

Liste der Technologien, für das Projekt „Yours-Cosmic-Guide“

Frontend (Benutzeroberfläche)

Technologien:

1. React:

- Erlaubt, interaktive Benutzeroberflächen zu erstellen.
- Einfach zu lernen und anzuwenden.

2. TypeScript:

- Eine Erweiterung von JavaScript, die hilft, Fehler zu vermeiden und den Code sauber zu halten.

3. Vite:

- Ein modernes Build-Tool, das schneller und einfacher ist
- Zum Starten und Entwickeln der React-App.

4. CSS (oder TailwindCSS):

- **CSS:** Für das Styling der App.
- **TailwindCSS:** Eine Utility-basierte CSS-Bibliothek, mit der man schneller arbeiten kann

Wofür?

- Darstellung der Seiten (z. B. Homepage, Sternzeichen-Seite, Heilsteine-Seite, Mondphasen-Seite).
 - Dropdown-Menüs, Listen und Buttons für Interaktivität.
 - Styling für ein sauberes und ansprechendes Design.
-

Backend (Logik und Datenverarbeitung)

Technologien:

1. Node.js mit Express.js:
 - Node.js: Führt JavaScript außerhalb des Browsers aus.
 - Express.js: Ein Framework, um einfache APIs zu erstellen.
2. SQLite:
 - Eine kleine, leichtgewichtige Datenbank
 - Alternativ geht auch JSON-Dateien verwenden(wenn man keine Datenbank möchtest.)
3. Flask (Optional):
 - Wenn du man Python statt Node.js verwenden möchtest, ist Flask eine gute Wahl.

Wofür?

- Logik für die Berechnung des Sternzeichens basierend auf dem Geburtsdatum.
- Speicherung und Abruf von Sternzeichen, Heilsteinen und Mondphasen.
- API-Endpunkte, um Daten an das Frontend zu liefern.

Datenquellen

Technologien:

1. Statische Daten (JSON oder SQLite):
 - Beispiel:
 - JSON-Datei mit Sternzeichen (Name, Beschreibung, Datumsspannen).
 - JSON-Datei mit Heilsteinen und deren Zuordnung.
 - Alternativ: SQLite-Datenbank für strukturierte Abfragen.
2. Mondphasen-Daten:
 - **Statisch:** Manuell in einer Datei speichern.

- **Dynamisch (Optional):** Über eine API wie Time and Date API.

Deployment (Veröffentlichung der App)

Technologien:

1. Vercel (Frontend):
 - Kostenloses Hosting für React-Apps.
 - Einfach zu bedienen.
2. Render (Backend):
 - Kostenloses Hosting für Node.js- oder Flask-APIs.
 - Unterstützt auch SQLite-Datenbanken.

Tools für die Entwicklung

Technologien:

1. Visual Studio Code (VS Code):
 - Eine IDE (Editor), um den Code zu schreiben und zu verwalten.
 - Erweiterungen für React, TypeScript, Node.js und Flask.
2. Git und GitHub:
 - Git: Zum Verwalten des Codes und Nachverfolgen von Änderungen.
 - GitHub: Zum Speichern und Teilen des Codes online.
3. Postman (Optional):
 - Zum Testen der APIs.