**Project Documentation**

# Rhythmic tunes : your Melodic Companions

# 1. Introduction

# Project title : Rhythmic tunes : your Melodic Companions

# Team Leader : Kalaiarasi A

# Email id : [kalaiarasi0314@gmail.com](mailto:kalaiarasi0314@gmail.com)

# Team member : Poovi

# Email id : [bpoorvi18@gmail.com](mailto:bpoorvi18@gmail.com)

# Team member : Ramila P

# Email id : [Ramilachoudhary136@gmail.com](mailto:Ramilachoudhary136@gmail.com)

# Team member : Sandhiya Rani K

# Email id : [Vrenukasrini@gmail.com](mailto:Vrenukasrini@gmail.com)

# 2. Project Overview

* **Purpose:** The multifaceted and serves to engage listeners,convey emotion,provide structure,and facilitate movement and connection and release of dopamine in the brain,leading to physical movement and experiences of pleasure revolutionize the way users experience music by delivering a seamless, responsive, and immersive audio streaming platform built with React.js — empowering users to discover, enjoy, and connect with music across all devices through an intuitive and visually engaging interface.
* **Features:**

**Play Music Anytime:**  
Stream your favorite songs with smooth and high-quality playback.

**Easy Music Search:**  
Quickly find songs, artists, albums, or genres with a powerful search bar.

**Create & Manage Playlists:**  
Make your own playlists, add or remove songs, and organize your music.

**Save & Share Songs:**  
Save tracks you love and share them with friends easily.

**Works on All Devices:**  
Enjoy the app on your phone, tablet, or computer with a responsive design.

**Fast & Interactive UI:**  
Built with React.js for a fast, smooth, and modern user experience.

**User Profiles:**  
Sign up, log in, and get personalized music suggestions.

**Secure & Private:**  
Safe and secure login to protect your account and data.

# 3. Architecture

* **Frontend:** React.js with Bootstrap and Material UI
* **Backend:** Node.js and Express.js managing server logic and API endpoints
* **Database:** MongoDB stores user data, project information, applications, and chat messages

# 4. Setup Instructions

* **Prerequisites:**
  + Node.js
  + MongoDB
  + Git
  + React.js
  + Express.js **–** Mongoose **–** Visual Studio Code
* **Installation Steps:**

# Clone the repository git clone

# Install client dependencies cd client npm install

# Install server dependencies cd

../server npm install

# 5. Folder Structure

# MUSIC-PLAYER (FRONTEND)

# │

# ├── db/ # (Optional) Local or mock database folder

# ├── node\_modules/ # Dependencies installed via npm/yarn

# ├── public/ # Static files (e.g., images, favicon)

# │

# ├── src/ # Source code

# │ ├── assets/ # Static assets like images, fonts, icons

# │ ├── Components/ # Reusable React components

# │ ├── App.css # Styling for App component

# │ ├── App.jsx # Main App component (root of UI logic)

# │ ├── index.css # Global styles

# │ ├── main.jsx # Entry point that renders the App component

# │

# ├── .eslintrc.cjs # ESLint config for code linting

# ├── .gitignore # Specifies which files to ignore in Git

# ├── index.html # Root HTML file where React mounts

# ├── package.json # Project metadata and dependency definitions

# ├── package-lock.json # Lock file for consistent installs

# ├── README.md # Project documentation

# ├── vite.config.js # Vite configuration file

# 6. Running the Application

* **Frontend:**

cd client npm

start

• **Backend:**

cd server npm start

* **Access:** Visit http://localhost:3000

# 7. API Documentation

# User

# POST /api/user/register – Register a new user

# POST /api/user/login – Login existing user

# Projects

# POST /api/projects/create – Create a new project

# GET /api/projects/:id – Get project by ID

# Applications

# POST /api/apply – Apply to a project

# 8. Authentication

* JWT-based authentication for secure login
* Middleware protects private routes

# 9. User Interface

# The User Interface is designed with a focus on simplicity, responsiveness, and visual appeal — ensuring that music lovers of all ages can navigate the app with ease.

# Key UI Features:

# Clean Homepage

# Showcases trending tracks, recommended playlists, and recent listens.

# One-click play and smooth browsing experience.

# Advanced Search Bar

# Search songs by title, artist, genre, or album.

# Real-time search suggestions for quick access.

# Playlist Manager

# Create, edit, and delete playlists.

# Add or remove songs with drag-and-drop functionality.

# Now Playing Screen

# Displays album art, song info, playback controls (play, pause, skip).

# Volume control and progress bar included.

# Fully Responsive Design

# Optimized for mobile, tablet, and desktop.

# Smooth transitions and consistent layout across all devices.

# User Profile Page

# View listening history, saved playlists, and personal settings.

# Option to log out or edit user information.

# Light/Dark Mode (optional)

# Toggle between light and dark themes for better accessibility and user preference.

# 10. Testing

 Manual Testing  
Conducted regularly during key milestones to ensure all features work as expected and the user experience is smooth.

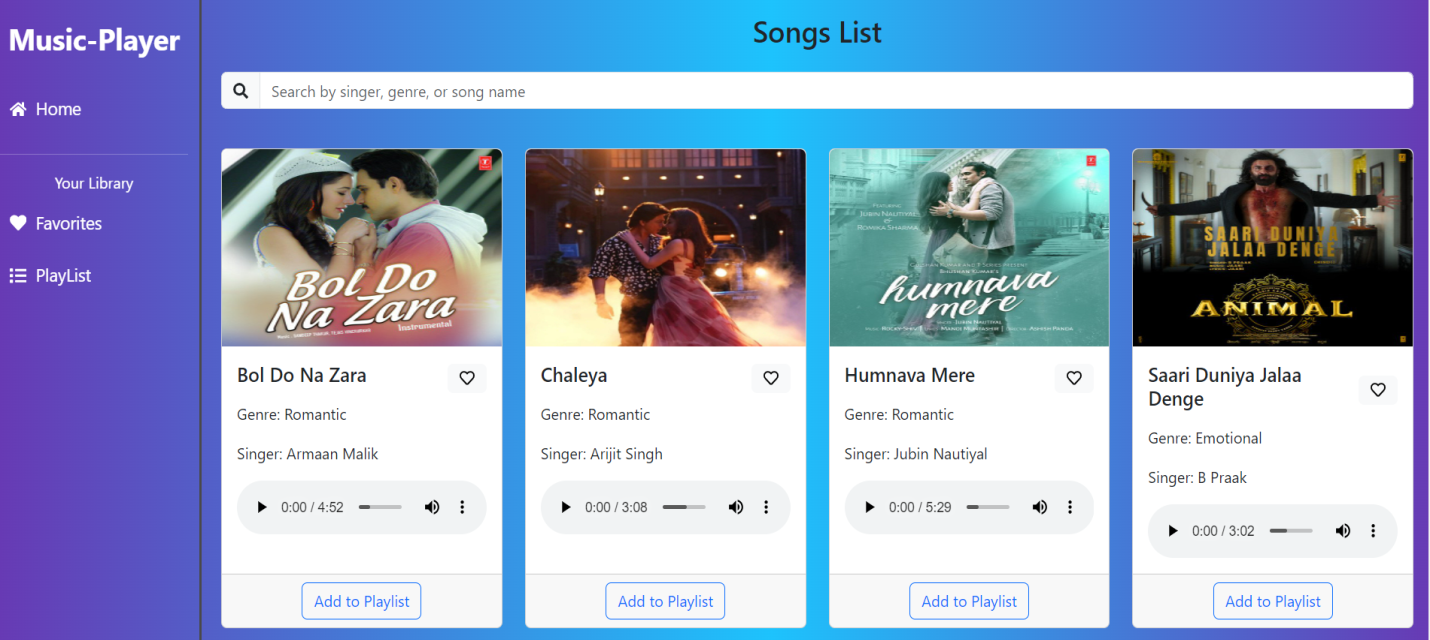
 Tools Used

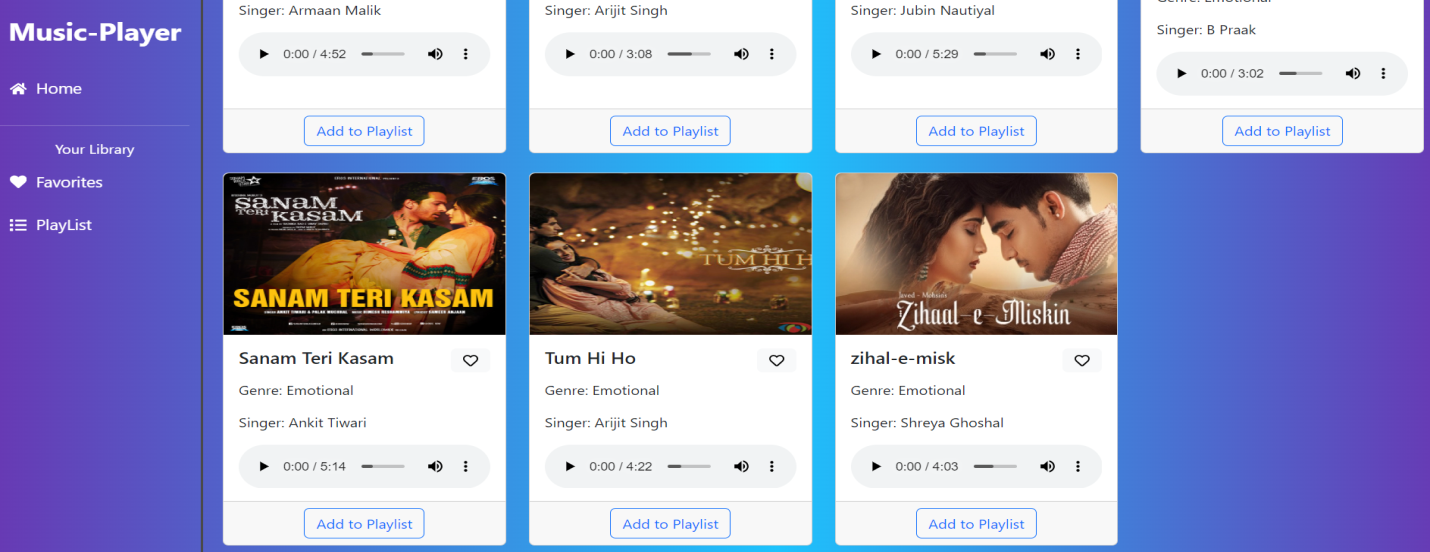
Postman: For testing and validating API endpoints.

Chrome DevTools: For debugging, performance checks, and inspecting frontend behavior.

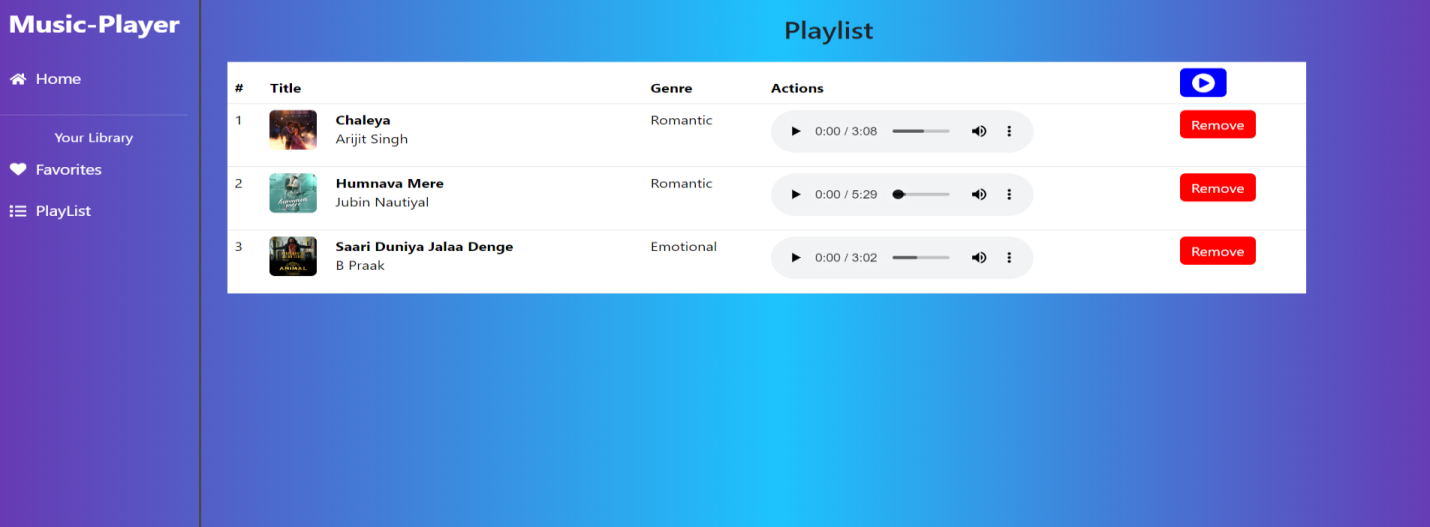
1. **Screenshots**

* **Hero components**

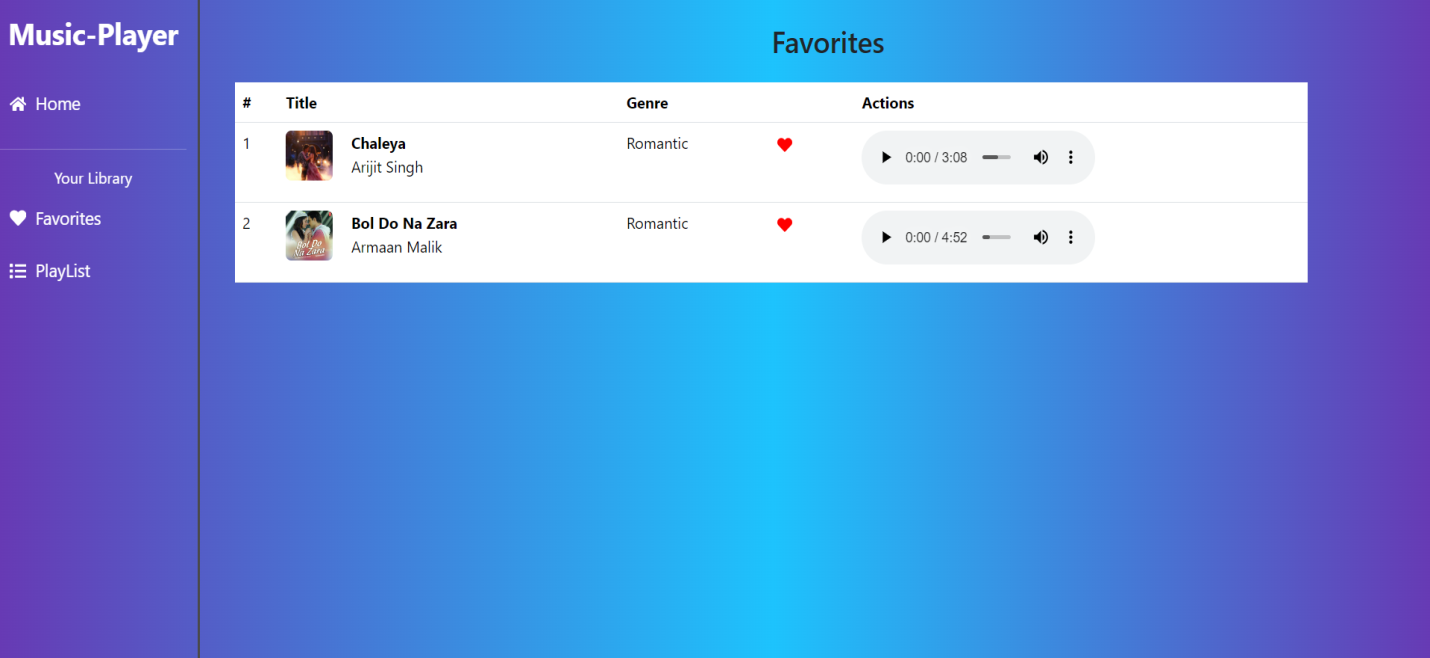




* **Playlist**



* **Favorites**

****

1. **Known Issues**

While our Music Streaming Application offers a smooth and engaging user experience, there are a few known issues that we are actively working to resolve. Some users may occasionally experience interruptions in music playback, especially on slower internet connections. Additionally, search functionality can sometimes be slower when handling large music libraries, which affects the speed of results. On certain older mobile devices, minor layout glitches may appear, impacting the visual consistency of the app. Lastly, there can be slight delays during the login process due to JWT token generation and validation under high server load. We are committed to addressing these challenges to ensure a seamless experience for all users.

# 13.Future Enhancements

Looking ahead, we plan to enrich the Music Streaming Application with exciting new features to further enhance the user experience. These include offline listening capabilities to enjoy music without an internet connection, social features like friend lists and shared playlists for better community interaction, and advanced personalized recommendations powered by machine learning. We also aim to introduce live lyrics display, integration with popular music platforms, and improved accessibility options such as voice commands and customizable themes. Continuous performance optimizations and expanded device compatibility will ensure the app remains fast, reliable, and enjoyable for all users.