RHYTHMICTUNES : YOUR MELODIC COMPANION

(MUSIC STREAMING APPLICATION)

NAANMUDHALVAN PROJECT REPORT

Submitted by

D. DEEPESHA	**222209354**
S. DIVYA	**222209356**
K. ERUCHAMMAL	**222209358**
A. HARINI	**222209359**

DEPARTMENT OF COMPUTER SCIENCE

TAGORE COLLEGE OF ARTS AND SCIENCE

(Affiliated to the University of Madras)

CLC WORKS ROAD, CHROMPET, CHENNAI – 600 044

APRIL – 2025

ABSTRACT

Rhythmic Tunes Your Melodic Companion is a React-based music streaming application designed to offer an unparalleled audio designed to offer an unparalleled audio experience. With a focus on user-centric design and seamless performance, the app allows users to discover, stream, and manage music effortlessly. By combining a modern tech stack with an intuitive UI, Rhythmic Tunes aims to redefine how users connect with music.

By combining a user-friendly interface with efficient data handling and secure authentication, Rhythmic Tunes aims to deliver an engaging and reliable platform for music enthusiasts. Future enhancements include AI-based recommendations and social media integration to enrich user experience further.

INTRODUCTION

Rhythmic Tunes is a React-based Music streaming application designed to transform how users discover, play, and manage their favourite songs. With a focus on seamless performance and user-centric design, the app offers a rich set of features, including playlist creation, playback control, offline listening, and an advanced search function.

Built using React.js for the frontend, Node.js for backend APIs, and MongoDB for data storage, Rhythmic Tunes ensures a smooth and responsive user experience. The platform is designed for a diverse audience, from music enthusiasts seeking personalized recommendations to casual listeners looking for a simple and effective way to explore music.

The primary goal of Rhythmic Tunes is to provide a secure, efficient, and enjoyable music streaming experience that adapts to users' preferences. By leveraging modern web technologies and a well-organized architecture, the app promises to set a new standard for music streaming.

PROJECT OBJECTIVES

The main objective of Rhythmic Tunes is to develop a user-friendly music streaming platform that offers a seamless and personalized listening experience. Built with React.js for the frontend and Node.js for the backend, the project aims to provide efficient and secure management of user data through MongoDB. Key features include secure user authentication, playlist creation, playback control, offline listening, and a powerful search function for quickly finding songs, artists, and albums. The platform is designed to be intuitive and responsive, ensuring smooth navigation and interaction. Additionally, Rhythmic Tunes is built with future enhancements in mind, such as AI-based recommendations and social media sharing, to offer a more engaging and interactive music experience.

- To develop a user-friendly music streaming platform using React.js for the frontend and Node.js for the backend.
- To implement secure user authentication for protecting user data and enhancing privacy.
- To enable seamless music playback and management with features like playlist creation, playback control, and offline listening.
- To integrate a powerful search functionality for quickly finding songs, artists, and albums.
- To ensure efficient data storage and retrieval using MongoDB for managing user profiles, songs, and playlists.
- To create an intuitive and responsive UI with React components and CSS for a smooth user experience.
- To support future enhancements such as AI-based recommendations and social media sharing for a personalized and interactive experience.

KEY FEATURES

- **Song Listings:** Displays a comprehensive list of songs with title, artist, genre, and release date for easy browsing.
- **Playlist Creation:** Allows users to create, edit, and manage personalized playlists effortlessly.
- **Playback Control:** Provides seamless play, pause, skip, and volume control options for an uninterrupted music experience.
- **Offline Listening:** Enables users to download songs for offline playback, enhancing accessibility and convenience.
- **Search Functionality:** Integrates a powerful search feature to quickly find songs, artists, or albums.
- **User Authentication:** Implements secure login and registration to protect user data.
- **Responsive UI:** Built with React.js and styled using CSS and Bootstrap for a smooth and consistent user experience across devices.
- **Favourites and Wishlist:** Allows users to mark songs as favourites and manage a wishlist for easy access.
- **Admin Panel:** Provides tools for managing users and content effectively.
- **Scalability:** Designed to support future enhancements like AI-based recommendations and social media integration.

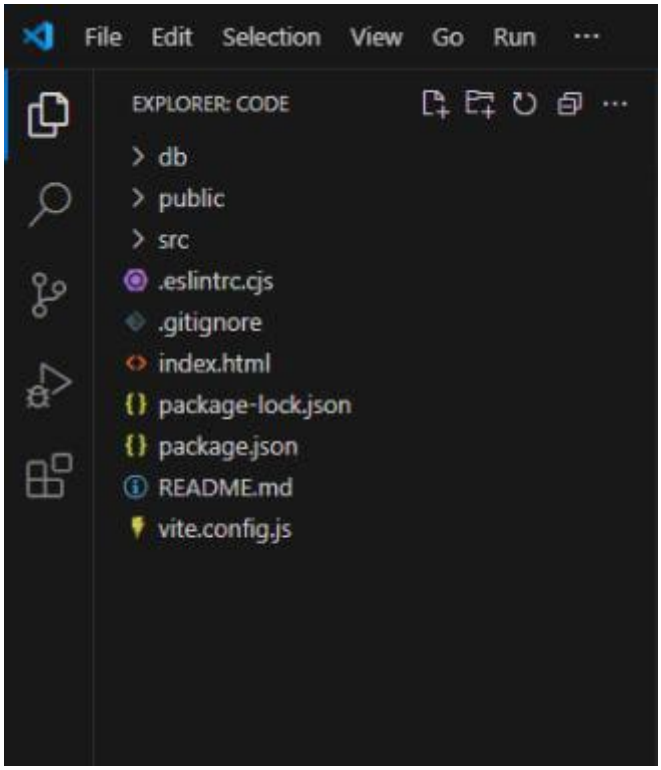
These features aim to make Rhythmic Tunes a secure, customizable, and engaging music streaming platform.

TECHNOLOGIES USED

Rhythmic Tunes is built using a modern tech stack to ensure a smooth and efficient music streaming experience. The frontend is developed with React.js, providing a responsive and interactive user interface, while CSS and Bootstrap are used for styling and a clean design. On the backend, Node.js and Express.js handle server-side logic and API requests, ensuring efficient data management and communication between the client and server. MongoDB serves as the database, storing user profiles, songs, and playlists securely and efficiently.

For API integration, Axios is used to fetch and update data seamlessly. To enhance security, the app employs JSON Web Tokens (JWT) for user authentication and password hashing for protecting credentials. Development is managed with Node Package Manager (npm) for dependencies and Git and GitHub for version control and collaboration. Postman is used for testing APIs, ensuring reliable data handling. This combination of technologies makes Rhythmic Tunes a secure, scalable, and user-friendly music streaming platform.

PROJECT FLOW



The project structure may vary depending on the specific library, framework, programming language, or development approach used. It's essential to organize the files and directories in a logical and consistent manner to improve code maintainability and collaboration among developers.

app/app.component.css, src/app/app.component: These files are part of the main AppComponent, which serves as the root component for the React app. The component handles the overall layout and includes the router outlet for loading different components based on the current route.

MODULES DESCRIPTION

User Authentication Module:

This module handles user registration, login, and authentication using JSON Web Tokens (JWT). It ensures secure access by hashing passwords and managing sessions efficiently. Users can create accounts, log in securely, and access their personalized playlists and settings.

Music Upload and Storage Module:

This module allows users to upload audio files with metadata such as title, artist, and genre. Uploaded files are stored securely in the MongoDB database along with their information. It also supports efficient retrieval and management of music files for seamless playback.

Music Streaming and Playback Module:

Responsible for real-time audio streaming, this module uses React components to provide a smooth playback experience. It includes play, pause, skip, and volume control options, ensuring an uninterrupted listening experience.

Playlist Management Module:

This module allows users to create, edit, and delete playlists. Users can add songs to their playlists, rearrange tracks, and organize their music efficiently. The playlists are stored in MongoDB and can be accessed or modified anytime.

Search and Discovery Module:

Integrates a robust search functionality to help users find songs, artists, or albums quickly. It uses filters and keywords to deliver accurate search results, enhancing user experience and music discovery.

Favourites and Wishlist Module:

Allows users to mark songs as favourites and manage a wishlist for quick access to their preferred tracks. This module saves favourite songs in the database, making it easy for users to revisit their top tracks.

Analytics and Recommendation Module:

Aims to integrate AI-based song recommendations by analyzing user listening patterns. This module will help in offering personalized suggestions and enhancing user engagement.

TESTING AND DEBUGGING

Rhythmic Tunes follows a comprehensive testing and debugging strategy to ensure smooth performance and reliability. The testing process includes unit testing, API testing, UI testing, and security testing to identify and fix issues efficiently.

1. **UNIT TESTING**

   Conducted using Jest and React Testing Library to test individual React components like Login, SearchBar, and Player.

   Verifies if components render correctly and handle user interactions as expected.

2. **API TESTING**

   Performed with Postman to test authentication, playlist management and music upload APIs.

3. **UI TESTING**

   Focuses on form validation, navigation, and responsiveness across devices.

   Uses React Dev Tools and manual testing to check the alignment and functionality of UI elements.

4. **DEBUGGING TECHNIQUES**

   Consoles logs for tracking data flow and identifying rutime errors.

   Browser Developer Tools for inspecting API requests and resolving JavaScript errors.

5. **SECURITY TESTING**

   Implements password hashing and JWT for authentication to prevent unauthorized access.

6. **PERFORMANCE TESTING**

   Uses Chrome Lighthouse to evaluate page load speed and accessibility.

   Optimizing API response times and removes unused CSS/JavaScript.

CONCLUSION

RhythmicTunes successfully combines React.js for the frontend, Node.js for the backend, and MongoDB for data management to deliver a seamless and user-friendly music streaming experience. The application's features, such as playlist creation, playback control, offline listening, and secure authentication, provide users with a comprehensive platform to explore and enjoy music effortlessly.

The project's modular architecture and the use of modern technologies ensure scalability, security, and efficient performance. Through comprehensive testing and debugging, potential issues were identified and resolved, enhancing the overall reliability of the application

In the future, RhythmicTunes aims to incorporate AI-based song recommendations, social media integration, and advanced analytics to further personalize andenrich the user experience. Overall, the project achieves its goal of offering a secure, responsive, and engaging music streaming platform for diverse audiences.