



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name : Melvina Rajvee R C
Student ID : au311121104036

College Name

Loyola ICAM College of
Engineering and Technology

CAPSTONE PROJECT SHOWCASE

Project Title

Music Web Application using Django Framework

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



Abstract:

This project involves development of a music application using Django. This application will allow users to create accounts, manage music libraries and stream music. Key features include user authentication, a responsive user interface, secure file storage, and a dynamic music player interface. Additionally, the application will utilize Django's built-in ORM for database management and RESTful APIs for seamless integration with frontend technologies.

Problem Statement

Creation of Music Web Application using Django Framework

Project Overview

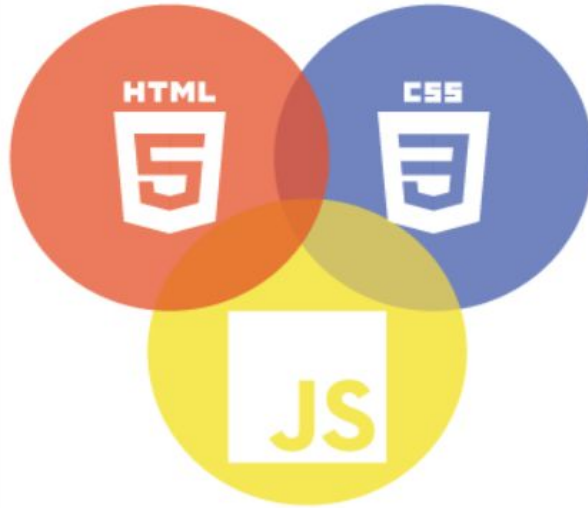
This project aims to develop a modern and user-friendly music streaming application using Django. The application will enable users to create personalized accounts, manage their music collections, and enjoy seamless streaming of their favorite tracks. Key features include user authentication, responsive design for various devices, secure file storage, and an intuitive music player interface. Leveraging Django's ORM and RESTful APIs, our goal is to deliver a reliable platform that enhances the music listening experience for all users.

Proposed Solution

Our solution involves building a music streaming application using Django, integrating essential features like user authentication, music upload and management, and a responsive music player. We will implement secure file storage for user uploads and ensure compatibility across different devices through responsive design. The use of Django's ORM will facilitate efficient database management, while RESTful APIs will enable seamless communication with the frontend. Our aim is to create a streamlined and enjoyable music streaming experience for users, emphasizing simplicity and functionality.

Technology Used

Front-end



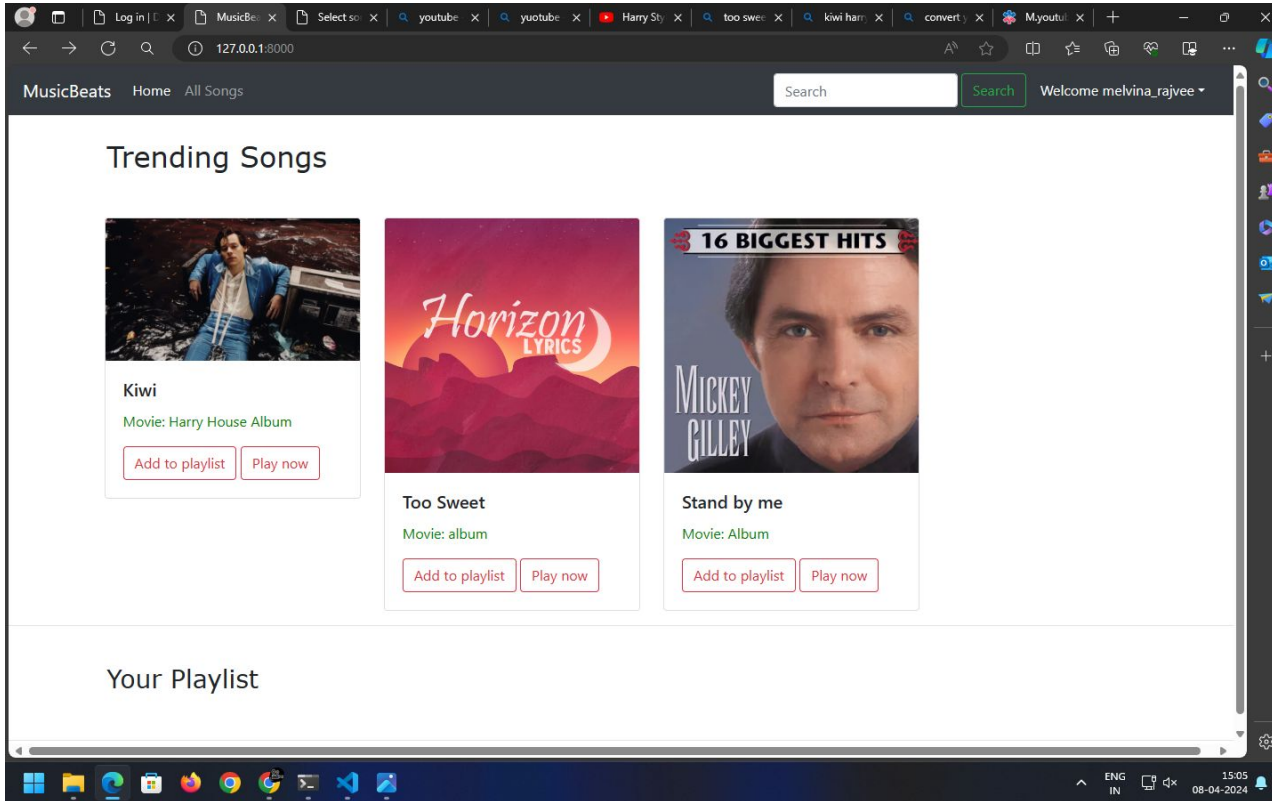
Back-end



Modelling & Results

The project will involve designing a Django data model to represent users, music tracks, and playlists. Utilizing Django's ORM, we'll ensure efficient data storage and retrieval. User authentication will be implemented using Django's built-in authentication system. For the frontend, responsive templates will be created using HTML, CSS and JavaScript for seamless user interaction. The music streaming functionality will be achieved using Django's file handling capabilities and a custom music player interface.


Homepage




The screenshot shows a web browser window displaying the MusicBeats homepage. The browser's address bar shows the URL 127.0.0.1:8000. The website has a dark header with the MusicBeats logo, navigation links (Home, All Songs), a search bar, and a user greeting (Welcome melvina_rajvee). The main content area features a 'Trending Songs' section with three song cards. Each card includes an album cover, the song title, the artist, and buttons to 'Add to playlist' and 'Play now'. The first card is for 'Kiwi' by Harry House, the second is 'Too Sweet' by album, and the third is 'Stand by me' by Mickey Gilley. Below the trending songs is a section titled 'Your Playlist'.

MusicBeats Home All Songs Search Welcome melvina_rajvee


Trending Songs



Kiwi
Movie: Harry House Album
Add to playlist Play now



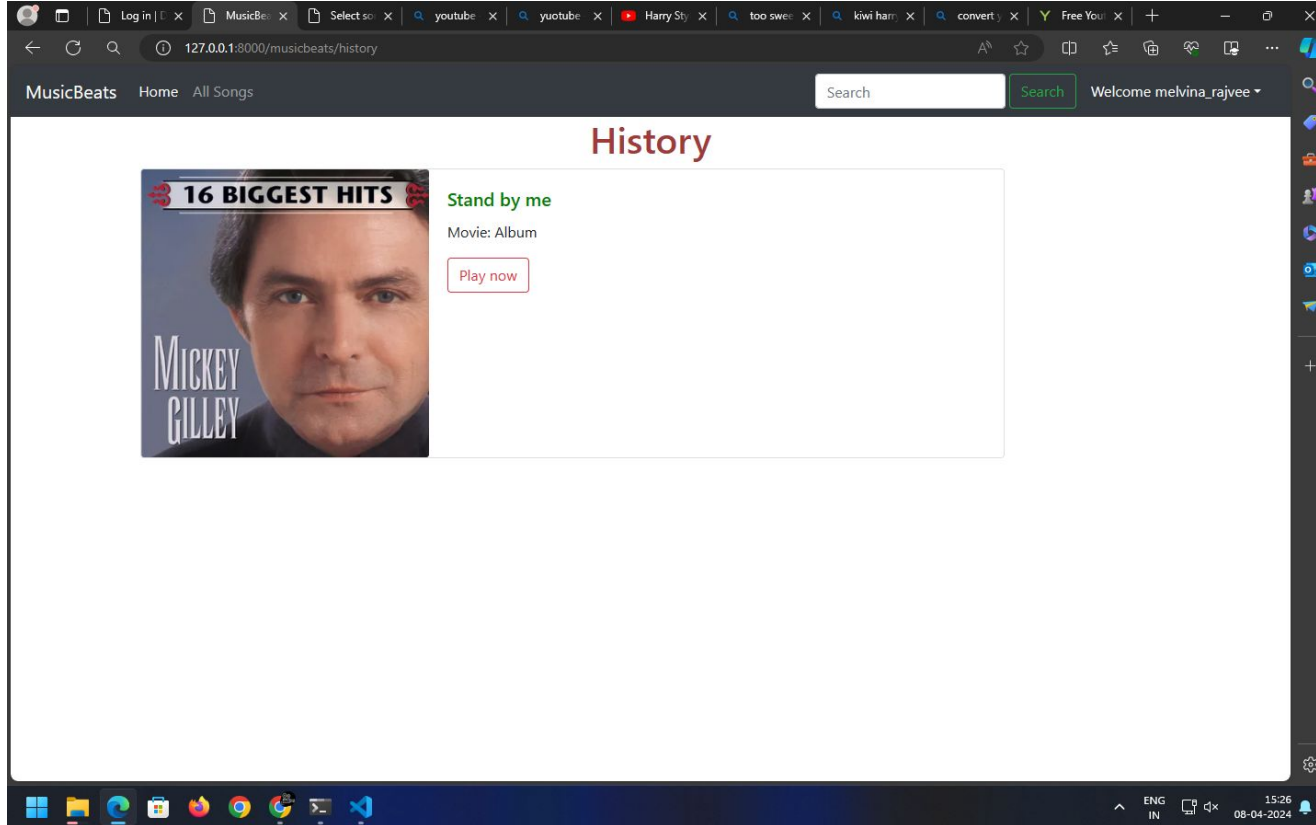
Too Sweet
Movie: album
Add to playlist Play now



Stand by me
Movie: Album
Add to playlist Play now

Your Playlist

History Page

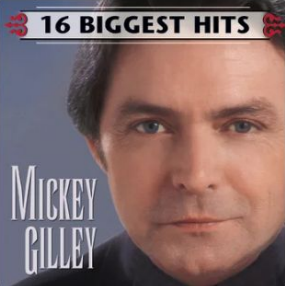


Log in | x MusicBeats x Select x youtube x youtube x Harry Sty x too swee x kiwi harr x convert x Free You x

127.0.0.1:8000/musicbeats/history

MusicBeats Home All Songs Search Search Welcome melvina_rajvee

History



16 BIGGEST HITS

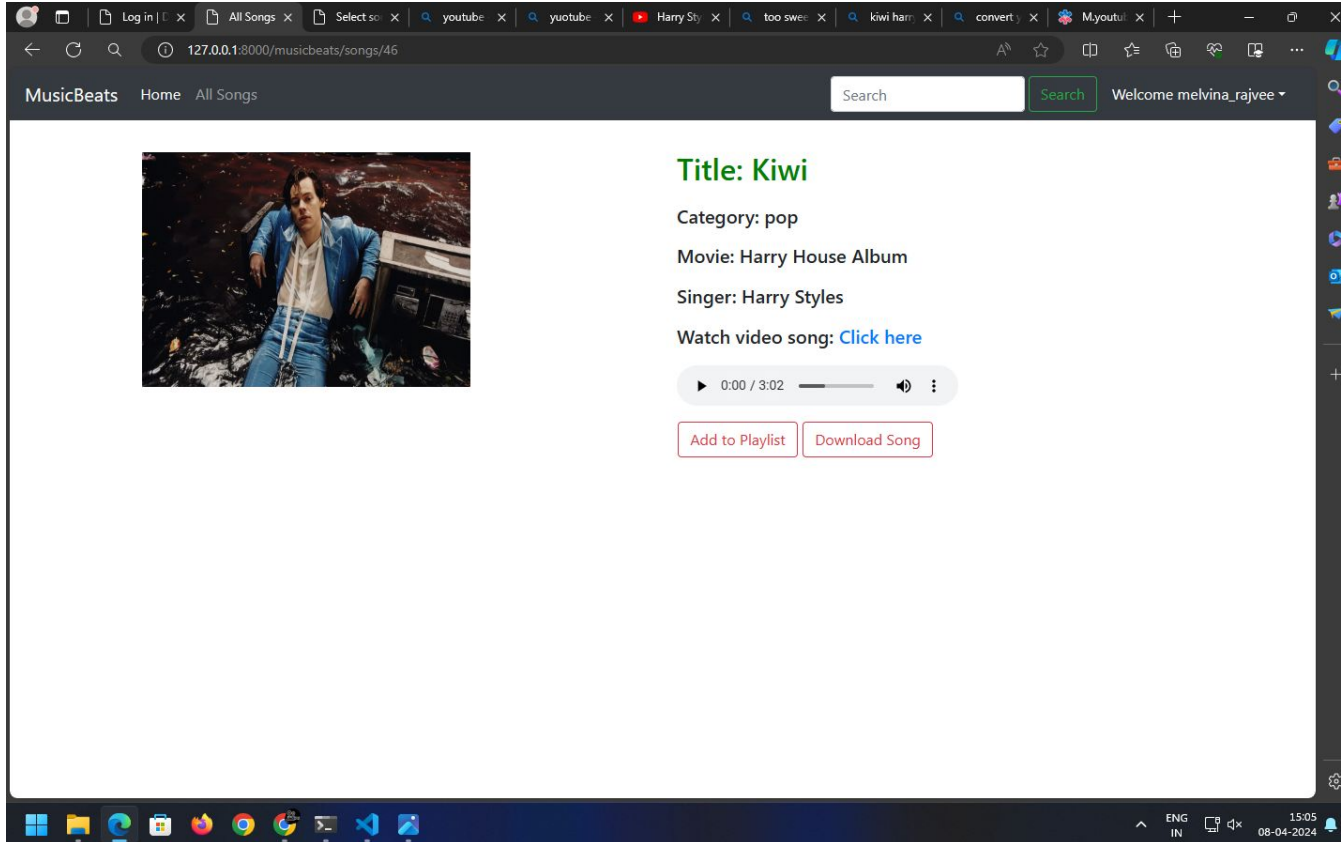
Stand by me

Movie: Album

Play now

Windows taskbar: ENG IN 15:26 08-04-2024

Audio Player Page



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "127.0.0.1:8000/musicbeats/songs/46". The browser's address bar shows the URL. The page content includes a navigation bar with "MusicBeats", "Home", and "All Songs" links. A search bar with a "Search" button is present. The main content area displays the song "Kiwi" by Harry Styles, categorized as "pop". It also lists the "Movie: Harry House Album" and provides a link to "Watch video song: Click here". Below the text is a video player interface showing a progress bar at 0:00 / 3:02. At the bottom of the page, there are two buttons: "Add to Playlist" and "Download Song". The Windows taskbar at the bottom shows various application icons and the system clock indicating 15:05 on 08-04-2024.

MusicBeats Home All Songs

Search

Welcome melvina_rajvee

Title: Kiwi

Category: pop

Movie: Harry House Album

Singer: Harry Styles

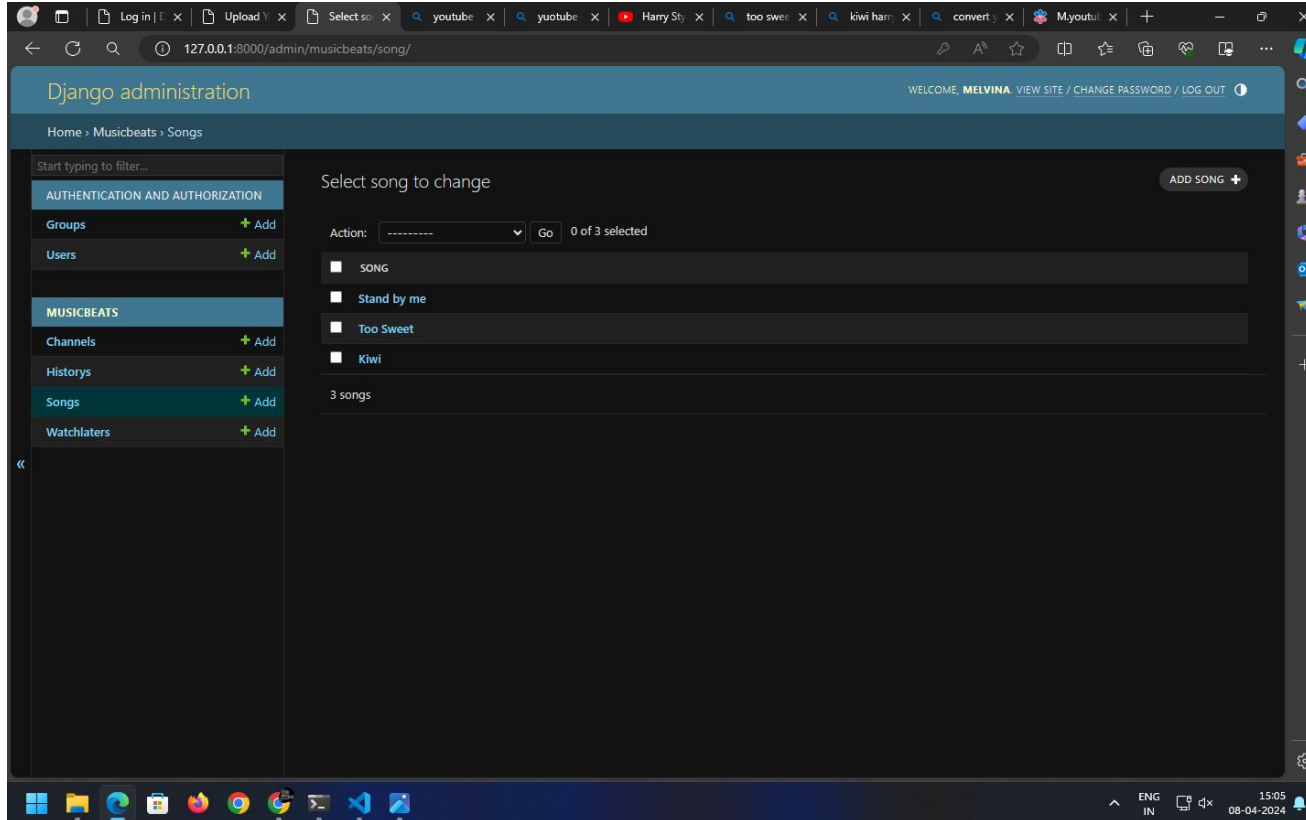
Watch video song: [Click here](#)

0:00 / 3:02

Add to Playlist Download Song

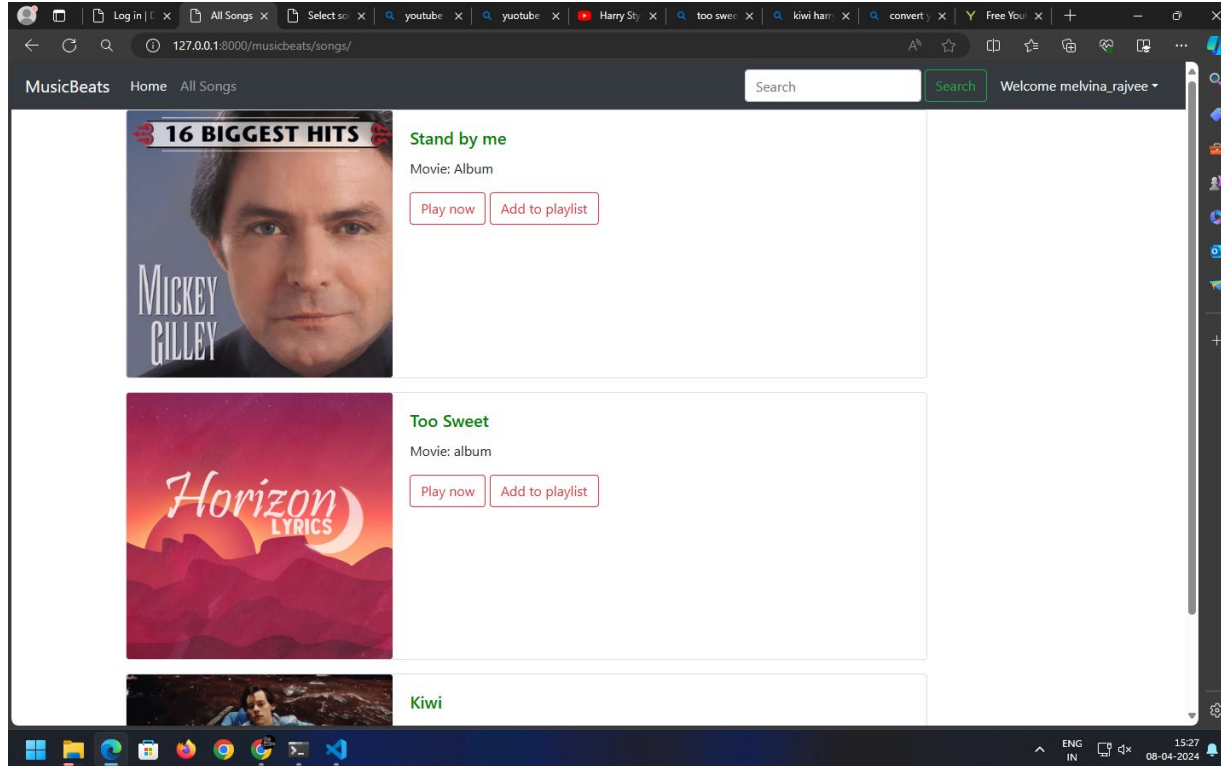
ENG IN 15:05 08-04-2024

Backend Admin Page



The screenshot displays the Django administration interface for the Musicbeats application. The browser's address bar shows the URL `127.0.0.1:8000/admin/musicbeats/song/`. The interface features a dark-themed sidebar on the left with a search bar and a list of menu items: **AUTHENTICATION AND AUTHORIZATION** (Groups, Users), **MUSICBEATS** (Channels, Historys, Songs, Watchlaters), and a back arrow. The main content area is titled "Select song to change" and includes an "ADD SONG +" button. Below this, there is an "Action:" dropdown menu and a "Go" button, followed by a list of three songs: **SONG**, **Stand by me**, and **Too Sweet**, each with a checkbox. At the bottom of the list, it says "3 songs". The top of the interface shows the "Django administration" header with a welcome message for "MELVINA" and links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 15:05 on 08-04-2024.

Songs List



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "127.0.0.1:8000/musicbeats/songs/". The page displays a music player interface with the following elements:

- Navigation Bar:** Includes "MusicBeats", "Home", "All Songs", a search bar, a "Search" button, and a welcome message "Welcome melvina_rajeev".
- Song Recommendations:**
 - 16 BIGGEST HITS:** A section featuring a portrait of Mickey Gilley. The song "Stand by me" is highlighted in green. Below it, it says "Movie: Album" and provides "Play now" and "Add to playlist" buttons.
 - Too Sweet:** A section featuring a sunset landscape with the text "Horizon LYRICS". The song "Too Sweet" is highlighted in green. Below it, it says "Movie: album" and provides "Play now" and "Add to playlist" buttons.
 - Kiwi:** A section featuring a portrait of a man. The song "Kiwi" is highlighted in green.
- Taskbar:** The Windows taskbar at the bottom shows various application icons and the system clock displaying "15:27" on "08-04-2024".

Future Enhancements:

In the future, we could implement features such as personalized recommendations based on listening history, social sharing of playlists, integration with external music APIs for expanded content access, and support for offline playback. We could also enhance the user interface with more interactive elements. Continuous performance optimization and scalability improvements could also be done for growing user base.

Conclusion:

In conclusion, this Django-based music streaming application presents a scalable and feature-rich platform for music enthusiasts. By leveraging Django's robust framework, I have developed a responsive and secure solution that prioritizes user experience. With planned future enhancements, we are committed to evolving this application to meet the dynamic demands of music streaming while maintaining a seamless and enjoyable experience for our users.

Thank You!