MAD Project Report *Building a Music Streaming App Tune Library.*

Name - Aditya R Kapadia**, Roll number -** 21f2001090

Email - 21f2001090@student.onlinedegree.iitm.ac.in

Author

I am Aditya Kapadia, currently pursuing a B.S. in Data Science at the Indian Institute of Technology, Madras (IIT-M). My academic journey includes the successful completion of a Data Science diploma, and I am poised to attain my Programming diploma. With a strong foundation in data science, I have also developed a sentiment analysis model. As I advance in my academic pursuits, I am actively diversifying my skill set, with a primary focus on achieving mastery in web application development. My recent project is building a music streaming web app which involved database modeling, feature development for various user profiles, and route management, contributing significantly to my growth and learning.

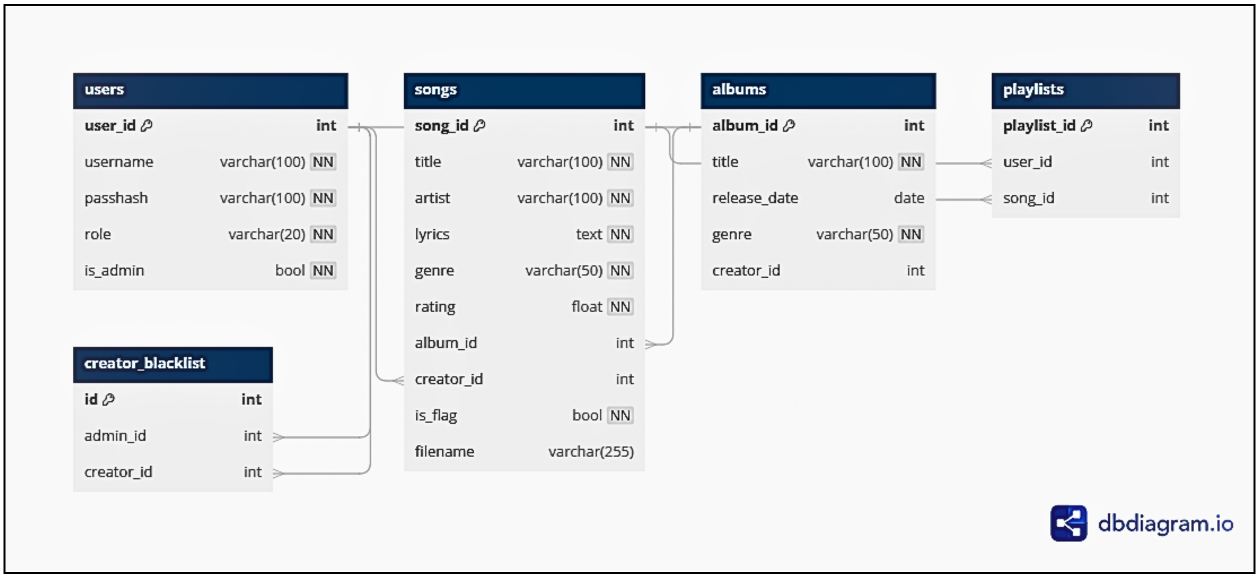
Description

I have developed a music streaming app called "Tune Library." Tune Library allows multiple users to explore, listen to, and make their music collections. Users can add songs to their playlists to list their favorite songs, rate songs by like/unlike button, and discover new music. Tune Library also offers a platform for music creators to upload their work, allowing them to create new albums and songs for their audience. Additionally, an admin oversees the app, monitoring trends and having the ability to flag songs and blacklist creators when necessary.

Technologies Used

- Flask for application code and Jinja2 templates + Bootstrap for HTML generation and CSS for styling and layout.
- SQLite for data storage and Flask-SQLAlchemy for creating and manipulating database models.
- JavaScript for data visualization through Chart.js.

Database Schema Design



users		
user_id	int	
username	varchar(100)	NN
passhash	varchar(100)	NN
role	varchar(20)	NN
is_admin	bool	NN

songs		
song_id	int	
title	varchar(100)	NN
artist	varchar(100)	NN
lyrics	text	NN
genre	varchar(50)	NN
rating	float	NN
album_id	int	
creator_id	int	
is_flag	bool	NN
filename	varchar(255)	

albums		
album_id	int	
title	varchar(100)	NN
release_date	date	
genre	varchar(50)	NN
creator_id	int	

playlists	
playlist_id	int
user_id	int
song_id	int

creator_blacklist	
id	int
admin_id	int
creator_id	int

Architecture

1. **Backend -** Flask is utilized as the backend framework to handle routing and requests. Custom authentication using decorators is implemented. These decorators serve as gatekeepers for specific routes, guaranteeing that only authenticated and authorized users can access them
2. **Frontend** - Jinja2 is used to embed data from the backend into your HTML render dynamic pages, Bootstrap for responsive design/pre-built UI components, and CSS for styling. Chart.js is also used to generate charts to monitor trends related to music.
3. **Database -** SQLite is used as the database for storing information related to users, songs, albums, etc and Flask-SQLAlchemy is used for interacting with the database and defining database models/relations.

Video Link:- link