NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name : Leo William G

Student ID : 311121104034

College Name

Loyola-ICAM College of Engineering and Technology

CAPSTONE PROJECT SHOWCASE

Project Title

Notes Sharing Web Application using Django Framework

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion

Abstract

Django Music Application : HarmoniX

The Django Music Application is a web-based platform designed for seamless music discovery, playback, and management. With user authentication, playlist creation, and personalized recommendations, users can explore a vast music library, interact with fellow music enthusiasts, and enjoy their favorite tunes across devices. Powered by Django, this application aims to deliver a user-centric music streaming experience with a responsive interface and robust backend database integration. Through this project, we aim to deliver a high-quality music player solution that enhances the user's enjoyment and convenience in accessing and enjoying music online.

Problem Statement

To create a music application using Django, which is a Python-based web framework.

The main challenge is to develop a Django-based music application that addresses the shortcomings and offers a user-centric solution for discovering, organizing, and enjoying music online.

The several use cases includes:

- User Experience
- Backend functionality
- Personalization
- Database Integration
- Scalability
- Performance

Project Overview

- Initially there is a login/signup page provided in order to handle the user sessions.
- Once a new user logs in initially a new playlist and all songs will be available.
- But if the user has already logged in then it will show their own playlist.
- Only the administrator will have the authority to provide or upload the songs.
- Django administration is used to handle the data.
- Ensures compatibility across devices and offers customization features.
- Utilizes HTML, CSS, JavaScript, Django and APIs for development.
- Emphasizes usability, performance, and compatibility.

Proposed Solution

- Develop a modern web interface using HTML, CSS, JavaScript and Django.
- Integrate with music streaming services via APIs for a vast music library.
- Include playlist management and playback controls for user convenience.
- Implement robust search functionality and user profiles for customization.
- Proper database management and propose the user.
- Ensure compatibility across devices and prioritize usability and performance.
- Gather user feedback for iterative improvements.
- Deliver a high-quality music streaming experience online.

Proposed Solution

- Technical Stack: Frontend: HTML, CSS, JavaScript, React.js
- Backend: Django Framework, Django REST framework
- Database: PostgreSQL Hosting: AWS EC2, S3 for media storage
- Machine Learning: Python, TensorFlow for recommendation engine
- Target Audience: MeloTune targets music enthusiasts who seek a sophisticated and personalized music streaming platform that goes beyond conventional offerings.
- Outcome: MeloTune aims to set a new standard in music streaming by combining cutting-edge technology with a user-centric approach, providing a unique and immersive music experience for all users.

Technology Used

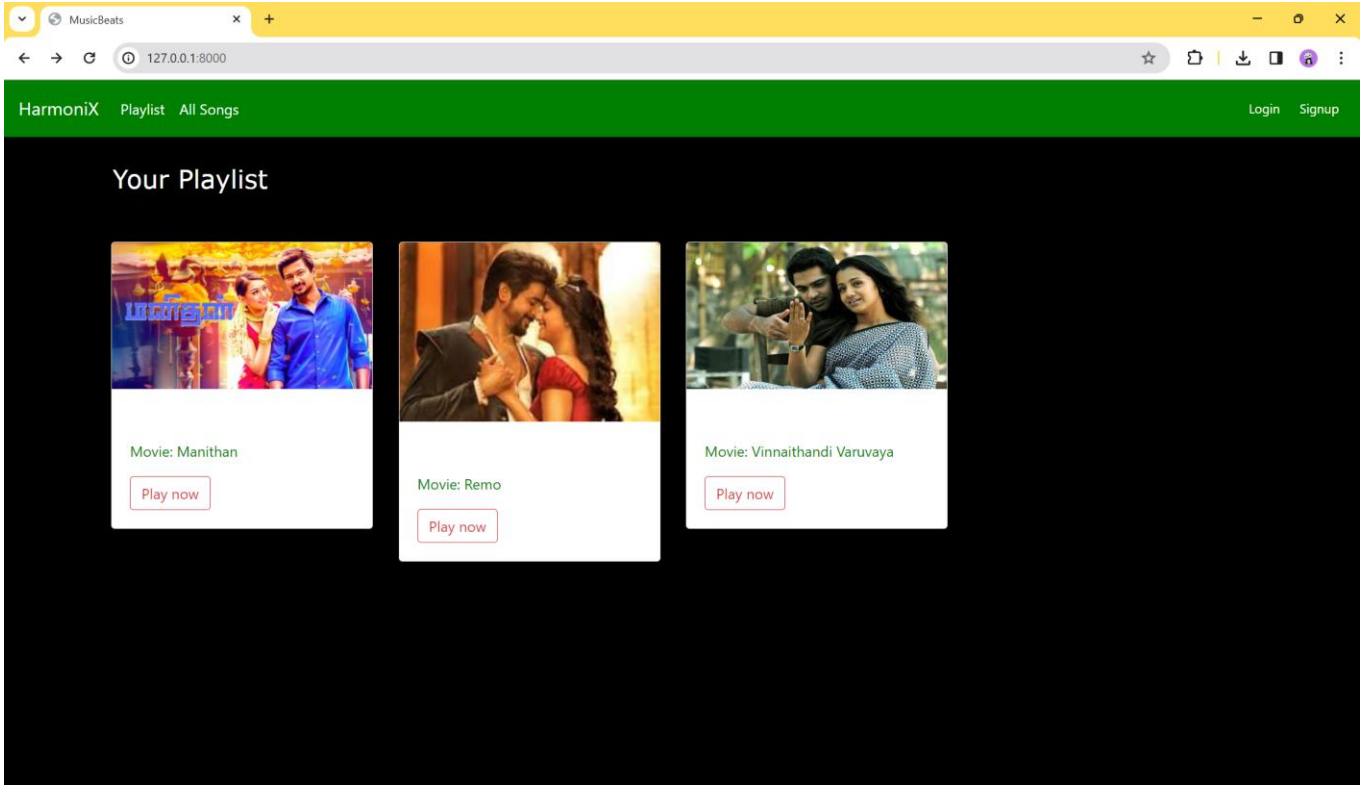
Front-end

Back-end

Modelling & Results

- **Data Model:** Utilize Django's ORM to define models for songs, albums, artists, playlists, and user interactions (e.g., likes, shares).
- **Machine Learning Model:** Use TensorFlow or scikit-learn to develop a recommendation engine based on user listening history and preferences.
- **Results:** Dynamic Music Library: Users will have access to a diverse and continuously updated music library, ensuring a wide selection of songs, albums, and artists.
- **Personalized Playlists:** The recommendation engine will enhance user experience by suggesting songs and artists based on their listening habits, leading to more personalized playlists.
- **Engaging User Interface**: The modern and responsive design will provide a visually appealing and intuitive user interface, enhancing user engagement and satisfaction.
- **Social Integration:** Users will be able to connect with friends, share favorite tracks, and discover new music together, fostering a sense of community within the application.

Homepage


MusicBeats
127.0.0.1:8000
HarmoniX
Playlist
All Songs
Login
Signup
Your Playlist
Movie: Manithan
Play now
Movie: Remo
Play now
Movie: Vinnaithandi Varuvaya
Play now

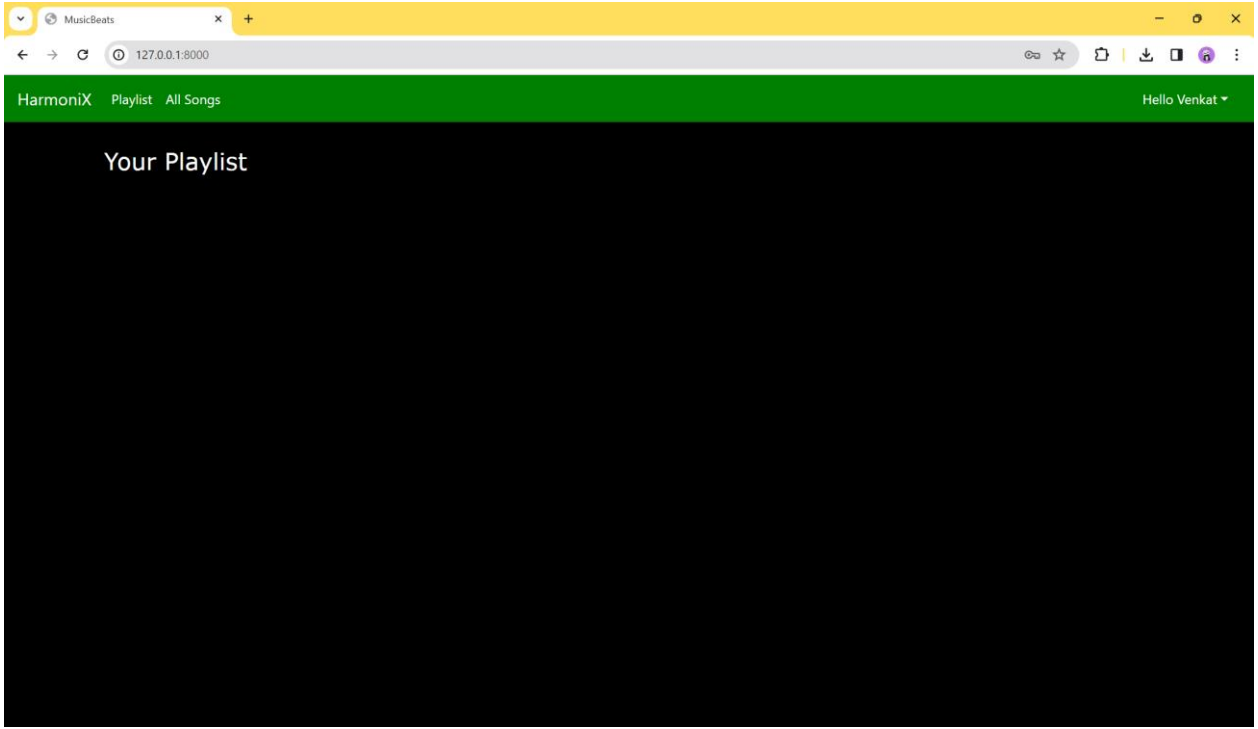
Sign-Up-Page


SignUp
127.0.0.1:8000/musicbeats/signup
HarmoniX
Playlist
All Songs
Login
Signup
Email address
name@example.com
Username
Venkat
First Name
Kirthik
Last Name
Venkat
Password
Renter Password
SignUp

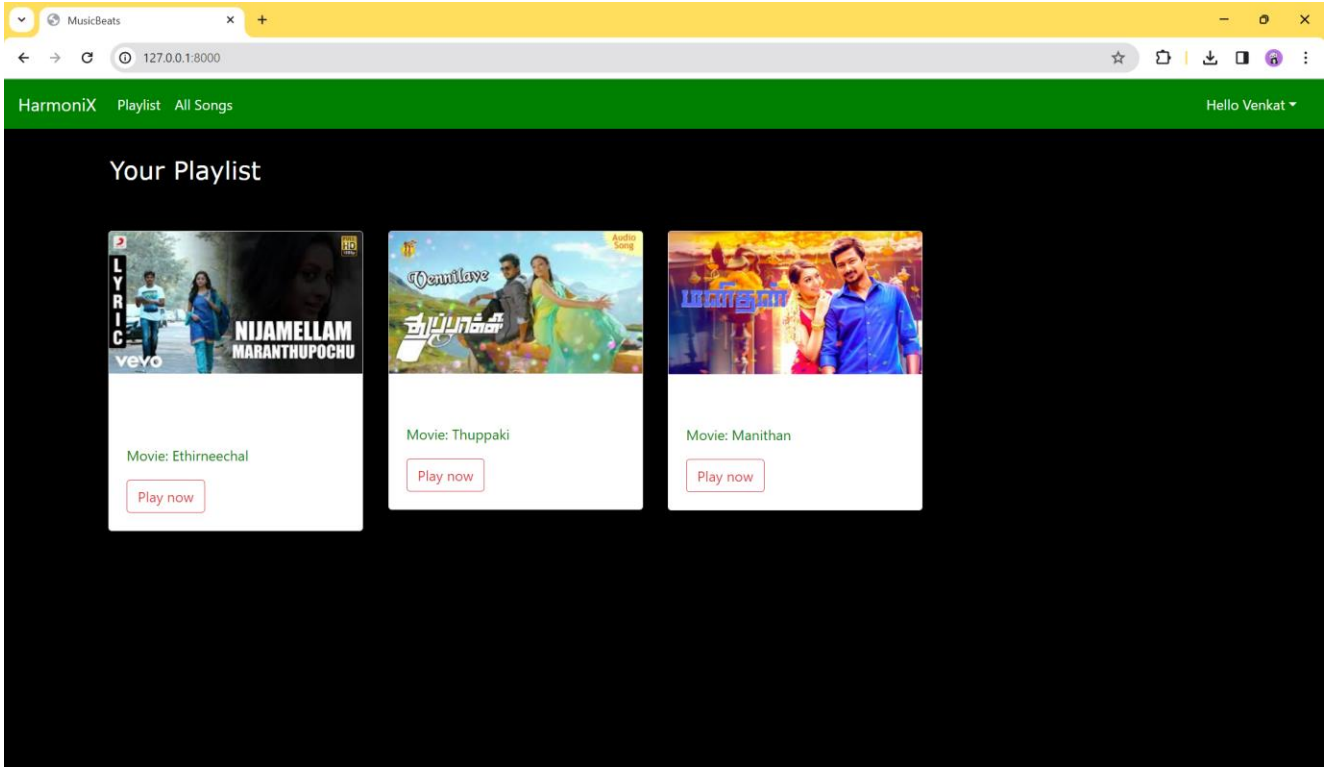
Log-In-Page


Login
127.0.0.1:8000/musicbeats/login
HarmoniX
Playlist
All Songs
Login
Signup
Username
Your Username
Password
Password
Login

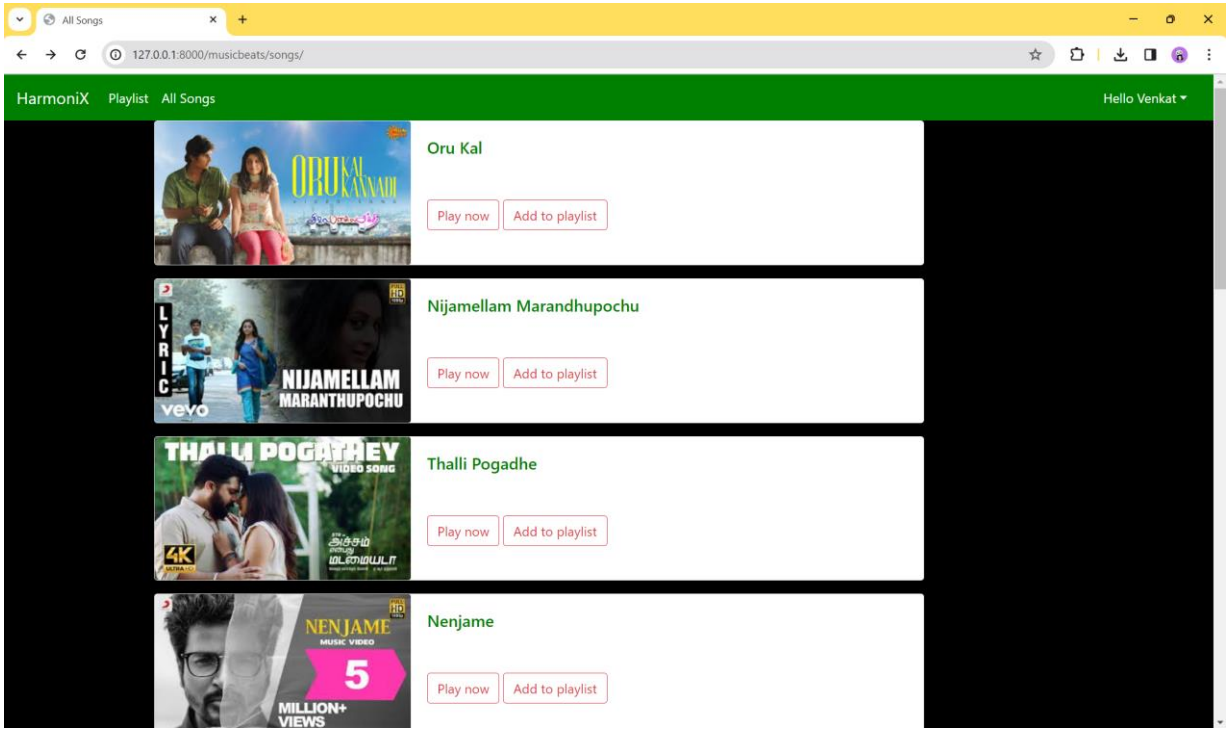
My Playlist Before Adding Songs



My Playlist After Adding Songs



All Songs Page



Future Enhancements:

Enhanced Recommendation System: Implement cutting-edge machine learning algorithms to enhance music recommendations, considering user preferences, listening history, and behavior for improved accuracy.

Expanded Social Features: Enhance social integration by enabling users to follow others, create public playlists, and view real-time updates on listening activities, fostering a vibrant community.

Lyrics Display: Integrate a feature to display song lyrics, elevating the music listening experience by providing additional context and engagement for users.

Comprehensive Artist and Album Insights: Offer users in-depth information about artists and albums, including biographies, discographies, and related content, enriching their understanding and appreciation of music.

Podcast and Audiobook Inclusion: Extend the platform's scope to encompass podcasts and audiobooks, broadening users audio content options for a more diverse listening experience.

Innovative Music Discovery Tools: Develop intuitive tools for users to explore new music based on genres, moods, or themes, facilitating discovery and expanding their musical horizons.

Conclusion

Django provides a better framework which is very responsive. Django's built-in features, such as the ORM (Object-Relational Mapper) and admin interface, accelerate development, allowing for quick prototyping and iteration of features in music applications. It supports multiple databases, including PostgreSQL, MySQL, and SQLite, offering flexibility in choosing the most suitable database backend for storing music metadata, user preferences, and playlists.The application's technical stack, including HTML, Bootstrap, Django Framework, Django REST framework, and sqlite, ensures a robust and scalable platform. Harmonix target audience, comprising music enthusiasts seeking a personalized music streaming experience, is well served by its innovative features and user-friendly interface.

Thank You!