MAD 1 Project Report

Music Streaming Web Application

Author:

Hi! I’m Charvi Maini, from Delhi. I’m a dual degree student, pursuing BSc in Programming and Data Sciences from IIT Madras, and BA Economics (Honors) from Delhi Technological University, Delhi.

Roll no: 22f1001106

Email: 22f1001106@ds.study.iitm.ac.in

Description:

‘GroovyGrid’ is a music streaming web application, where a user can play songs, read song lyrics, create playlists, add songs to playlists, as well as rate songs. It is a multi-user app, wherein a user can also become a creator and upload news songs and albums. There is also one admin who can flag songs as well as albums, remove them, as well as blacklist creators. There is separate interface for user and admin, along with a separate one for creator as well.

Technologies Used:

- Flask: main python framework used backend development
- Flask Login: For user and login management
- Flask – SQLAlchemy: Used for database management
- HTML/CSS, Jinja2, Bootstrap: To display the frontend content and styling
- SQLite: Database used for modelling of the web application

DB Schema Design:

User: User_ID: integer, primary key Username: string, not nullable Creator_ID: Boolean, default = false Password: string, not nullable Is_admin: Boolean, default = false, not nullable Is_flagged: Boolean, nullable, default = false	**Album:** Album_ID: integer, primary key Album_name: string, not nullable Genre: string, nullable Artist: string, nullable Is_flagged: Boolean, default = false, not nullable	**Song:** Song_ID: integer, primary key Song_name: string, not nullable Lyrics: string, not nullable Duration: integer, not nullable Is_flagged: Boolean, not nullable, default = false Date_created: date, nullable User_ID: integer, foreign key(user.user_ID)
Album Songs: ID: integer, primary key Album_ID: integer, foreign key (album.album_ID) Song_ID: integer, foreign key(song.song_ID)	**Playlist:** playlist_ID: integer, primary key playlist_name: string user_ID: integer, foreign key(user.user_ID)	**Playlist Songs:** ID: integer, primary key playlist_ID: integer, foreign key(playlist.playlist_ID) song_ID: integer, foreign key(song.song_ID)
Song Rating: Rating_ID: integer, primary key Song_ID: integer, foreign key(song.song_ID) Song_rating: float, nullable Song_count: integer, nullable	- Intentionally left blank	- Intentionally left blank

Architecture:

The architecture of the folder is as follows:

- App.py: main file with all controllers for user, creator and admin
- Models.py: contains a copy of all the models
- Templates: includes all HTML pages for user, creator and admin
- Static: includes images used and songs uploaded (.mp3 files)
- Instance folder: contains database.sqllite3
- Project report and requirements.txt are also included

Video Link:

https://drive.google.com/file/d/1dfWeHEirVGJdU_AIoUFCVy8Sc7g9X3nh/view?usp=sharing