Author: Uroosha Rahat
Roll Number. : 21f1002968
Email id: 21f1002968@ds.study.iitm.ac.in

Description:

The purpose of the music streaming app is to design and implement a multi-user music streaming app with lyrics reading, providing a seamless experience for users and creators. The application features an administrative role and multiple user roles, allowing creators to contribute and users to enjoy the content.

Technologies Used:

- Programming Language: Python
- Framework: Flask
- Database: SQLite
- Frontend: Jinja2 templates, CSS for HTML generation and styling

Features implemented:

- Admin: separate form for admin login; manages the overall system; oversees user roles; content additions; total users/creators/playlist and playlists; and can flag or remove songs.
- Admin can flag or remove Creators and Users.
- Creators: Users with the ability to add/edit/remove new songs, albums, and lyrics.
- Users can stream music seamlessly by viewing all available songs/playlists/albums.
- The application supports lyrics reading synchronised with the music playback for an enhanced user experience.
- Users can play songs, read lyrics, and rate them based on their preferences.
- Users can register themselves as creators and fill out a separate form for creators.
- Users can create personalised playlists by adding their favourite songs, offering a customised and enjoyable listening experience.
- The system automatically showcases the latest albums and songs added, creating a dynamic and up-to-date user environment.
- Users can search their songs based on their names, ratings, and playlists.
- CRUD on songs and playlists.

Video Link:

https://drive.google.com/file/d/1dMr88kOgrVhk51Tex8PVjxzkuAEjRMD6/view?usp=drive_link

Tables (4)

Name	Type	Schema
Admin		CREATE TABLE "Admin" (admin_id INTEGER NOT NULL, adminname VARCHAR(20) NOT NULL, password VARCHAR(80) NOT NULL, email VARCHAR(255) NOT NULL, PRIMARY KEY (admin_id), UNIQUE (adminname), UNIQUE (email))
admin_id	INTEGER	"admin_id" INTEGER NOT NULL
adminname	VARCHAR(20)	"adminname" VARCHAR(20) NOT NULL
password	VARCHAR(80)	"password" VARCHAR(80) NOT NULL
email	VARCHAR(255)	"email" VARCHAR(255) NOT NULL
User		CREATE TABLE "User" (user_id INTEGER NOT NULL, username VARCHAR(20) NOT NULL, password VARCHAR(80) NOT NULL, email VARCHAR(255) NOT NULL, PRIMARY KEY (user_id), UNIQUE (username), UNIQUE (email))
user_id	INTEGER	"user_id" INTEGER NOT NULL
username	VARCHAR(20)	"username" VARCHAR(20) NOT NULL
password	VARCHAR(80)	"password" VARCHAR(80) NOT NULL
email	VARCHAR(255)	"email" VARCHAR(255) NOT NULL
playlist		CREATE TABLE "playlist" ("id" INTEGER NOT NULL, "name" VARCHAR(255) NOT NULL, "user_id" INTEGER, FOREIGN KEY("user_id") REFERENCES "User"("user_id"), PRIMARY KEY("id"))
id	INTEGER	"id" INTEGER NOT NULL
name	VARCHAR(255)	"name" VARCHAR(255) NOT NULL
user_id	INTEGER	"user_id" INTEGER
song		CREATE TABLE song (id INTEGER NOT NULL, track VARCHAR(255) NOT NULL, creator VARCHAR(255) NOT NULL, lyrics TEXT, song_url TEXT, playlist_id INTEGER, ratings FLOAT, PRIMARY KEY (id), FOREIGN KEY(playlist_id) REFERENCES "playlist" (id) ON DELETE CASCADE)
id	INTEGER	"id" INTEGER NOT NULL
track	VARCHAR(255)	"track" VARCHAR(255) NOT NULL
creator	VARCHAR(255)	"creator" VARCHAR(255) NOT NULL
lyrics	TEXT	"lyrics" TEXT
song_url	TEXT	"song_url" TEXT
playlist_id	INTEGER	"playlist_id" INTEGER
ratings	FLOAT	"ratings" FLOAT

DB Schema Design