

Rhythmic Tunes Project Documentation

1. Introduction

Project Title: Rhythmic Tunes ■ Your melodic companion

Team ID: NM2025TMID40448

Team Leader:

Name: Bhavadharani M

Mail ID: shadowqueenbhava2006@gmail.com

Team Members:

Name: JAYASHREE V | Mail ID: jayashreee2007@gmail.com

Name: MANIMEGALAI S | Mail ID: manimegalai.s638075@gmail.com

Name: KAMALI M | Mail ID: kamalimageshkumar@gmail.com

2. Project Overview

Purpose: Rhythmic Tunes is designed to provide a simple and interactive platform where users can explore melodic tunes with ease.

Features: Guided Project flow with demo video and code access, Local execution with Node.js, JSON Server setup for data handling, Browser-based application accessible at <http://localhost:5173>

3. Architecture

Frontend: React.js (executed with Vite)

Backend: JSON Server for database simulation

Database: db.json (local JSON-based database)

4. Setup Instructions

Prerequisites: Node.js (LTS version), Visual Studio Code, PowerShell (Windows) or Terminal (Mac)

Installation Steps: Download & install Node.js → Configure execution policy → Download project from SmartInternz portal → Extract & open in VS Code → Run npm install → Run npm run dev

5. Folder Structure

Rhythmic-Tunes/

- code/ # Frontend React.js code
- db/ # JSON Server database
- db.json

6. Running the Application

Frontend: npm run dev → Access at <http://localhost:5173>

Backend: cd db → npm i -g json-server → json-server --watch db.json --port 3000 → Runs at <http://localhost:3000>

7. API Documentation

Default REST API from JSON Server: GET /songs, POST /songs, PUT /songs/:id, DELETE /songs/:id

8. Authentication

Not implemented (uses open JSON server).

9. User Interface

Application opens in browser at <http://localhost:5173>

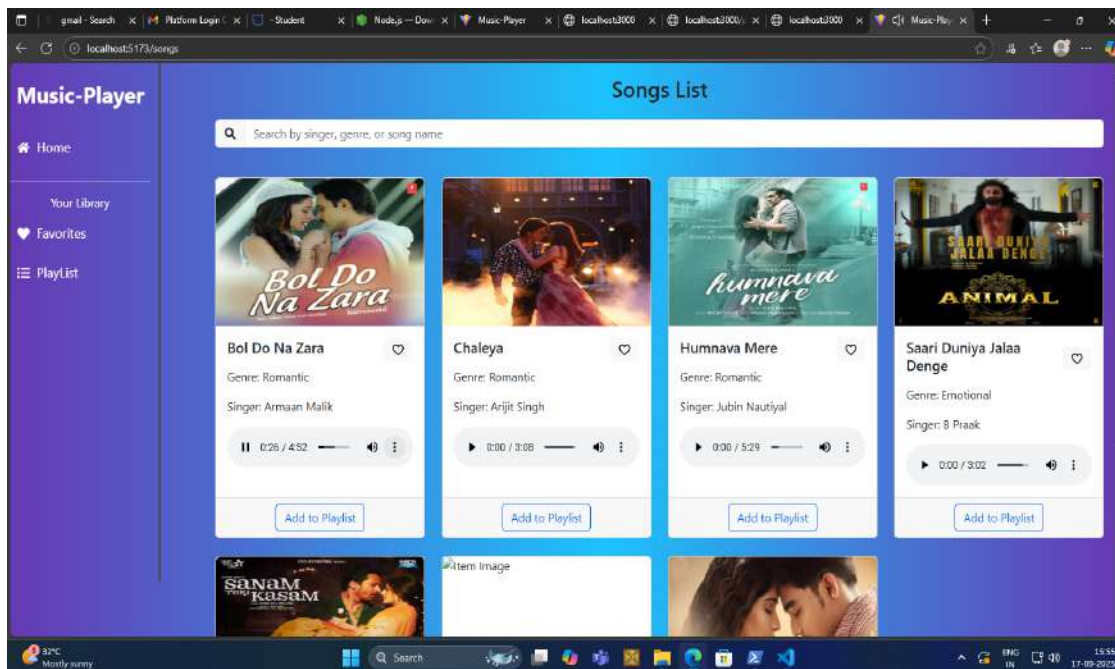
UI built with React.js and Vite; basic navigation and demo pages included in the frontend code folder.

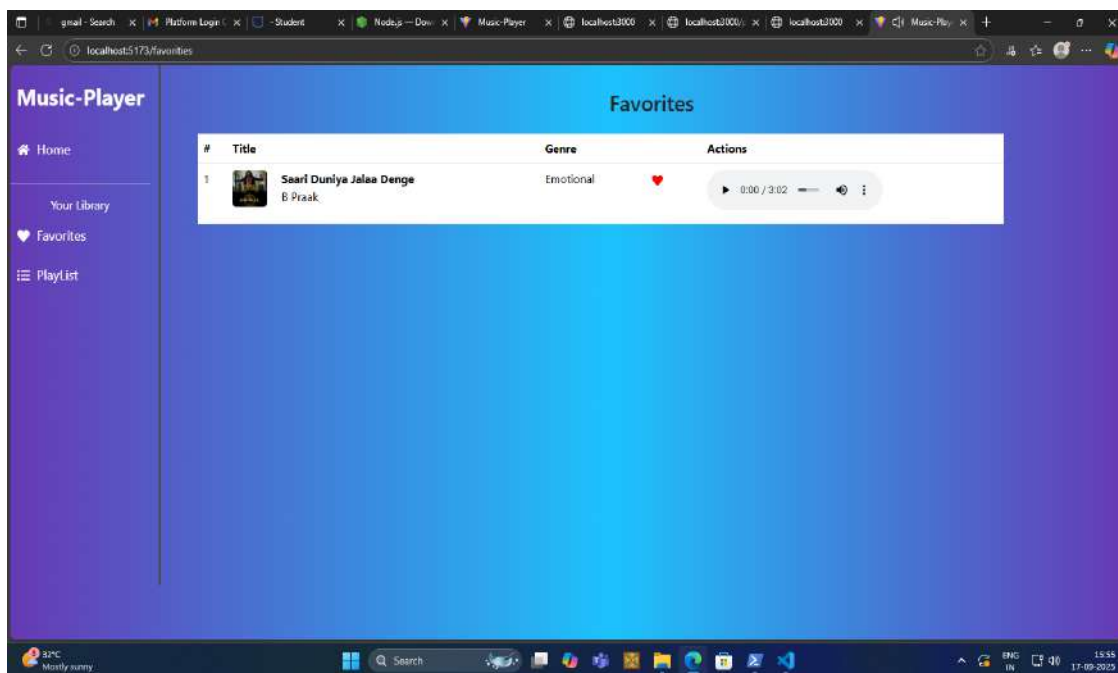
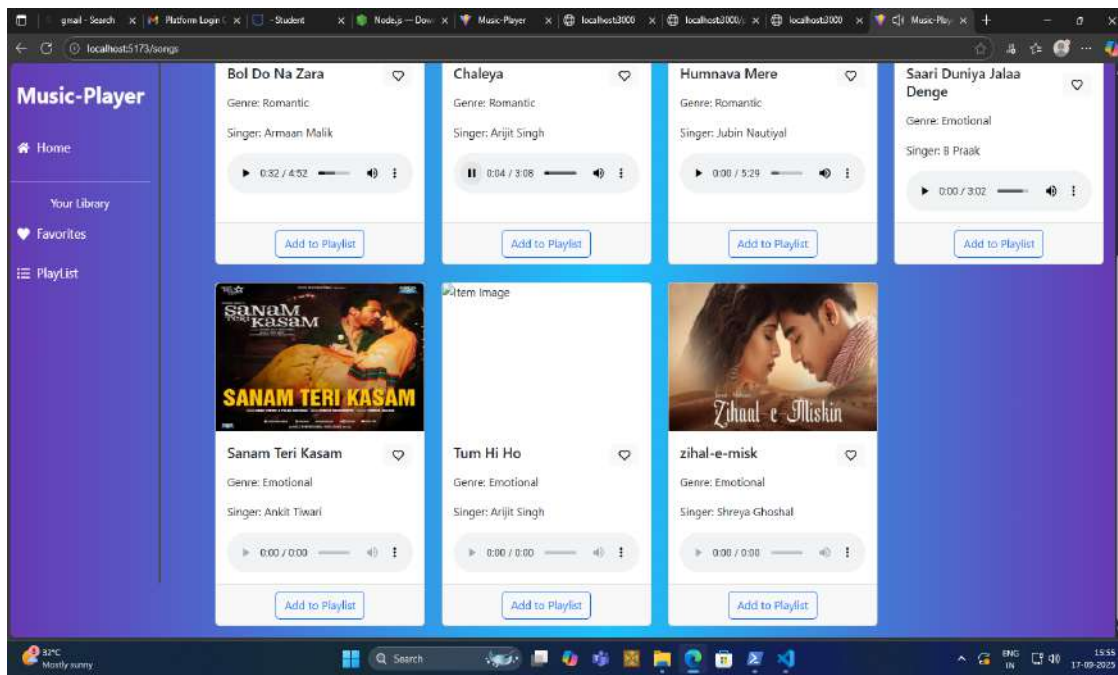
10. Testing

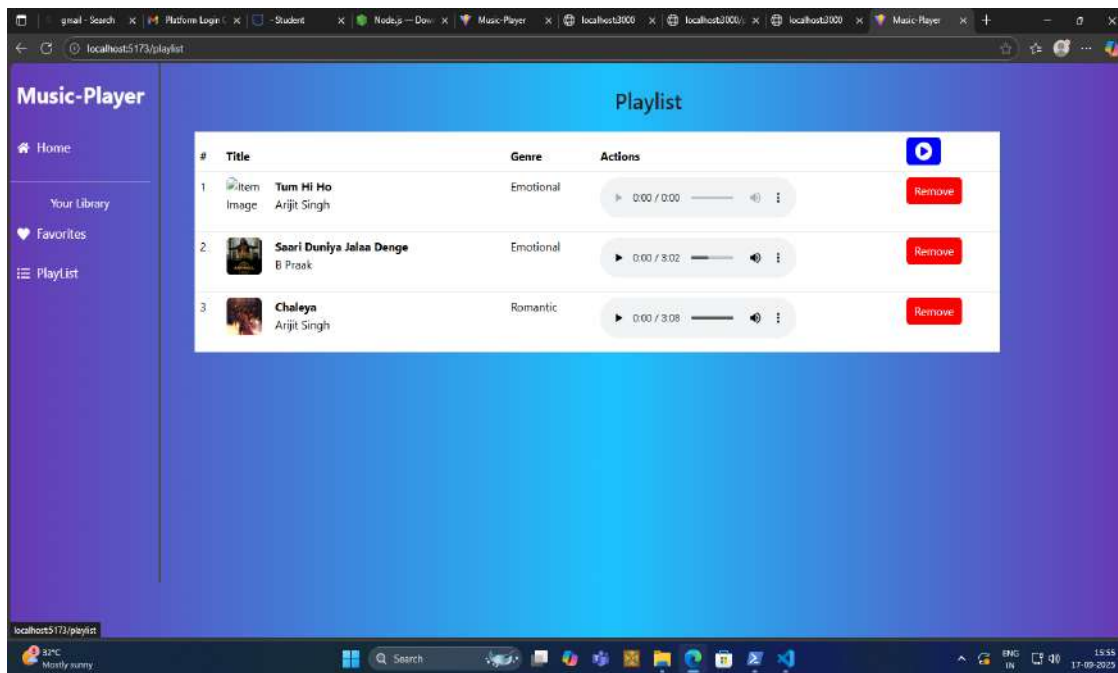
Manual testing using Browser Dev Tools & Postman for API validation

Verify endpoints: GET /songs, POST /songs, PUT /songs/:id, DELETE /songs/:id

11. Screenshots or Demo







12. Known Issues

- JSON server needs to be manually started
- Only works on local machine setup

13. Future Enhancements

- Add proper backend with database integration (MongoDB/MySQL)
- Implement authentication & user management
- Deploy project to cloud for remote access