MAD – 1 PROJECT

Author

E DEVANANADA
21f3002612
21f3002612@ds.study.iitm.ac.in
I am currently pursuing Integrated MSc course in Pondicherry University along with the online degree course by IITM

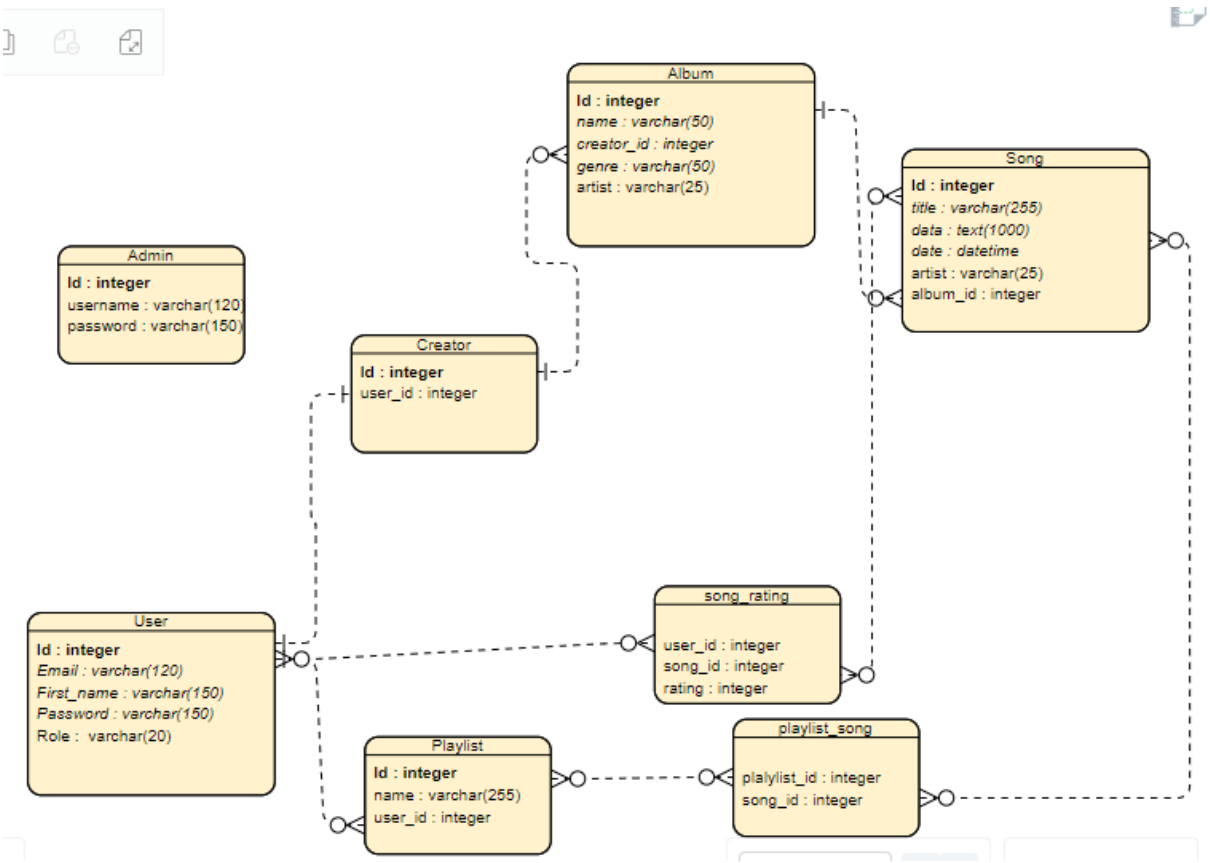
Description

I wanted to develop a web-application that focuses on the lyrics of the songs. Most of the music apps that are available today, doesn't provide the users in the sense proper lyrics. I used python leveraging and flask framework for light weight and modular design and SQL Alchemy fo r database

Technologies used

Flask- framework for python to develop the app
Flask-login – for user verification and authorization along with managing user session
SQL Alchemy – For database management and facilitating interaction with database
werkzeug.security: Part of the Werkzeug library, it provides utilities for password hashing and verification, enhancing security in web applications.

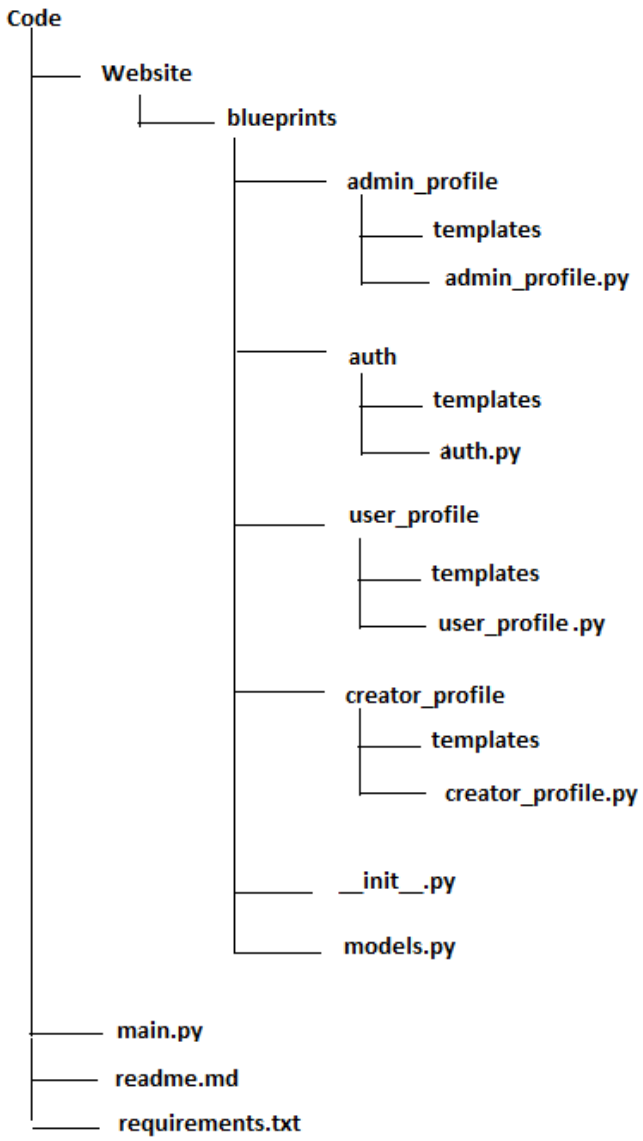
DB Schema Design



MAD – 1 PROJECT

The web application features key models: Song, Album, User, Admin, Creator, and Playlist. These models establish relationships, with the exception of Admin. Users and Creators are differentiated, and UserMixin simplifies user access. Song and User models have a rating relationship, and a Song will belong to an Album. Albums and Songs have a one-to-many relation. Playlists use a many-to-many relation with Songs. Users can transition to Creators, altering their role. The application has four routes (authentication, user, creator, admin) with corresponding templates. Flask-Login manages user sessions, requiring login for access. The organized structure aids development clarity.

Architecture and Features



Here I have used Blueprints for modularity (admin, user, authentication, creator). Templates are organized within their respective blueprint folders. Controllers, managing functions, are spread across main.py and route folders in the website directory. This approach helps code organization and development efficiency
The app focuses around song lyrics, offering a versatile search for music, artists, and albums.
Main attraction is that users can explore songs based on specific words or themes, access popular and recent releases, and discover genres and playlists. Creating, adding, and deleting playlists is seamless, allowing users to rate lyrics based on their enjoyment. Users can also contribute as creators, uploading albums and songs with a personalized dashboard reflecting their impact. Admins have full control, managing content, ensuring policy adherence, and conducting searches by artist names, rating values, and admin-specific criteria. The login page directs users to their respective roles—admin or user—providing a secure and tailored experience.

Video

https://drive.google.com/file/d/11HgDLIU7II0Un5L5T85Z6jpFK0Xszjrl/view?usp=sharing