**Project Documentation**

# Introduction

* Project Title: Rhythmic Tunes - Music Player
* Team ID:NM2025TMID28799
* Team Leader: [M.ilakkiya&ilakkiyailakkiya2006@gmail.com](mailto:M.ilakkiya&ilakkiyailakkiya2006@gmail.com)
* Team Members:

[M.Abinaya & abinayamaruthiyan95@gmail.com](mailto:M.Abinaya%20&%20abinayamaruthiyan95@gmail.com)

[R.Shivani & shivanirajasekar21@gmail.com](mailto:R.Shivani%20&%20shivanirajasekar21@gmail.com)

A.Anusiya & anushiyavempu@gmail.com

# Project Overview

* Purpose: Rhythmic Tunes is a web-based music player that allows users to browse, play, and manage their favorite songs.
* Features:
* Play, pause, and navigate through songs
* Add songs to favorites
* Create and manage playlists
* Responsive UI with smooth design

# Architecture

* Frontend: React.js with Vite
* Styling: CSS for layout and design
* Backend: JSON-based db.json (can be extended with Node.js/Express)
* Database: db.json (simulated), extendable to MongoDB or SQL

# Setup Instructions

* Prerequisites: Node.js, npm, Vite, Visual Studio Code
* Installation Steps:
* 1. Clone the repository
* 2. Run 'npm install' to install dependencies
* 3. Run 'npm run dev' to start the development server
* 4. Access the app at http://localhost:5173

# Folder Structure

* code/
* |-- public/ (Static assets, songs)
* |-- src/ (React source code)
* |-- Components/ (Playlist, Favorites, UI components)
* |-- App.jsx
* |-- main.jsx
* |-- db/ (db.json sample data)
* |-- package.json (project dependencies)

# Running the Application

* Frontend:
* cd code
* npm run dev
* Access: Visit http://localhost:5173

# API Documentation

* Since the project currently uses db.json as a local database, APIs can be extended with Node.js/Express.
* Future API endpoints may include:
* /api/songs
* /api/playlists
* /api/favorites

# Authentication

* Currently, no authentication implemented.
* Future enhancement: JWT-based authentication for secure login.
* Middleware to protect playlists and favorites routes.

# User Interface

* Home Page with song list
* Playlist management page
* Favorites section
* Music controls (play, pause, next, previous)

# Testing

* Manual testing performed during development.
* Browser Developer Tools for debugging.
* Tested on Chrome and Edge browsers.

# Screenshots or Demo

• Screenshots can be added to showcase UI pages like Home, Playlist, Favorites, and Music Controls. A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Known Issues

* Limited functionality without backend integration.
* Songs must be placed in /public/Songs manually.
* No authentication system implemented yet.

# Future Enhancements

* Integrate backend with Node.js and MongoDB.
* Add user authentication and login system.
* Implement search and filter for songs.
* Add album art and lyrics display.
* Improve UI with advanced animations and themes.