

React Abschlussprojekt

DAKOPO

Portfolio von Danny Köhler-Poetsch

[Hier ansehen](#)

Einleitung

Portfolio DaKoPo

☞ Das ist mein Portfolio, gebaut in React.

Das Portfolio wurde im September 2024 erstellt, nachdem ich erste Kenntnisse in React erlangt habe. Ich habe vor, dieses Projekt mit wachsenden Fähigkeiten in der Webprogrammierung laufend zu erweitern. In erster Linie dient das Portfolio dazu, Ihnen einen ersten Überblick über mich zu geben.

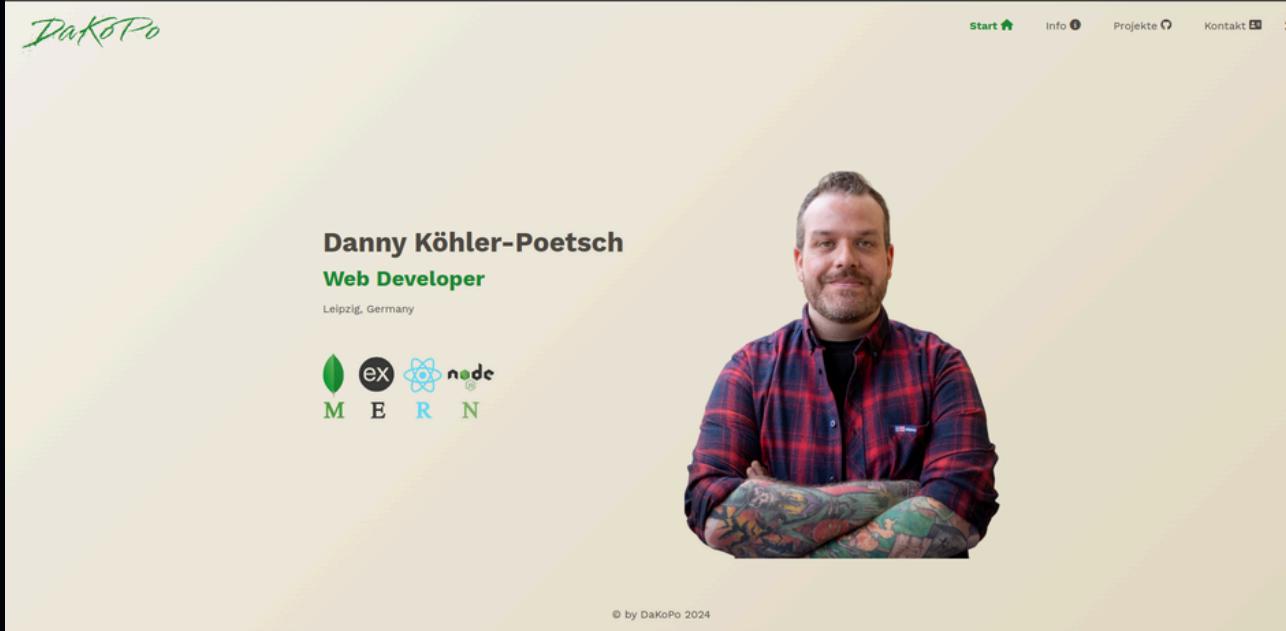
Genutzte Techniken:

- React-Router-Dom
- Context API
- Controlled Input
- useContext
- useState
- CSS-Modules

Features:

- Responsives Webdesign
- Wahl zwischen Dark Mode und Light Mode
- Webseite auf Deutsch und Englisch verfügbar

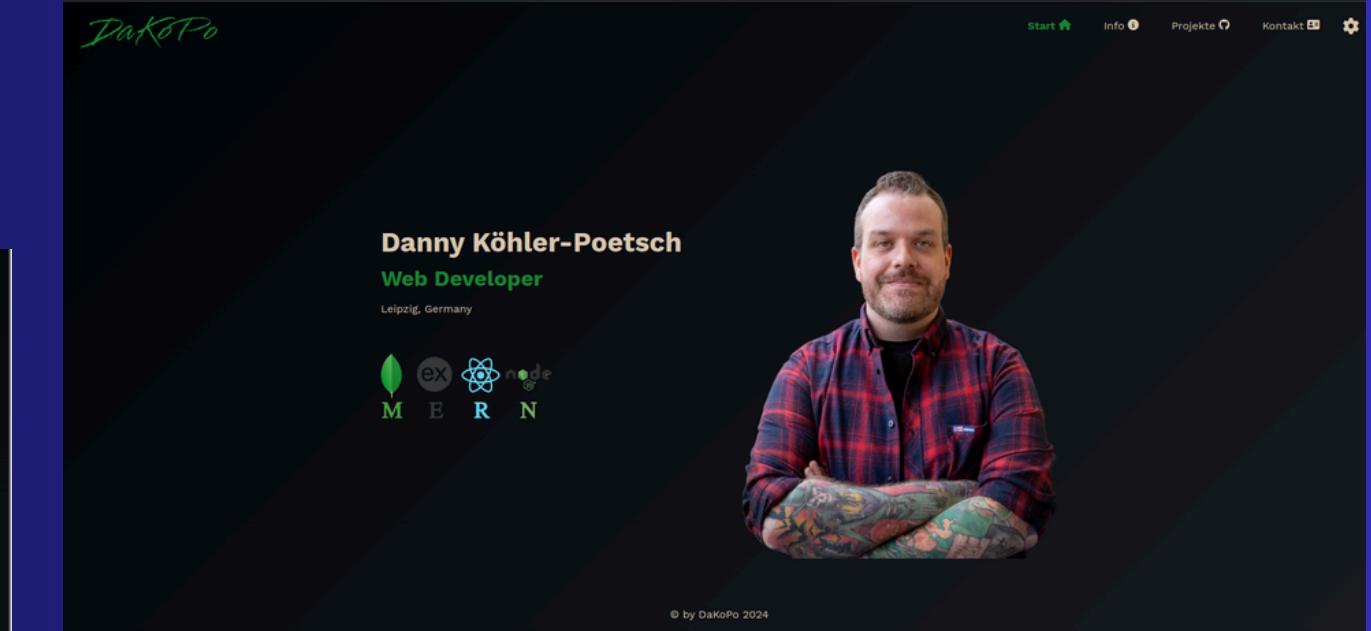
Features



Lightmode



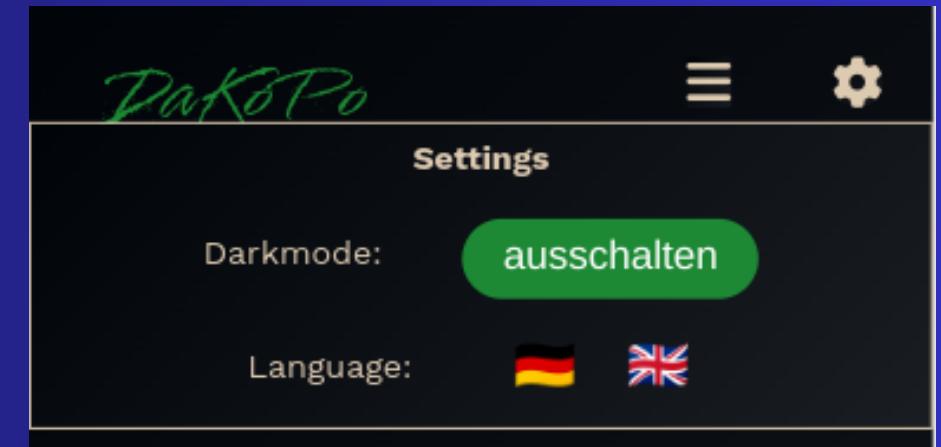
Responsive



Darkmode



Sprachen



Ordnerstruktur

```
└ dakopo
  └ node_modules
  └ public
  └ src
    └ assets
    └ components
      └ BackToTopLink
      └ Footer
      └ Header
        └ Header.jsx
        └ # Header.module.css
      └ Impressum
      └ Navigation
      └ Published
    └ context
    └ pages
      └ About
        └ About.jsx
        └ # About.module.css
      └ Contact
      └ Home
      └ Main
      └ Projects
      └ # App.css
      └ # App.jsx
      └ JS data.js
      └ # index.css
      └ JS links.js
      └ # main.jsx
      └ .gitignore
      └ # eslint.config.js
      └ < index.html
      └ # package-lock.json
      └ # package.json
      └ # README.md
      └ # vercel.json
```

Meine Struktur umfasst separate Ordner für die Komponenten, die jeweiligen Seiten, den Kontext und für die Assets (Schriften, Bilder)

Hell oder Dunkel?

```
⌘ ThemeContext.jsx X  
10 - Meine Projekte > dakopo > src > context > ⌘ ThemeContext.jsx > ...  
You, gestern | 1 author (You)  
1 import { createContext, useState, useEffect } from "react";  
2  
3 export const ThemeContext = createContext();  
4  
5 export default function ThemeProvider({ children }) {  
6   const [darkMode, setDarkMode] = useState(  
7     () => JSON.parse(localStorage.getItem("darkMode")) || false  
8   );  
9  
10  useEffect(() => {  
11    localStorage.setItem("darkMode", JSON.stringify(darkMode));  
12    [darkMode];  
13  }  
14  return (  
15    <ThemeContext.Provider value={{ darkMode, setDarkMode }}>  
16      {children}  
17    </ThemeContext.Provider>  
18  );  
19}  
20
```

Für den Zugriff auf den Zustand des darkmode habe ich die Context API benutzt. Der Zustand wird im localStorrage gespeichert.

```
1 import { BrowserRouter as Router } from "react-router-dom";  
2 import Header from "./components/Header/Header";  
3 import Main from "./pages/Main/Main";  
4 import Footer from "./components/Footer/Footer";  
5 import ThemeProvider from "./context/ThemeContext";  
6 import { ThemeContext } from "./context/ThemeContext";  
7 import LanguageProvider from "./context/LanguageContext";  
8 import { useContext } from "react";  
9  
10 import "./App.css";  
11  
12 function App() {  
13  return (  
14    <Router>  
15      <ThemeProvider>  
16        <LanguageProvider>  
17          <MainWrapper />  
18        </LanguageProvider>  
19      </ThemeProvider>  
20    </Router>  
21  );  
22}  
23  
24 function MainWrapper() {  
25  const { darkMode } = useContext(ThemeContext);  
26  
27  return (  
28    <div className={`app-wrapper ${darkMode ? `dark-mode` : `light-mode`} `}>  
29      <Header />  
30      <Main />  
31      <Footer />  
32    </div>  
33  );  
34}  
35  
36 export default App;
```

Hier habe ich bereits den Hauptcontainer den dark - oder lightmode mitgegeben



In der App.css habe ich grundlegende Stile für dark-oder lightmode definiert und habe Variablen für meine Farben genutzt

```
● ● ●
1 @font-face {
2   font-family: "work sans";
3   src: url("./assets/fonts/WorkSans-VariableFont_wght.ttf");
4 }
5
6 @font-face {
7   font-family: "splash";
8   src: url("./assets/fonts/Splash-Regular.ttf");
9 }
10
11 *
12   padding: 0;
13   margin: 0;
14   box-sizing: border-box;
15   scroll-behavior: smooth;
16 }
17
18 :root {
19   /* Light Mode Farben */
20   --background-color-light: linear-gradient(135deg, #f5f1e9, #e0d9c1);
21   /* --background-color-light: linear-gradient(135deg, #defaed, #ffffff); */
22   --text-color-light: #4e4a45;
23   /* --secondary-color-light: #c2783c; */
24   --accent-color-light-1: #7cb360;
25   /* --accent-color-light-1: #99312c; */
26   --secondary-color-light: #1d8934;
27
28   --accent-color-light-2: #ecdac9;
29   --highlight-color-light: #d1c1b2;
30
31   /* Dark Mode Farben */
32   --background-color-dark: linear-gradient(135deg, #010409, #1a1d22);
33   --text-color-dark: #dccab1;
34   --secondary-color-dark: #7d5a50;
35   --background-dark-item-even: #444;
36 }
37
38 body {
39   overflow-x: hidden;
40   max-width: 100vw;
41 }
```

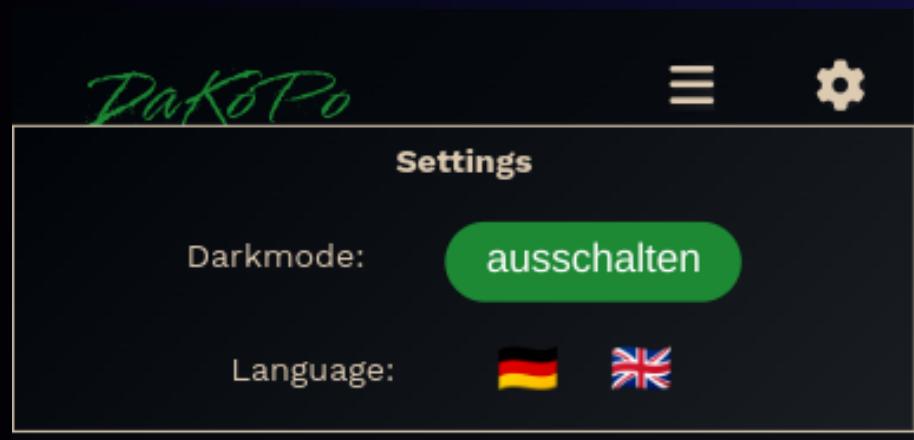
```
● ● ●
1 /* Light Mode */
2 .light-mode {
3   background: var(--background-color-light);
4   color: var(--text-color-light);
5 }
6
7 /* Dark Mode */
8 .dark-mode {
9   background: var(--background-color-dark);
10  color: var(--text-color-dark);
11 }
12
13 .app-wrapper {
14   display: flex;
15   flex-direction: column;
16   width: 100%;
17   min-height: 100vh;
18   font-family: "Work Sans", sans-serif;
19   background-attachment: fixed;
20   background-size: cover;
21   overflow-x: hidden;
22 }
23
24 h1 {
25   font-size: clamp(2rem, 0.3043rem + 3.4783vw, 2.5rem);
26 }
27
28 h2 {
29   font-size: clamp(1.6rem, 0.3043rem + 3.4783vw, 2rem);
30   color: var(--secondary-color-light);
31 }
32
33 h3 {
34   font-size: clamp(1.2rem, 0.3043rem + 3.4783vw, 1.5rem);
35   color: var(--secondary-color-light);
36 }
37
38 a {
39   /* font-size: clamp(1rem, 0.3043rem + 3.4783vw, 1.1rem); */
40   color: var(--secondary-color-light);
41   transition: transform 0.2s linear;
42   text-decoration: none;
43   position: relative;
44   transition: all 0.3s;
45
46   &:hover {
47     transform: scale(1.05);
48   }
49 }
50
51 p,
52 span {
53   /* font-size: clamp(1rem, 0.3043rem + 3.4783vw, 1.1rem); */
54   line-height: 3.5vh;
55 }
56
```

```
10 const { language, switchLanguage } = useContext(LanguageContext);
11 const [menuOpen, setMenuOpen] = useState(false);
12 const [settings, setSettings] = useState(false);
13
14 const handleDarkMode = () => {
15   setDarkMode((prevState) => !prevState);
16 };
17
18 const toggleMenu = () => {
19   setMenuOpen((prevState) => !prevState);
20 };
21
22 const closeMenu = () => {
23   setMenuOpen(false);
24 };
25
26 const toggleSettings = () => {
27   setSettings((prevState) => !prevState);
28 };
29
30 const closeSettings = () => {
31   setSettings(false);
32 };
33
34 return (
35   <>
36     <header className={styles.header}>
37       <div className={styles.headerLogo}>
38         <NavLink to="/">
39           <span id="top">DaKoPo</span>
40         </NavLink>
41       </div>
42       <Navigation closeMenu={closeMenu} menuOpen={menuOpen} />
43       <div
44         className={`${styles.burger} ${darkMode ? "dark-mode" : "light-mode"}`}
45         onClick={() => toggleMenu()}
46       >
47         <i className="fa-solid fa-bars"></i>
48       </div>
49
50       <button
51         className={`${styles.settings} ${darkMode ? "dark-mode" : "light-mode"}`}
52         onClick={toggleSettings}
53         title={language === "de" ? "Einstellungen öffnen" : "open settings"}
54       >
55         <i className="fa-solid fa-gear"></i>
56       </button>
57     </header>
58   </>
59 
```



Im Header läuft dann einiges zusammen. Der dark/light-Mode und die Sprachauswahl sind ersichtlich

Excuse me, sprechen Sie deutsch?



In den Einstellungen kann man zwischen den Sprachen deutsch und englisch wählen.

Die Settings öffnen sich per Klick

A screenshot of a code editor showing the 'LanguageContext.jsx' file. The code defines a context provider for language switching, using useState and useEffect hooks to manage the language state and persist it to localStorage.

conditional
rendering
macht's
möglich

A screenshot of a code editor showing the 'Navigation' component. It uses conditional rendering to show different navigation items based on the language context. An orange arrow points from the word 'möglich' in the explanatory text below to the conditional rendering logic in the code.

Bereitstellung der Daten in 2 Sprachen

Vorbereitung der Daten

```
5   titleDe: "Programmiersprachen",
6   infoDe:
7     "Ich lerne derzeit den MERN-Stack, der aus MongoDB, Express.js, React
8     und Node.js besteht. Es ist ein leistungsstarkes
9     Full-Stack-JavaScript-Framework, das für den Aufbau dynamischer
10    Webanwendungen verwendet wird.",
11
12  {
13    title: "Professional Background",
14    info: "From 2011 to 2023, I worked as a technical laboratory assistant
15    in the field of microbiology. Unfortunately, I wasn't able to fully
16    utilize all of my skills in this role, which led me to pursue a career
17    in web development to help shape the digital future. I am currently
18    enrolled in the Web and Software Development course at the DCI - Digital
19    Career Institute, where I serve as both class representative and tutor.
20    Thanks to my quick learning ability and talent for simplifying complex
21    topics, I have already started earning money as a tutor during the
22    course.",
23    titleDe: "Beruflicher Hintergrund",
24    infoDe:
25      "Von 2011 bis 2023 arbeitete ich als technischer Laborassistent im
26      Bereich Mikrobiologie. Leider konnte ich in dieser Rolle nicht all
27      meine Fähigkeiten vollständig einsetzen, was mich dazu veranlasste,
28      in der Webentwicklung zu verfolgen, um die digitale
29      Zukunft mitzugestalten. Derzeit bin ich im Kurs für Web- und
30      Softwareentwicklung am DCI - Digital Career Institute eingeschrieben,
31      wo ich sowohl als Klassensprecher als auch als Tutor tätig bin. Dank
32      meiner schnellen Auffassungsgabe und meines Talents, komplexe Themen
33      zu vereinfachen, habe ich bereits während des Kurses angefangen, als
34      Tutor Geld zu verdienen.",
```

```
7  export const myProjects = [
8    {
9      title: "Frightful Fun - my Halloween-HTML-Project",
10     titleDe: "Frightful Fun - mein Halloween-HTML-Projekt",
11     url: "https://dannykoehlerpoetsch.github.io/frightful-fun/",
12     src: frightful,
13     github: "https://github.com/dannykoehlerpoetsch/frightful-fun",
14     alt: "Screenshot from Frightful-Fun Website",
15     altDe: "Screenshot von der Frightful-Fun Website",
16     info: "`Frightful Fun` is a Halloween-themed website offering everything
17     for a magical and spooky Halloween experience. Visitors can explore
18     sections on Halloween history, costumes, films, music, and even (fake)
19     shop for Halloween items. The site invites users to dive into the eerie
20     and enchanting world of Halloween, with interactive elements and fun
21     surprises for a thrilling experience.",
22     infoDe:
23       "`Frightful Fun` ist eine Halloween-Website, die alles für ein
24       magisches und gruseliges Halloween-Erlebnis bietet. Besucher können
25       Bereiche zur Halloween-Geschichte, Kostümen, Filmen, Musik und sogar
       einen (fiktiven) Halloween Shop erkunden. Die Seite lädt die Nutzer
       ein, in die unheimliche und bezaubernde Welt von Halloween
       einzutauchen, mit interaktiven Elementen und spaßigen Überraschungen
       für ein aufregendes Erlebnis.",
```

Ich habe die Daten für meine “About” Seite und meine “Projects” Seite zuvor in einem Array, bestehend aus Objekten, gespeichert - damit ich später mit der map-Methode arbeiten kann

Schreib mir eine Nachricht!

KISS - keep it simple stupid



```
1 export default function Contact() {  
2   const id = userId();  
3   const navigate = useNavigate();  
4   const { darkMode } = useContext(ThemeContext);  
5   const { language } = useContext(LanguageContext);  
6   const [name, setName] = useState("");  
7   const [mail, setMail] = useState("");  
8   const [message, setMessage] = useState("");  
9   const [submitted, setSubmitted] = useState(false);  
10  const [error, setError] = useState("");  
11  const [showImpressum, setShowImpressum] = useState(false);
```

useReducer() ? Nein, Danke.

Da ich über “event.Preventdefault()” das Standardverhalten von Submit verhindere, musste ich für eine funktionierende Kontaktaufnahme den Code etwas modifizieren. Das sehen wir im Backend aber sicher noch genauer



```
1 try {  
2   const response = await fetch("https://formspree.io/f/xyzganan", {  
3     method: "POST",  
4     headers: {  
5       "Content-Type": "application/json",  
6     },  
7     body: JSON.stringify({  
8       name: name,  
9       email: mail,  
10      message: message,  
11    }),  
12  });  
13  if (response.ok) {  
14    setSubmitted(true);  
15    setName("");  
16    setMail("");  
17    setMessage("");  
18  
19    setTimeout(() => {  
20      navigate("/");  
21    }, 5000);  
22  } else {  
23    setError(  
24      language === "de"  
25        ? "Etwas ist schiefgelaufen. Bitte versuchen Sie es später erneut"  
26        : "Something went wrong. Please try again later."  
27    );  
28  }  
29} catch (error) {  
30  console.error("Error submitting the form", error);  
31  setError(  
32    language === "de"  
33        ? "Etwas ist schiefgelaufen. Bitte versuchen Sie es später erneut"  
34        : "Something went wrong. Please try again later."  
35    );  
36  }  
37};  
38};  
39};
```

Das Modal in der Projektkomponente



```
1 export default function Links() {  
2   const { darkMode } = useContext(ThemeContext);  
3   const [showImage, setShowImage] = useState(null);  
4   const { language } = useContext(LanguageContext);  
5  
6   const openModal = (src) => {  
7     setShowImage(src);  
8   };  
9   const closeModal = () => {  
10     setShowImage(null);  
11   };
```

Logik schreiben



```
1 <div className={styles.cardImage}>  
2   <img  
3     src={project.src}  
4     alt={language === "de" ? project.altDe : project.alt}  
5     onClick={() => openModal(project.src)}  
6     title="click to show large image"  
7   />  
8 </div>
```

Logik übergeben

“e.stopPropagation”...blubbert da was?

Hier jedenfalls wird verhindert, dass ein Klick auf das geöffnete Bild das Modal direkt wieder schließt.



```
1 {showImage && (  
2   <div className={styles.modal} onClick={closeModal}>  
3     <div  
4       className={styles.modalContent}  
5       onClick={(e) => e.stopPropagation()}  
6     >  
7     <button  
8       className={styles.closeButton}  
9       onClick={closeModal}  
10      title="close"  
11    >  
12    X  
13  </button>  
14  <img  
15    src={showImage}  
16    alt="Selected"  
17    className={styles.modalImage}  
18  />  
19  </div>  
20 </div>  
21 )}
```

Logik anwenden

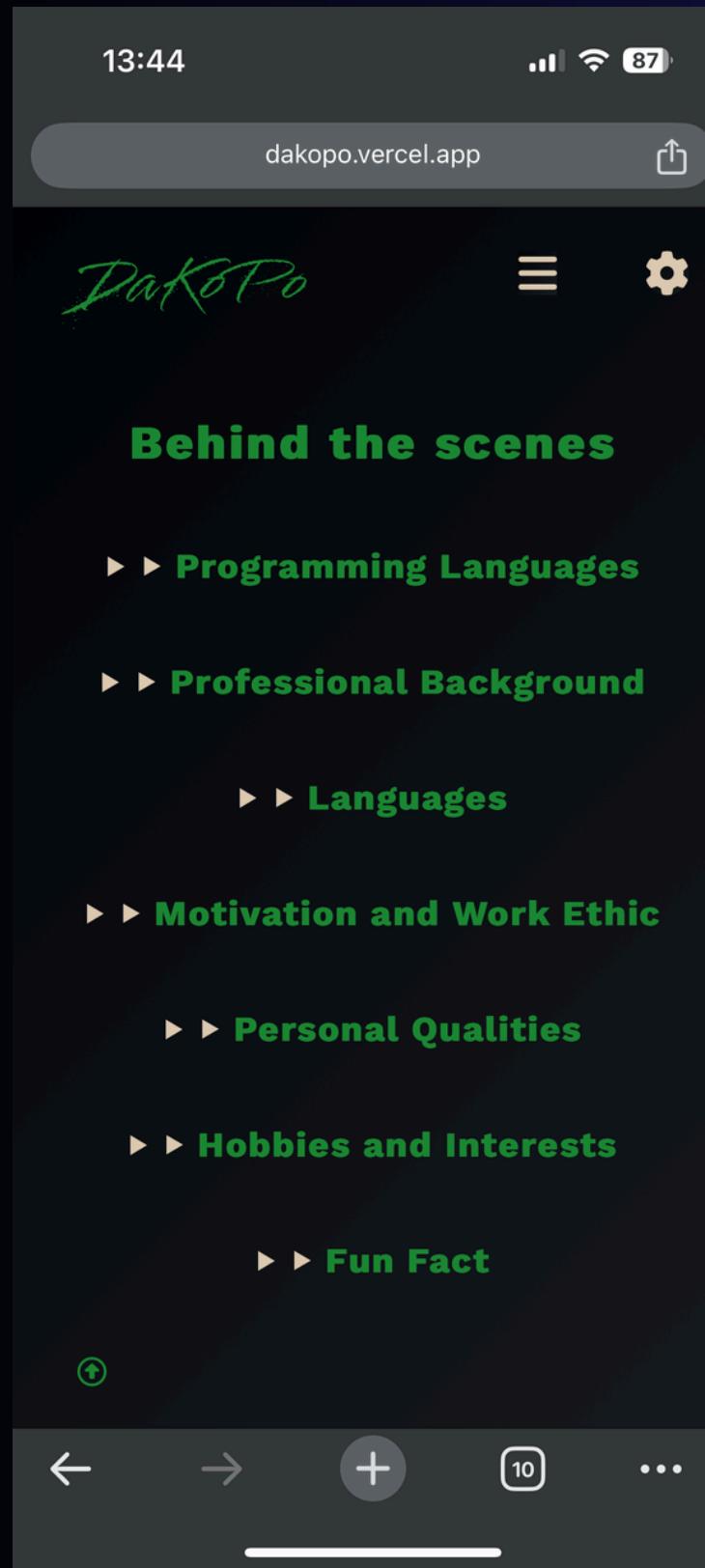
Routing

```
● ● ●  
1 import React from "react";
2 import { Routes, Route } from "react-router-dom";
3 import Home from "../Home/Home";
4 import Projects from "../Projects/Projects";
5 import About from "../About/About";
6 import Contact from "../Contact/Contact";
7 import styles from "./Main.module.css";
8
9 export default function Main() {
10   return (
11     <main className={styles.main}>
12       <Routes>
13         <Route path="/" element={<Home />} />
14         <Route path="/projects" element={<Projects />} />
15         <Route path="/about" element={<About />} />
16         <Route path="/contact" element={<Contact />} />
17       </Routes>
18     </main>
19   );
20 }
```

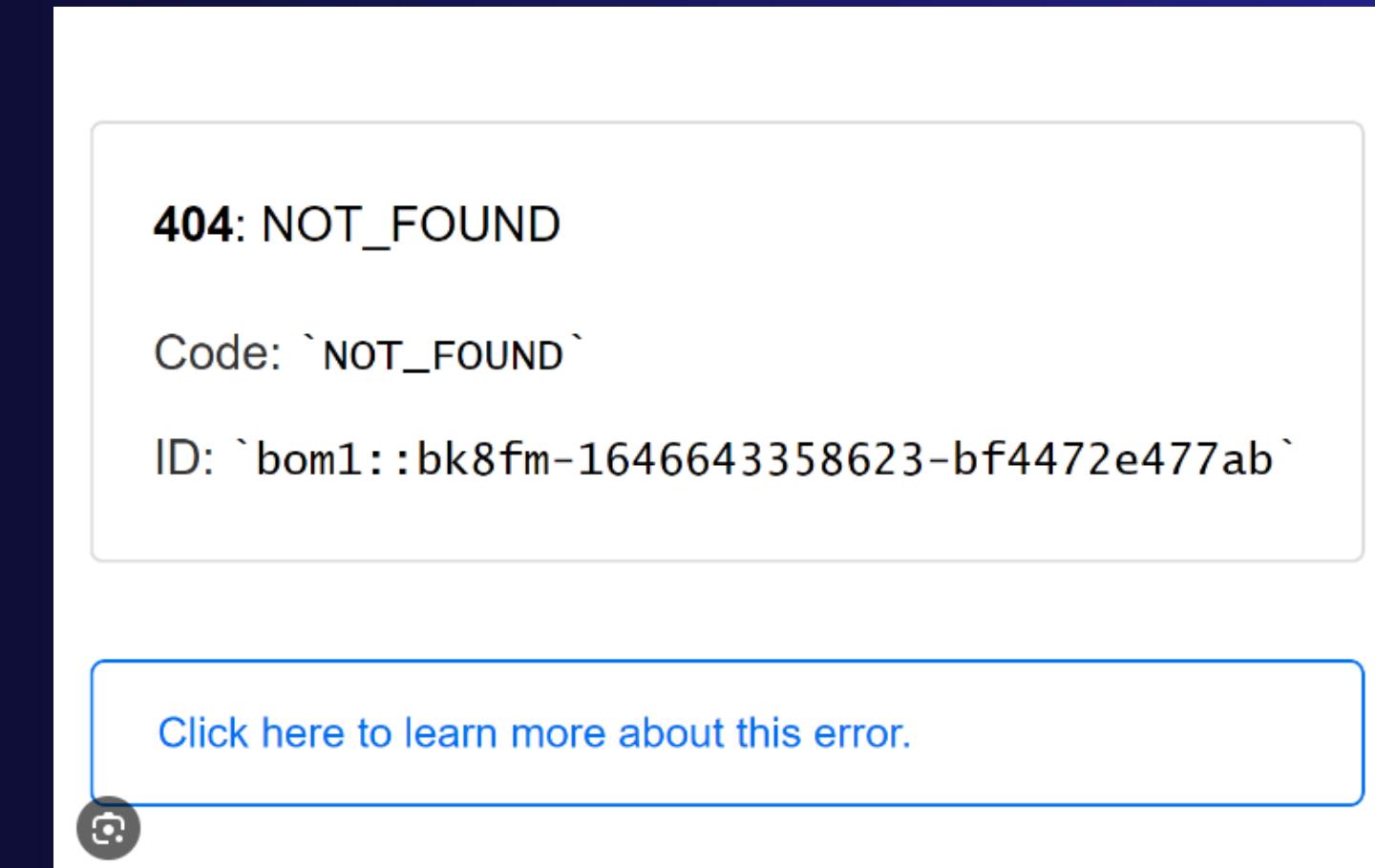
```
48           {links.map((link, index) => (
49             <li key={index}>
50               <NavLink
51                 to={link.path}
52                 onClick={closeMenu}
53                 className={({ isActive }) =>
54                   isActive ? styles.activeLink : styles.link
55                 }
56               >
57                 {language === "eng" ? link.name : link.nameDe}{ " "}
58                 <i className={link.classname}></i>
59               </NavLink>
60             </li>
61           )}
```

Die simpelste Lösung für dieses Projekt

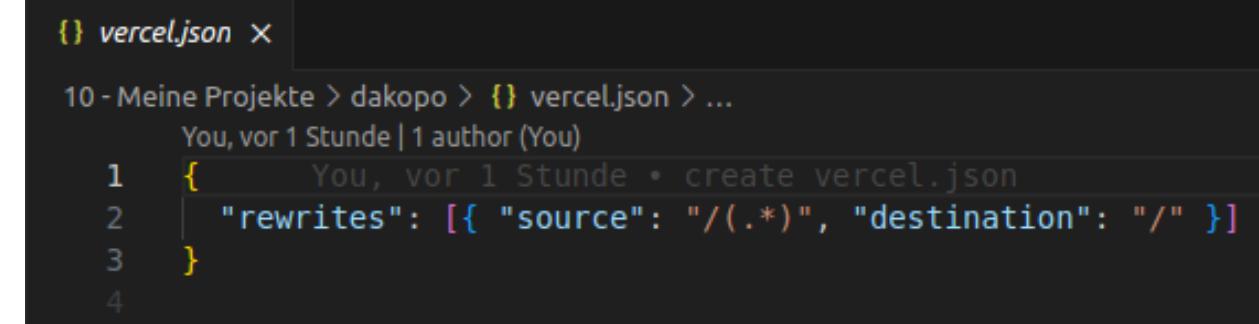
Probleme



Problem 1



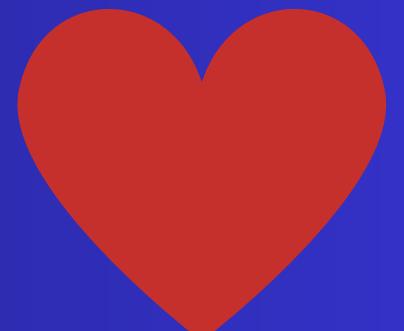
Problem 2



Lösung 2



=



Was nicht fehlen darf?

Um die größtmögliche Kompatibilität mit allen Browsern zu gewährleisten, empfiehlt sich, die CSS-Dateien einmal von einem auto-Prefixer bearbeiten zu lassen

```
1  /*  
2  * Prefixed by https://autoprefixer.github.io  
3  * PostCSS: v8.4.14,  
4  * Autoprefixer: v10.4.7  
5  * Browsers: last 4 version  
6  */  
7  
8 .about {  
9  display: -webkit-box;  
10 display: -ms-flexbox;  
11 display: flex;  
12 -webkit-box-orient: vertical;  
13 -webkit-box-direction: normal;  
14 -ms-flex-direction: column;  
15 flex-direction: column;  
16 -webkit-box-align: center;  
17 -ms-flex-align: center;  
18 align-items: center;  
19 -webkit-box-pack: justify;  
20 -ms-flex-pack: justify;  
21 justify-content: space-between;  
22 }  
23  
24 .opener {  
25  -moz-text-align-last: center;  
26  text-align-last: center;  
27 }
```

THANK YOU!