

# Persönlichkeitstest App

## Frameworks:

### - **React:**

Für die Entwicklung dieser App habe ich mich für Next JS entschieden, Ein React Framework, welches mit low code sowohl Serverside rendering bietet, als auch die Entwicklung von Web-site-Modelle mit mehreren Seiten unterstützt.

### - **Styling:**

Meine Entscheidung war hier für Tailwind als CSS-Framwork. In Taiwind kann man von einer guten Dokumentation, einer großen Auswahl von Komponenten und einem Token basierten CSS-System profitieren, um einen übersichtlichen Styling-Code schnell erstellen zu können.

### - **State Management:**

Um einen Globalen State für alle Seiten der App zu erstellen, habe ich Redux benutzt. Und da ich das Resultat für einen spätere Zugriff local speichern möchte, habe ich Redux-Persist für diesen Zweck verwendet.

## Local Storage:

Die Email-Adresse, der Nutzername und die ausgewählten Antworten im Formular werden im Browser local gespeichert mit der möglichkeit zur Veränderung.

## Schwierigkeit:

Da es leider keine API gibt, aus der man den Text der 40 Fragen fischen kann, musste ich diesen Text hartkodieren.

Eine Alternative wäre die Api mithilfe von NodeJS im NextJS zu erstellen und danach den Text fetchen. Da dieser Prozess aber den Schreibaufwand nicht sparen würde, habe ich auf die Api verzichtet.

## Verbesserungen:

Da Die Zeit leider knapp war, fehlen noch die Unit-Test mit dem Jest-Framework.