NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name : Byleen Janet Roy J
Student ID : 311121205015

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

This Music Streaming Application, built on the Django framework, delivers a comprehensive and user-centric platform for seamless music discovery and streaming. Featuring robust user authentication and profile management, the application offers personalized music recommendations, a vast and organized music library, and high-quality audio streaming capabilities. Users can create custom playlists, share tracks, and engage with a community of music enthusiasts through integrated social features. Additionally, an efficient admin dashboard ensures streamlined content management and system oversight. Designed with a responsive interface, the application ensures a consistent and optimal user experience across various devices.

Source :

Problem Statement

Designing an efficient music application using Django poses challenges such as implementing robust user authentication, managing complex music metadata, enabling seamless audio streaming, and ensuring a responsive frontend interface. The project aims to address these challenges by developing a scalable and user-friendly platform that simplifies music discovery and organization. Key objectives include optimizing database performance for large music catalogs, integrating secure audio streaming capabilities, and delivering a visually appealing and intuitive user experience. The solution will focus on leveraging Django's features effectively to overcome these hurdles and provide a compelling music application that meets modern user expectations.

Source :

Project Overview

This project involves building a music application using Django, integrating user authentication, music catalog management, playlist creation, and audio streaming functionalities. The application will leverage Django's ORM for database operations, ensuring data integrity and scalability. Frontend development will utilize Django's templating system and JavaScript for interactive features. Key goals include delivering a responsive and visually appealing interface for users to explore and enjoy music seamlessly. Emphasis will be placed on optimizing performance, security, and user experience throughout the development process.

Source :

Proposed Solution

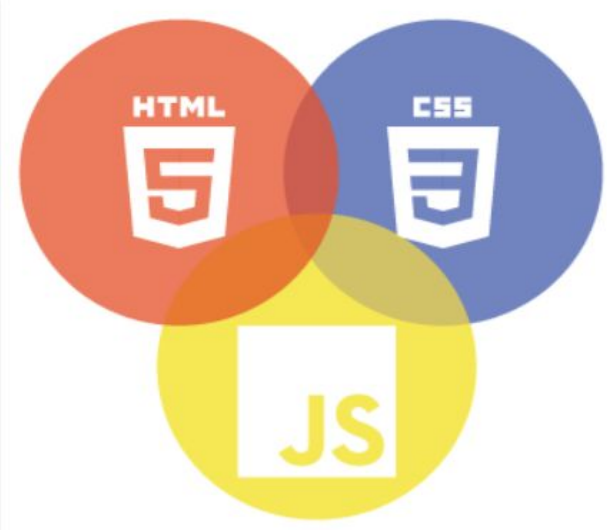
- User Authentication and Authorization:
  Implement secure user registration and login functionality using Django's authentication system. Ensure proper authorization levels for users and administrators to control access to features like song management.
  Song Catalog and Management:
  Develop a robust backend system to manage a catalog of songs, albums, and artists. Utilize Django models to define database schemas for storing music metadata and implement CRUD (Create, Read, Update, Delete) operations for song management.
  Admin Dashboard for Song Administration:
  Create an intuitive admin dashboard accessible only to authorized users (admins). This dashboard allows admins to add new songs, update existing songs (including metadata like title, artist, and genre), and delete songs as needed

Source :

User-Facing Home Page with Song Playback: Design a user-friendly home page where authenticated users can browse through available songs and play them directly on the website. Implement audio streaming functionality to deliver a seamless listening experience. Responsive Frontend Design: Develop responsive frontend views using Django templates, HTML, CSS, and JavaScript to ensure a visually appealing and intuitive interface. Focus on usability and responsiveness across different devices and screen sizes to enhance user experience.

Technology Used

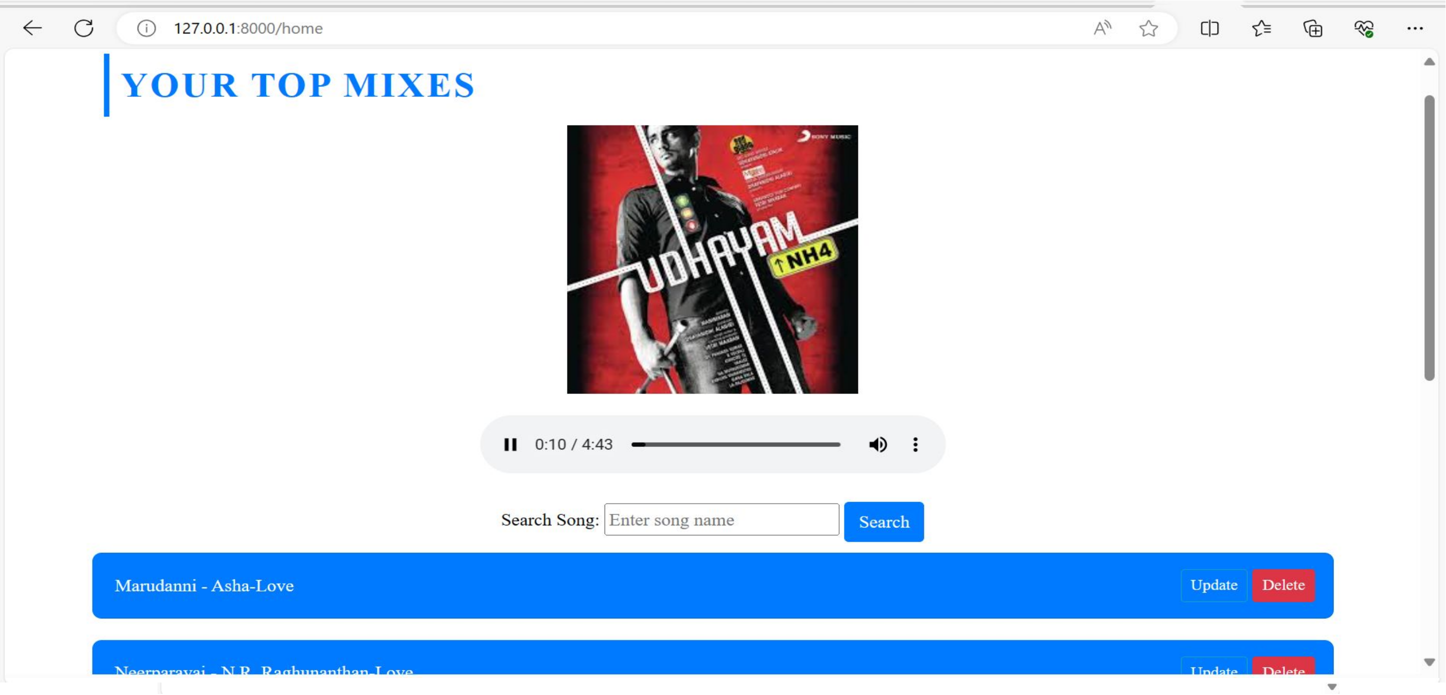
Front-end



Back-end

Source :

Modelling & Results



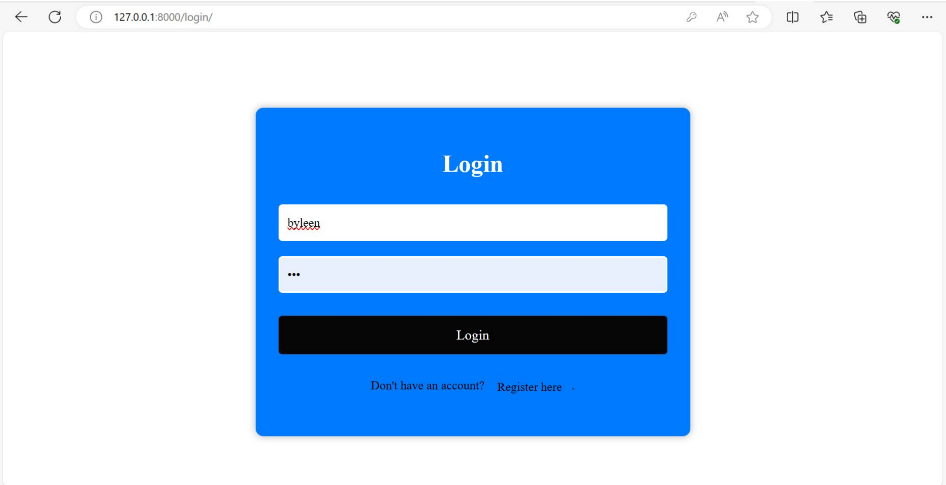
YOUR TOP MIXES

Search Song: Enter song name Search

Marudanni - Asha-Love Update Delete

Neerparavai - N.R. Raghunanthan-Love Update Delete

Source :



127.0.0.1:8000/login/

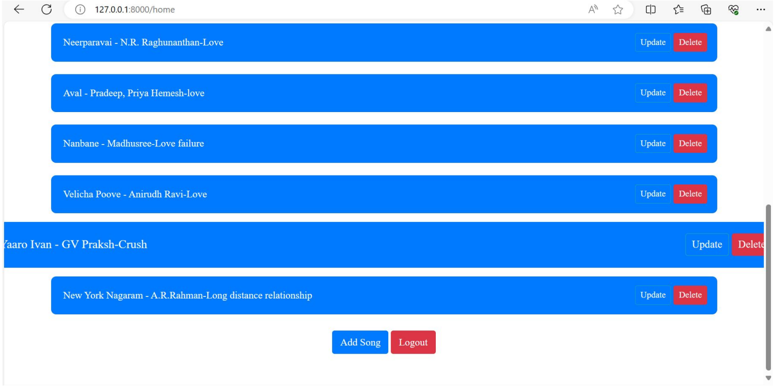
Login

byleen

•••

Login

Don't have an account? Register here



127.0.0.1:8000/home

Neerparavai - N.R. Raghunanthan-Love Update Delete

Aval - Pradeep, Priya Hemesh-love Update Delete

Nanbane - Madhusree-Love failure Update Delete

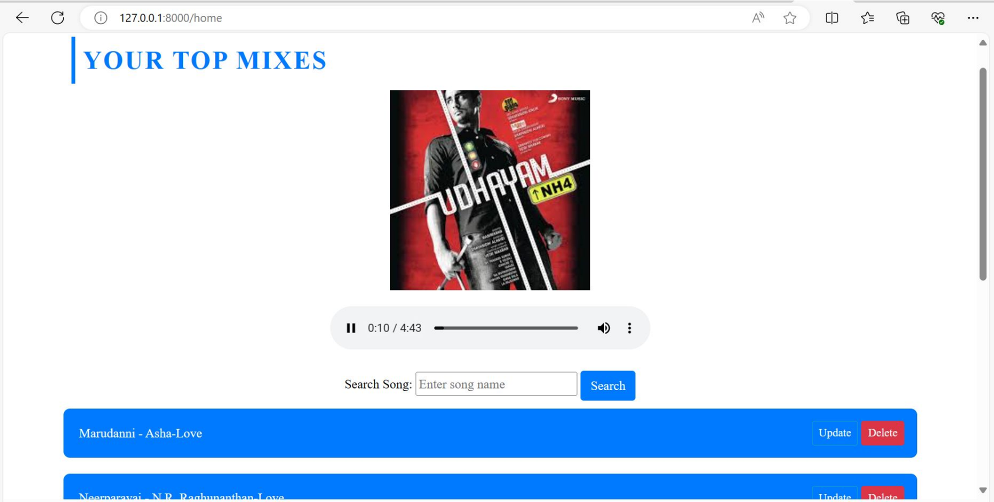
Velicha Poove - Anirudh Ravi-Love Update Delete

Yaaro Ivan - GV Praksh-Crush Update Delete

New York Nagaram - A.R.Rahman-Long distance relationship Update Delete

Add Song Logout

Next Gen Employability Program



127.0.0.1:8000/home

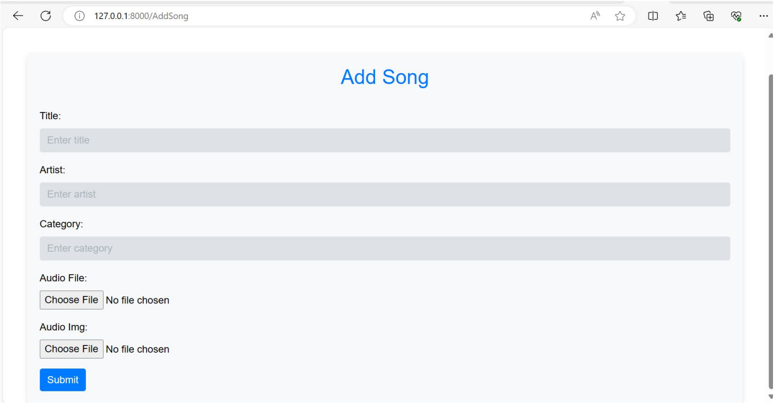
YOUR TOP MIXES

0:10 / 4:43

Search Song: Enter song name Search

Marudanni - Asha-Love Update Delete

Neerparavai - N.R. Raghunanthan-Love Update Delete



127.0.0.1:8000/AddSong

Add Song

Title:

Enter title

Artist:

Enter artist

Category:

Enter category

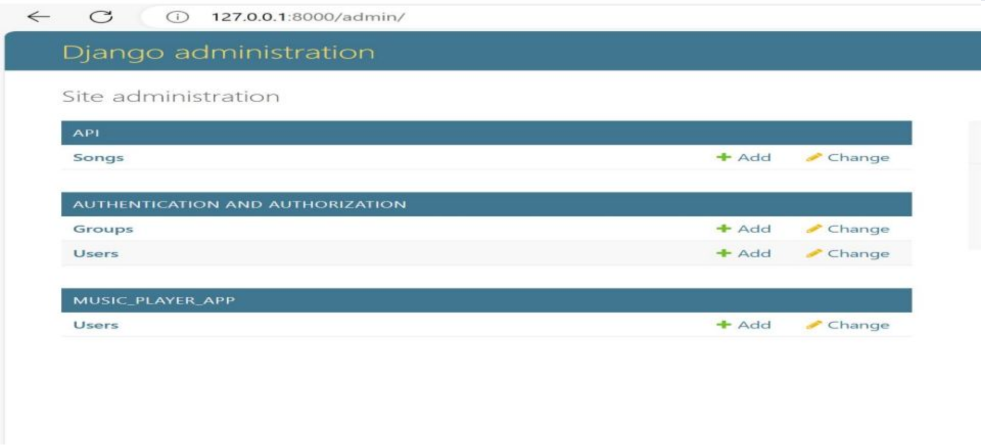
Audio File:

Choose File No file chosen

Audio Img:

Choose File No file chosen

Submit



127.0.0.1:8000/admin/

Django administration

Site administration

API

Songs + Add ✎ Change

AUTHENTICATION AND AUTHORIZATION

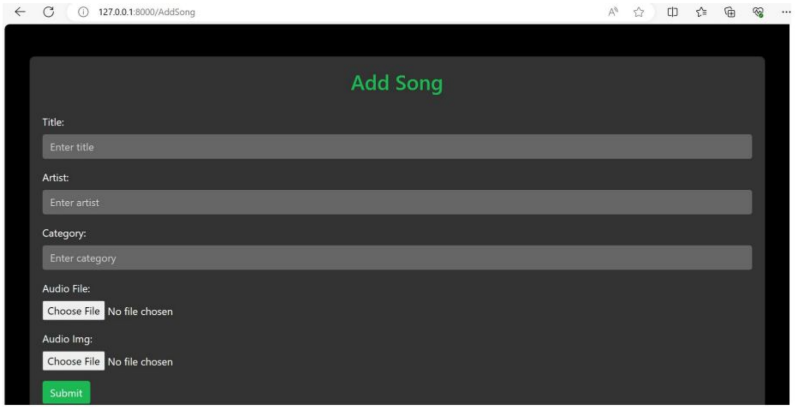
Groups + Add ✎ Change

Users + Add ✎ Change

MUSIC_PLAYER_APP

Users + Add ✎ Change

AddSong Page



127.0.0.1:8000/AddSong

Add Song

Title: Enter title

Artist: Enter artist

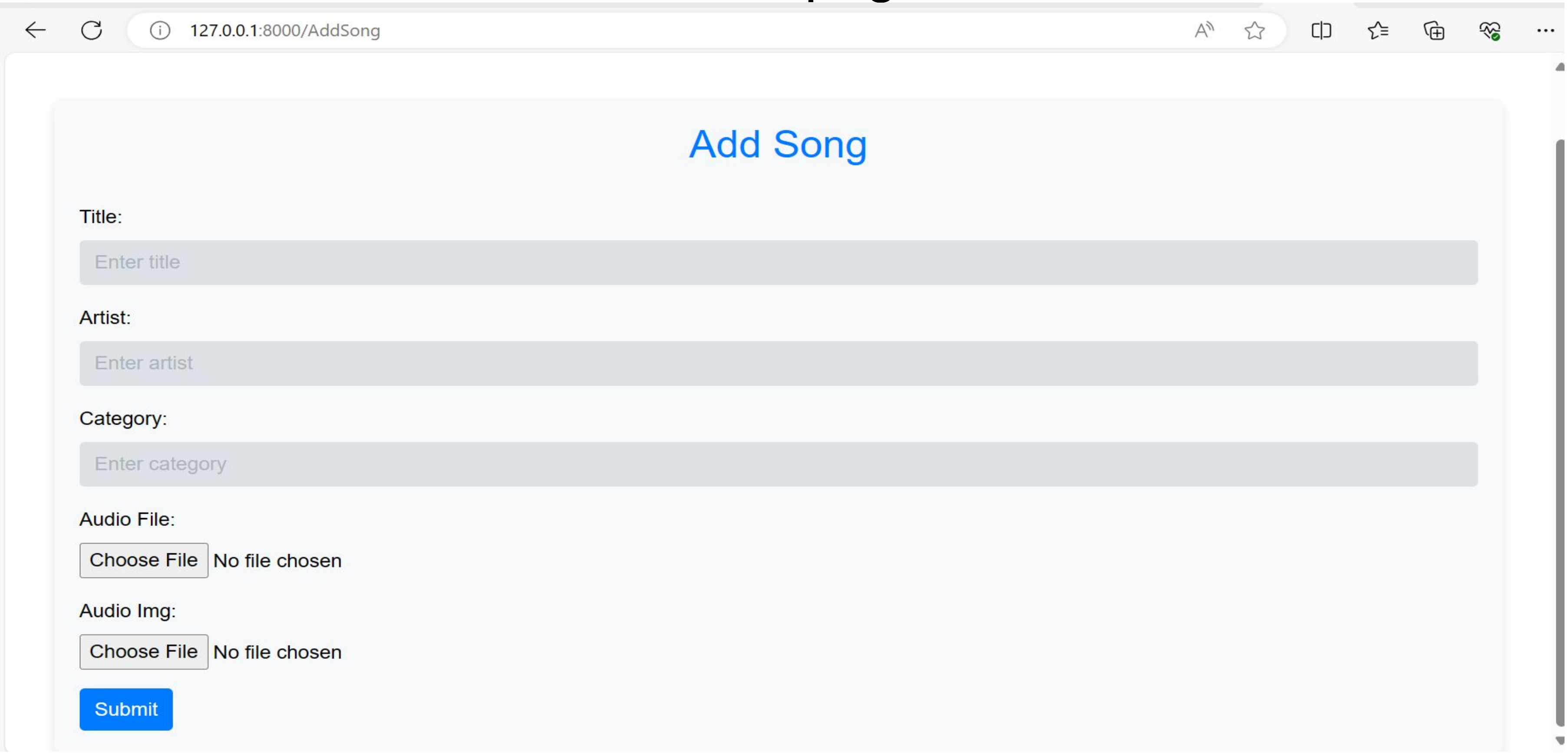
Category: Enter category

Audio File: Choose File No file chosen

Audio Img: Choose File No file chosen

Submit

Homepage



127.0.0.1:8000/AddSong

Add Song

Title:

Enter title

Artist:

Enter artist

Category:

Enter category

Audio File:

Choose File No file chosen

Audio Img:

Choose File No file chosen

Submit

Future Enhancements:

Advanced Search and Filtering:
Enhance the search functionality to support advanced queries such as filtering songs by genre, artist, album, release date, and popularity.
User Profiles and Social Features:
Implement user profiles where users can customize their preferences, create public playlists, follow other users, and share music recommendations.
Integration with External APIs:
Integrate with external music APIs (e.g., Spotify, Apple Music) to import additional music metadata, album artwork, and potentially allow users to link their accounts for cross-platform functionality.
Cloud Storage Integration:
Provide options for users to upload their music files securely to the cloud (e.g., AWS S3, Google Cloud Storage) and stream personal collections alongside the platform's catalog.

Conclusion

Developing a music application using Django offers a robust foundation for a user-friendly and scalable platform. Through secure user authentication, comprehensive song catalog management, and intuitive frontend design, we've created a functional prototype.

Looking forward, planned enhancements such as advanced search, social features, and mobile app integration promise to elevate the application's utility and appeal. Continuous integration, accessibility improvements, and machine learning-driven recommendations will further enhance the user experience.

This project highlights the versatility of Django in building complex web applications and underscores the importance of iterative development based on user feedback. With these enhancements, the music application is poised to evolve into a comprehensive and engaging platform for music enthusiasts worldwide.

Source :

Thank You!