DOCUMENTATION FOR RHYTHMIC\_TUNES

# Introduction

 Project Title: [RHYTHMIC\_TUNES]

 Team Members:

* VIJAY.G -(Webpage Development)
* DHINESH KUMAR.V -(Documentation)
* ADARSH.P -(Web Design)
* SENTHIL KUMAR.S -(Demo Video)

# Project Overview

 Purpose: Rhythmic Tunes is a music streaming platform designed to provide a seamless experience for discovering, listening to, and managing music playlists. It aims to combine an initiative user interface with powerful search and recommendation features.

 Features:

 User authentication and profile management  Music search and playlist creation

 Streaming music with playback controls  Personalized music recommendations

 Offline listening support

# Architecture

 Component Structure:

 Header Component – Displays navigation and user info

 Sidebar Component – Provides access to playlists and settings  Player Component – Handles music playback and controls

 Search Component – Allows users to search for songs and artists  Playlist Component – Displays user-created playlists and songs

 State Management:

 Global State: Managed using Redux Toolkit for centralized state management

 Local State: Managed within components using the useState and useReducer hooks

 Routing: Uses react-router-dom for navigation between pages (e.g., Home, Search, Playlist)

# Setup Instructions

 Prerequisites:

 Software Requirements:

 Node.js – JavaScript runtime environment  npm – Package manager

 React.js – JavaScript library for building user interfaces

Installation:

 Install Node.js and npm:  Download Node.js

 Create a new React app using Vite:

Navigate to the project directory and install dependencies:

npm create vite@latest

Start the development server:

cd project-name npm install

 Open the app in the browser:

npm run dev

[http://localhost:5173](http://localhost:5173/)

# Folder Structure

src/

├── components/

│ ├── Header.js

│ ├── Sidebar.js

│ ├── Player.js

├── pages/

│ ├── Home.js

│ ├── Search.js

│ ├── Playlist.js

├── assets/

├── hooks/

├── utils/

1. Running the Application

├── state/

└── index.js

 Start the React application:

Start the JSON server:

npm run dev

Open the app in the browser:

json-server --watch ./db/db.json

 Frontend:

[http://localhost:5173](http://localhost:5173/)

 React.js – For building the user interface and state management.  React Router Dom – For navigation and routing.

 React Icons – For embedding icons.

 Bootstrap/Tailwind CSS – For styling components.  Axios – For HTTP requests and data fetching.

# Component Documentation

 Key Components:

 Songs.js – Displays songs and manages playback.

 Favorites.js – Handles adding and removing favorite songs.  Playlist.js – Allows playlist creation and modification.

 App.js – Main component managing routing and state.

# State Management

 Global State: Managed using Redux Toolkit for centralized state management

 Local State: Managed within components using the useState and useReducer hooks.

# User Interface

 Provide screenshots or Home page, search page, playlist view and playback controls.

# Styling

 CSS Frameworks/Libraries: Tailwind CSS for component styling Custom themes defined using CSS variables

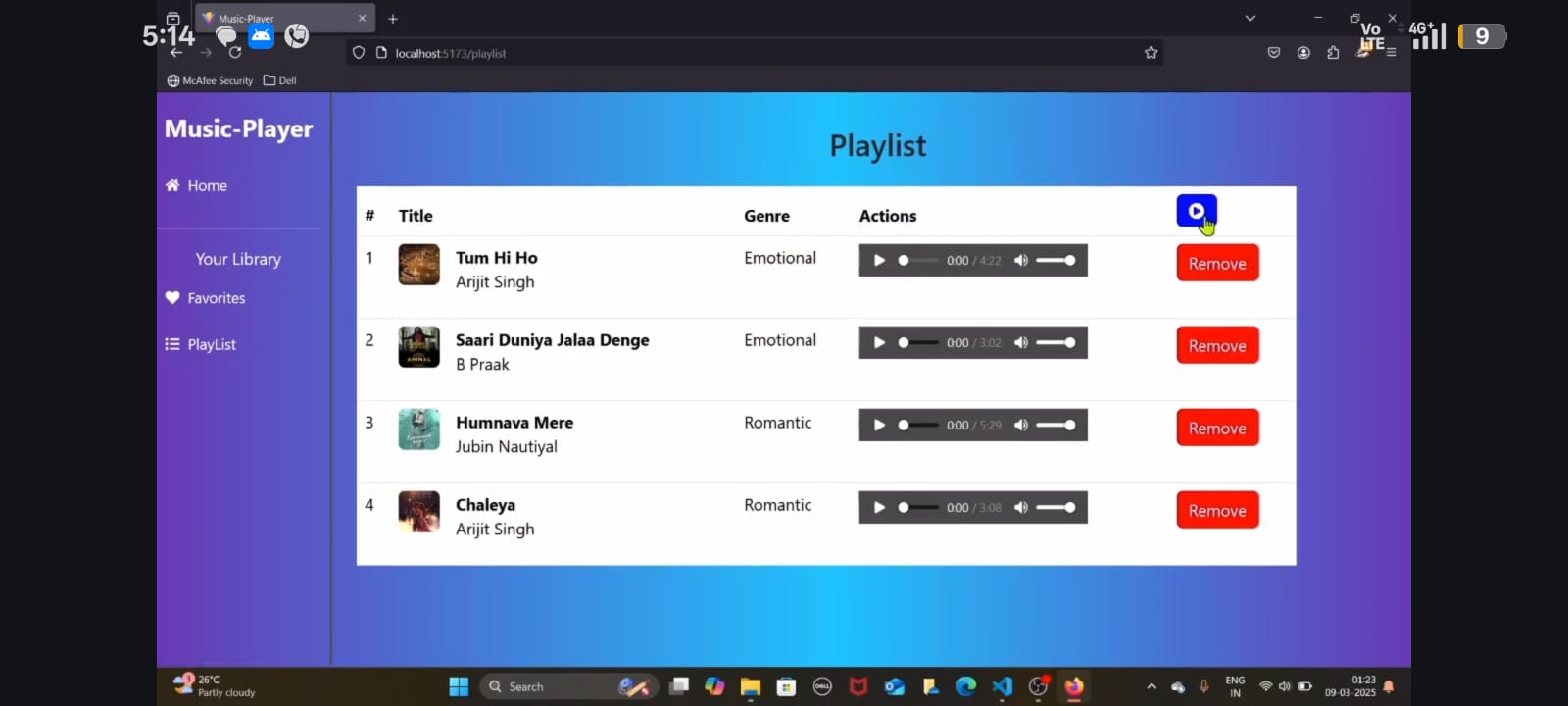
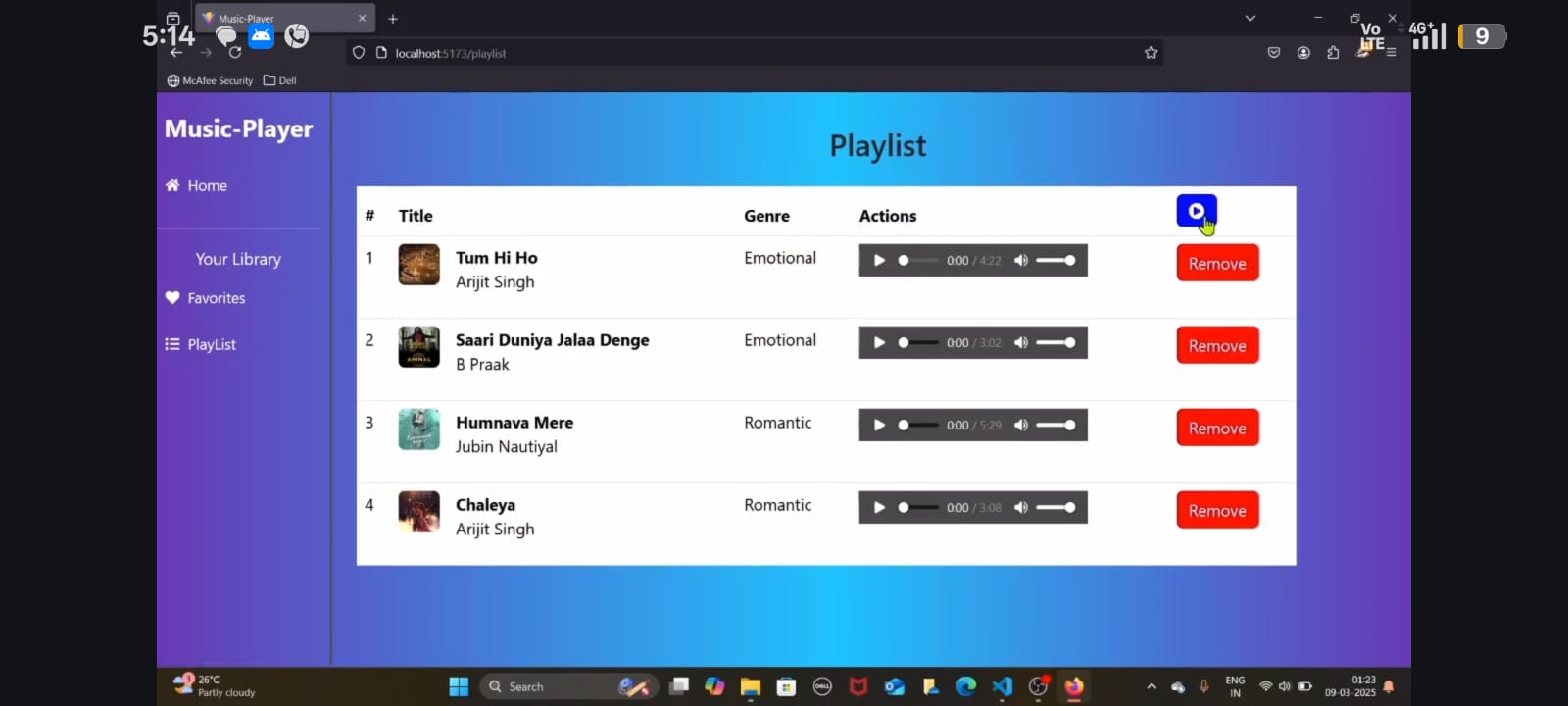
 Theming: Dark mode and light mode available Testing

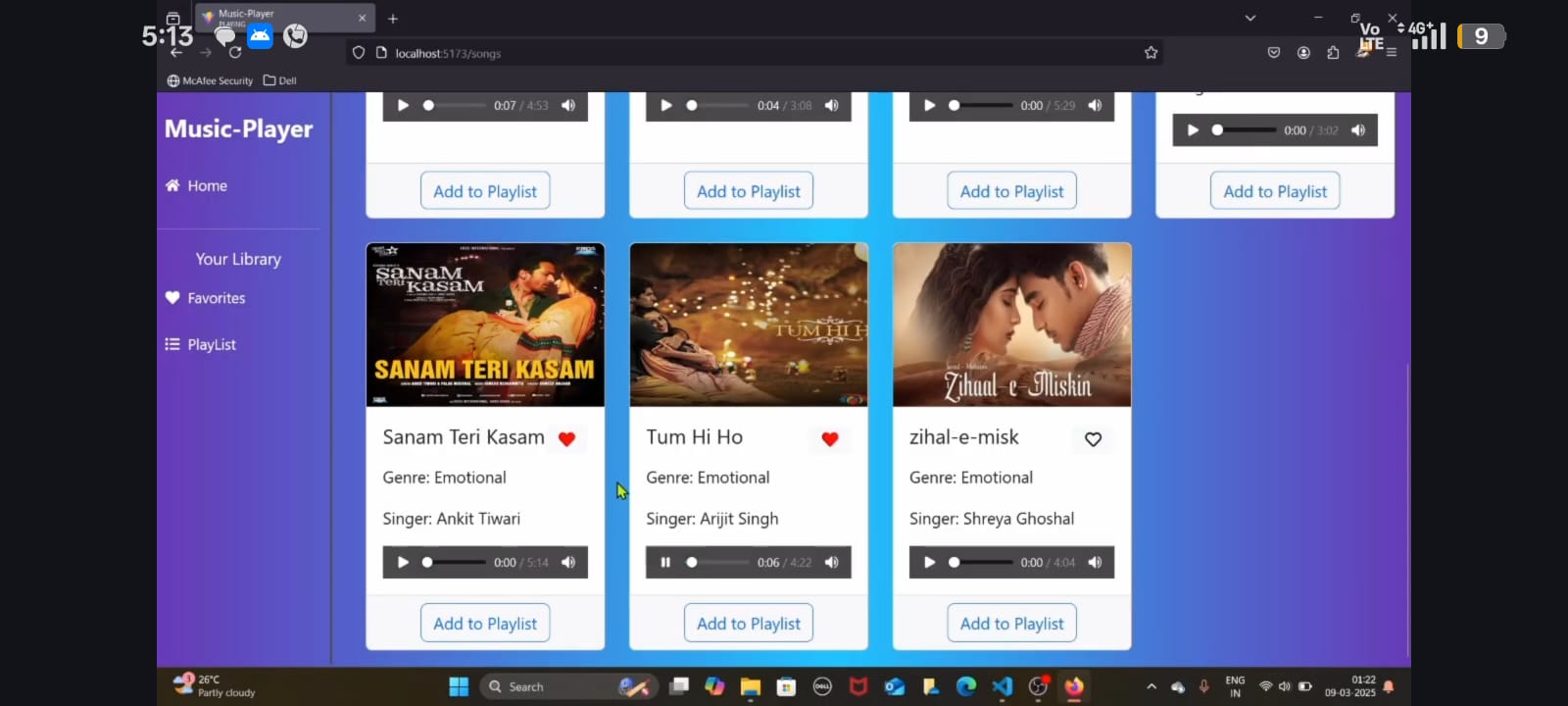
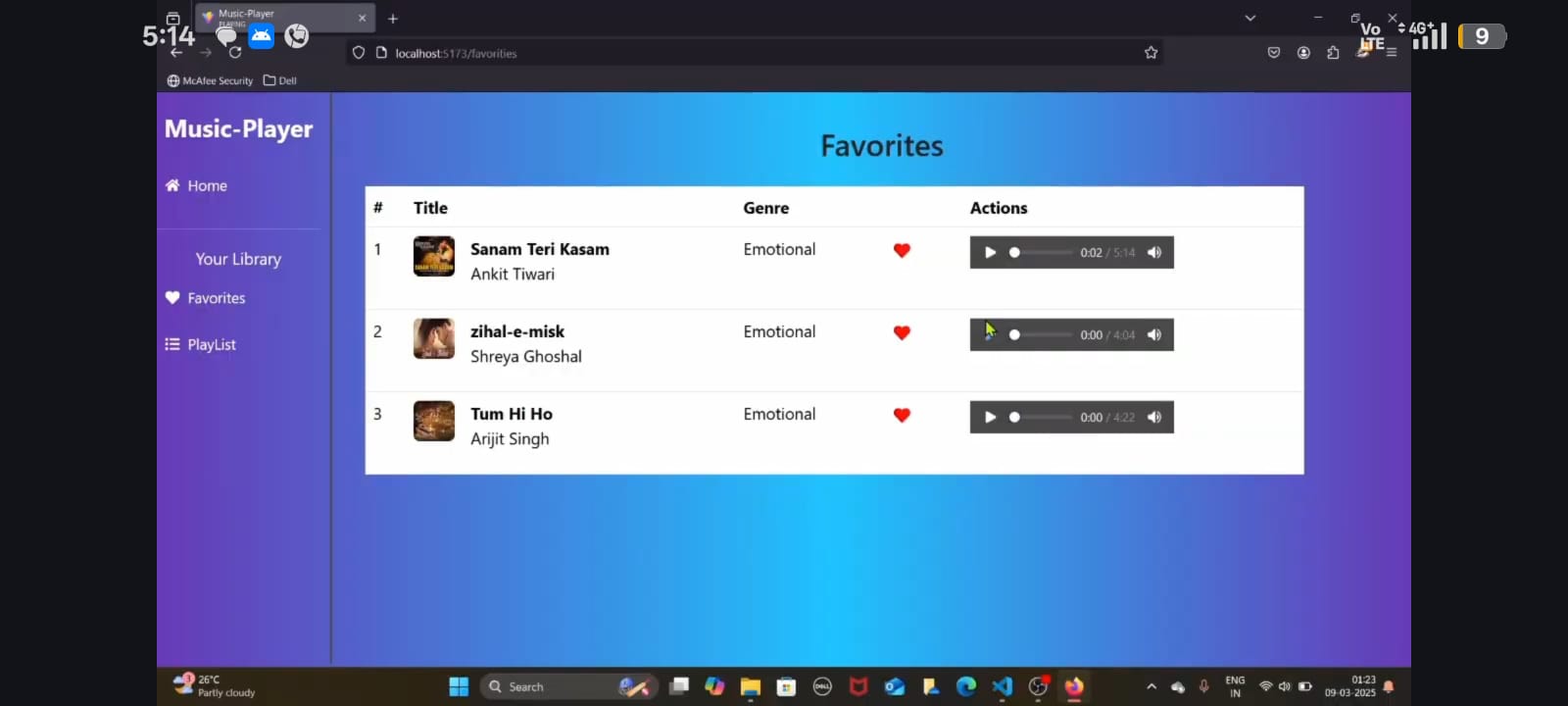
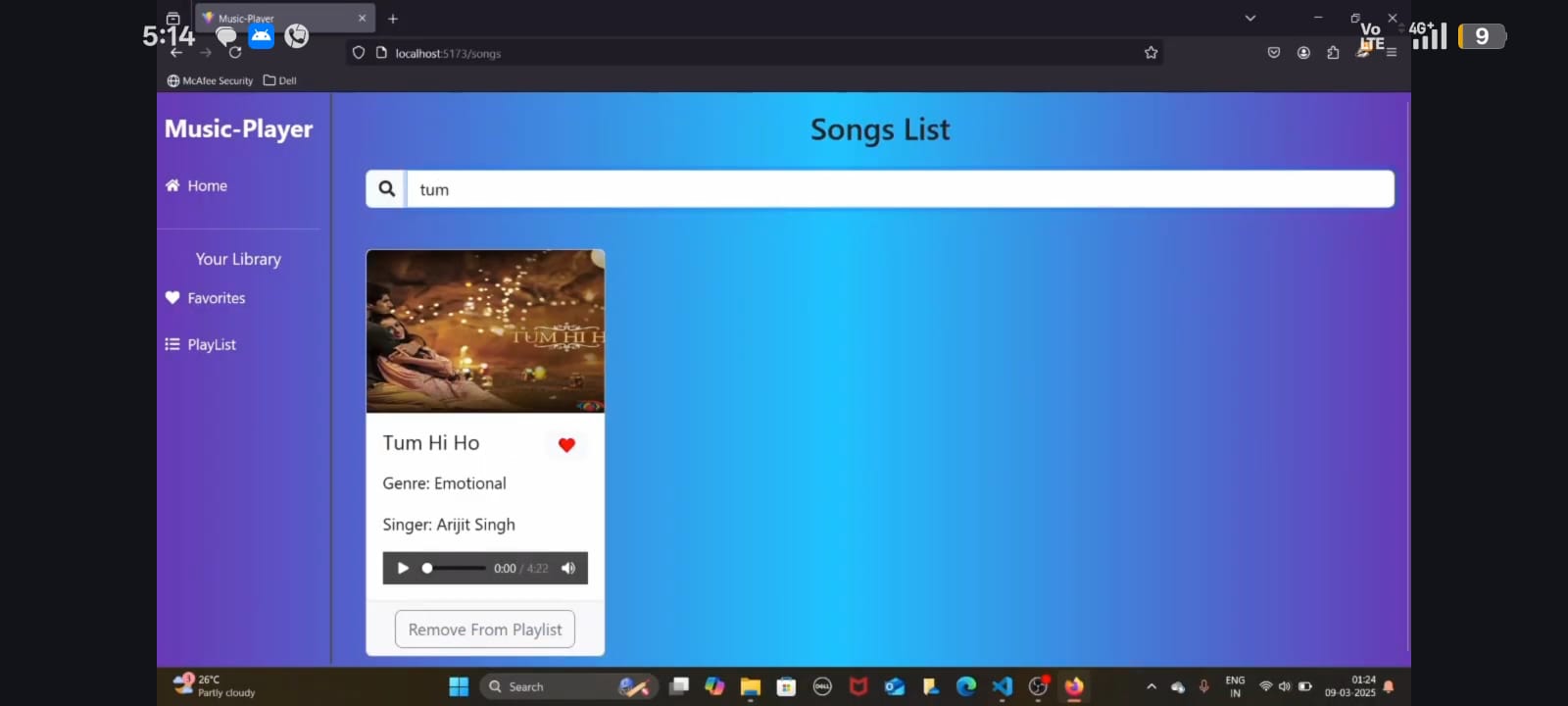
# Testing

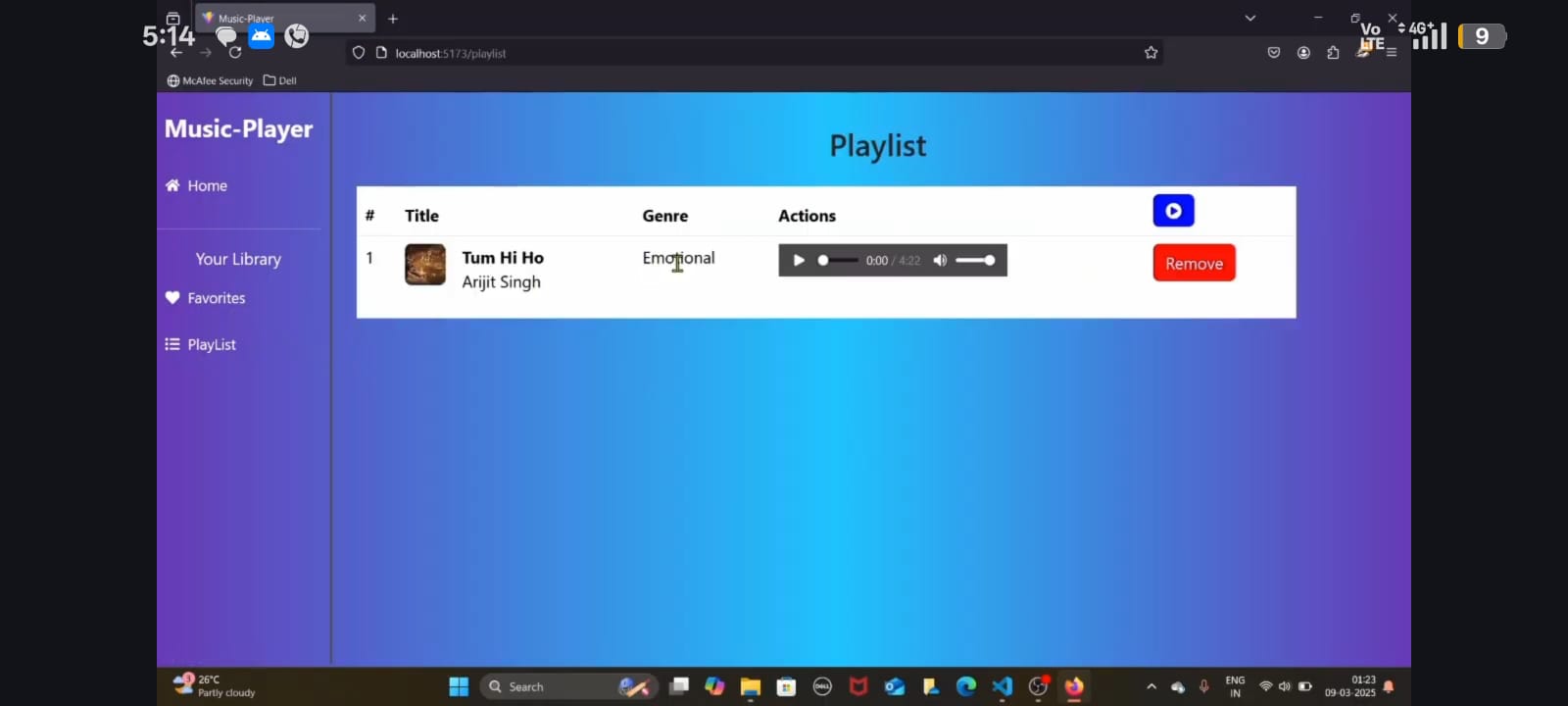
 Testing Strategy: Unit testing with Jest Component testing with React Testing Library

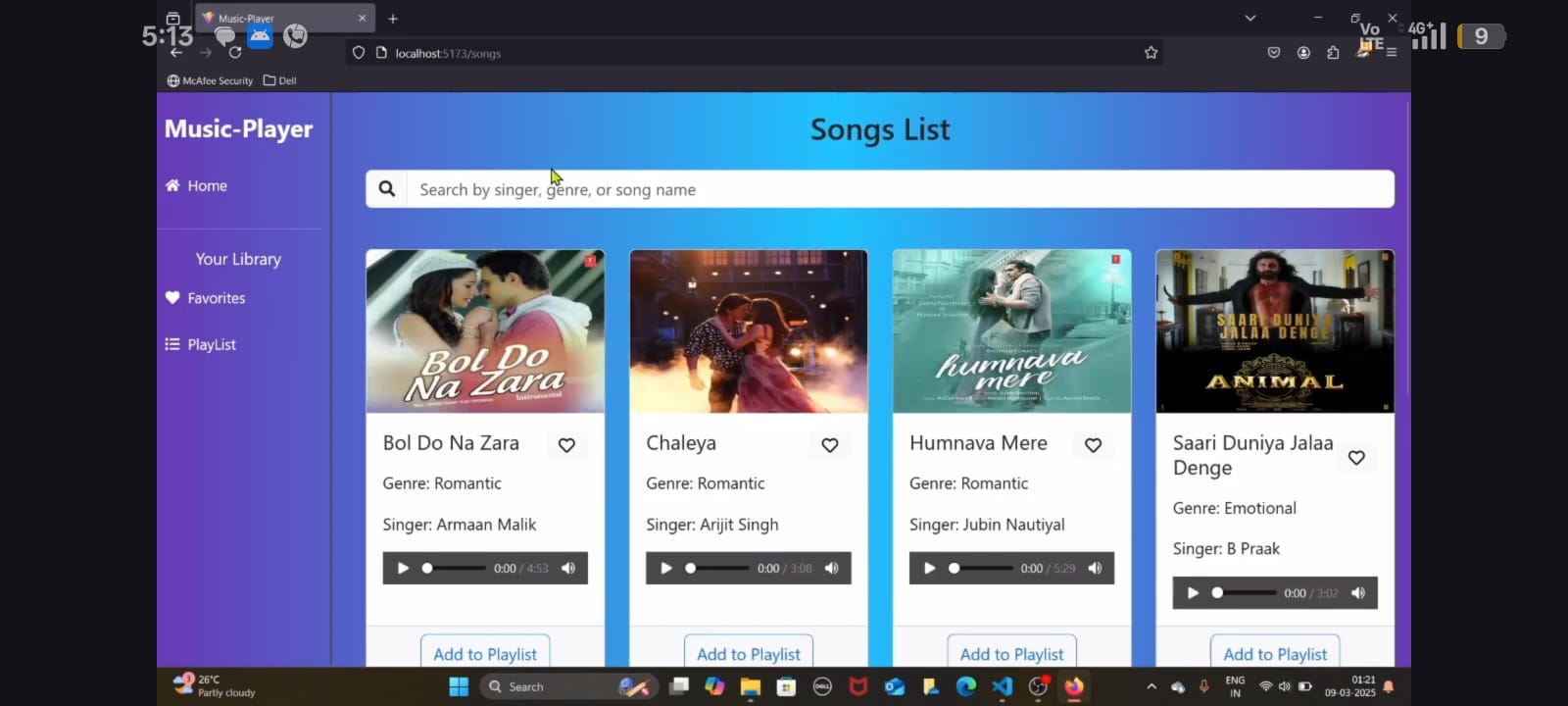
 Code Coverage: Code coverage tracked using Jest

# Screenshots or Demo







# Known Issues

Issues

 Playback stuttering on slow connections  Minor visual bugs in dark mode

Future Enhancements

 Add social sharing for playlists  Integrate lyrics display feature

 Implement a recommendation algorithm based on listening habits Let me know if you'd like to modify or add any details!

# Future Enhancements

Rhythmic Tunes is a dynamic and interactive music streaming platform designed to provide a seamless and engaging user experience. Its use of modern web technologies like React.js ensures a fast, scalable, and responsive application.