MUSIC STREAMING APPLICATION

MAD - 1

Author

Name: Rahul Sharma
Roll Number: 23f1001879
Email-id: 23f1001879@ds.study.iitm.ac.in
Currently, I am a full time student pursuing this degree.

Project Description

This project is based upon making a music streaming app where users can register as a user and a creator. Users can interact with listening to songs, making playlists, rating songs and make themselves as a creator. Creators can use all functionality of a normal user and they can also upload songs and check their dashboard and delete their recent songs. Admins are kind of backend users which can view various statistics for the app. They can also read and delete tracks.

Technologies Used

- Flask - for application used
- Jinja2 templates,Bootstrap and HTML/CSS for styling
- SQLite and SQLAchemy for Database work

Architectures and Features

- My project is organised using the MVC architecture(Model-View-Controller), with the controllers handling all the logic and routing, templates for display and model for database interaction.
- New users can register themselves as a normal user or creator and Existing users can login and enjoy.
- After logging into the application, users can listen to songs and rate them, make their own playlists and check genres and if the user wants to register as a creator, they can do it and start uploading songs and check their dashboard
- *Admin Login* The admin registration can be done through the database AS WELL AS admin can register MANUALLY too. But it is better to use database pathway. For admin features, they can view different statistics, view songs and can delete the songs if they want.

DataBase Schema Design

The database has been designed to store the information about users, songs, playlist and admin.

__Tables_

User: user_id, username, email, password

Admin: admin_id, adminname, email, password

Song: id, track, creator, lyrics, song_url, playlist_id

Playlist: id, name

Major Challenges Faced

The major challenge I faced during making this project was making the song api, search api and the graph api. It required a lot of concentration to integrate with other templates and show in good way.

Future Improvements

This project can be improved by adding functionalities for editing playlists, genres and api for admin and creator dashboard statistics. Additionally, the UI can be furnished much more than this if required.

Conclusion

In conclusion, the project was a success in creating a platform for user to listen to songs and make playlists and much more. Use of Flask, jinja2 and SQLAlchemy made the development much more effective.

Project Demonstration Links:

-Youtube link: https://youtu.be/YVXzTCGwdiQ