

Project Documentation

Project Title: Rhythmic Tunes: Your Melodic Companion

1.Introduction:

- **Project Title:** Rhythmic Tunes: Your Melodic Companion
- **Team ID:** NM2025TMID35072
- **Team Leader:** Jayavarthini T
(jayavarthinitheivendran@gmail.com)
- **Team Members:**
 - 1) Navmitha afrin S
(navmithaafrin@gmail.com)
 - 2) Palani praba D
(palanipraba15@gmail.com)
 - 3)Abinaya R
(abinaya8648@gmail.com)

2.Project Overview:

Purpose:

Rhythmic Tunes is a web-based music player application that allows user to browser, search, and play songs. It provides playlist management, favorites, and a smooth user interface for a personalized music experience.

Features:

- Browse and search songs
- Add songs to playlists and favorites
- Play, pause, skip songs with an interactive player
- Song details (genre, singer, duration)
- Modern responsive UI

3.Architecture:

- Frontend: React.js with Bootstrap for styling
- Backend: JSON Server (simulated REST API with db.json)
- Database/ Data Source: db.json file stores songs, genres, singers, playlists

4.Technologies used:

- React.js
- JSON Server (db.json)
- Visual Studio Code
- Git

5.Installation:

➤ Go to project folder

cd Rhythmic-Tunes

➤ Install frontend dependencies

npm install

➤ Start the React app

npm run dev

In a separate terminal:

- Start JSON Server on port 3000 (or 5174/5173 if React is using 3000 already)

npm json-server--watch db.json--port 3000

6.Folder Structure:

Rhythmic-Tunes/

| --code/ #Main project folder

| | --db/ #Database

Configuration (MongoDB)

| | --node_modules/ #Installed

Dependencies

| | --public/ #Static files

| | --src/ #Source code

| | --assets/ #Media assets

(images,songs,etc.)

| | --Components/ #React

Components

| | --App.css

| --App.jsx
| --index.css
| --main.jsx
| --.eslintrc.cjs
| --.gitignore
| --index.html
| --package-lock.json
| --vite.config.js
|--README.md

7. Running the Application:

Fronted:

npm run dev

Access at: <http://localhost:5173>

8. API Documentation

- **Playlist**

- /api/playlist (Get all playlist)
- /api/playlist/:id (Get playlist by ID)
- /api/playlist/create (Add new playlist)
- /api/playlist/:id/update (Update playlist by ID)
- /api/playlist/:id/delete (Delete playlist by ID)

- **Songs:**

- /api/songs (Get all songs)
- /api/songs/:id (Get song by ID)
- /api/songs/create (Add new song)
- /api/songs/:id/update (Update song by ID)
- /api/songs/:id/delete (Delete song by ID)

- **Artists:**

- /api/artists (Get all artists)
- /api/artists/:id (Get artist by ID)
- /api/artists/create (Add new artist)

- **Albums:**

- /api/albums (Get all albums)
- /api/albums/:id (Get album by ID)

9.Authentication:

No authentication implemented (future enhancement)

10.User Interface:

- Landing Page (all songs)

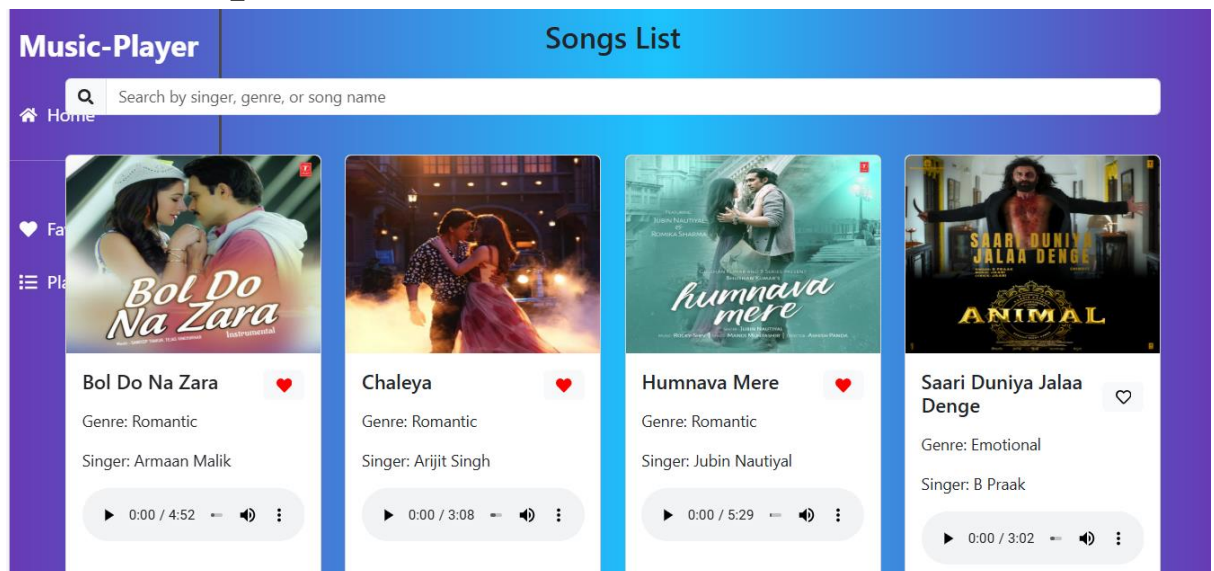
- Sidebar Navigation (Home, Favorites, Playlists)
- Song Cards (title, singer, genre, play button)
- Music Player (play/pause, next, volume)

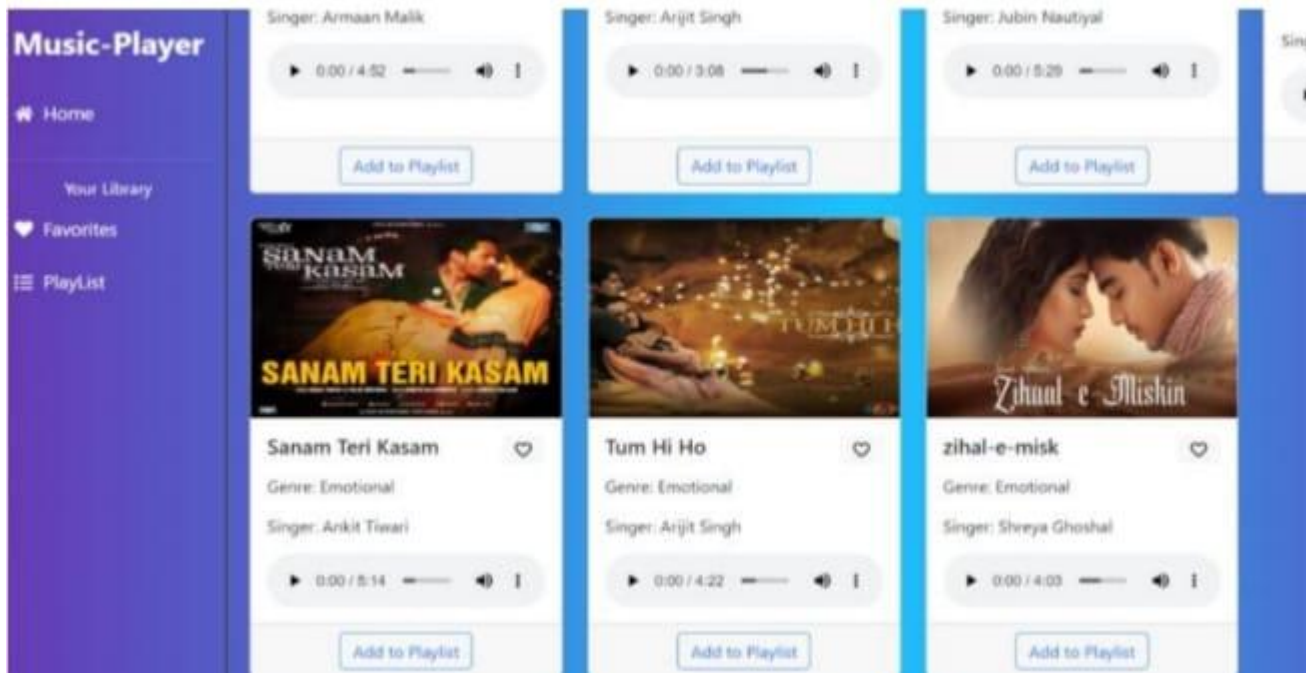
11. Testing:

- Manual Testing: Verified song listing, search, play/pause
- Tools: Chrome Dev Tools, JSON Server

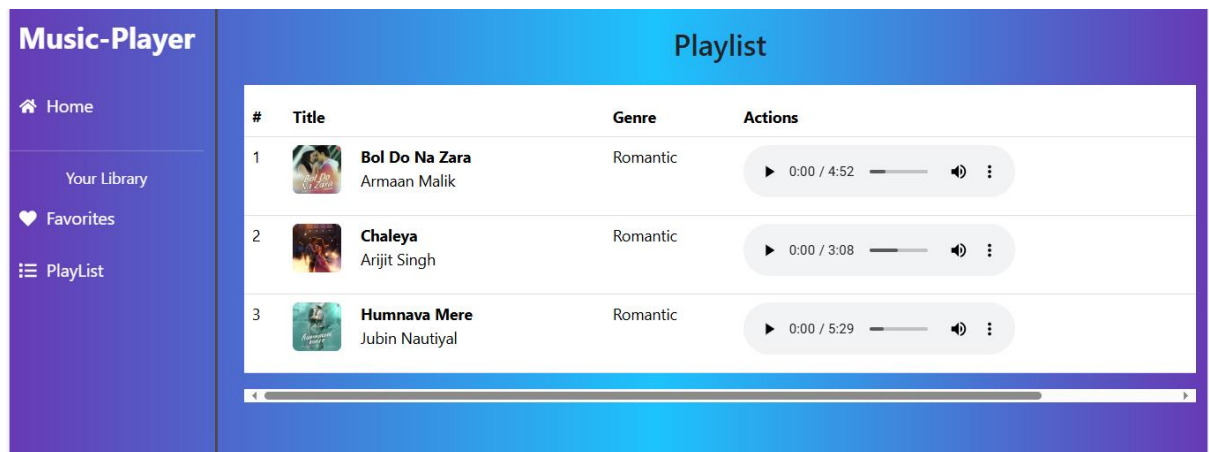
12. Screenshot:

- Hero components

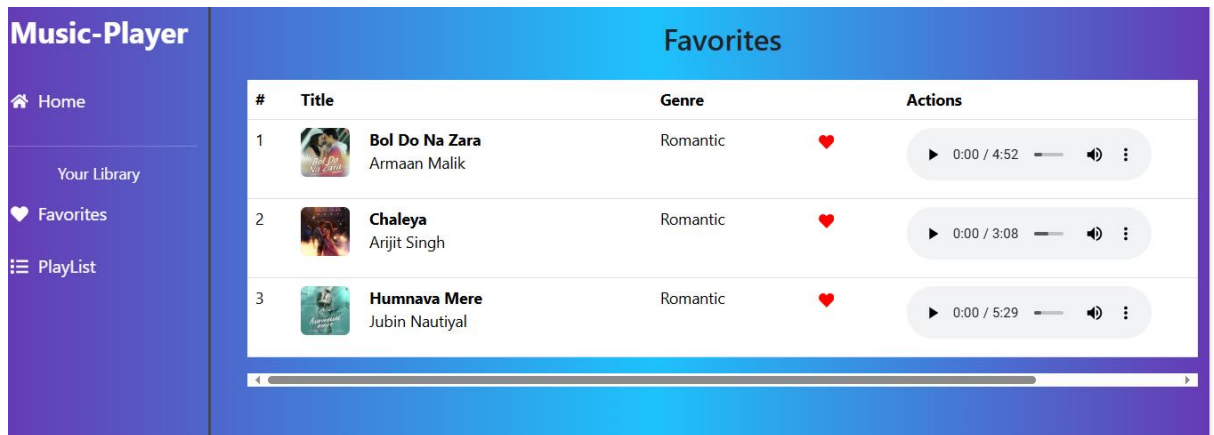




• Playlist:



• Favorites:



13. Known Issues:

- No login system
- Limited to static JSON data
- JSON server must run separately

14. Future Enhancement:

- User authentication with JWT
- Cloud-based music database
- AI-powered songs
- Offline playback support
- Mobile app (React Native)

