# Author

**Harsh Kumar**
22f3002198
22f3002198@ds.study.iitm.ac.in
2022 batch BS Data Science & Application student, pursuing as a standalone degree.

# Description

**Heal** , a dynamic multi-user music streaming app, redefines the digital music experience. Users enjoy seamless streaming, lyric reading, and personalized playlists. Creators can register, craft albums, and upload songs, fostering a collaborative musical community. The admin panel provides insights into user and song performance metrics. Boasting a secure Flask Login system, Heal ensures data integrity. With a robust search feature, users effortlessly discover albums, artists, and songs. Heal stands out as a comprehensive and innovative platform, bridging the gap between music enthusiasts and aspiring artists.

# Technology used

- **Flask**: for basic backend Implementation.
- **Flask_sqlalchemy**: for implementing Database.
- **datetime**: for storing the date for the deadline.
- Some inbuilt libraries like jinja2, render_template, redirect, and url_for displaying HTML content.
- **flash**: to show an alert
- **Flask_Login**: for implementing the login functionality
- **werkzeug.security**: for hashing the password
- **Flask-Restful**: to create Api

# API USED

There are three API:

**1. UserAPI** -

- It is linked to two endpoints:
  - /api/user – With this endpoint, we can get all users in database, creating user.
  - /api/user/<int:id> – With this endpoint, we can get user with input id, updating user with respective id, deleting user with respective id

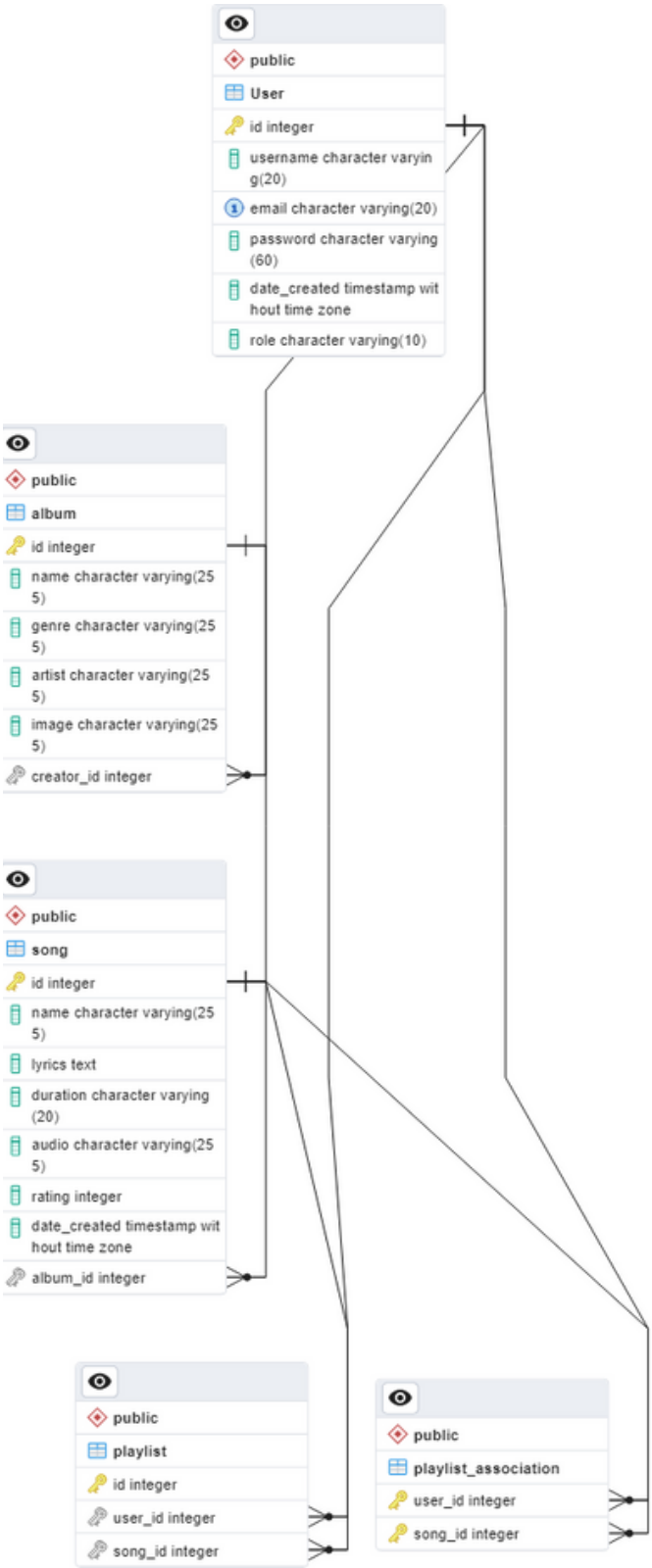
**2. AlbumAPI**-

- It is linked to three endpoints:
  - /api/album/<int:album_id> – With this endpoint, we can get all album with respective id from database.
  - /api/album/<int:user_id>/album/<int:album_id> – With this endpoint, we can get album with respective id under respective user, delete album with respective id under respective user, update album with respective id under respective user.
  - /api/album/<int:user_id>/album – With this endpoint, we can create album under respective user

**3. SongAPI**-

- It is linked to three endpoints:
  - /api/album/song/<int:album_id> – With this endpoint, we can get all songs with respective id from database.
  - /api/album/song/<int:album_id>/album/<int:song_id>/song – With this endpoint, we can get songs with respective id under respective album.
  - /api/album/song/<int:song_id>/song – With this endpoint, we can get song under respective id.

# DATABASE SCHEMA DESIGN



# ARCHITECTURE AND FEATURES

• There are 2 controllers
    1. **auth**: It is used for authorization purpose
    2. **view**: It is used for all other purpose like dashboard, profile, creating albums and songs , CRUD, admin dashboard, Login system, user-registration etc.
• There are 2 folders
    1. **static**: It contains some CSS files, images files and audio files.
    2. **templates**: It contains all HTML templates used in Project.

• **login / sign-up system**: Here user can fill in all details for creating a new account and after that user will be able to do login.
• **Profile view with basic stats**: This page has user details, email-id, role, number of albums uploaded(if creator).
• **Admin panel**: Here admin will be able to see and manipulate with the User, creator , Album and songs information. Here, the performance of songs (i.e. - average ratings for particular genre) is also shown.
• **Search**: Here the user will be able to search a album ,song, artist with name which consists of that.
• **Playlist Management**: Here the user will be able to create a playlist and add songs to it.
• **Ratings**: This functionality allows the user to give ratings to the songs.
•**Top Songs**: This functionality shows the songs which has highest ratings.
• **API**: Created three API user, album, song with basic functionality

# VIDEO

**https://drive.google.com/file/d/1FE-4LrIahl8VTOXswGEcg3giAirKd_LM/view?usp=sharing**