



KalpRaag

by Saransh Saini [22F1001123]

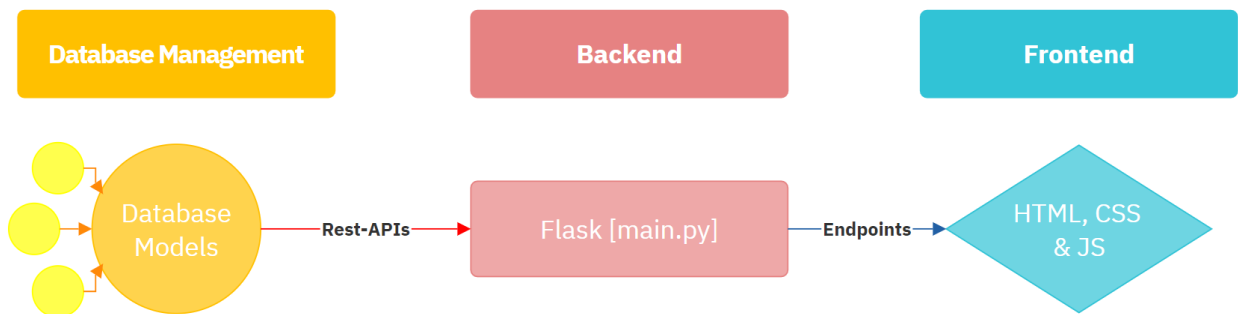
Introduction

A multi-user music streaming platform for lovers of music, passionate creators, and moderated by admins.

Technologies Used



System Architecture



The **main.py** file serves as the nexus of the application, directing vital CRUD operations through the employed REST APIs. The orchestrated integration of app routes ensures the rendering of HTML templates at specific endpoints, embodying a deliberate code separation strategy. This approach is akin to the synergy observed in frameworks, fostering better comprehensibility and operational efficiency within the application.

Database Management

The database is powered by SQLite, serving as the foundational software system. The models, crafted through **Flask-SQLAlchemy**, are meticulously designed with appropriate constraints, ensuring the steadfast integrity and consistency of the stored data. Provided below is a succinct code snippet illustrating the structure of the Creator model.

```
class Creator(db.Model):
    creator_id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    user_name = db.Column(db.Integer, unique=True, nullable=False)
    song_ids = db.Column(db.String, nullable=False)
    album_ids = db.Column(db.String, nullable=False)
```

API Management

The database's CRUD operations are exclusively executed through APIs, leveraging Flask-Restful to implement essential functionalities such as **GET**, **POST**, **PUT**, and **DELETE** operations at specific endpoints. Flask-Restful facilitates the seamless integration of these operations, allowing for efficient communication between the application and the database. This approach ensures a robust and standardized mechanism for interacting with the database, promoting scalability and maintainability in the overall system.

Multi-User Functionalities and Features

User

- Read lyrics
- Rate and like songs
- Create & Delete Playlists
- Save albums
- Register for Creator access

Creator

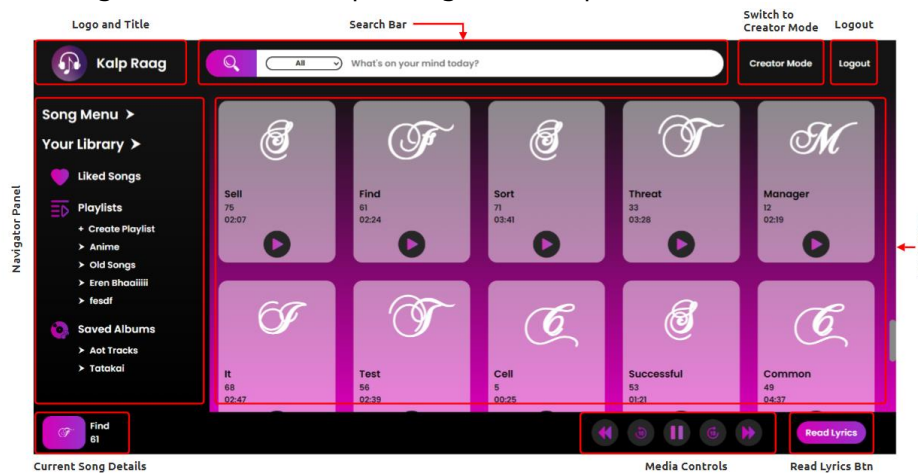
- All User access
- View statistics
- Add/Edit/Delete songs
- Add/Edit/Delete albums

Admin

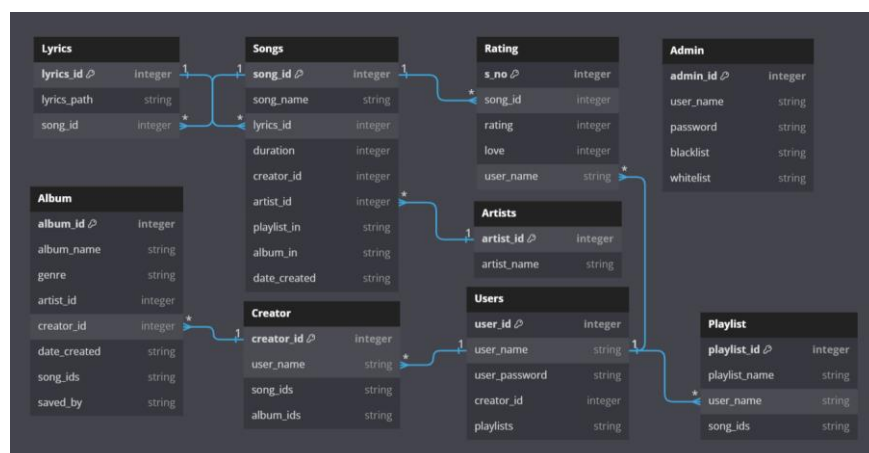
- View all statistics
- Delete/Flag Songs
- Delete/Flag Albums
- Ban Users temp./perm.
- Blacklist/whitelist Creators

User Interface

The web application presents an intuitive User Interface designed to accommodate the distinct requirements of three user categories: Users, Creators, and Admins. Striving for a harmonious User Experience (UX), the design prioritizes a relaxing ambiance for users, fostering a positive and enjoyable interaction with the platform. This approach aims to enhance engagement and satisfaction across all user roles, contributing to a seamless and pleasing overall experience.



Database Schema



Conclusion and Extra Info

Overall, I had a lot of fun while creating this project, and also, I learned a lot.

- Only one song is loaded in the app, I added it to debug the app.
- **How to run the app?:** In the terminal type `[python -m app.main]` this runs the app as a module.

- **Presentation Video Link:**

<https://drive.google.com/file/d/1o0sdUVMwG27UjXLHg784Qswcl1zV3U4b/view?usp=sharing>