Project Documentation
Rythamic Tunes

1.INTRODUCTION

- **Project Title:** Rythamic Tunes
- **Team ID:** NM2025TMID38595
- **Team Leader:** Rajeshwari G& Mail ID:@gmail.com
- **Team Members:**
  - Manisha R & Mail ID:manisharamamoorthy20@gmail.com
  - Jenifer christiya S & Mail ID: @gmail.com
  - Saranya M & Mail ID: @gmail.com
  - Phebe Lois D & Mail ID:

2.PROJECT OVERVIEW

Purpose:

Rythmic Tunes is a digital platform designed to connect music enthusiasts, learners, and creators. It enables users to access, share, and collaborate on musical content in an interactive way. The platform provides features like song uploads, playlists, collaborations, and real-time engagement to create a vibrant musical community.

Features:

♪ Core Features (For Users)

- Music Upload & Sharing – Users can upload their songs, compositions, or recordings.
- Playlist Creation – Create personal playlists or collaborative playlists with friends.

- Real-time Collaboration – Musicians can work together on tracks (compose, remix).
- Streaming & Download – Listen online or download tracks (based on permissions).
- Search & Filter – Find songs, artists, or genres easily.
- Secure Chat System – Built-in chat for artists, fans, and collaborators.
- Feedback & Reviews – Users can rate songs and provide constructive feedback.

👨‍💼 Admin Features

- Admin Dashboard – Central control panel for managing users and content.
- Content Moderation – Approve/reject uploaded tracks and handle reports.
- Analytics – Track most played songs, top artists, and user activity.
- User Management – Add, remove, or block users if necessary

3.ARCHITECTURE

- **Component structure:**
  The component structure of Rythamic Tunes follows a modular and reusable approach, making the app easy to maintain and scale. Each UI feature is built as an independent component.

Main Components:

1. NavbarComponent – Handles app navigation.
2. HeroBanner – Displays the app's main intro section.
3. TrendingTracks – Shows popular songs or albums.
4. GenresSection – Lists different music genres.
5. NowPlayingBar – Sticky music player with controls.
6. PlaylistPanel – Displays and manages user playlists.
7. MusicCard – Reusable card for individual songs/albums.
8. Footer – Contains links and app info.

These components communicate via props and state, ensuring smooth interaction and a dynamic user interface.

- **State Management:**

  The statement structure defines the logical flow and core functions of the application. It helps in understanding how the app processes data and responds to user actions.

Key Statement Areas:

1. Input Statements
   - Handles user interactions like clicking play, selecting songs, or choosing genres.

2. Conditional Statements
   - Used to control behavior, e.g., if a song is playing, show pause button; else, show play.

3. Looping Statements
   - For rendering multiple songs, playlists, or genres dynamically.

4. Function Calls
   - Reusable functions for play, pause, skip, volume control, and navigation.

   This structured approach improves readability, reusability, and simplifies debugging and maintenance.

- **Routing:**

  Routing in Rythamic Tunes is handled using React Router to enable smooth navigation between different pages without reloading the app.

Main Routes:

1. / – Home page with banner, trending tracks, and genres
2. /playlists – Displays user-created or featured playlists
3. /genres – Shows music categorized by genres
4. /about – App information and developer credits
5. /track/:id – Dynamic route to display details and play a specific song

Routing helps keep the app organized and allows users to easily explore different sections with a seamless experience**.**

4.SETUP INSTRUCTIONS

α. Prerequisites:

- Node.js & npm
- React.js
- VS code
- Git

β. Installation Steps:

Clone the repository git clone

- Git clone <your-repo-url>
- Cd insightstream
- Npm install
- Npm start

5.FOLDER STRUCTURE

|--db/

|__db.json/

|--node_modules/ # all the node modules installed

|code/

|--public/

|__songs/

|__vite.svg/

|__app.css/

|__app.jsx/

|__index.css/

|__main.jsx/

|--.eslintrc.cjs/

|--gitignore/

|--index.html/

|--package-lock.json/

|--package.json/

|--README.md/

|--vite.config.js/

6.RUNNING THE APPLICATION

χ. **Install Dependancies:**

i. npm install

δ. **Start the app:**

i. npm start

ε. **Access:**

i. Visit http://localhost:5173

7. STATE MANAGEMENT

Rythamic Tunes uses React's useState and useContext for managing the application's state efficiently.

Key Areas of State Management:

1. Current Track State
- Stores the song currently being played (title, artist, audio URL).

2. Playback Control
- Manages play, pause, skip, and progress bar states.

3. Playlist Data
- Tracks user-selected playlists and dynamically updates them.

4. Theme or UI State
- Maintains settings like dark mode, volume level, or view preferences.

Using React's built-in state tools ensures real-time updates and smooth in.

8. OMPONENT DOCUMENTATION:

. Navbar Component:

- Displays navigation links (Home, Genres, Playlists, About).
- Visible on all pages.

2. HeroBanner:

- Landing section with app title and quick links.
- Attracts user attention with visuals.

3. TrendingTracks

- Shows a list of popular or latest songs.
- Dynamically fetches and displays tracks.

4. GenresSection

- Displays various music genres in card format.
- Allows users to explore songs by type.

5. NowPlayingBar

- Fixed bottom player showing current track and controls (play, pause, skip).
- Updates dynamically with each song change.

6. PlaylistPanel

- Displays user-created or featured playlists.
- Allows adding or removing songs.

Each component is styled separately and communicates with others using props and state for a smooth user experience

9. USER INTERFACE

α. Now Playing Bar

i. A fixed bottom player showing the current song, play/pause, next/previous, and progress bar.

ii. Ensures users can control music from any page.

β. Genres Section

i. Displays various music genres with icons or images.

ii. Helps users quickly explore and discover songs by their preferred style.

10.STYLING

χ. CSS Modules / Styled Components: For modular and reusable styles across components.

δ. Responsive Design: Ensures compatibility across devices (mobile, tablet, desktop).

ε. Dark Mode Theme: Provides a smooth visual experience ideal for music apps.

ϕ. Animations & Transitions: Used for smooth playback effects, hover actions, and interactive UI feedback.

11.TESTING

To ensure the reliability and smooth performance of Rythamic Tunes, the following testing methods were applied:

Unit Testing: Individual components like the music player, playlist, and controls were tested using Jest to verify their functionality

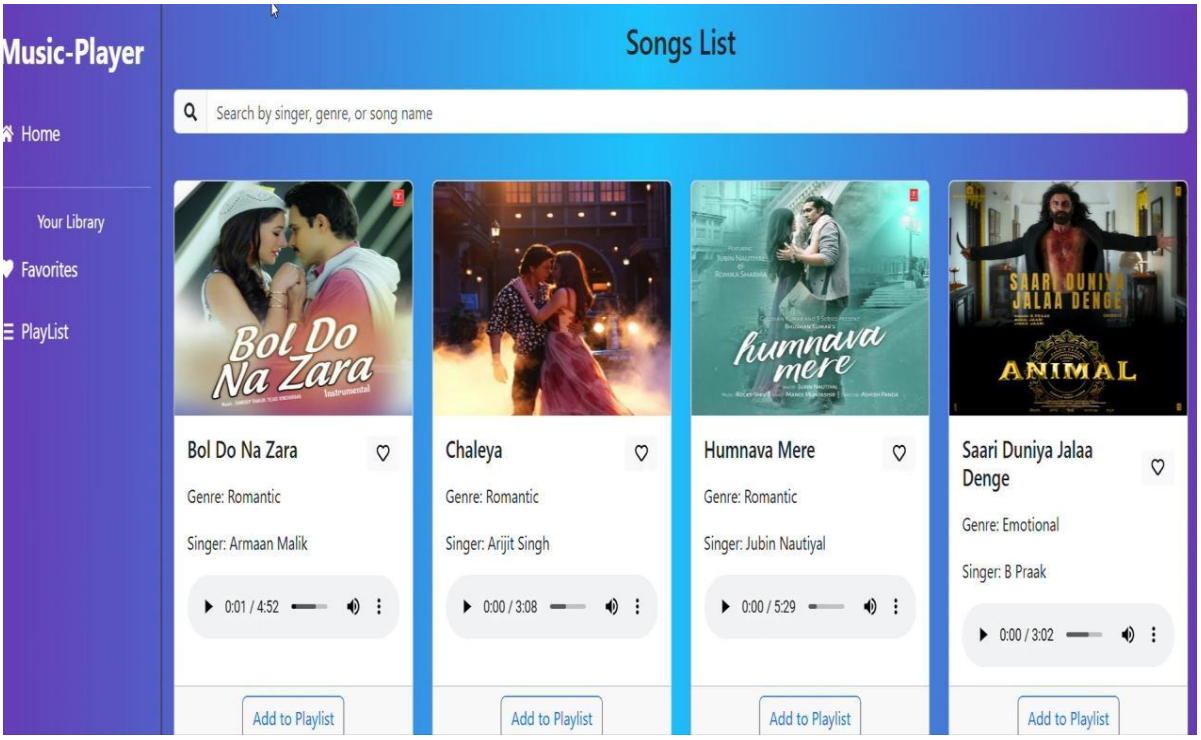
γ. **UI Testing**: User interactions such as play/pause, volume control, and navigation were tested with tools like React Testing Library.

η. **Manual Testing**: The application was manually tested on various devices and browsers to ensure responsiveness and compatibility.

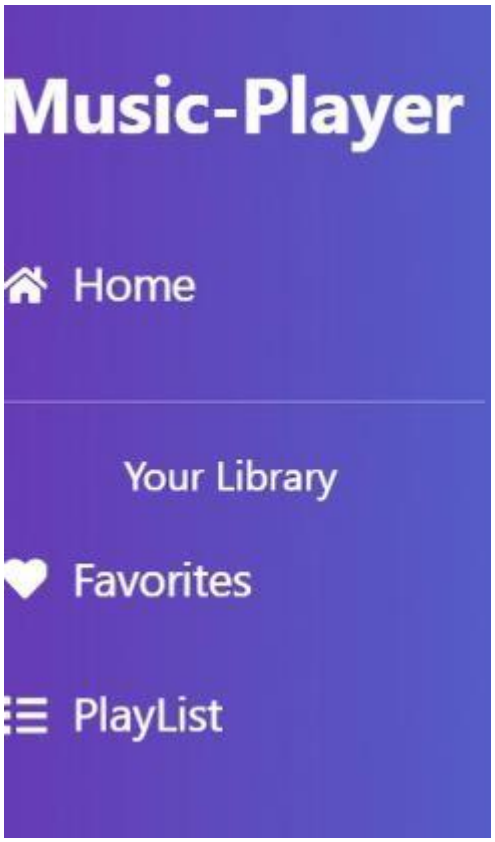
ι. **Bug Fixing & Debugging**: Console monitoring and error handling were done throughout the development process.

12.SCREENSHOTS

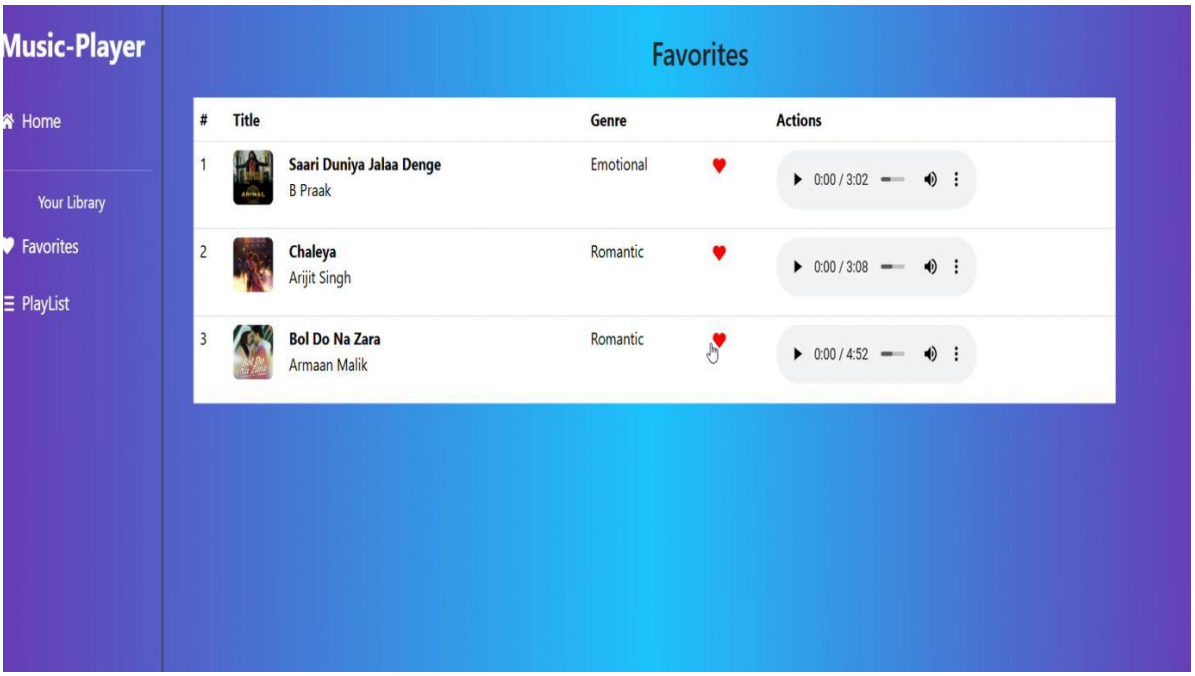
HOME PAGE



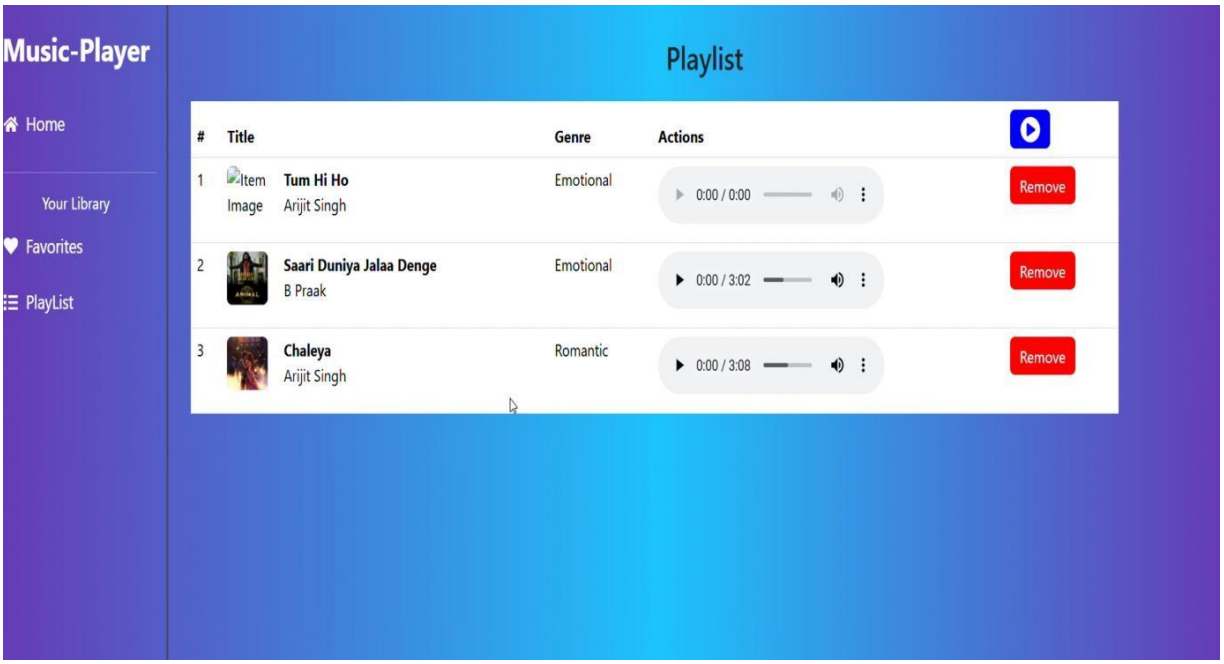
NAV BAR



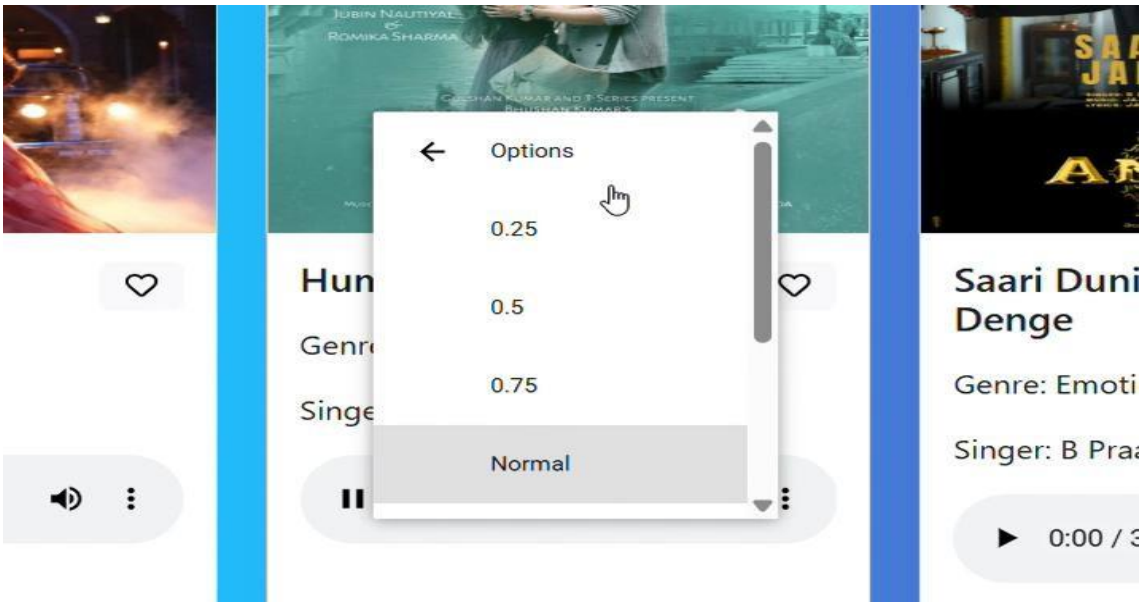
FAVORITES

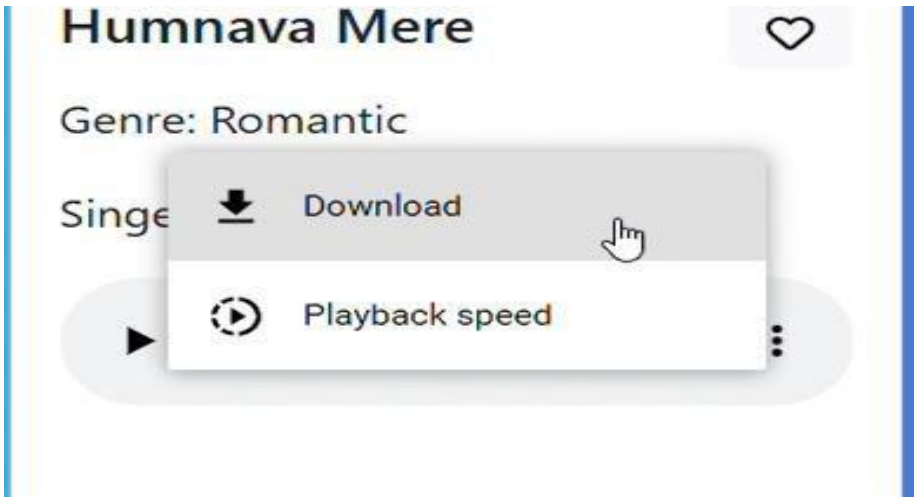


PLAYLIST

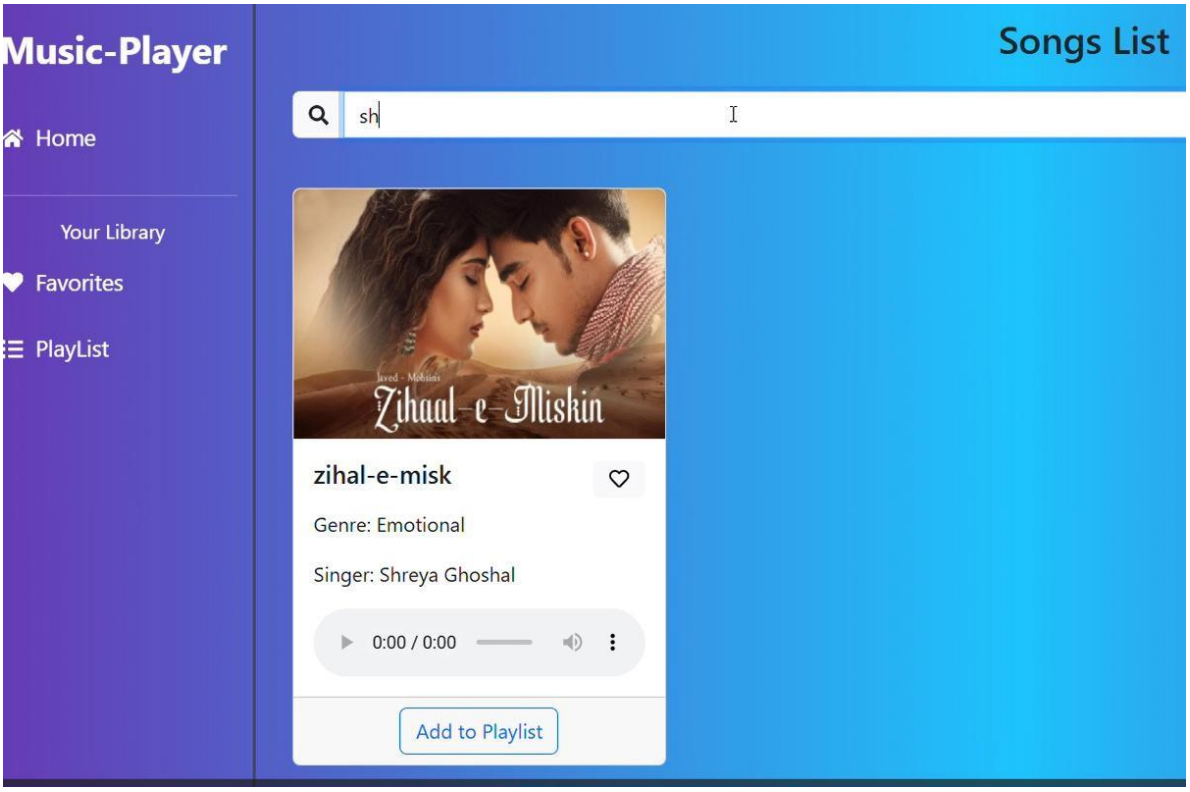


OPTIONS




Humnava Mere
Genre: Romantic
Singe
Download
Playback speed

SEARCHBAR


Music-Player
Songs List
Home
sh
Your Library
Favorites
PlayList
Zihaal-e-Miskin
zihal-e-misk
Genre: Emotional
Singer: Shreya Ghoshal
0:00 / 0:00
Add to Playlist

13. KNOWN ISSUES

- **Audio Playback Delay** :Slight delay occurs when switching between tracks, affecting smooth playback.
- **No User Login System :**Users can't save playlists permanently since authentication is not implemented**.**

14. FUTURE ENHANCEMENT

α. User Authentication – To save playlists and preferences permanently.

β. Advanced Recommendation System – AI-based personalized song suggestions.

Offline Mode – Download music for listening without internet.