

# Project Documentation

## Rhythmic Tunes:Your melodic companion

### 1. Introduction

- Project Title:Rhythmic Tunes:Your melodic companion
- Team ID:NM2025TMID37420
- Team Leader: Gunavathi S & [gunapoppz@gmail.com](mailto:gunapoppz@gmail.com)
- Team Members:
  - Sivasakthi T & [sivasakthithennarasu76@gmail.com](mailto:sivasakthithennarasu76@gmail.com)
  - Sivasankari T & [sivasankarithennarasu23@gmail.com](mailto:sivasankarithennarasu23@gmail.com)
  - Logitha P & [logiprasad30@gmail.com](mailto:logiprasad30@gmail.com)

### 2. Project Overview

- **Purpose:**Rythamic Tunes is a web-based music application that allows users to listen to tunes, manage playlists, and simulate a music streaming experience using a simple JSON-based backend.
- **Features:**
  - Play, pause, and navigate songs
  - Song list stored in JSON server
  - Local development setup with Node.js
  - Responsive frontend built with modern tools

### 3. Architecture

- **Frontend:** React.js with Vite (UI development)
- **Backend:** JSON Server (simulates REST API for song data)
- **Database:** db. json file stores songs and playlist data

## 4. Setup Instructions

### Prerequisites:

- Node.js LTS (<https://nodejs.org>)
- Visual Studio Code
- JSON Server (npm package)

### Installation Steps:

1. Download project code from SmartInternz portal  
Guided Projects-Workspace
2. Extract the ZIP folder.
3. Open the project in VS Code.
4. Open terminal and run:

```
Bash Copy code
npm install
```

5. Run the frontend:

```
Bash Copy code
npm run dev
```

6. Start JSON server (in another terminal):

```
Bash Copy code
cd db
npm i -g json-server
json-server --watch db.json --port 3000
```

## 5. Folder Structure

```
Rythamic-Tunes/
|-- src/           # React frontend code
|  |-- components/
|  |-- pages/
|
|-- db/           # JSON Server data
|  |-- db.json
```

```
|  
|-- package.json # Project dependencies
```

## 6. Running the Application

- **Frontend:**

Bash

```
npm run dev
```

- **Backend (JSON Server):**

Bash

```
cd db
```

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

## 7. API Documentation

- **Songs:**

- GET /songs – Get all songs
- POST/songs -Add a new song
- PUT/songs/:id– Update a song
- DELETE /songs/:id- Delete a song

## 8. Authentication

The current version has no authentication. Can be extended with Firebase or JWT in future.

## 9. User Interface

- Landing Page: Displays song list

- Player Controls: Play, pause, skip
- Playlist Management (Future Enhancement): Add/remove songs

## 10. Testing

- **Manual testing:** Check in browser for proper song rendering
- **Tools:** Chrome DevTools, Postman (for API)

## 11. Demo Link

[https://drive.google.com/file/d/1O2bNbeE8u\\_Z0zFdDUlfinDs4I7YH8YLD/view?usp=sharing](https://drive.google.com/file/d/1O2bNbeE8u_Z0zFdDUlfinDs4I7YH8YLD/view?usp=sharing)

## 12. Known Issues

- Limited to local JSON server
- No real-time streaming or authentication

## 13. Future Enhancement

Your Melodic Companion blends rhythm and melody into an interactive partner for creators and listeners. Future enhancements will add AI-powered tune generation, personalization, and collaborative features to elevate the musical journey.