Project Documentation :

Rhythmic Tunes Your Melodic Companion

# 1. Introduction

* Project Title: Music Player Frontend
* Team ID: NM2025TMID28777
* Team Leader: RAJESHWARI A & [arajeshwari365@gmail.com](mailto:arajeshwari365@gmail.com)
* Team Members: DHIVYADHARSHANI B [balur1876@gmail.com](mailto:balur1876@gmail.com) & ,DEVISRI B & [devisridevika2007@gmail.com](mailto:devisridevika2007@gmail.com) , VAISHNAVI S & [ajayavp2008@gmail.com](mailto:ajayavp2008@gmail.com)

# 2. Project Overview

* Purpose: The Music Player Frontend provides an interactive interface for users to play, pause, and manage their songs with a clean and responsive design.
* Features: Play, pause, and skip tracks; Responsive design with TailwindCSS and Bootstrap; React Router for navigation; JSON Server integration for mock API; Icon-based controls with react-icons

# 3. Architecture

* Frontend: React.js + Vite with TailwindCSS and Bootstrap
* Backend (Mock API): JSON Server for simulating REST APIs
* Database: Mock JSON data (JSON Server)
* Development Tools: ESLint, Vite

# 4. Setup Instructions

* Prerequisites: Node.js, Git, Visual Studio Code
* Installation Steps:
* git clone <repo-url>
* cd music-player-frontend-
* npm install
* npm run dev

# 5. Folder Structure

* music-player-frontend-/
* │-- public/ # Static assets
* │-- src/ # React source code
* │ │-- components/ # UI components
* │ │-- pages/ # Application pages
* │ │-- App.jsx # Main app component
* │ │-- main.jsx # Entry point
* │-- index.html # Root HTML file
* │-- package.json # Dependencies & scripts
* │-- vite.config.js # Vite configuration
* │-- .eslintrc.cjs # Linting rules
* │-- .gitignore # Git ignored files
* │-- README.md # Basic documentation

# 6. Running the Application

* Frontend: npm run dev → Access at http://localhost:5173
* Mock Backend (JSON Server): npx json-server --watch db.json --port 3001 → Access at http://localhost:3001

# 7. API Documentation

* Songs:
* GET /songs → Fetch all songs
* GET /songs/:id → Fetch song by ID
* POST /songs → Add new song
* PUT /songs/:id → Update song
* DELETE /songs/:id → Delete song

# 8. Authentication

* Currently, no authentication system is implemented.
* Future enhancement: JWT-based authentication for user login and playlists.

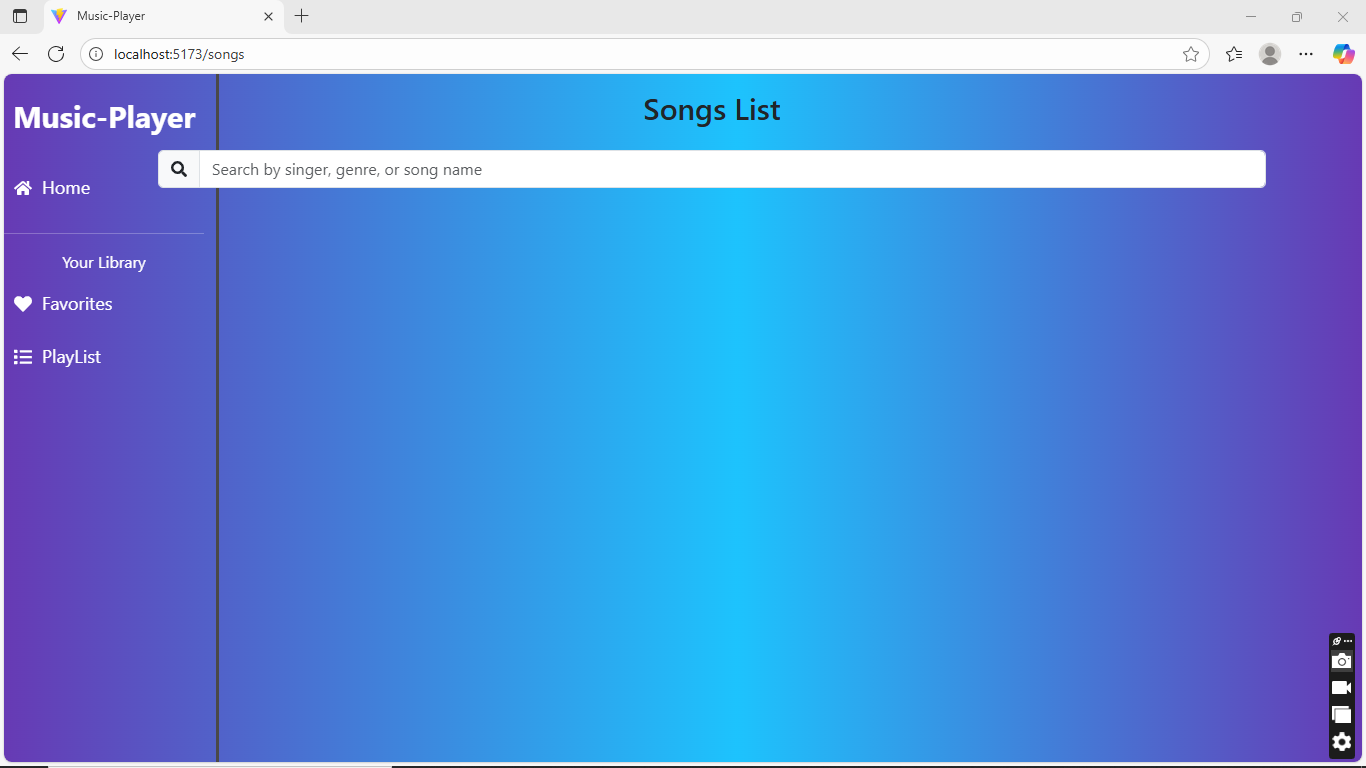
# 9. User Interface

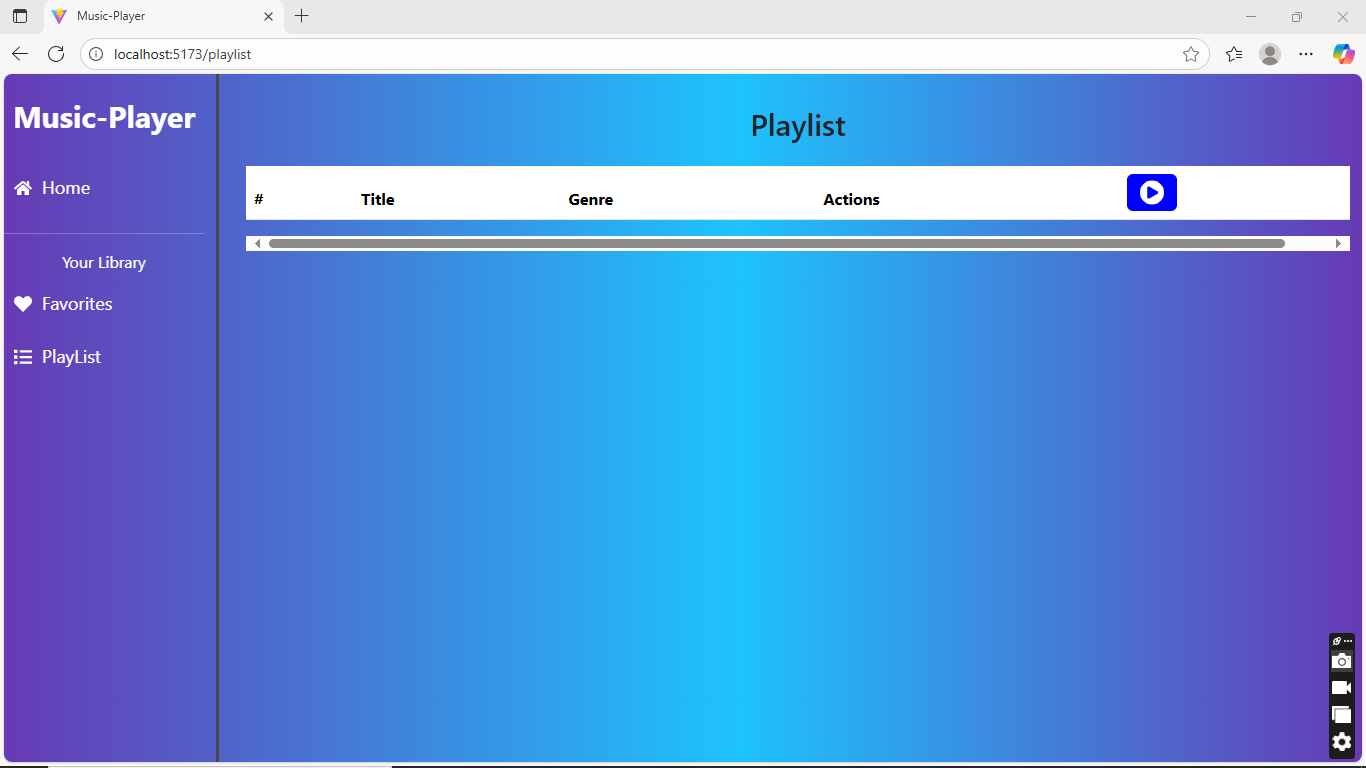
* Landing Page with music list
* Music Player controls (play, pause, skip)
* Playlist view
* Responsive layout with TailwindCSS & Bootstrap

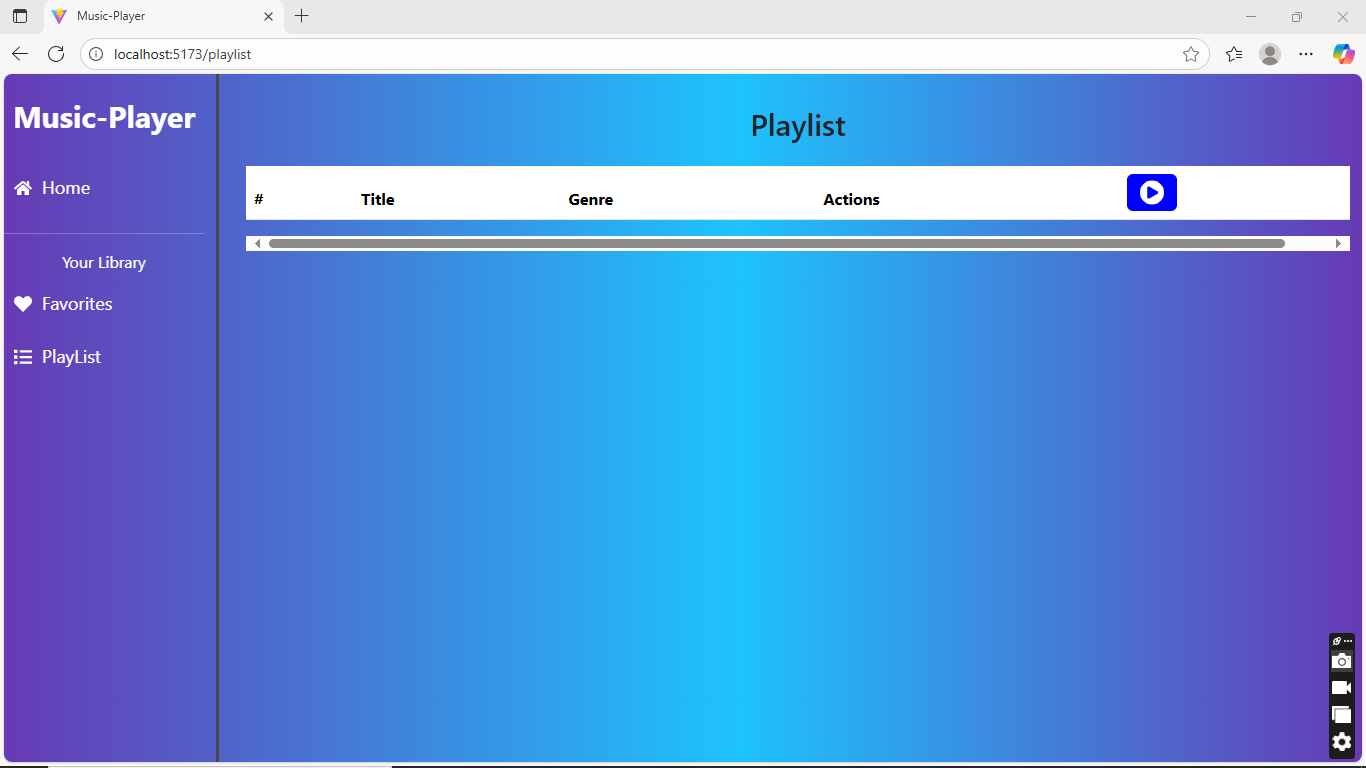
# 10. Testing

* Manual testing during development
* Tools: Chrome DevTools, Postman (for API testing with JSON Server)

# 11. Screenshots or Demo







# 13. Future Enhancements

* Implement real backend with Node.js & MongoDB
* Add JWT authentication
* Support user playlists and favorites
* Mobile app integration