Rithamic Tunes - Project Documentation

# Introduction

Project Title: Rithamic Tunes

Team Members:

1. T. Deivanai – Project Head, and JavaScript Developer
2. S. Devadharshana – HTML Developer
3. M. Swetha– CSS Developer
4. G. Sandhiya– Tester & Debugger
5. S. Swetha – UI/UX Designer

# Project Overview

## Purpose

The purpose of this project is to create a simple web-based Music Player system that allows users to listen to songs, manage favorites, and maintain playlists.

## Features

* Home Page (music.html) to view and play songs.
* Add songs to ❤️ Favorites (fav.html).
* Organize and manage 🎶 Playlists (playlist.html).
* Central hub 📂 Your Library (library.html) with counters for favorites and playlists.
* Custom CSS for user interface styling.
* JavaScript for dynamic functionality and localStorage handling.

# Architecture

## Component Structure

The system consists of four main HTML pages:  
  
music.html – Displays songs with search, audio preview, and add-to-favorites/playlist options.  
  
fav.html – Shows songs added to favorites.  
  
playlist.html – Displays songs added to playlist in table format.  
  
library.html – Central page to access Favorites and Playlists with counters.

## State Management

The project uses JavaScript variables and localStorage for temporary data storage. Actions such as adding to Favorites or Playlist automatically update counters and reflect changes across pages.

## Routing

Navigation between pages is achieved via HTML links in the sidebar. No advanced routing library is used.

# Setup Instructions

## Prerequisites

A modern web browser (Chrome, Firefox, Edge).  
Visual Studio Code (VS Code)

## Installation

1. Clone or download the project folder.  
2. Ensure all HTML, CSS, and JS files are in the same directory.  
3. Open music.html in your web browser to start the application.

# Folder Structure

Project Folder:  
- music.html (Songs Page)  
- fav.html (Favorites)  
- playlist.html (Playlist)  
- library.html (Your Library)  
- music.css, fav.css, playlist.css, library.css  
- music.js, fav.js, playlist.js, library.js

# Running the Application

Option 1 – Run directly in a Web Browser (Recommended):  
Open the project folder on your computer.  
Double-click on music.html to open it in your default web browser.  
Navigate to Favorites, Library, or Playlist using the sidebar.  
  
Option 2 – Run via Live Server in VS Code:  
Open the project folder in VS Code.  
Right-click on music.html and select 'Open with Live Server'.  
The application will run locally in your browser.

# Component Documentation

## music.html

Provides navigation links to all other modules (Library, Favorites, Playlist).  
Displays the Songs Catalog with search functionality.  
Each song has an audio player, Add to Playlist button, and Favorite icon.  
Uses music.css for styling and music.js for dynamic functionality.

## fav.html

Displays songs marked as Favorites by the user.  
Uses fav.css for styling and fav.js for handling Favorites data.

## playlist.html

Displays user’s playlist in a table with Title, Singer, Preview, and Action (remove).  
Uses playlist.css for styling and playlist.js for managing playlist state.

## library.html

Centralized page to access Favorites and Playlists.  
Shows counters for number of songs in Favorites and Playlists.  
Uses library.css for styling and library.js for functionality.

# State Management

The project uses JavaScript variables and localStorage to persist data temporarily.  
For example:  
- Adding songs to Favorites updates the favorite counter in sidebar and library page.  
- Adding songs to Playlist updates the playlist counter and displays songs in playlist.html.  
These updates are reflected across all pages via localStorage.

# User Interface

The UI consists of HTML pages with a sidebar for navigation, styled using custom CSS. The design is clean with counters for Favorites and Playlists, and audio previews for songs.

# Styling

The project uses custom CSS styling for each page:  
- music.css, fav.css, playlist.css, library.css.  
Emojis are used for intuitive navigation (🏠, 📂, ❤️, 🎵).

# Testing

Manual testing has been performed:  
- Adding/removing Favorites updates correctly.  
- Adding/removing songs from Playlist works properly.  
- Search functionality works on music.html.  
- Navigation between pages works correctly.

# Screenshots or Demo

Screenshots of the UI or Demo video will be added later.

# Known Issues

Data does not persist after browser cache is cleared.  
No authentication system implemented.

# Future Enhancements

- Add backend support with a real database for persistent storage.  
- Implement user authentication and roles.  
- Improve UI with frameworks like Bootstrap or Tailwind CSS.  
- Add advanced filtering and sorting options in songs list.  
- Enable playlist export/import.