**Frontend Development with React.js**

**Project Documentation**

**Introduction**

**Project Title:**  
Rhythmic Tunes: Your Melodic Companion

**Team Members:**

* **Team Leader:** RAGUPATHI S
* **Team Member:** PRANAV V
* **Team Member:** MURALI M
* **Team Member:** MUNIRAJ M

**Project Overview**

**Purpose:**  
The purpose of Rhythmic Tunes is to create a user-friendly platform where music enthusiasts can discover, listen, and manage tunes effortlessly. It serves as a **melodic companion** for users, combining simple design with powerful functionality.

**Features:**

* Browse and play different music tracks.
* Add, view, and manage playlists.
* Interactive UI with smooth navigation.
* Local JSON server to simulate backend for storing data.

**Architecture**

**Component Structure:**

* **App.js** – Root component that manages routing and layout.
* **Header / Footer** – Common layout components.
* **Home Page** – Displays music categories or featured tracks.
* **Playlist Page** – Allows users to view and manage their playlists.
* **Player Component** – Plays the selected track with controls (play, pause, skip).

**State Management:**

* Local state with React hooks (useState, useEffect).
* Data fetched from **JSON server (db.json)** for persistence.

**Routing:**

* Implemented with **react-router-dom**:
  + / → Home
  + /playlist → Playlist Page
  + /player/:id → Music Player

**Setup Instructions**

**Prerequisites:**

* Node.js (LTS version)
* VS Code

**Installation Steps:**

1. Download project source code from SmartInternz resources.
2. Extract the zip file.
3. Open the project in **VS Code**.
4. Run in terminal:
5. npm install
6. npm run dev

→ App runs at: http://localhost:5173

**JSON Server Setup:**

1. Open a new terminal, navigate to **db** folder:
2. cd db
3. npm i -g json-server
4. json-server --watch db.json --port 3000

→ JSON server runs at: http://localhost:3000

**Folder Structure**

project-folder/

│── src/

│ ├── components/

│ ├── pages/

│ ├── assets/

│ ├── App.js

│ └── main.js

│── db/

│ └── db.json

**Running the Application**

* **Frontend:**
* npm run dev
* **Backend (JSON Server):**
* json-server --watch db.json --port 3000

**Component Documentation**

* **Header** – Navigation bar for pages.
* **Playlist** – Displays user’s saved songs.
* **Player** – Music controls for play, pause, skip.
* **SongCard** – Reusable component to display a single track.

**User Interface**

* Homepage → Displays featured tracks.
* Playlist Page → User-created music collections.
* Music Player → Interactive player with controls.

**Styling**

* **CSS:** Default CSS with modular structure.
* **Frameworks/Libraries:** Tailwind CSS (if included).
* **Design:** Clean and minimal for easy usability.

**Testing**

* Basic testing with React’s built-in error boundaries.
* Future plan: Jest & React Testing Library for unit tests.

**Screenshots or Demo**

* Demo Video → Available in SmartInternz portal.
* Live Demo → Runs on http://localhost:5173

**Known Issues**

* Currently works with JSON server only (not connected to real backend).
* Limited number of music tracks (demo data).

**Future Enhancements**

* User authentication & personalized playlists.
* Dark/Light theme toggle.
* Integration with real-time music APIs.
* Advanced player features (loop, shuffle, equalizer).