Author

- Name: Anuj Gupta
- Roll Number: 21f3001598
- Student email: 21f3001598@ds.study.iitm.ac.in

Description

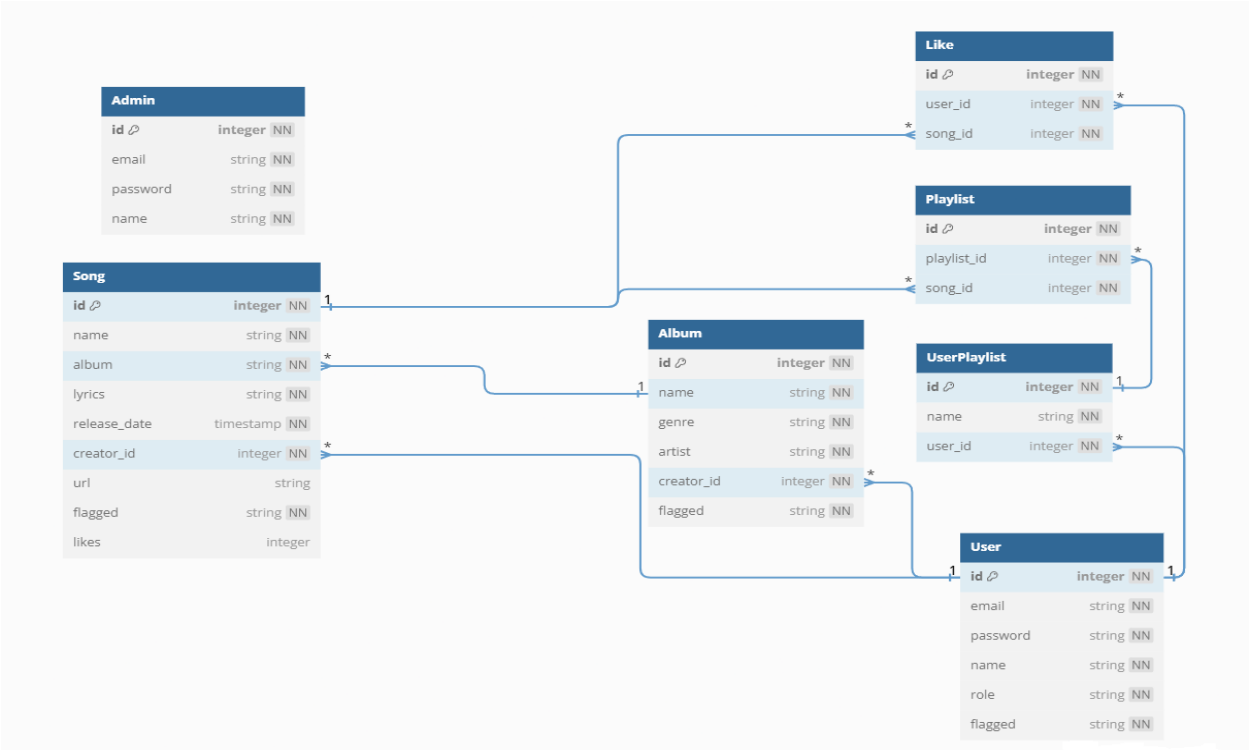
- The problem statement for this project was to develop a web application, ‘Music Streaming App’. It is a multi-user app (one required admin and other users or creators) enabling a user to login/signup with the application and streaming music/reading lyrics. Users can see all the existing songs/albums and can perform CRUD operations on the playlists. Creators can perform CRUD operations on the albums/songs/lyrics.

Technologies used

- The technologies used in this project are: − Python libraries, HTML/CSS, Jinja, Bootstrap, Flask, Flask-SQLAlchemy, and SQLite.
- Python is the core programming language used.
- Flask is the main framework used for the web app.
- Flask-SQLAlchemy is the SQL toolkit used to connect with the database file.
- SQLite is the database used for the modeling of the application.

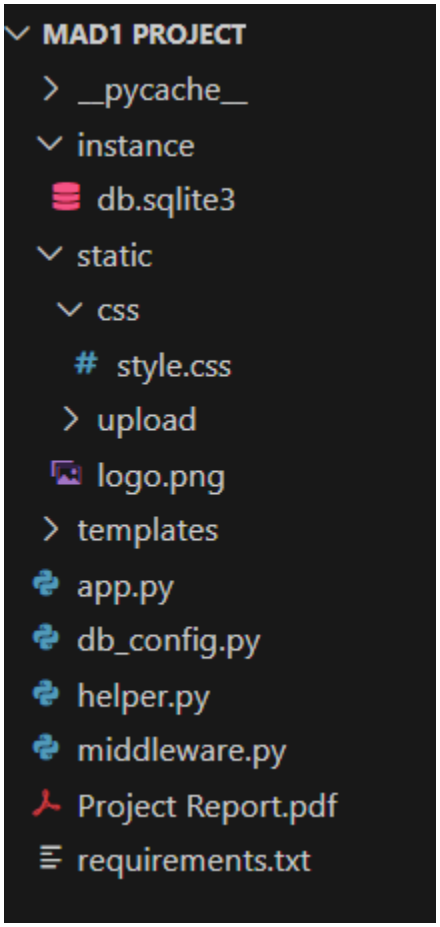
DB Schema Design

- Given below is the detailed description of the DB schema along with the ERD



Architecture

- Shown below is the layout of the web app architecture.



Project Structure:

1. Application folder contains:
    a. db_config.py where database initialization has been done.
    b. app.py with all controllers.
    c. middleware.py with decorators to prevent unauthorized access.
    d. helper.py with helper functions.
2. All HTML files are present in the templates folder.
3. Static folder contains:
    a. css folder which contains css files.
    b. upload folder where songs files(.mp3) are stored.
    c. logo.png file which is our application logo.
4. Instance folder contains database file db.sqlite3.
5. Main application is running in app.py.
6. Project Report.pdf and requirements.txt also included.

Features

- Hashed password to enhance security.
- User Authentication: Users can register and log in using their credentials.
- User Dashboard: Logged-in users have access to a dashboard where they can view all the existing & trending songs/albums. Can also search for song(s)/album(s) by name, genre, artist etc.
- Creator Dashboard: Creator can see analytics of their song(s)/album(s) and can manage them with CRUD functionalities.
- Admin Dashboard: Admin can flag/unflag/delete, see the analytics & count of song(s)/album(s)/user(s)/creator(s).

Video Link: MAD1 Project Video