Project Report: Multi-User Music Streaming Platform

OSMUSIC

SHARE YOUR MUSIC

OS Music MAD 2 PROJECT - DEC 2023	
Author:	**Rishabh Prakash (21f1001626)**
Email	**21f1001626@ds.study.iitm.ac.in**
Institution:	**Indian Institute of Technology, Madras**
Date:	**December 14, 2023**

Aim

This project aims to develop a multi-user music streaming platform where users can listen to music, create playlists, become creators, and upload their own songs. The platform also includes features for admins to manage content and track platform usage.

Description

The platform offers a wide range of functionalities for users and creators:

User Features:

- Register new user
- Login as User using RBAC
- View existing songs and albums
- Search for songs, albums, and genres
- Listen to songs
- View lyrics while playing songs
- Like songs
- Flag a song
- Create, delete, and modify playlists
- Add and remove songs from playlists
- Become a creator

Creator Features:

- Login as Creator using RBAC
- Upload songs
- Edit song information
- View song performance statistics
- Manage creator profile

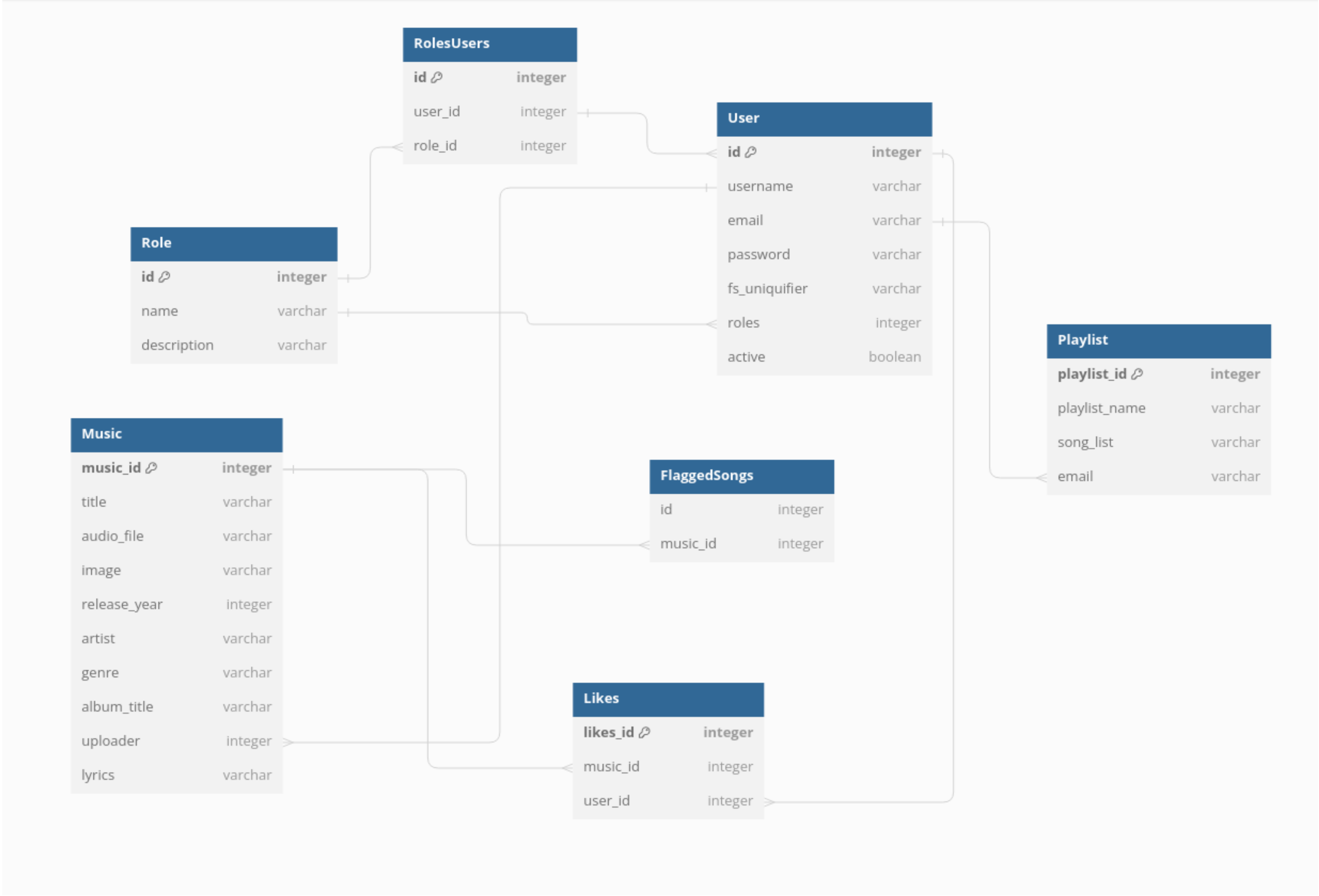
Admin Features:

- Login as Administrator using RBAC
- View platform statistics
- Manage creators
- Review and take action on reported songs

- Delete songs in violation of terms and conditions

Additionally, the platform features scheduled email reminders sent to users and creators.

Database Schema Design



RolesUsers	
id	**integer**
user_id	integer
role_id	integer

User	
id	**integer**
username	varchar
email	varchar
password	varchar
fs_uniquifier	varchar
roles	integer
active	boolean

Role	
id	**integer**
name	varchar
description	varchar

Playlist	
playlist_id	**integer**
playlist_name	varchar
song_list	varchar
email	varchar

Music	
music_id	**integer**
title	varchar
audio_file	varchar
image	varchar
release_year	integer
artist	varchar
genre	varchar
album_title	varchar
uploader	integer
lyrics	varchar

FlaggedSongs	
id	integer
music_id	integer

Likes	
likes_id	**integer**
music_id	integer
user_id	integer

Technologies Used

- Backend:
    - Python: Programming language
    - Flask: Micro web framework
    - SQLAlchemy: ORM library
    - Celery: Distributed task queue system
    - Flask-Security: Security and authentication extension
    - Redis: In-memory data store
- Frontend:
    - HTML
    - JavaScript
    - Vue.js: JavaScript framework
- Database:
    - SQLite3: Relational database

Architecture and Features

- Backend:
    - `app.py`: Initializes the server and runs the application
    - `config.py`: Configuration files
    - `cache.py`: Caching mechanisms
    - `resource_marshal.py`: Data marshaling for frontend communication
    - `task.py`: Scheduled and asynchronous tasks (e.g., email sending)
    - `workers.py`: Celery worker initialization
    - `api.py`: API routes for backend functionality
    - `database.py`: Database connection and initialization
    - `models.py`: ORM models for database interaction

- Frontend:
  - Vue.js components and base files
  - Form Validation using HTML and JS

Features Implemented

User and Creator:

- User registration
- Universal login for users, creators and admins
- Creator role switching
- Song listening, searching, liking and flagging
- Playlist creation, deletion, and modification
- Song upload, editing, and deletion
- Song lyric display during playback
- Creator profile with performance statistics

Admin:

- Access to platform statistics
- Creator list management
- Song review and action on reported content
- Song deletion for terms of service violations

Scheduled Reports, Notification and Reminders:

- Generate scheduled reports and send them by scheduled mail
- Daily and monthly email reminders sent to users and creators

Challenges Faced

- Setting up celery and redis
- Vue state management and routing the website
- Routing the website

Project presentation - https://drive.google.com/file/d/1azj1jojHyqfz3828K3NuoPzl2acgU47M/view?usp=sharing