

MODERN APPLICATION DEVELOPMENT - I

MUSIC STREAMING APP

NAME - MOHIT MAHUR

ROLLNO. - 22f1000344@ds.study.iitm.ac.in

PROJECT NAME – MUSIC STREAMING APP

Project Description:-

Title :- **MelodyMingle**

This Project is a multi-user app(One required admin and other required creators or general Users. The Use of this app is to play MP3 songs and read lyrics of songs. The aim of this app is to revolutionize the way people discover, listen to , and share their favorite music, The Melody Mingle app is built to provide users a seamless music experience.For Making this app i follow the guidelines and rules given in AppDev 1 project. These are some steps that I have followed.

- First step for creating the database schema and tables I used flask - sqlalchemy.
- Created routes using flask and html with the help of bootstrap templates.
- Some basic flask security for login pages,
- All the necessary routes and html pages linked with the flask and database models using flask- sqlalchemy.
- For styling used css styling with the help of bootstrap.

Frameworks:-

- Flask is used for building the application code.
- Jinja2 templates for displaying the dynamic data and bootstrap for styling.
- For storing data SQLite is used.
- Bootstrap - for making navbar and tables , aesthetic styling and also bootstrap cards.
- Separate login form for general users , admin, and creators.

Database:-

- Db models using flask-sqlalchemy and for storing sqlite3.
- Creating all the necessary Models like User, Song, Albums, playlist.
- One - to - many relationships between user and songs.
- Many - to - many relationships between playlist and songs.
- There is a separate table for the relationship between song and playlist.
- For admin and creator making the columns are in user table

System Architecture:-

- This Music streaming project strongly follows the MVC design pattern.
- In the Templates folder all the necessary html files are present .
- In the instance folder the sqlite database file is present.
- All the audio files are present in the templates/static/audio folder.
- In the models.py all the db models for this project are present.
- App.py – all the necessary routes for this project are present in the app.py folder
- Requirements.txt – in this folder all necessary requirements are import like flask, flask-sqlalchemy, jinja2, flask security etc....
- All CSS styling is integrated in the html page with bootstrap CSS link.

Features and functionality:-

- This web application is a multi-user application which contains separate login pages for admin, user and creators.
- Admin dashboard can see all the dynamic data like how many general users, creators and how many black creators etc..
- General users can listen to songs, then can like or dislike the songs, they can make the playlist by available songs.
- Creators are able to upload the new songs they can make the albums also they have all functionalities of general users.
- Creators also have the ability to edit the song information and are able to edit the album and playlist information.
- Search functionality is also present in which users can search any song, any album and they can search any playlist.
- General user can register himself as a creator .
- Proper flash alert messages are also present for the task performance.

To run the application:-

Unzip the folder and download all necessary dependencies present in the requirements.txt folder and run models.py and then run app.py file.

Project's presentation video link:-

https://drive.google.com/file/d/1gHW2bPTz9VKPuhhzPV-L6cLdaCKmdIT_/view?usp=drive_link