Project Documentation

Project Title

Rhythmic Tunes: Your Melodic Companion

1. Introduction

- Project Title: Rhythmic Tunes: Your Melodic Companion
- Team ID: NM2025TMID30482
- Team Leader: ANUSRI B M (Email ID: anusribm07@gmail.com)
- Team Members:
  - NARMATHA A (Email ID: sweetrawdygirl7@gmail.com)
  - JANANI S (Email ID: jananijanani143207@gmail.com)
  - PREMA K (Email ID: premasudha082@gmail.com)

2. Project Overview

Purpose:

Rhythmic Tunes: Your Melodic Companion is a music streaming platform designed to provide a seamless and enjoyable listening experience. It allows users to discover, enjoy, and share diverse musical content while offering a user-friendly interface and smooth navigation.

Target Audience:

- **Music Enthusiasts:** Individuals passionate about listening to music during their free time to relax, enjoy, and explore melodies.
- **Casual Listeners:** Users who want quick access to trending and favorite tracks.
- **Playlist Lovers:** People who enjoy curating and sharing playlists.

Project Objectives:

- **User-Friendly Interface:** Develop an intuitive interface that allows users to effortlessly explore, save, and share their favorite tracks and playlists.
- **Comprehensive Music Streaming:** Provide robust features for organizing and managing music content, including advanced search options for easy discovery.
- **Modern Tech Stack:** Harness cutting-edge web technologies (React.js, etc.) to ensure an efficient and enjoyable user experience while navigating and interacting with the music streaming platform.

Key Features:

- Music streaming (play, pause, skip)
- Playlist creation and management

- Advanced search functionality
- Personalized recommendations
- User account management (register, login)
- Admin panel for managing content

3. Architecture

- **Frontend:**
  Developed using **React.js** with **Vite** for fast development and build optimization. Styling handled with **CSS** (can also add Bootstrap/Material UI if used).
- **Backend:**
  Powered by **Node.js** and **Express.js**, responsible for handling API requests, authentication, and interaction with the database.
- **Database:**
  MongoDB stores user details, playlists, songs metadata, and preferences.

4. Setup Instructions

Prerequisites

Here are the key prerequisites for developing and running the Rhythmic Tunes frontend application:

- **Node.js and npm:**
- **React.js (with Vite.js for faster builds)**
- **JSON Server**
- **Git**
- **HTML, CSS, JavaScript (basic knowledge)**
- **Code Editor (Visual Studio Code recommended)**

Installation Steps

1.Clone the repository

git clone < GitHub - anusri-07/RhythmicTunes-Your-Melodic-Companion >

cd RhythmicTunes

2.Install dependencies

npm install

3.Start the development server

npm start

4.Start the React App

npm run dev

5.Access the app

Open your web browser and visit: http://localhost:3000

5. Folder Structure

```
RHTYMICTUNES/              # Project root directory
| -- code/                 # Project-related code
|    └──── db/             # Database-related files (if any)
|    └──── node_modules/   # Dependencies for backend/DB
|    └──── public/         # Public assets (images, icons, etc.)
|    └──── Songs/          # Songs or audio file storage
|    └──── vite.svg        # Vite logo (default asset)
|
| -- src/                  # Source code for React frontend
|    └──── assets/         # Static assets (images, fonts, etc.)
|    └──── Components/     # React components
|    └──── App.css         # Main CSS file
|    └──── App.jsx         # Root component
|    └──── index.css       # Global styles
|    └──── main.jsx        # Entry point of React app
|
| -- .eslintrc.cjs         # ESLint configuration
| -- .gitignore            # Git ignore file
| -- index.html            # HTML entry point
| -- package.json          # Project metadata and dependencies
| -- package-lock.json     # Dependency lock file
| -- README.md             # Project documentation (if any)
| -- vite.config.js        # Vite configuration
```

```
|
|-- my-app/              # (Secondary app directory if used)
|   └──── node_modules/    # Dependencies for secondary app
|   └──── public/          # Public files
|   └──── src/             # Source code
|   └──── .gitignore       # Git ignore file
```

6.Running the Application

Project Implementation & Execution

After completing the code, follow the steps below to run the application:

Frontend (React.js Application):

- Run the React application by using the command, if you are using Vite.js:

  ```
  npm run dev
  ```

Backend (JSON Server):

- Open a new terminal and type the command:

  ```
  json-server --watch ./db/db.json
  ```

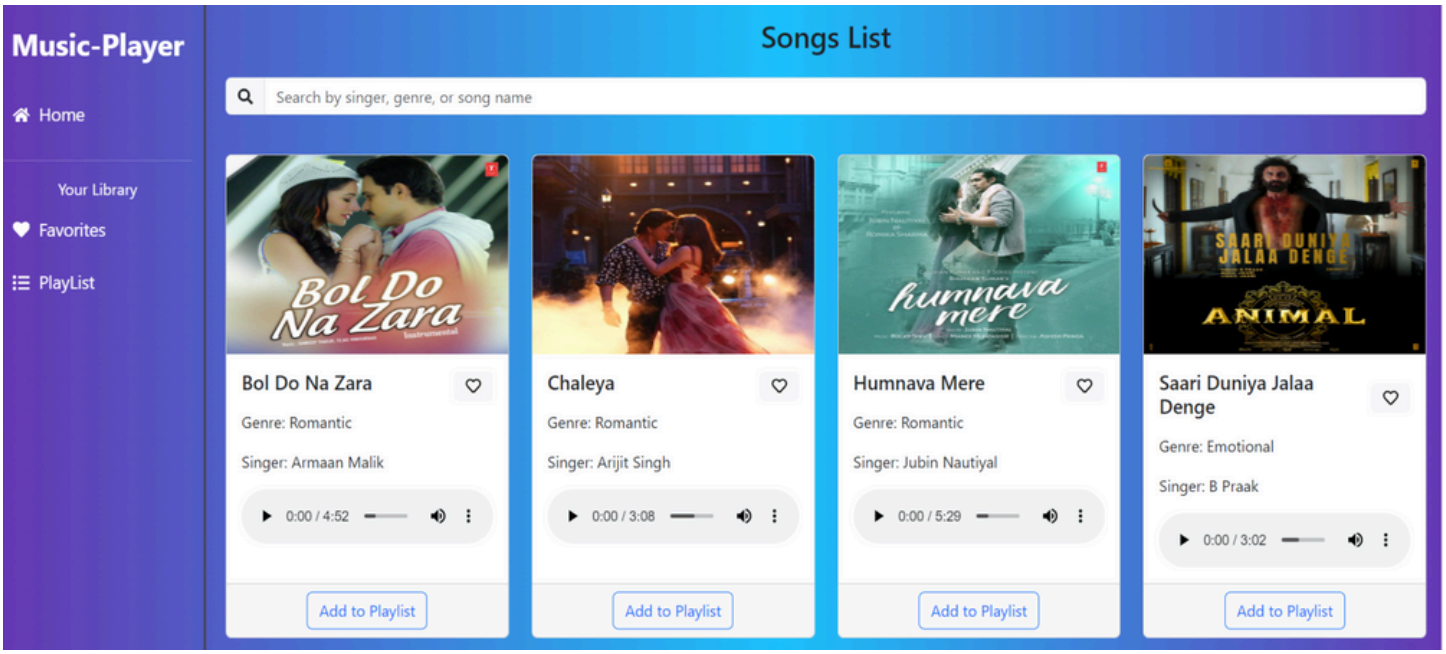
- This will start the JSON server and provide the backend for data handling.

Launch the Application:

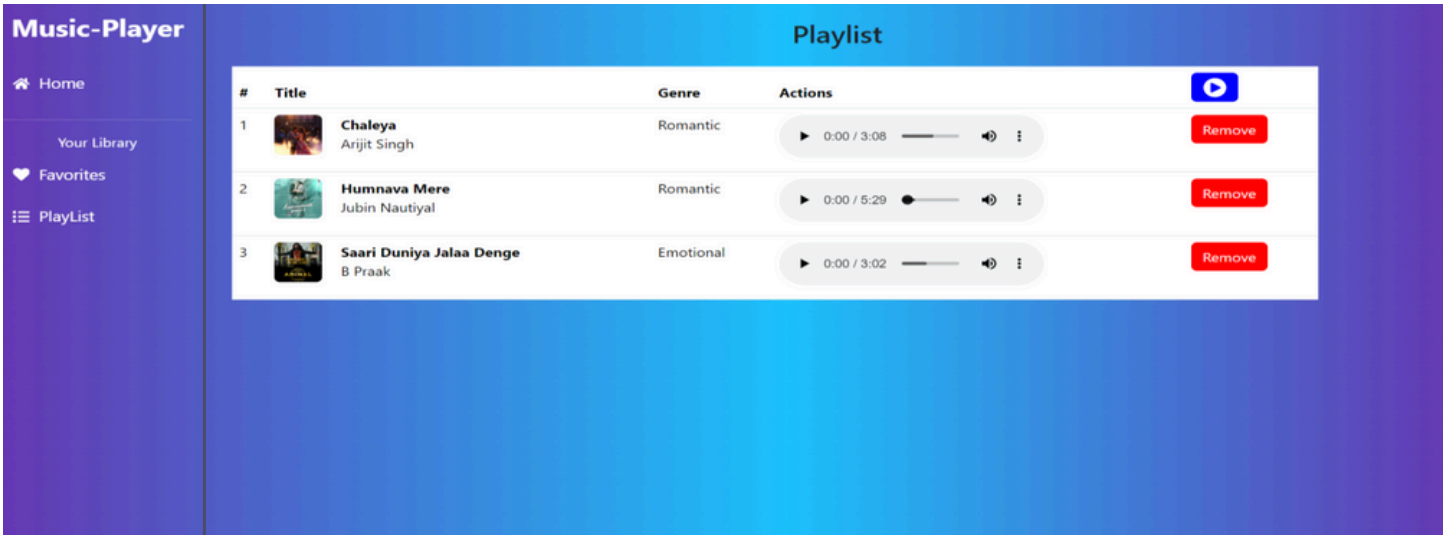
- Once both the React application and JSON server are running, launch **Rhythmic Tunes** in your browser to explore and test the features.

Screenshots:

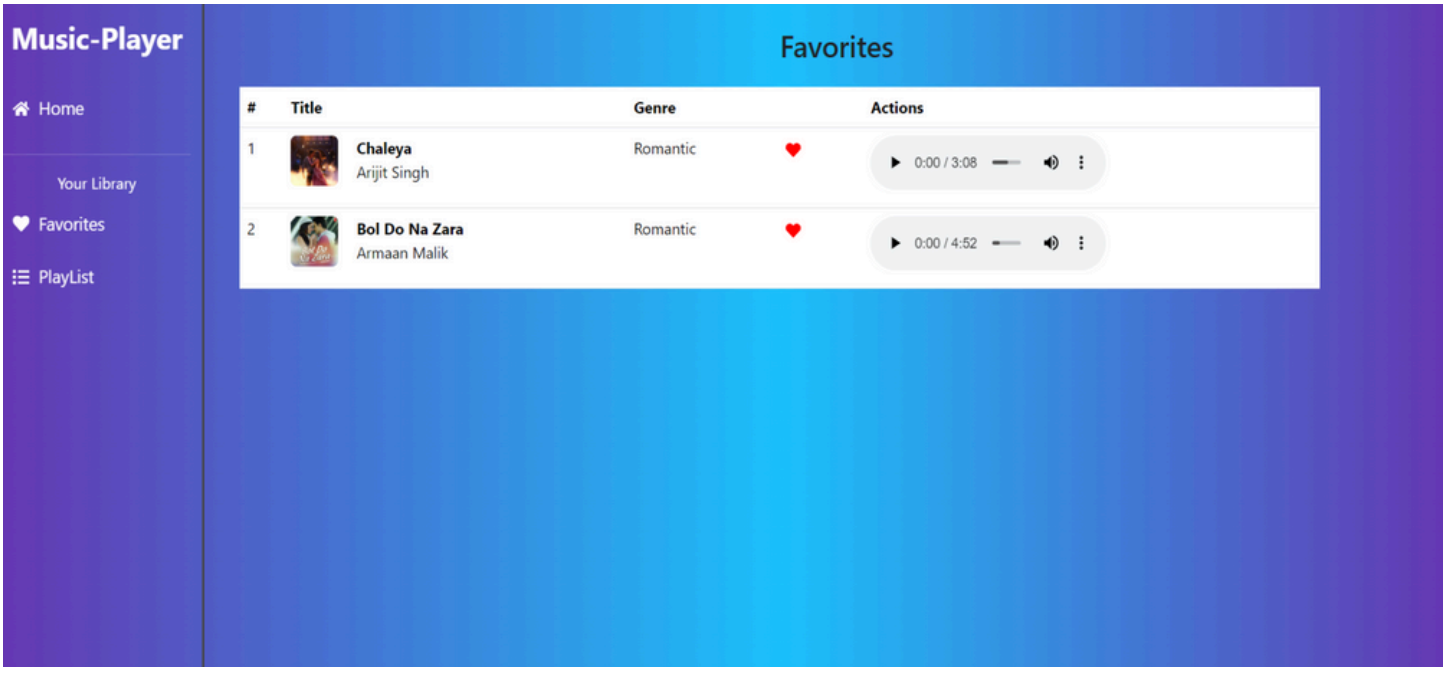
- **Hero components**



- Playlist



- Favorites



7. API Documentation

The backend uses **JSON Server** to simulate REST APIs for handling music, playlists, and user data.

Users

- POST /users – Register a new user
- GET /users – Get all users
- GET /users/:id – Get user details by ID
- PUT /users/:id – Update user information
- DELETE /users/:id – Delete a user

Songs

- GET /songs – Fetch all songs
- GET /songs/:id – Fetch details of a song by ID
- POST /songs – Add a new song (admin only)
- PUT /songs/:id – Update song details
- DELETE /songs/:id – Delete a song

Playlists

- GET /playlists – Fetch all playlists
- GET /playlists/:id – Get details of a playlist
- POST /playlists – Create a new playlist
- PUT /playlists/:id – Update playlist information
- DELETE /playlists/:id – Delete a playlist

Favorites

- GET /favorites – Get all favorite songs of a user
- POST /favorites – Add a song to favorites
- DELETE /favorites/:id – Remove a song from favorites

8. Authentication

- **Basic Authentication (Mock):**
  - Users can register and log in using mock APIs provided by **JSON Server**.
  - User credentials (username, email, password) are stored in the db.json file.
  - Login is validated by checking credentials against stored user data.
- **Protected Routes (Frontend):**
  - Certain features like creating playlists or adding favorites are only available to logged-in users.
  - A simple session or local Storage mechanism is used to track whether a user is logged in.
- **Note:**
  - This project currently uses **mock authentication** for demonstration purposes.
  - Future enhancement may include **JWT-based authentication** with a Node.js/Express backend for secure login and token-based session handling.

9. User Interface

The application provides a simple and user-friendly interface with the following key screens:

- **Landing Page:**
  The homepage introducing Rhythmic Tunes, highlighting features, and guiding users to explore or sign up.
- **Music Player:**
  Core page where users can play, pause, skip, and control songs. Displays current track details and album art.
- **Playlist Page:**
  Allows users to create, manage, and view playlists. Supports adding/removing songs and organizing favorites.
- **Favorites Page:**
  Dedicated section where users can view and manage their marked favorite songs.
- **Admin Panel:**
  A simple backend management page for adding, updating, or removing songs and managing user data.

10. Testing

Testing Approach:

- Manual testing was carried out during the development of each milestone.
- The primary focus was on verifying core functionalities such as song playback, playlist creation, and favorites management.

Tools Used:

- **Chrome DevTools:** For debugging UI, checking console errors, and testing responsiveness.
- **JSON Server Logs:** To verify API requests and responses.
- **Browser Testing:** Application was tested on multiple browsers (Chrome, Edge, Firefox) for compatibility.

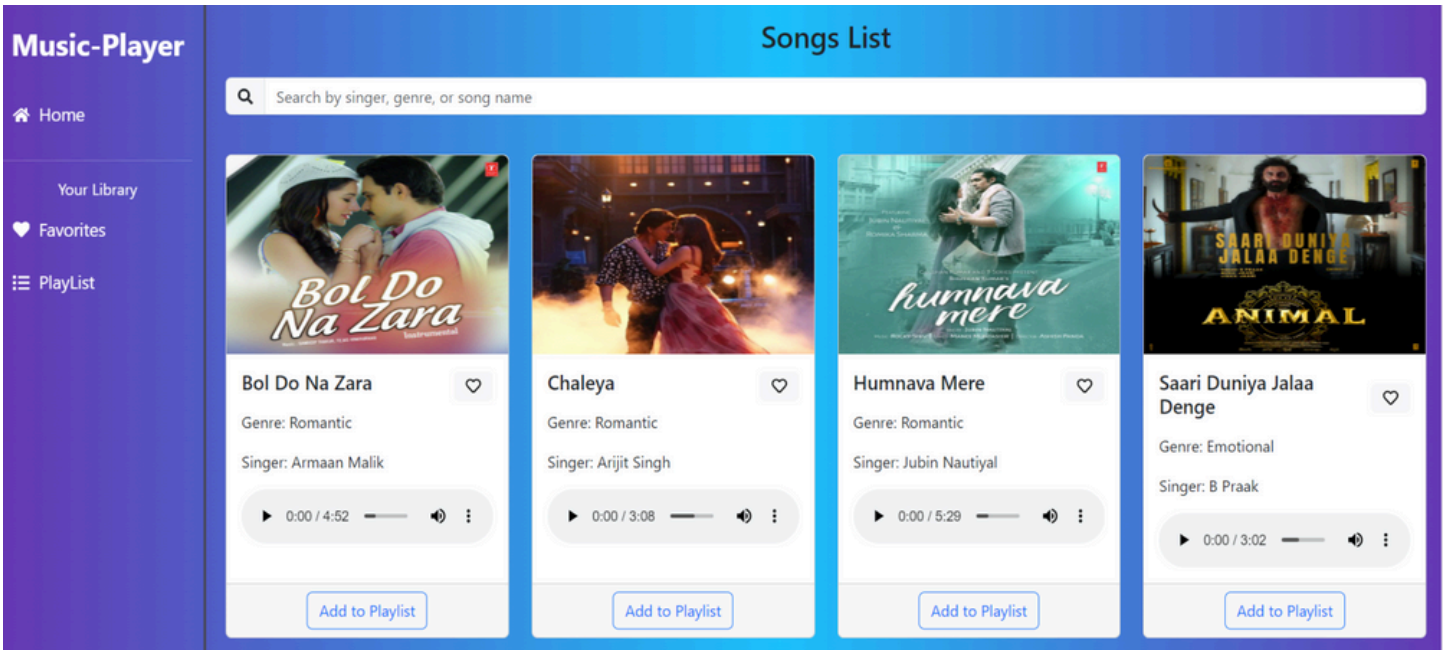
Key Tests Performed:

- Verified user registration and login with mock authentication.
- Tested song streaming (play, pause, skip).
- Checked playlist creation, update, and deletion features.
- Confirmed favorite songs could be added and removed.
- Ensured that UI elements (buttons, links, forms) worked as expected.

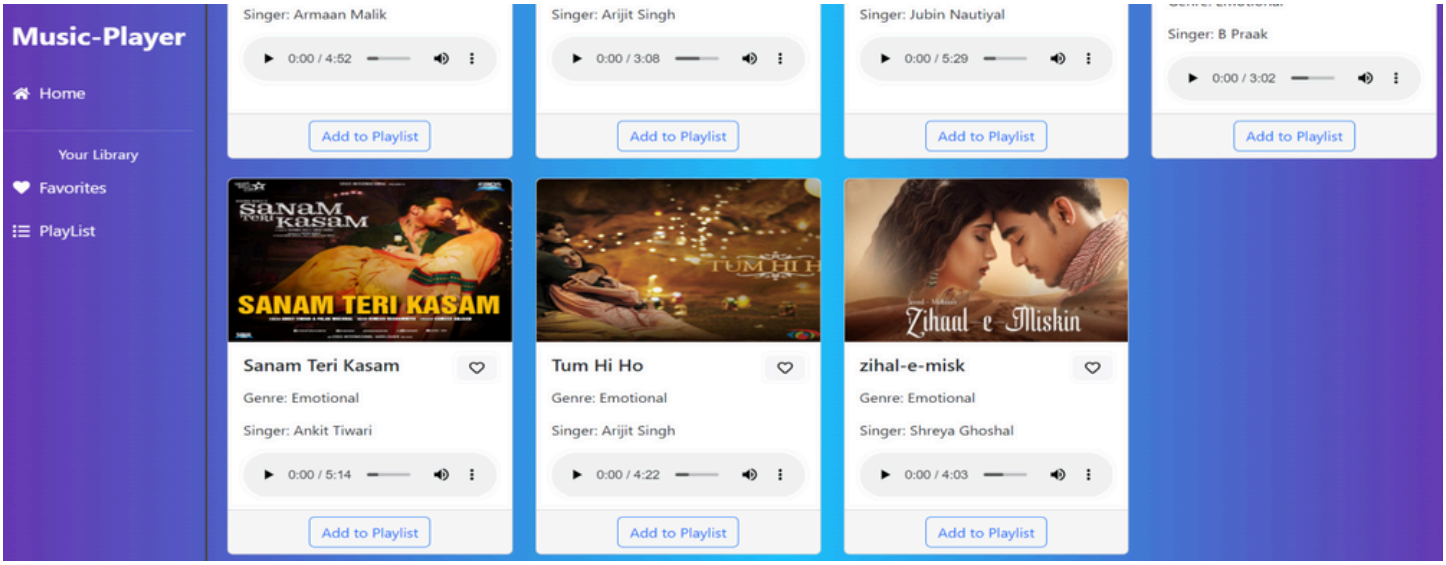
11. Screenshots or Demo

Below are the screenshots of the **Rhythimic Tunes** Music Streaming Application:

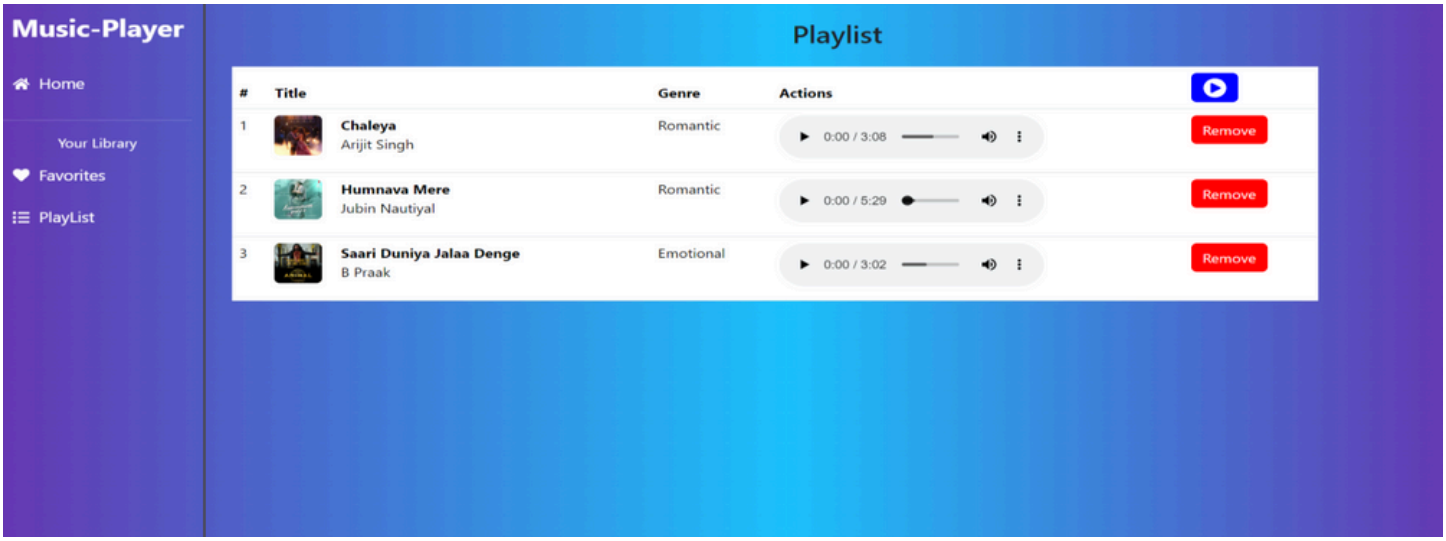
- **Home Page:**



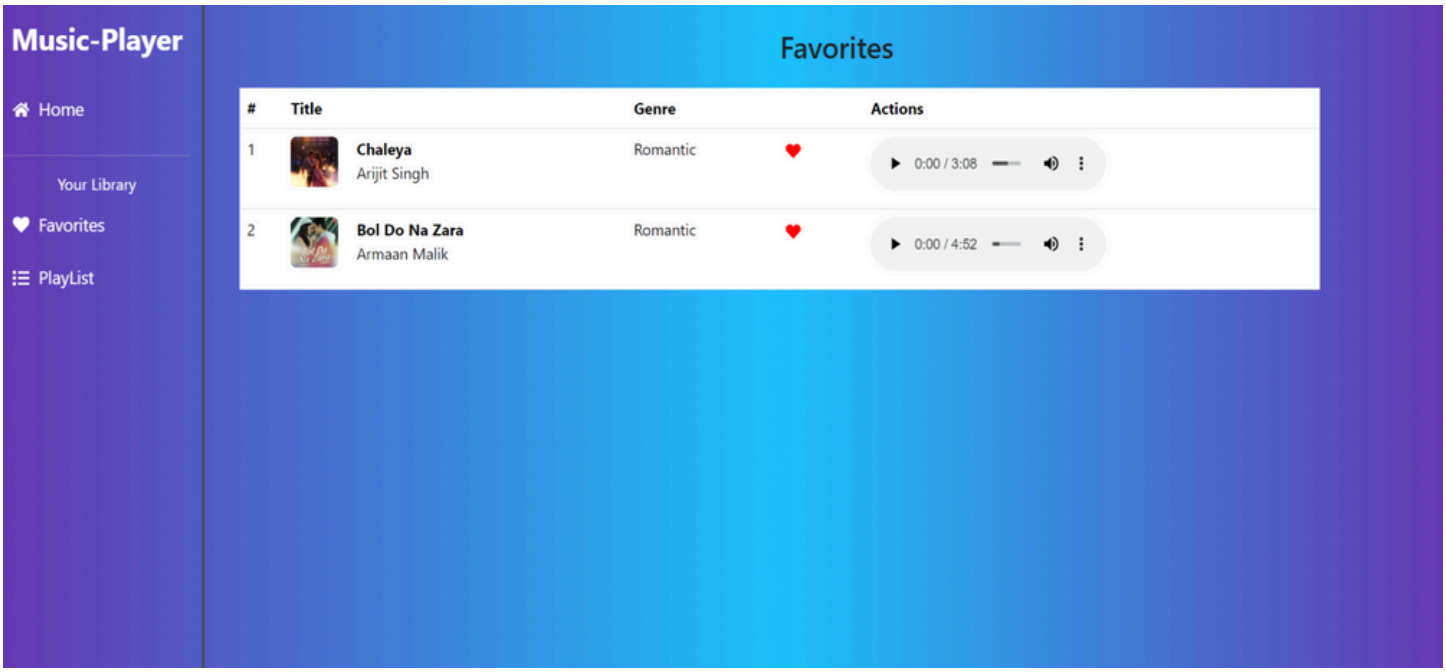
- Music Player:



- Playlist Management:



- Favorites Section:



12. Conclusion & Future Work

The **Rhythmic Tunes: Your Melodic Companion** project successfully delivers a user-friendly music streaming platform, allowing users to explore, play, and organize their favorite tracks seamlessly. With its simple interface, responsive design, and integration of JSON server for backend data, it provides an enjoyable music experience for enthusiasts.

Future Work:

- **Mobile Application:** Extend the platform to Android/iOS for better accessibility.
- **AI-Powered Recommendations:** Suggest songs and playlists based on user preferences.
- **Offline Mode:** Enable downloading and playing music without internet connectivity.
- **Social Features:** Allow users to share playlists and connect with other music enthusiasts.
- **Integration with Cloud Database:** Replace the JSON server with scalable cloud solutions for production-ready deployment.