NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Done by,

Student Name : PRAVEENA G

Student ID :au311121205045

Loyola-ICAM College of engineering and technology

Chennai

CAPSTONE PROJECT SHOWCASE

Project Title

Music Web Application using Django Framework

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

Abstract

In the era of digital music streaming, there's a growing demand for web applications that provide users with a seamless music listening experience. This project aims to develop a music web application using the Django web framework to cater to this demand. The application will allow users to discover, stream, and manage their favorite songs and playlists.

Source :

Problem Statement

The existing music streaming platforms may not always meet the specific preferences of users. Additionally, some users may prefer a more personalized and customizable music listening experience. Therefore, there's a need to develop a music web application that offers a wide range of features while providing users with greater control over their music choices.

Source :

Project Overview

The music web application will include features such as user authentication, song browsing, playlist creation, song playback, search functionality, user profile management, and social sharing. Users will be able to create accounts, browse through a vast catalog of songs, create personalized playlists, and share their favorite music with friends. The application will provide a user-friendly interface and a smooth navigation experience.

Source :

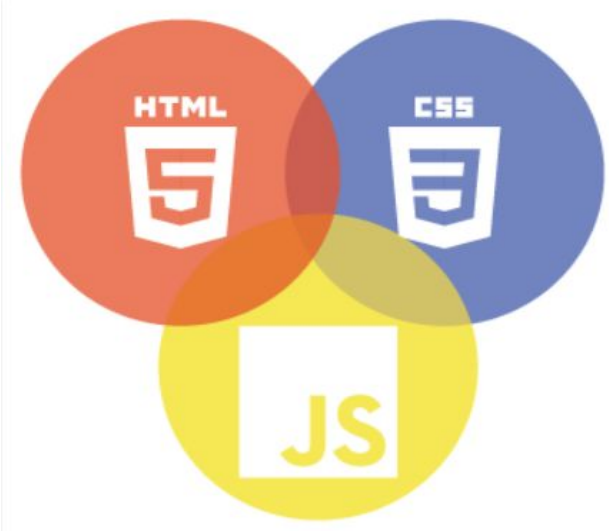
Proposed Solution

To address the problem statement, we propose developing a music web application using the Django web framework. Django's built-in features such as authentication, ORM (Object-Relational Mapping), and template engine will be leveraged to build the core functionality of the application. Additionally, we'll integrate third-party APIs for fetching song metadata and streaming music. The application will be designed with a responsive and intuitive user interface to ensure compatibility across various devices.

Source :

Technology Used

Front-end



Back-end

Source :

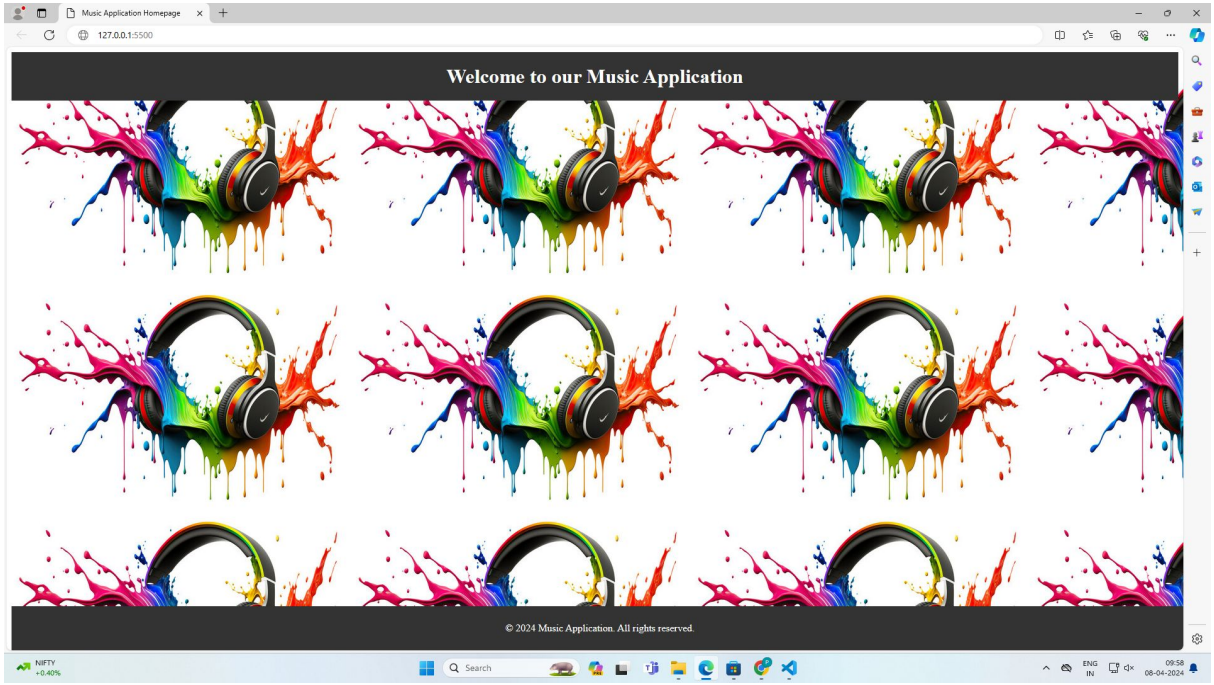
Modelling & Results

- Data Model: The application will include models for User, Song, Playlist, and other relevant entities. Relationships between these models will be established to maintain data integrity.
- User Interface: The user interface will be designed to be intuitive and user-friendly, allowing users to easily navigate through the application and access desired features.
- Functionality: The application will allow users to register and login, browse songs, create playlists, search for specific songs or artists, play songs, and share music with others.
- Testing: The application will undergo rigorous testing to ensure functionality, security, and performance. Testing will include unit tests, integration tests, and user acceptance testing.

Source :

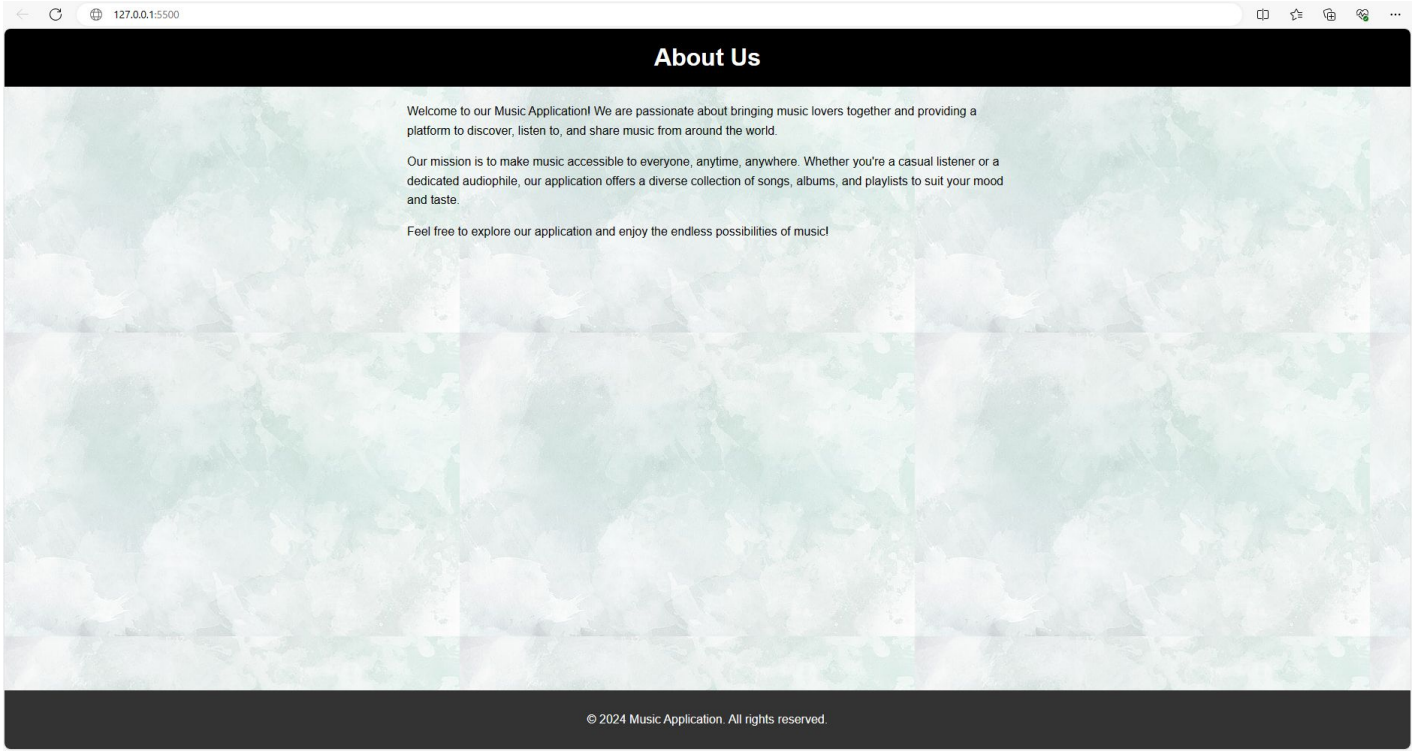
edunet foundation

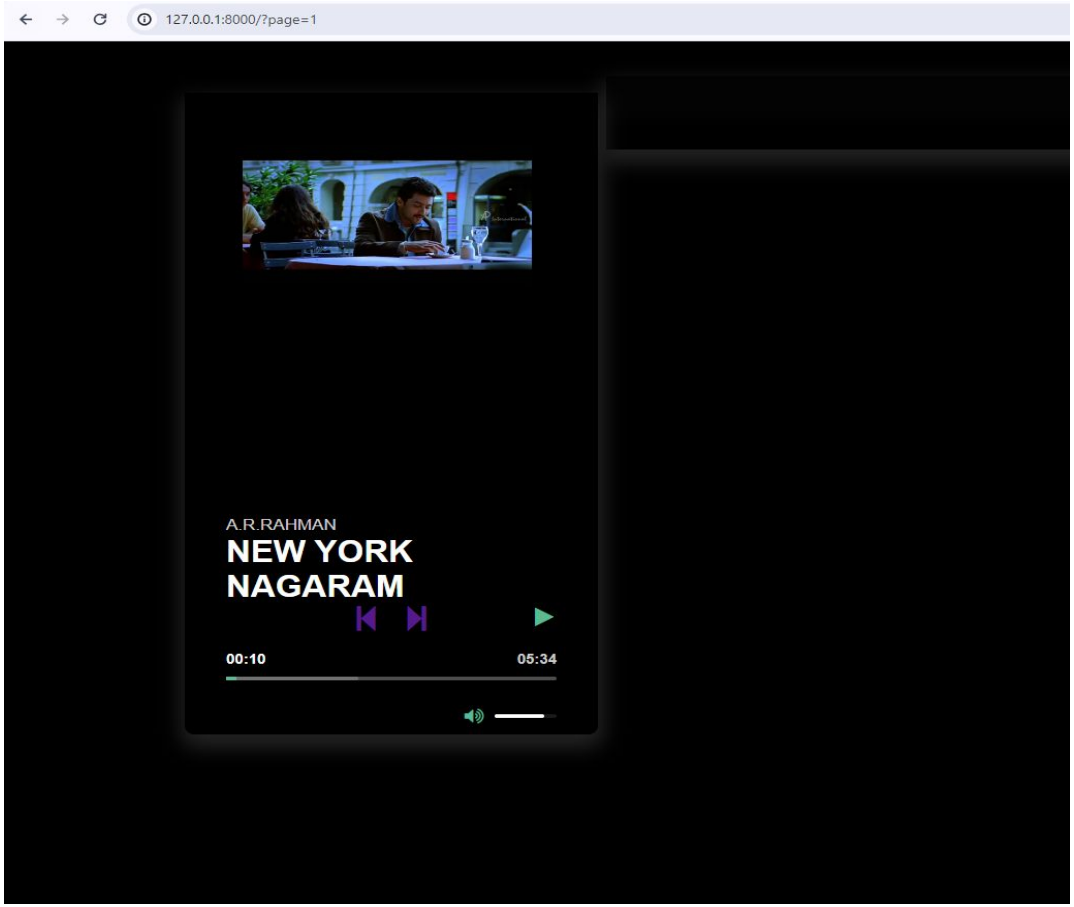
Homepage

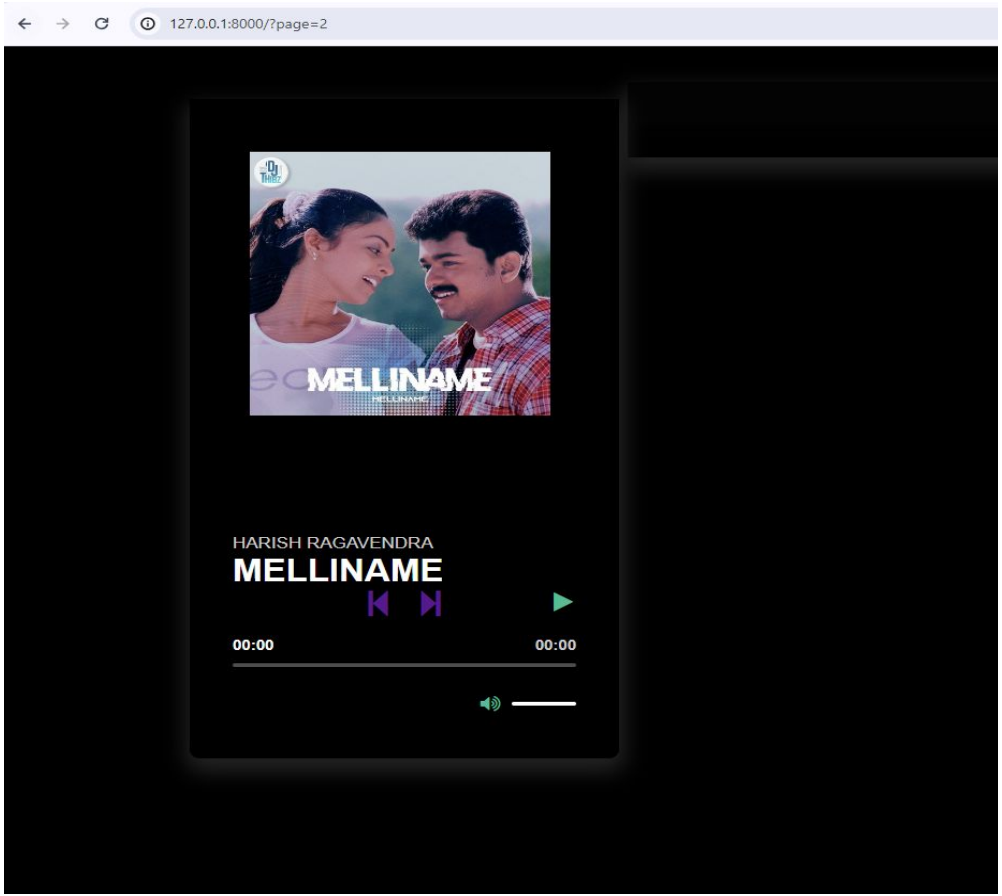


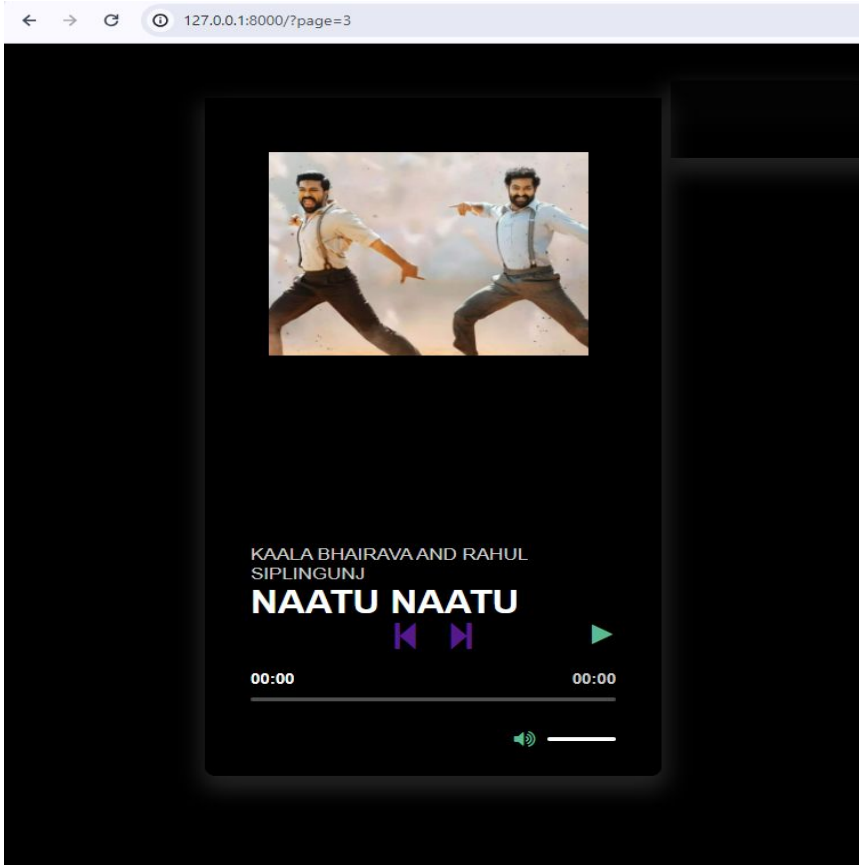
edunet foundation

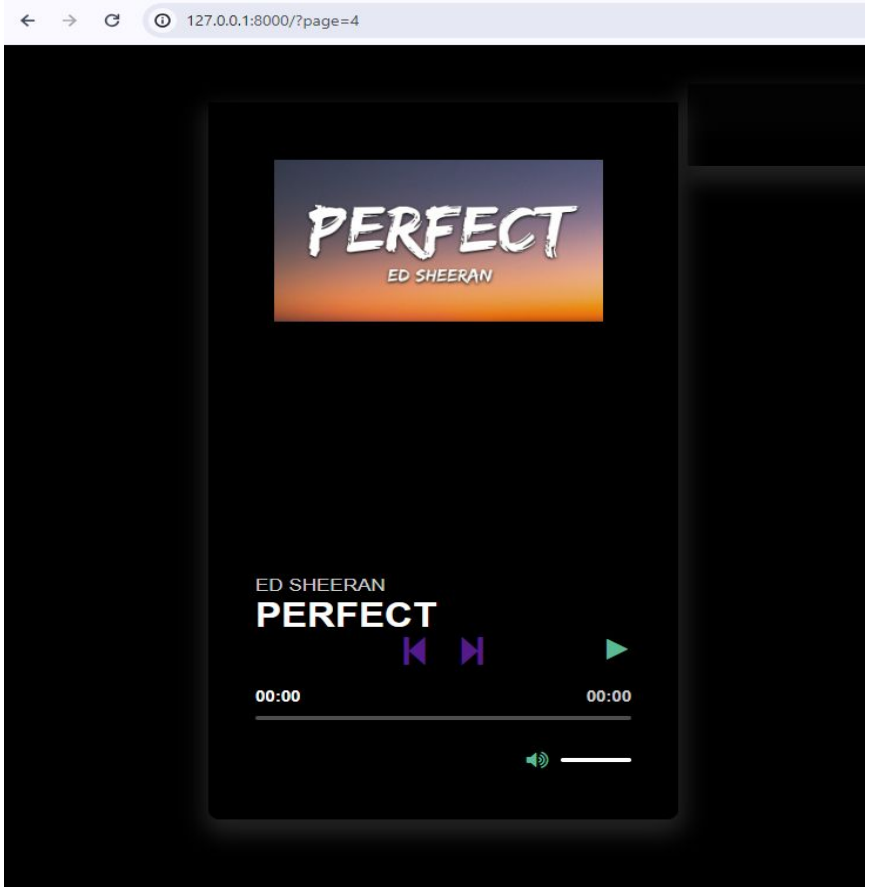
About-Us-Page




127.0.0.1:8000/?page=1
A.R.RAHMAN
NEW YORK
NAGARAM
00:10
05:34


127.0.0.1:8000/?page=2
MELLINAME
HARISH RAGAVENDRA
MELLINAME
00:00
00:00


127.0.0.1:8000/?page=3
KAALA BHAIRAVA AND RAHUL SIPLIGUNJ
NAATU NAATU
00:00
00:00


127.0.0.1:8000/?page=4
PERFECT
ED SHEERAN
ED SHEERAN
PERFECT
00:00
00:00

Future Enhancements:

1. **Personalized Recommendations:** Utilize advanced algorithms to analyze user listening habits, preferences, and contextual data to offer tailored music recommendations.
2. **Social Integration:** Implement features that enable users to connect with friends, share playlists, view what others are listening to, and collaborate on music discovery.
3. **AI-Driven Playlist Creation:** Employ artificial intelligence algorithms to curate playlists based on user preferences, mood, activity, and even contextual factors like weather or location.
4. **Enhanced Discovery Tools:** Develop tools such as interactive maps showcasing local music scenes, virtual concerts, or immersive experiences for users to explore new genres and artists.
5. **Lyric Integration**: Integrate lyrics into the app, allowing users to follow along with songs, search based on specific lyrics, and even provide translations for multilingual lyrics.
6. **Live Streaming and Virtual Concerts:** Partner with artists and venues to offer live streaming of concerts, virtual reality experiences, and exclusive behind-the-scenes content for users.
7. **Music Education Resources:** Provide tutorials, instrument lessons, music theory quizzes, and interactive challenges to help users learn more about music and improve their skills.
8. **Voice Control and Integration:** Enable voice-controlled commands for hands-free operation, allowing users to search for songs, control playback, and navigate the app using voice commands.

Conclusion

In conclusion, the development of a music web application using Django offers a solution to the demand for personalized music streaming platforms. By leveraging Django's powerful features and integrating third-party APIs, the application provides users with a seamless music listening experience while offering flexibility and customization options. With further refinement and enhancements, the application has the potential to become a popular choice among music enthusiasts.

edunet foundation

Thank You!