Project Documentation

Rhythmic Tunes-Your melodic companion

# 1. Introduction

• Project Title: Rhythmic tunes-Your melodic companion

• Team ID: NM2025TMID28784

• Team Leader: R.Priyadharshini & [priyadharshinirajeshkannan1423@gmail.com](mailto:priyadharshinirajeshkannan1423@gmail.com)

• Team Members:  
– M.Sasnila & [sasnila851@gmail.com](mailto:sasnila851@gmail.com)   
– R.Swathisri & [swathisriswathisri028@gmail.com](mailto:swathisriswathisri028@gmail.com)

-P.Parkavi & [sramchandren51@gmail.com](mailto:sramchandren51@gmail.com)

# 2. Project Overview

• Purpose: The application allows users to manage their personal music playlists, search songs, add to favorites, and organize playlists with an intuitive interface.

• Features:  
– Playlist creation and management  
– Favorites and Wishlist  
– Song search functionality  
– Sidebar navigation  
– Responsive design with React

# 3. Architecture

• Frontend: React.js (Vite build tool)

• Backend: Local DB / Mock DB (db folder)  
• Server may be extended via Node.js/Express if required

• Database: Local JSON/Mock database (db folder)

# 4. Setup Instructions

• Prerequisites:  
– Node.js  
– Git  
– Visual Studio Code  
– npm or yarn

• Installation Steps:  
# Clone the repository  
git clone <repo\_url>  
  
# Install dependencies  
cd code  
npm install  
  
# Run the application  
npm run dev

# 5. Folder Structure

code/  
 |-- public/ # Static assets  
 |-- db/ # Mock/Local database  
 |-- src/ # React source code  
 |-- Components/ # React components  
 |-- App.jsx # Main App component  
 |-- main.jsx # Entry point  
 |-- assets/ # Images and other static assets  
 |-- package.json # Dependencies and scripts  
 |-- vite.config.js # Vite configuration

# 6. Running the Application

• Start the development server:  
npm run dev  
  
• Access the app via:  
http://localhost:5173

# 7. API Documentation

This project currently uses a mock/local database.  
Possible API endpoints (if backend is extended):  
– /api/songs  
– /api/playlists  
– /api/favorites  
– /api/wishlist

# 8. Authentication

Currently, the app does not implement authentication.  
Future versions may add JWT-based authentication or OAuth.

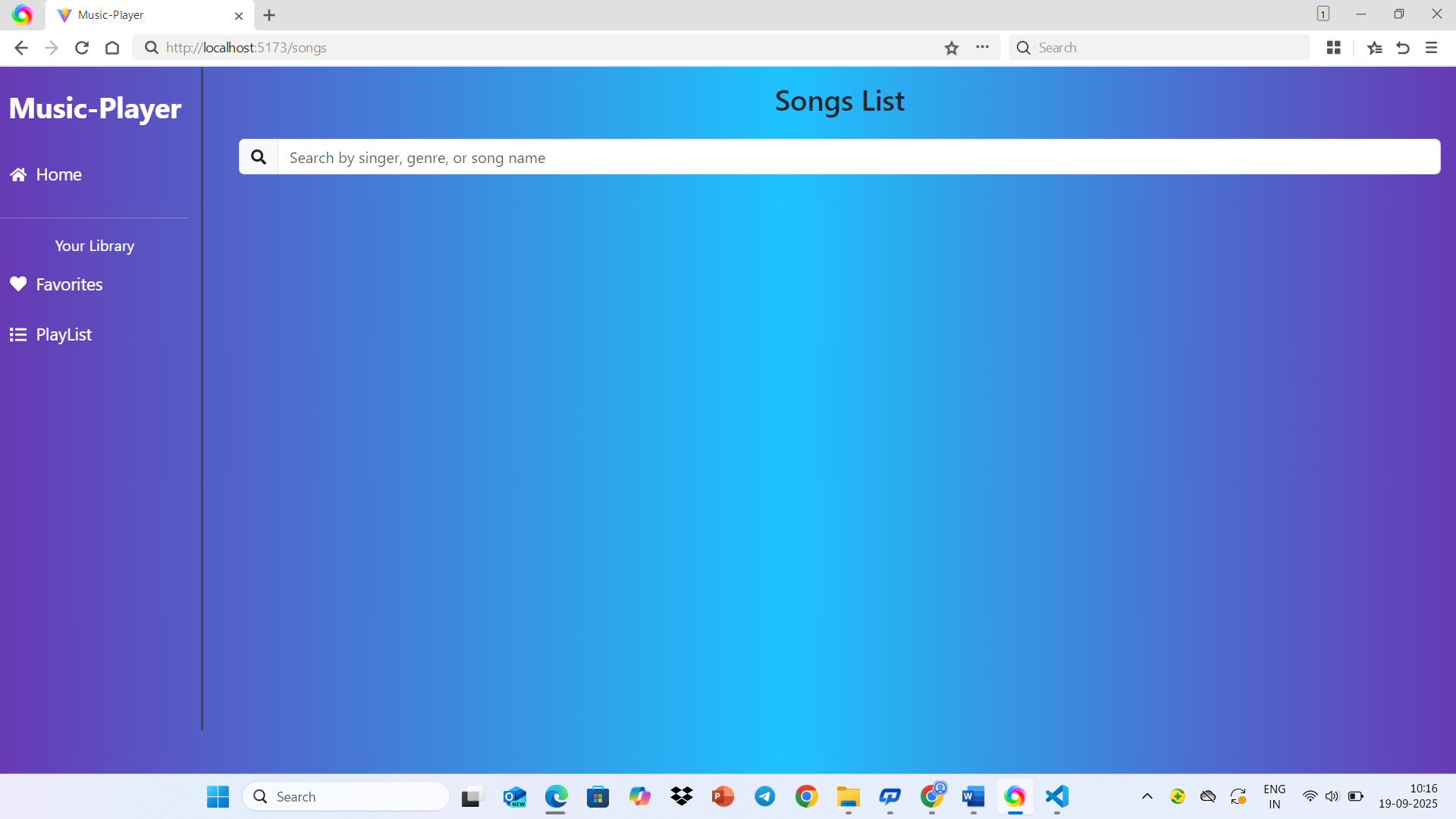
# 9. User Interface

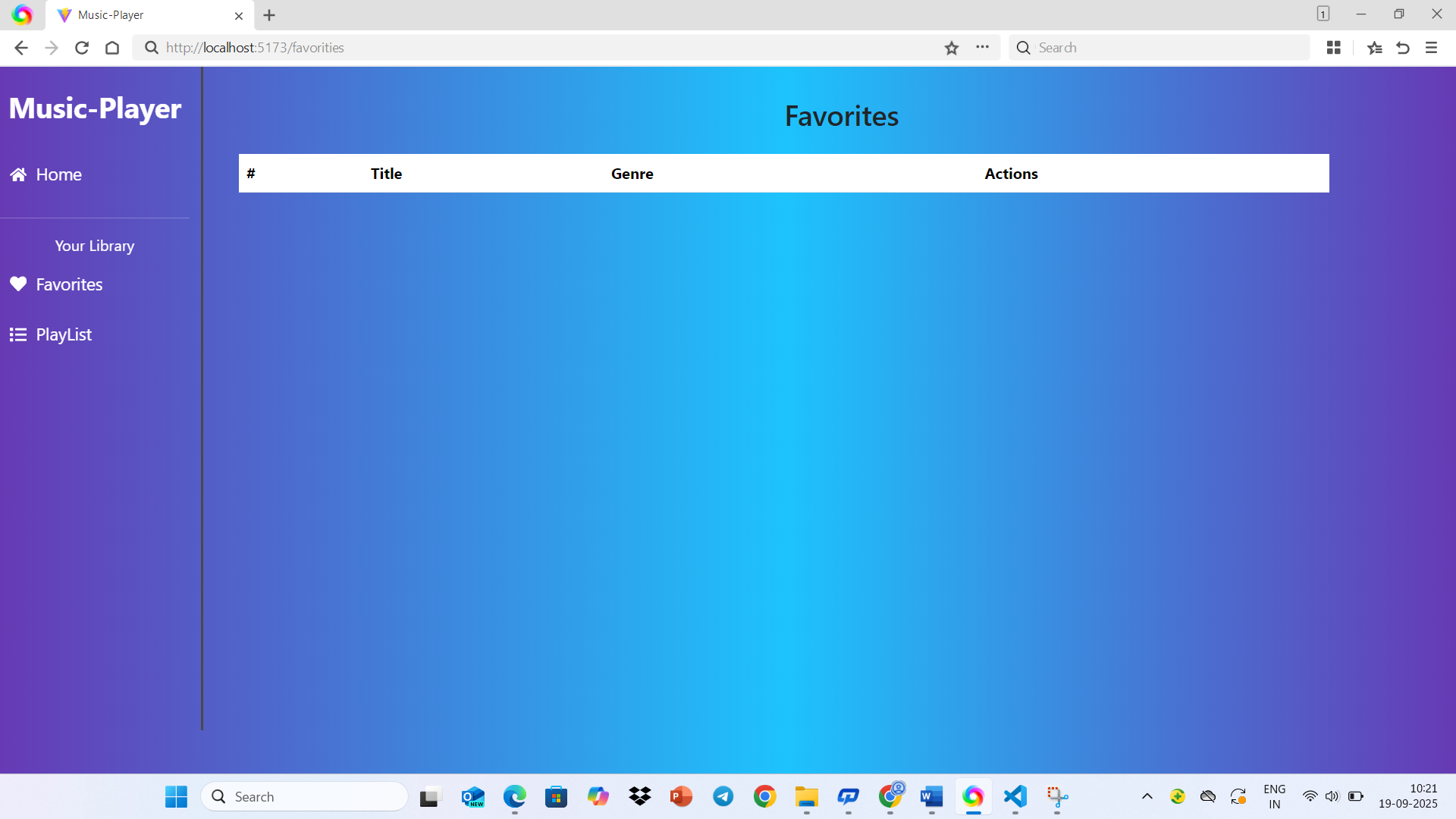
• Home Page  
• Sidebar Navigation  
• Songs Page  
• Playlist Page  
• Favorites Page  
• Wishlist Page  
• Search Page

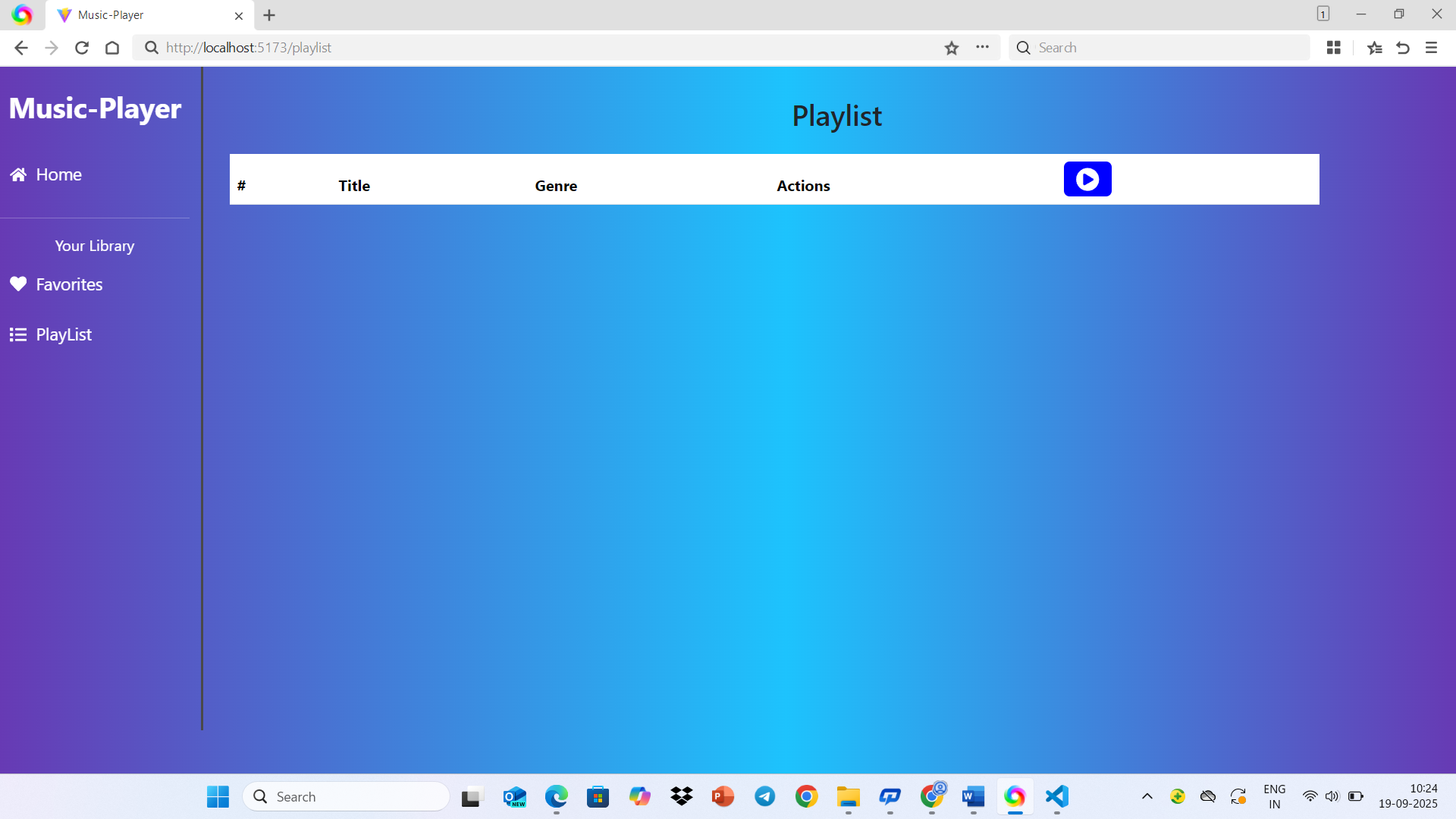
# 10. Testing

• Manual testing during development.  
• Tools: Chrome Dev Tools, React Developer Tools, Postman (if backend APIs are added).

# 11. Screenshots or Demo







# 12. Known Issues

• Authentication not implemented.  
• Local DB only, no persistent backend.  
• Limited error handling.

# 13. Future Enhancements

• Implement full backend with Node.js and MongoDB.  
• Add JWT-based authentication.  
• Improve UI/UX with Material UI.  
• Add real-time music streaming integration.  
• Deploy application to cloud platforms.