# Project documentation

**Rhythmic Tunes :**

**Your Melodic Companion**

**1.Introduction:**

**Project Title**: Rhythmic Tunes

**Team ID** : NM2025TMID35179

**Team Leader** : BOOMIKA B

boomikababup@gmail.com

**Team members :**

SUBITHA BEGUM S

subithabegum9@gmail.com

HEMAVATHI K

hemahemavathi046@gmail.com

KEERTHANA S

sagakeerthana0@gmail.com

SARUMATHI T

[thiruppathithiruppathi246@gmail.](mailto:thiruppathithiruppathi246@gmail.com)com

**2.Project overview:**

**Purpose:**

1. Learning & Practice:

To practice HTML5 for structuring webpages.

To apply CSS3 for styling, layouts, and design consistency.

To understand how to link multiple webpages and connect them using navigation.

2. User Engagement:

Provide a homepage introducing the platform.

Allow users to navigate to a main content page where they can see:

Artists (with images and names).

Playlists (linked to real Spotify content).

Search bar for better music discovery (UI only).

**Features:**

1. Homepage (rhythmictunes.html)

Spotify-style landing page with project logo and title.

“Go to Spotify” button that navigates to the main content page.

Clean black-and-white theme to match Spotify’s branding.

2. Main Page (spotify.html)

Sidebar Navigation

Home, Your Library, Favorites, and Playlist sections for easy navigation.

Input field to search by singer, genre, or song name (currently static).

3. Design & Styling (style.css)

Consistent theme with black background

**3.Architecture:**

1. Frontend: Html,Css

Role:Defines the structure and design of the website.

Homepage (rhythmictunes.html)

Main Page (spotify.html) with sidebar, search bar, artists, about section, and footer

Applies consistent styling via style.css.

2. Backend :(Application Layer) (Future Scope)

Future Enhancement:

Backend can be developed using Node.js, Django, or Flask.

Manages user requests (e.g., searching songs, handling favorites, creating playlists).

3. Database: (Data Layer) (Future Scope)

Current Project Status: No database used (static demo).

Future Enhancement:Store user information, playlists, and favorites in a database.

Suggested databases: MySQL, MongoDB, or Firebase.

Maintains persistent data for personalized music.

**4.Setup Instructions:**

Prerequisites:

1.Git hub

2.React.js

3.visual studio code

##### **Installation Steps:**

1. Download / Copy the Project Folder

Save the folder RHYTHMIC TUNES on your computer.

2. Verify Folder Structure

Ensure the folder looks like this:

RHYTHMIC TUNES/

├── rhythmictunes.html

├── spotify.html

├── style.css

└── img/

├── spotify.png

├── HARRISH.jpg

├── HIPHOP.jpg

├── Vijay Antony.jpg

3. Open the Project in a Browser

Double-click rhythmictunes.html to launch the homepage in your default browser.

Alternatively, open the project in Visual Studio Code → Right-click → “Open with Live Server” (requires Live Server extension).

4. Navigate Through the Website

From the homepage, click “Go to Spotify” to access the main page (spotify.html).

**5.Folder structure:**

Rhythmic-Tunes/ ← Main project folder

│

├── rhythmictunes.html ← Homepage (main entry page)

├── spotify.html ← Songs/Spotify linked page

├── style.css ← External CSS stylesheet

│

├── img/ ← Images folder

│ ├── spotify.png

│ ├── ARR.jpg

│ ├── D.IMAN.jpg

│ ├── Ilayaraja.png

│ ├── HARRISH.jpg

│ ├── HIPHOP.jpg

│ ├── Vijay Antony.jpg

│ ├── YUVAN.jpg

│ ├── GVP.jpg

│ └── other images…

│

└── README.md ← (optional, for project description)

Benefits of this structure:

All HTML files are in the root folder → easy navigation.

CSS styles are in one place → linked via <link rel="stylesheet" href="style.css">.

All images are grouped inside img/ → avoids clutter and makes linking simpler.

Example of how you should link files:

<!-- Linking CSS -->

<link rel="stylesheet" href="style.css">

<!-- Linking Images -->

<img src="img/ARR.jpg" alt="A.R. Rahman">

**6.Running the Application** :

Step 1: Open the Project Folder

Locate the folder named RHYTHMIC

TUNES on your computer.

Ensure it contains the required files:

rhythmictunes.html

spotify.html

style.css

img/ folder with images.

Step 2: Launch the Homepage

Double-click on rhythmictunes.html.

It will automatically open in your default web browser (Chrome, Firefox, Edge, Safari).

Step 3: Navigate to the Main Page

On the homepage, click the “Go to Spotify” button.

This will open spotify.html, the main content page of the project.

Step 4: Explore the Features

Inside spotify.html, you can:

Use the sidebar navigation (Home, Your Library, Favorites, Playlist).

Enter text in the search bar (UI only, no backend search).

View artist cards (Harris Jayaraj, HipHop Tamizha, Vijay Antony) with images and names.

Click artist links → Redirects to \*\*real Spotify profiles/play

**7.API Documentation:**

User:/api/users

Methods:

POST /api/users/register → Register a new user.

POST /api/users/login → Authenticate user and return token.

GET /api/users/{id} → Fetch user details.

PUT /api/users/{id} → Update user profile.

DELETE /api/users/{id} → Delete user account.

Sample Response:

{

"id": 101,

"username": "musiclover",

"email": "user@example.com",

"favorites": ["HipHop Tamizha", "Harris Jayaraj"]

}

Projects:/api/projects

Methods:

GET /api/projects → Fetch all music projects/playlists.

GET /api/projects/{id} → Fetch details of a specific project.

POST /api/projects → Create a new music project/playlist.

PUT /api/projects/{id} → Update project details.

DELETE /api/projects/{id} → Remove a project.

Sample Response:

{

"id": 201,

"title": "Tamil Hits",

"description": "Best of Tamil music collections",

"songs": [

{"title": "Vaseegara", "artist": "Harris Jayaraj"},

{"title": "Sodakku", "artist": "HipHop Tamizha"}

]

}

Chats:/api/chats

Methods:

GET /api/chats/{userId} → Fetch chat history for a user.

POST /api/chats → Send a new chat message.

DELETE /api/chats/{chatId} → Delete a chat message.

Sample Response:

{

"chatId": 301,

"sender": "musiclover",

"receiver": "admin",

"message": "I love Harris Jayaraj songs!",

"timestamp": "2025-09-11T12:45:00Z"

}

**8.Authentication:**

Purpose of Authentication:

Ensure only authorized users can access certain features.

Allow users to save favorites, create playlists, and manage profiles.

Protect sensitive user data such as email, password, and playlists.

Enable role-based access (e.g., normal user vs. admin).

Types of Authentication: (Planned)

1. Username & Password Login

Users create an account with email and password.

**9.User Interface:**

The User Interface (UI) of Rhythmic Tunes is designed to be simple, clean, and Spotify-inspired, ensuring ease of navigation for music lovers.

Usability → Sidebar + main content area for smooth user flow.

**10.Testing:**

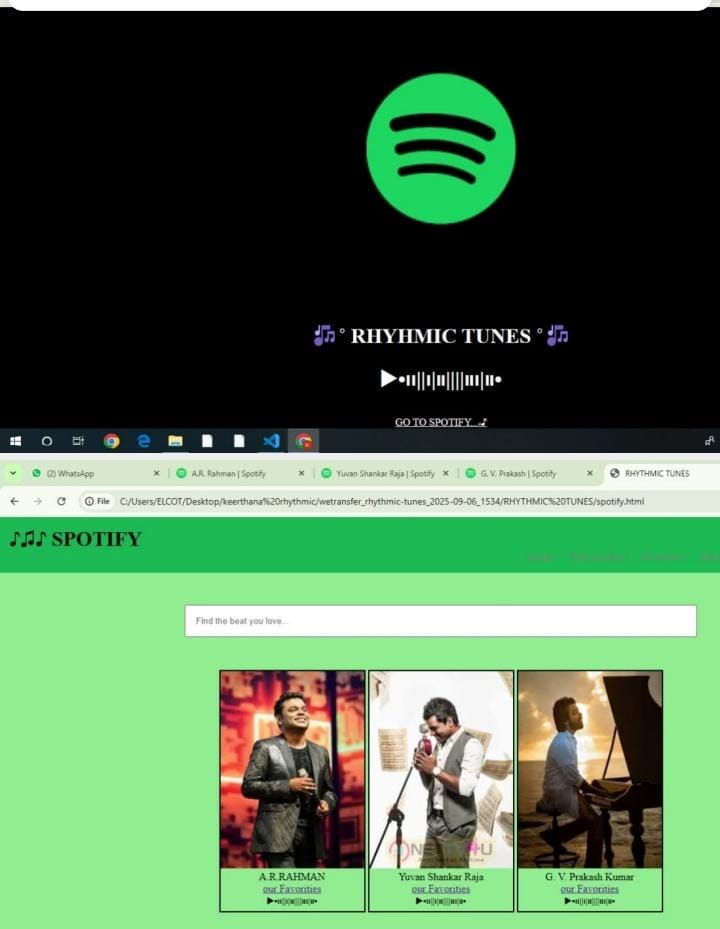
1. \*Check HTML Structure\*: Ensure the HTML structure for the "products" section is rendering correctly with artist names (like Ilaiyaraaja and Imman) and links to their Spotify pages.

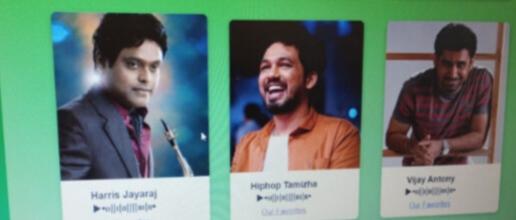
2. \*Image Rendering\*: Verify that images for each artist (`img/D.IMAN.jpg` and `IMG/Ilaiyaraaja.png`) are loading properly with the correct alt text ("song").

3. \*Links Functionality\*: Test if the "Our Favorites" links redirect correctly to the respective Spotify artist pages.

4. \*About Section\*: Check if the "About" section displays the heading "The Vibe" and the paragraph describing the project's belief about music.

**11.Screenshort or Demo:**







1. Beat – The basic unit of time in music, like the heartbeat of a song

2. Rhythm – The pattern of sounds and silences (arrangement of long and short notes).

3.Melody – A sequence of notes that sound plea

Search functionality (by singer, genre, song).

Library, Favorites, Playlist features.

Focus on Rhythm (tempo variations, beats, background scores).

Artist showcase (A. R. Rahman, Yuvan, G. V. Prakash, etc.).

**12. Known Issues:**

Issue 1: The volume control slider may sometimes be unresponsive on certain browsers.

Issue 2: Slow loading time for large playlists.

**13.Future Enhancements:**

Offline Mode: Allow users to listen to previously downloaded songs when they’re offline.

Expanded Playlist Features: Add more advanced playlist options, such as collaborative playlists and auto-generated playlists based on user preferences.