**Frontend Development with React.js**

**Project Documentation**

**Introduction**

**Project Title:** Music-Player (React + Vite + TailwindCSS)

**Team Members:**

* Aravind -Team leader
* Bigwin -Team member
* Boomivasagan -Team member
* Jaikishore -Team member

**Project Overview**

**Purpose:**  
The project is a modern music player built with React.js, bundled using Vite for fast development, and styled with TailwindCSS. Its purpose is to provide users with an interactive and visually appealing interface for playing and managing music.

**Features:**

* Responsive music player UI
* Fast rendering with React + Vite
* Styled using TailwindCSS
* Component-based structure for scalability

**Architecture**

**Component Structure:**

* **App.jsx** – Root component managing routes and global state
* **Player Component** – Handles play/pause, next, previous actions
* **SongList Component** – Displays a list of available tracks
* **Controls Component** – Music controls (play, pause, seek)
* **Navbar/Footer** – Navigation and layout organization

**State Management:**

* React **useState** and **useEffect** for local state
* Props drilling for communication between parent and child components
* (Optional future) Context API for global state management

**Routing:**

* Currently a single-page application mounted in index.html via #root
* Future enhancement: React Router for multiple views

**Setup Instructions**

**Prerequisites:**

* Node.js (v14 or above)
* npm or yarn

**Installation:**

# Clone repository

git clone <your-repo-url>

# Navigate to project folder

cd music-player

# Install dependencies

npm install

# Start development server

npm run dev

**Folder Structure**

project-root/

│── node\_modules/ # Installed dependencies

│── public/ # Static files

│── src/ # React source code

│ ├── components/ # Reusable UI components

│ ├── pages/ # Page-level components

│ ├── assets/ # Images, icons, media

│ ├── App.jsx # Root component

│ └── main.jsx # Entry point

│── index.html # Root HTML file

│── .gitignore # Ignored files and directories

│── .eslintrc.cjs # ESLint configuration

│── package.json # Project metadata and dependencies

**Utilities:**

* TailwindCSS utility classes for styling
* ESLint for maintaining code quality

**Running the Application**

npm run dev

This starts the Vite development server. Open http://localhost:5173/ in the browser.

**Component Documentation**

**Key Components:**

* **Player.jsx** – Core music player logic
* **SongList.jsx** – Displays all available songs
* **Controls.jsx** – Play/pause, skip, volume control

**Reusable Components:**

* Buttons with Tailwind utility classes
* Card layout for songs

**State Management**

**Global State:**

* Not yet implemented (planned with Context API/Redux)

**Local State:**

* useState for managing current track, play/pause state, and volume
* useEffect for updating UI based on user interaction

**User Interface**

* **TailwindCSS** styled responsive design
* Mobile-friendly layout
* Dark/light mode possible with Tailwind’s theming

**Styling**

**CSS Frameworks/Libraries:**

* TailwindCSS (via CDN in index.html)

**Theming:**

* Currently minimal custom styling, extendable with Tailwind config

**Testing**

**Testing Strategy:**

* Planned with **Jest** and **React Testing Library** for unit tests
* Integration testing for component interactions

**Code Coverage:**

* Not yet set up (can use Jest + Istanbul for coverage reports)

**Screenshots or Demo**

* (Add screenshots of the player UI once available)
* (Optional) Host demo on **Vercel/Netlify**

**Known Issues**

* No routing support yet
* Global state management missing
* No persistent music library

**Future Enhancements**

* Add React Router for multiple pages
* Implement Context API/Redux for state
* Add animations with Framer Motion
* Dark/light theme toggle
* Cloud-based music streaming integration