of songs, albums, posters, users, creators, genres*

Video: https://drive.google.com/file/d/1FqHtkpy-4MdMyAfW_4VYtiQyA-NQaWV7/view?usp=sharing