Project Documentation

**Rhythmic Tunes**

# Introduction

## Project Title: Rhythmic Tunes

* + **Team ID: NM2025TMID36996**
  + **Team Leader:** MARIA SELI SANTRA R ([mariaselisantra03@gmail.com](mailto:mariaselisantra03@gmail.com))

## Team Members:

* + - MAHANADHI M ([msmahanadhi@gmail.com](mailto:msmahanadhi@gmail.com))
    - MALAVIKA M ([mmalavikanirmala@gmail.com](mailto:mmalavikanirmala@gmail.com))
    - MANJU M ([manjudhivya19@gmail.com](mailto:manjudhivya19@gmail.com))

# Project Overview

* + **Purpose:** The primary goal of Music Streaming is to provide a seamless platform for music enthusiasts, enjoying, and sharing diverse musical experiences

## Features:

* User-Friendly Interface: Develop an intuitive interface that allows users to effortlessly explore, save, and share their favorite music tracks and playlists.
* Comprehensive Music Streaming: Provide robust features for organizing and managing music content, including advanced search options for easy discovery.
* Modern Tech Stack: Harness cutting-edge web development technologies, such as React.js, to ensure efficient and enjoyable user experience while navigating and interacting with the music streaming application.

# Architecture

* + **Frontend:** 
    - Framework: React
    - Tooling: Vite, which is a modern frontend build tool optimized for speed and performance.
    - Styling: CSS files such as App.css and index.css are used for styling components.
    - Component Structure: The src/Components folder likely contains reusable React components.
    - Assets: Static files like images, icons, or fonts are stored in src/assets.
  + **Backend:** Not used.
  + **Database:** Not used.

# Setup Instructions

## Prerequisites:

* + - Node.js and npm
    - React.js
    - HTML, CSS, and JavaScript:
    - Git
    - Visual Studio Code:

## Installation Steps:

## Node.js and npm:

## Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

## Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

## Download: https://nodejs.org/en/download/

## Installation instructions: https://nodejs.org/en/download/package-manager/

## React.js:

## React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

## Install React.js, a JavaScript library for building user interfaces.

## Create a new React app:

## npx create-react-app my-react-app

## Replace my-react-app with your preferred project name.

## Navigate to the project directory:

## cd my-react-app

## Running the React App:

## With the React app created, you can now start the development server and see your React application in action.

## Start the development server:

## npm start

## This command launches the development server, and you can access your React app at http://localhost:3000 in your web browser.

## Version Control:

## Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

## Git: Download and installation instructions can be found at: https://git-scm.com/downloads

## Development Environment:

## Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

## Visual Studio Code: Download from https://code.visualstudio.com/download

## Sublime Text: Download from https://www.sublimetext.com/download

## WebStorm: Download from https://www.jetbrains.com/webstorm/download

## Follow below steps to install Dependencies:

## Navigate into the cloned repository directory and install libraries:

## cd Code

## npm install

## Start the Development Server:

## To start the development server, execute the following command: npm start

## Access the App: Open your web browser and navigate to http://localhost:3000.

## You should see the rhythmic tunes app's homepage, indicating that the installation and setup were successful.

# Folder Structure

RHYTHMIC TUNES/

│

├── code/

│

├── public/ ← Frontend (static assets)

│ └── Songs/ ← Frontend (media files used in the app)

│ ├── chaleya.mp3

│ ├── Humnava Mere.mp3

│ ├── Saari Duniya Jalaa.mp3

│ └── vite.svg

│

├── src/ ← Frontend (React source code)

│ ├── assets/ ← Frontend (images, icons, etc.)

│ ├── Components/ ← Frontend (React components)

│ ├── App.css ← Frontend (styling)

│ ├── App.jsx ← Frontend (main app component)

│ ├── index.css ← Frontend (global styles)

│ └── main.jsx ← Frontend (entry point)

│

├── .eslintrc.cjs ← Frontend (linting config)

├── .gitignore ← General (Git config)

├── index.html ← Frontend (HTML entry point)

├── package-lock.json ← Frontend (dependency lock file)

├── package.json ← Frontend (project metadata and dependencies)

├── README.md ← General (documentation)

└── vite.config.js ← Frontend (Vite build configuration)

# Running the Application

## Frontend:

## cd Code

npm install

npm run dev

* + **Access:** Visit <http://localhost:5173/songs>

# User Interface

* + Home Page
  + Your Library
  + Favorites
  + Playlist
  + Songs list

# Testing

* + Manual testing done against the user interface

1. **Screenshots:**







