**Project Documentation**

**Project Title: Rhythmic Tunes**

**Introduction**

• Project Title: Rhythmic Tunes

**•** Team ID: NM2025TMID30057

**•** Team Leader**:** [**PUSHPAVALLI.S - 202400107@sigc.edu**](mailto:PUSHPAVALLI.S%20%20%20-%20%20%20%20%20202400107@sigc.edu)

**• Team Members:**

[**SIVAPRIYA.A - 202400200@sigc.edu**](mailto:SIVAPRIYA.A%20%20%20-%20%20%20202400200@sigc.edu)

[**SIRUMBAYEE.R - 202400718@sigc.edu**](mailto:SIRUMBAYEE.R%20%20%20%20-%20%20%20202400718@sigc.edu)

[**SOUNDARYA.K - 202400355@sigc.edu**](mailto:SOUNDARYA.K%20%20%20%20%20-%20%20%20%20202400355@sigc.edu)

**2.project overview**

Project view:

|  |  |
| --- | --- |
| Web Application src/components/  🡪Purpose: |  |
| To build an interactive music-themed web app |  |
| 🡪Technology Stack:  React, Vite, JavaScript (ES6+), HTML5, CSS3 |  |
| 🡪Main Entry Point  main.jsx (renders the root <App /> component)  🡪Assets: |  |
|  |  |

|  |
| --- |
| Images, icons, or media stored in src/assets/ |
| 🡪Build Tool  Vite (fast bundling, HMR) |  |
| 🡪Package Manager  npm (Node Package Manager)  🡪Configuration Files  .eslintrc.cjs, vite.config.js, .gitignore  🡪Documentation Files  README.md, PROJECT\_DOCUMENTATION.md (this file)  **3.Architecture**  **Frontend**: HTML for structure,CSS for styling,javascript for embedding.  React.js (Vite) with Node.js runtime  **Backend**: JSON Server (mock backend API)  **Database:** db.json file acting as a mock database  **4. Setup Instructions**  • Prerequisites: Node.js, Visual Studio Code, Web Browser, JSON Server  .Installation Steps:  1. npm install  2. npm run dev  3. cd db  4. json-server --watch db.json --port 3000  **5. Folder Structure**  **project-root/**  **├── node\_modules/ # Installed dependencies (ignored by Git)**  **├── public/ # Static public assets (served as-is)**  **├── src/ # Main source code for the application**  **│ ├── assets/ # Images, icons, and other static files**  **│ ├── components/ # Reusable React components**  **│ ├── App.jsx # Root app component**  **│ ├── main.jsx # Entry point that renders <App /> to the DOM**  **│ └── index.css # Global styles for the app**  **├── .eslintrc.cjs # ESLint configuration file**  **├── .gitignore # Files/folders ignored by Git**  **├── package.json # Project metadata and dependencies**  **├── vite.config.js # Vite configuration file**  **└── README.md # Project instructions**  **6.Running the Application**  **. Frontend:** npm run dev (default at http://localhost:5173)  • **Backend**: json-server --watch db.json --port 30007. API Documentation  • **Users:** /api/users (GET, POST)  • **Tracks**: /api/tracks (GET)  **• Playlists**: /api/playlists (GET, POST)  **7.Authentication**  Rhythmic Tunes uses a basic mock authentication system with JSON Server.  User login/logout is simulated.  Private routes are protected through conditional rendering.  User Experience Features:   * Smooth navigation between sections * Real-time updates using React state and props * Responsive design adaptable to different screen sizes   **8.Testing**  **Planned / Future Testing Implementation:**   * Add unit tests for individual React components using Jest * Add integration tests with React Testing Library * Use ESLint to maintain code quality and catch potential errors early   **Manual Testing Checklist:**   * Verify that the development server starts without errors * Confirm that all components render properly * Test user interactions (play, pause, add to playlist, etc.) * Check layout responsiveness on different screen sizes   **9.Screenshot**    **10.Known Issues**   Default README not updated   No authentication system   No automated testing setup  **11.Future Enhancements**  Add real authentication (JWT, OAuth).  Integrate real music streaming APIs.  Implement playlist sharing and collaboration.  Enhance UI/UX with animations and responsive design.  **12. Demo Link:**  https://drive.google.com/file/d/1gMTLtqG\_1EMMtygulZTZh8xRV7tbGhga/view?usp=drivesdk |  |

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  | |
|  | |

**Bottom of Form**