Nirvana Music Streaming App

IITM AppDev-1 Project Report

(Click the link to watch the Video: Project Video)

Name: Anwesha Ghosh

Student ID: 21f1000323

Email Id: 21f1000323@ds.study.iitm.ac.in

Introduction

Nirvana is an online music streaming application that brings hundreds of songs of different languages around the world at our fingertips. It is user friendly, fast and free of ads. It is multi-faceted in which a user can also opt to become a creator and enjoy a wide range of features in addition to regular features. All activities are monitored by an Admin who has the authority to flag songs as well as blacklist or whitelist creators.

Key Features:

The three dimensions of the app are:

1) **User:**
   A new user has to first register on the app by creating a username and password and entering their email address for communication and security purposes. The user can then login to the app at any time just by entering the registered username and password. The user will then be able to view the regular user homepage which has panels for Recommended Tracks, User's Playlists and Albums. Here the user can browse through a multitude of songs and albums, create playlists, add or remove songs from the playlist, play songs simultaneously while viewing lyrics, play songs on shuffle and repeat, like, comment as well as rate songs. The user can also make use of the Search bar to search for songs, albums, singer names and also genres. If a regular user wishes to become a creator, they can do so by registering as a creator and providing some additional information about themselves.

2) **Creator:**
   Creators can not only enjoy all the privileges of regular users, they can also upload new songs, lyrics and albums to the app. They can view various statistics such as the total number of songs and albums that they have uploaded, the average rating of the songs uploaded by them etc. on the Creator Dashboard. They can also view each song, and their corresponding lyrics and edit as well as delete the songs that were previously uploaded.

3) **Admin:**
An admin has a separate login authentication which cannot be accessed by regular users and creators. An admin has a separate dashboard in which he/she can keep track of various statistics such as total number of users, creators, songs, albums and genres. He/she can also analyse the data on the app through various graphs such as popular songs, popular genres and top creators. Additionally, he/she can view all the songs, segregated by genre, flag inappropriate songs, read lyrics and also delete the songs from the app permanently. The admin also has the authority to blacklist certain creators if they are repeatedly uploading unsuitable songs and lyrics.

DB Schema:

There are a total of 7 tables in my database which are as follows:

1) **User:** It contains 6 columns – id, username, email, password, role (user, creator, admin), is_creator which will either be 0 or 1.
2) **Creator:** It contains 6 columns – id, user_id, bio, genre, language, listed (0 for whitelisted and 1 for blacklisted).
3) **Playlist:** It contains 3 columns – id, title, user_id (foreign key reference to id from User table)
4) **Song:** It contains 13 columns – id, title, singer, created_date, genre, album, duration, lyrics, song_file, creator_id (foreign key reference to id from User table), flag, likes and rating.
5) **Rate:** It contains 6 columns – id, song_id (foreign key reference to id from Song table), user_id (foreign key reference to id from User table), rate, like and comment.
6) **Album:** It contains 3 columns – id, title and artist_id (foreign key reference to id from User table)
7) **PlaylistSongs:** It contains 3 columns – id, playlist_id (foreign key reference to id from Playlist table), song_id (foreign key reference to id from Song table)

Additional Features:

1) Search Bar: Provides with approximate results even for imperfect and incomplete searches in both upper and lower case.
2) Aesthetic and user-friendly interface with hassle free registration and login.
3) Songs can be played on loop, shuffle and autoplay
4) API Design: CRUD operations can also be performed using API endpoints for the same table schema using the HTTP requests – GET, POST, PUT, DELETE.

Architecture:

This project makes use of the Model-View-Controller (MVC) architecture. The controllers are stored in the project folder named “app.py” and handles the user requests and updates the model accordingly. The models are used to manage the data and interact with the database. The HTML templates are stored in the templates folder and are used to provide the user interface and display the data to the user.