Music Streaming App

Author

Yash Kumar
22f3000472
22f3000472@ds.study.iitm.ac.in
Web Dev Enthusiast | Interested in Linux and Lower Level Programming

Description

Through this project, I aim to build a music streaming app where any user can upload and listen to music. It also allows users to create playlists and share them publicly.

Technologies Used

The technologies used in this project include flask, flask-restful, Jinja2, Python, Mutagen and sqlite3. Flask has been used as the backend framework while Jinja2 has been used for templating the user interface.

SQlite has been used as the backend database. Mutagen has been used to find the length of uploaded songs.

DB Schema Design

The project makes use of an SQLite database known as database.db which is located in the db folder. This database consists of 5 tables. An extract from the schema can be found below. The full schema can be found in the root directory with file name “schema.pdf”

Playlists Table

Name	Type	Schema
UID	INTEGER	"UID" INTEGER PRIMARY KEY AUTOINCREMENT
PLAYLISTNAME	TEXT	"PLAYLISTNAME" TEXT,
DATE CREATED	TIME	"DATE CREATED" NOT NULL DEFAULT CURRENT_TIMESTAMP,
USER	INTEGER	“USER” INTEGER
SONGS	TEXT	"SONGS" TEXT,

API Design

There are 9 REST API Endpoints in this application. These API Endpoints support POST & GET requests. The POST request can be used to update and add information. No PATCH or DELETE Endpoints have been defined as HTML Forms don't support them.

Architecture and Features

The root directory contains `app.py` which serves as the entry point to this application. The functions have been grouped and defined in python files under `functions` directory.

The RESTful APIs can be found at the `api.py` under the api folder. While the database file can be found in the `db` directory which also contains python files for creating the database and connecting to it.

The templates have been grouped into their related directories and can be found in the `templates` folder.

The app also defines a rating feature where listeners can click either upvote or downvote buttons to rate the song. The total rating is calculated as Number of Upvotes
* 100 / Total Votes

Each user can only edit songs uploaded by them. An admin account has also been defined.

In the Admin Panel, one can view the graph related to app performance and manage users such as blocking and unblocking them.

Project Link :

https://music-vfpd.onrender.com/auth/login

Video Link :

https://drive.google.com/file/d/1V7R4mLp9PoQrP3tFC06m2_YAZsOy3k2s/view?usp=sharing