Report
Modern Application Development 1 (MAD1) - Project

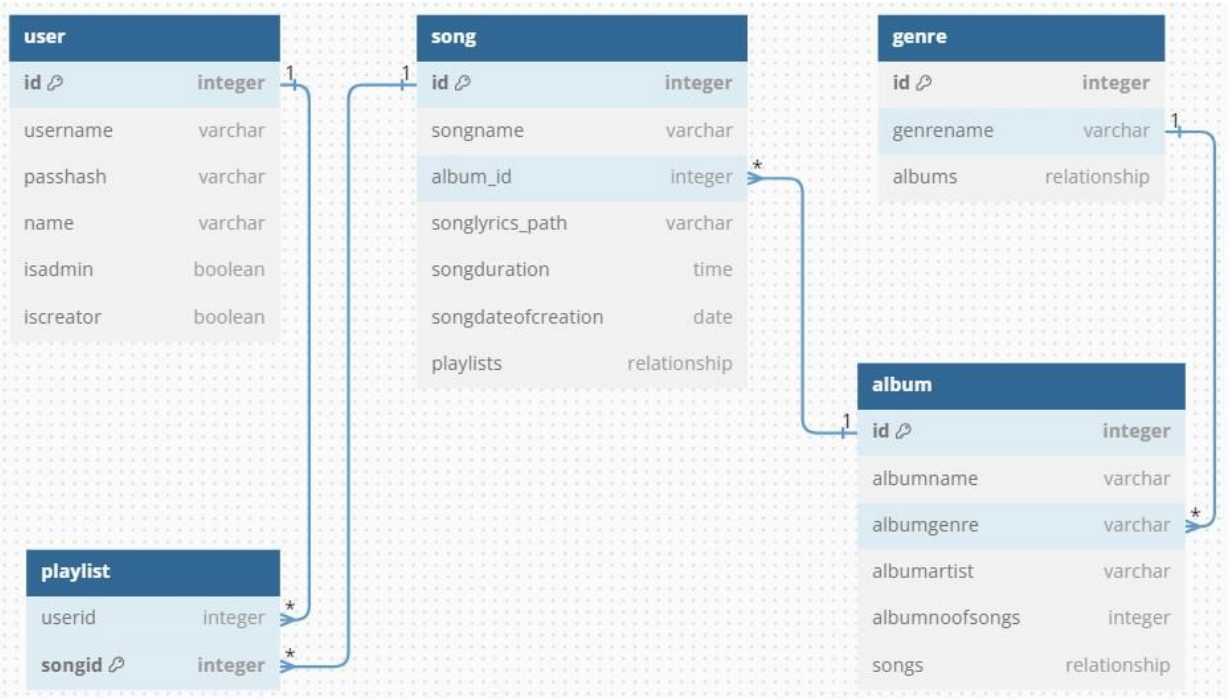
September 2023 Term - Music Streaming Application

Author
Name : MADHAVAN R MOHAN
Roll Number : 22F3000983
Student Email ID : 22f3000983@ds.study.iitm.ac.in
I am doing my undergraduation in Mathematics at Presidency College, Chennai.

Description
In this project, we have to create a music streaming application. There must be an admin managing the app data. There should be users who can play songs, read lyrics, create playlists and creators who can upload their own songs.

Technologies Used
In this app, technologies like Flask, Flask-SQLAlchemy, Werkzeug, Jinja2, SQLite, HTML/CSS, Bootstrap, Python in-built modules have been used. These technologies are used to develop a web application for users to manage and interact with a music database. Flask framework is employed for web development. Extensions and libraries enhance functionalities, including database integration, template rendering, and user authentication. This tech stack enables the creation of a dynamic and feature-rich web app.

DB Scheme Design
These design choices aim for normalisation, data integrity, and flexibility. ForeignKey relationships enforce referential integrity, and the use of primary and foreign keys helps establish relationships between tables. The nullable fields allow for optional data, while default values provide sensible defaults when applicable.



API Design

In this app, there are routes that can be considered as endpoints for an API. There are APIs for GET and POST methods for User-related, Admin-related, Profile-related, Login and Registration, Genre-related, Album-related, Song-related, Playlist-related, Creator-related routes. These routes represent the various operations that can be performed through the provided application.

Architecture and Features

The Flask application is organised following the Model-View-Controller (MVC) pattern. The models are defined in the ‘app.py’ file using SQLAlchemy, representing models such as User, Song, Album, Genre, and Playlist. The controllers, which handle the application's routes and logic, are also implemented in ‘app.py’. Each route function corresponds to a specific page or action, such as the home page, admin page, profile page, login page, registration page, and various CRUD operations for albums and songs. The templates, representing the views, are stored in the "templates" folder. The static files, such as CSS and images, are stored in the "static" folder. The project uses the Flask web framework, SQLAlchemy for database interactions, and Jinja2 for template rendering. In terms of features, the application includes user authentication, role-based access control (admin and user roles), CRUD operations for albums and songs, user profile management, song playlist functionality, and the ability for users to become creators and upload songs. Additional features like searching for albums, viewing and editing genres, and viewing user playlists are also implemented. The application provides feedback to users through flash messages for actions such as successful registrations, logins, and updates, as well as error messages for invalid inputs. The organisation of the code and the use of decorators such as ‘auth_required’ and ‘admin_required’ help maintain a structured and secure codebase.

Video

22F3000983_MAD1PROJ.mp4

https://drive.google.com/file/d/1FGUXvyXFk_moAvSzXOY6AIpFdjqWCukr/view?usp=drivesdk