**Technische Beschreibung**

**Projektüberblick:**  
Das Projekt ist ein moderner, modular aufgebauter Musikblog, der verschiedene Musikgenres (z. B. Rap, Rock, Jazz) behandelt. Es bietet für jedes Genre mehrere Unterseiten wie Geschichte, Künstler (Icons), Alben, Instrumente und Playlists – jeweils mit eigenen Inhalten und interaktiven Elementen wie Quizzen. Für bessere Überblick Siehe bitte der gezeichnete Projektbaum am Ende dieses Dokumentes.

**Technologien & Tools:**

* **React:** Für die UI-Entwicklung mit Komponenten, State-Management und JSX-Syntax.
* **React Router:** Für die Navigation zwischen den verschiedenen Seiten und Unterseiten (z. B. Home, GenreList, GenrePage und deren Subpages).
* **Vite:** Als moderner Build- und Entwicklungsserver, der schnelles Hot Module Replacement (HMR) und eine optimierte Entwicklungsumgebung bietet.
* **CSS:** Für das Styling der Komponenten. Dabei werden globale CSS-Dateien genutzt, die in den jeweiligen Komponenten importiert werden, um eine konsistente und zugleich flexible Optik zu gewährleisten.

**Datenmodell:**  
Jede Genre-Daten-Datei (z. B. rockData.js) exportiert ein Objekt, das strukturierte Inhalte für das jeweilige Genre enthält. Diese Objekte beinhalten:

* Ein allgemeines Genre-Feld (Name, Bild).
* Eine history-Sektion mit einem Titel, einem Array von Textblöcken (jeder Block enthält Überschrift, Inhalt, ggf. spezifische CSS-Klassen), einem Bild und Quiz-Daten (Frage, Optionen, Antwort).
* Ein Array icons mit Künstlerobjekten, die jeweils Informationen wie Name, Bild, Beschreibung und Quiz-Daten enthalten.
* Weitere Abschnitte wie albums, instruments und playlists, die jeweils themenspezifische Inhalte und interaktive Elemente (z. B. Abstimmungsfunktionen oder eingebettete Videos) beinhalten.

**Interaktivität & Logik:**

* **State-Management:** Komponenten wie GenreIcons nutzen lokale State-Hooks (z. B. useState) um die Auswahl eines Künstlers und die Anzeige von Quiz-Antworten zu steuern.
* **Event-Handler:** Durch Klick-Events auf Karten und Buttons werden Detailansichten und Antworten angezeigt.
* **Routing:** Die App verwendet React Router, um zwischen Seiten zu wechseln, wobei untergeordnete Routen in der generischen GenrePage definiert sind, um Unterseiten wie History, Icons, etc. anzuzeigen.

**Styling:**

* Die Styles sind in separaten CSS-Dateien organisiert, die in den jeweiligen Komponenten importiert werden.
* Es werden sowohl globale als auch spezifische Styles verwendet, um ein konsistentes, aber flexibles Design zu erreichen.
* Anpassungen wie maximale Breiten, Scrollbalken bei zu langem Text, zentrierte Überschriften und farbliche Hervorhebungen (z. B. oranger Überschriften, lila Quiz-Optionen) werden über CSS realisiert.

**Zusammenfassung:**  
Das Projekt ist ein modular aufgebauter Musikblog, der mit React, React Router und Vite entwickelt wurde. Es trennt Inhalt (in den data/ Dateien) von der Präsentation (in den components/ und pages/ Dateien) und verwendet separate CSS-Dateien zur feingranularen Steuerung des Erscheinungsbilds. Interaktive Elemente wie Quizze und Detailansichten werden über lokale States und Event-Handler realisiert, wodurch eine dynamische Benutzererfahrung entsteht.

Diese technische Beschreibung deckt die wesentlichen Aspekte des Projekts ab – von der Struktur über die verwendeten Technologien bis hin zu Datenmodell und Interaktivität.

**Aufgabenverteilung**

Projektstruktur & allgemeine Komponenten: *Maral*

Genre Rock: *Oona*

Genre Indie: *Harnoor*

Genre Jazz: *Hala*

Genre Rap: *Maral*

**Projektstruktur:**

music-blog/

├── package.json

├── public/

│ └── index.html

└── src/

├── App.jsx

├── index.jsx

├── index.css

├── data/

│ ├── rapData.js

│ ├── rockData.js

│ └── jazzData.js

├── components/

│ ├── Navbar.jsx

│ ├── GenreList.jsx

│ └── GenreSubpages/

│ ├── GenreHistory.jsx

│ ├── GenreIcons.jsx

│ ├── GenreAlbums.jsx

│ ├── GenreInstruments.jsx

│ ├── GenrePlaylists.jsx

│ └── SubpageLayout.jsx

├── pages/

│ ├── Home.jsx

│ ├── Posts.jsx

│ ├── About.jsx

│ ├── Sources.jsx

│ └── GenrePage.jsx

└── styles/

├── App.css

├── Navbar.css

├── GenrePage.css

├── Subpages.css

├── GenreHistory.css

├── GenreIcons.css

├── GenreAlbums.css

├── GenreInstruments.css

└── GenrePlaylists.css