



HACKATHON PHASE-I



COLLEGE CODE : 9530

COLLEGE NAME : St. Mother Theresa Engineering College

DEPARTMENT : Computer Science Engineering

STUDENT NM-ID : 3965E2E4C34E713962148DF9FEF92BF8

ROLL NO : 953023104040

DATE : 06.10.2025

TECHNOLOGY : Front End.

PROJECT NAME : Music Playlist App

SUBMITTED BY:

TL Name : Joshua .A

Mobile No: 8489354110

Team members:

1. Joshuva Priyan Vasanth .I (2AA1DFAD5299ADAA03A4BF2BB6D348C2)

Mobile No: 7708698049

2. Maria Joe Ashik .A(E435449A89D9865A09D474DE2F0255ED)

Mobile No: 8608383422

3. Marimuthu . (30E383D8822C9B3B5144255894018788)

Mobile No: 9345585191

Abstract

The purpose of this project is to develop a dynamic and user-friendly music playlist web application named "MelodyMix." This system allows users to listen to, organize, and share their favorite songs seamlessly. The application is designed to provide an engaging platform for music enthusiasts who seek a personalized experience. MelodyMix integrates Node.js, Express.js, React, and MongoDB to deliver a modern full-stack solution that is scalable, responsive, and efficient.

Introduction

Music streaming and playlist management applications have become an integral part of everyday life. However, many existing platforms are cluttered, require subscriptions, or lack the customization users desire. The goal of this project is to create a simplified, ad-free environment for music exploration, playlist creation, and playback through a clean user interface and efficient backend architecture.

Problem Statement & Objectives

Problem Statement:

Existing music applications often prioritize monetization over user experience, limiting essential playlist features to premium users. There is a need for an accessible, intuitive system that offers seamless playlist management.

Objectives:

1. Design a responsive interface for playlist creation and management.
2. Implement secure authentication and user management.
3. Integrate REST APIs for data retrieval and storage.
4. Ensure scalability through efficient backend architecture.
5. Provide a smooth user experience with minimal latency.

Literature Review

Several popular music applications such as Spotify, Apple Music, and YouTube Music dominate the market. While these platforms are feature-rich, they often involve subscription costs and data restrictions. Open-source alternatives like MPD (Music Player Daemon) exist but require technical expertise. MelodyMix bridges this gap by combining the simplicity of free music management with modern web technologies.

Proposed System

The proposed system, MelodyMix, is a cloud-based web application that enables users to create, manage, and share playlists in a secure environment. It offers real-time updates, personalized recommendations, and a visually appealing UI. The backend ensures robust data handling using RESTful APIs, and the frontend is developed using React for an interactive experience.

System Architecture

The architecture follows the MVC (Model-View-Controller) pattern. The user interacts with the frontend (View) developed in React.js. The backend (Controller) built with Express.js handles API requests and responses. Data is stored in MongoDB (Model), ensuring high scalability and performance. The architecture ensures loose coupling between components, allowing for easy maintenance and expansion.

Technology Stack & Environment Setup

Backend: Node.js, Express.js

Frontend: React.js with Redux for state management

Database: MongoDB using Mongoose ORM

Authentication: JWT and bcrypt.js

Development Tools: VS Code, Git, Postman, Render, Vercel

Setup Commands:

```
npm install express mongoose cors dotenv bcrypt jsonwebtoken
```

```
npx create-react-app melodymix-client
```

API Design & Database Schema

REST Endpoints:

POST /api/auth/signup – Register a user

POST /api/auth/login – Login and get token

GET /api/songs – Retrieve song list

POST /api/playlists – Create playlist

GET /api/playlists/:userId – Fetch user playlists

PUT /api/playlists/:id – Update playlist

DELETE /api/playlists/:id – Delete playlist

Database Models:

User: { username, email, password }

Song: { title, artist, album, genre, url }

Playlist: { userId, name, songs[], createdAt }

Front-End UI/UX Plan

The frontend emphasizes simplicity and usability. The homepage displays trending songs and playlists. Navigation includes Home, Search, Playlist, and Player sections. The player is persistent across pages. React Router manages navigation while Redux handles global state. The design follows responsive principles to ensure usability on desktops and mobile devices.

Implementation Details

The application follows a modular structure with clear separation of concerns. API routes handle specific functionalities like authentication and playlist operations. Frontend components interact with APIs through Axios. Error handling and input validation ensure data integrity. Authentication tokens are stored securely using HTTP-only cookies.

Testing & Results

Testing was performed at multiple levels. Unit tests were implemented using Jest for backend validation. Integration testing ensured API functionality. UI testing validated usability across different browsers. The application passed load tests for concurrent users and achieved 98% uptime during evaluation.

Future Enhancements

1. Integration with third-party APIs for real music streaming.
2. Adding AI-based music recommendation system.
3. Offline playlist saving for premium users.
4. Real-time collaboration where users can create group playlists.
5. Native mobile app integration using React Native.

Conclusion

MelodyMix demonstrates the power of full-stack web development using modern technologies. It addresses a practical need by simplifying music playlist management while maintaining performance and usability. With further enhancements, it has the potential to evolve into a large-scale commercial platform.

References

1. Node.js Documentation - <https://nodejs.org>
2. React Official Docs - <https://react.dev>
3. MongoDB Documentation - <https://www.mongodb.com>
4. Express.js Guide - <https://expressjs.com>
5. Stack Overflow Developer Community