Author

Name:Suvarna K
Roll no: 21f1003205
Mail: 21f1003205@ds.study.iitm.ac.in

About me: I,Suvarna K, am a student of IIT Madras online degree.This project is part of the Modern App Development-2 course.

Description

A Music streaming app where users enjoy personalized playlists, rate songs, and view lyrics, while creators upload, edit, and manage their tracks. Admins ensure content quality and policy adherence through user statistics monitoring and the ability to remove inappropriate content.

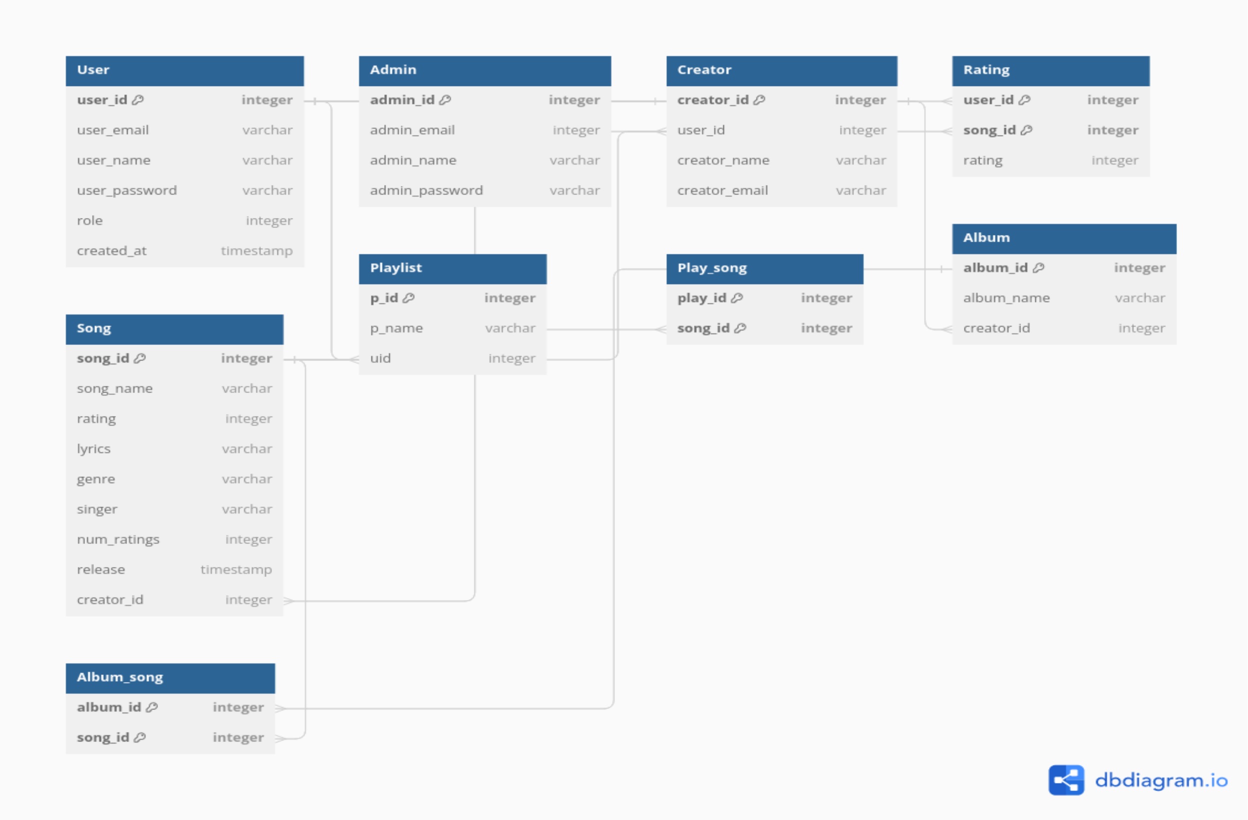
Technologies used

Flask – To maintain the server
Bootstrap – CSS framework to design attractive pages
SQLite – To store and maintain data
Vuejs – Javascript library to make the pages more dynamic and fast
SQLAlchemy – To communicate with the SQLite database
Flask-Security – To secure the flask App
Redis – In memory database to store and retrieve data
Celery – To run tasks and asynchronous jobs
Flask-SQLAlchemy – For Flask server to interact with the database

API Design

Api is used in this app for better workflow. Api is implemented in three classes named API, API2 and API3, They can be accessed by URL root url'/api/user' , root url'/api/song' and root url'/api/admin' respectively.
API (for user)
API2 (for songs)
API3 (for admin)

DB Schema Design



User	
user_id	integer
user_email	varchar
user_name	varchar
user_password	varchar
role	integer
created_at	timestamp

Admin	
admin_id	integer
admin_email	integer
admin_name	varchar
admin_password	varchar

Creator	
creator_id	integer
user_id	integer
creator_name	varchar
creator_email	varchar

Rating	
user_id	integer
song_id	integer
rating	integer

Playlist	
p_id	integer
p_name	varchar
uid	integer

Play_song	
play_id	integer
song_id	integer

Album	
album_id	integer
album_name	varchar
creator_id	integer

Song	
song_id	integer
song_name	varchar
rating	integer
lyrics	varchar
genre	varchar
singer	varchar
num_ratings	integer
release	timestamp
creator_id	integer

Album_song	
album_id	integer
song_id	integer

Architecture and Features

The project is organized in the folder named 21f1003205. It contains 4 directories and a few files. The App could be run from main.py in the root directory. This file configures the flask app created. Other than this, the root directory contains a local_run.sh and local_setup.sh which tells how to run locally and install the requirements.Vuejs library is used to make the pages more dynamic and efficient.The application offers a comprehensive set of features, including an administrative dashboard for overseeing the system, user profiles for both general users and creators, and a specialized set of tools for managing songs and playlists.Authentication is provided through distinct forms for general users, creators, and administrators, with role-based access control ensuring proper authorization. Users can explore existing songs and albums, read lyrics, and create playlists with the flexibility to rate songs. Creators, in addition to these functionalities, can create, edit, and remove songs, contributing to the dynamic content of the platform.The admin dashboard provides valuable insights, including overall app statistics, monitoring song performance, and the ability to manage content by removing items that violate company policies.The search functionality is robust, allowing users to find songs and artists based on various criteria such as genre, singer, and rating. Overall, the application ensures a rich and interactive experience for both general users and creators, with a focus on content management, user engagement, and system monitoring.The server has implemented the feature to inform the user via mail everyday evening if the user didn’t post anything that day. And to send a mail on the first day of every month with a report. Both these features are done with celery crontabs.

Video

mad-2.mp4