Frontend Development with React.js

Project Documentation – Rhymetic Tunes

1. Introduction

Project Title: Rhymetic Tunes

Team Members: [List names and roles, e.g., Developer, Designer, Tester]

2. Project Overview

Purpose: Rhymetic Tunes is a web-based music platform designed to let users explore, create, and enjoy rhythm-based tunes interactively.

Features:

User-friendly interface for browsing and playing tracks.

Playlist creation and management.

Interactive visualization of tunes with rhythmic animations.

Responsive design for mobile and desktop.

3. Architecture

Component Structure:

Header (Navigation, Logo, Search)

Music Player (Play, Pause, Next, Previous, Progress Bar)

Playlist Manager (Add, Remove, Edit Playlists)

Tune Visualizer (Animated rhythm display)

State Management: Context API for user sessions and playlist data.

Routing: React Router for navigation (Home, Playlists, Player, About).

4. Setup Instructions

Prerequisites: Node.js, npm/yarn.

Installation:

git clone https://github.com/your-repo/rhymetic-tunes.git

cd rhymetic-tunes

npm install

npm start

5. Folder Structure

Client:

/src/components → UI Components (Player, Playlist, Header, etc.)

/src/pages → Home, Playlists, About

/src/assets → Images, Icons, Audio samples

Utilities:

Custom hooks (e.g., useAudioPlayer)

Helper functions for audio formatting

6. Running the Application

Start the frontend server:

npm start

7. Component Documentation

Key Components:

Player.js: Controls playback and displays track info.

Visualizer.js: Shows rhythmic animations.

Reusable Components:

Button.js, Card.js, Modal.js

8. State Management

Global State: User session, active track, and playlist data.

Local State: Play/pause toggle, volume, track progress.

9. User Interface

Screenshots/GIFs of:

The music player in action

Playlist creation

Visualizer animations

10. Styling

CSS Frameworks/Libraries: Tailwind CSS + Styled Components.

Theming: Dark/Light mode toggle with custom theme support.

11. Testing

Unit testing with Jest + React Testing Library.

Manual testing for audio playback and routing.