

IU – Internationale Hochschule – Akademie

Modul: IPWA01-01 - Programmierung von Webanwendungsoberflächen

Tutor: Alexander Christopher Bock

Studiengang: IU Akademie Weiterbildung zum Agile Java Web Developer

GitHub Repository: <https://github.com/Rondidon/climate-impact-dashboard/tree/main>

Fallstudie

Entwicklung einer Webanwendung zur Darstellung des
CO2-Fußabdrucks von Unternehmen und Ländern

Eingereicht am 23. April 2024

Verfasser der Fallstudie:

Robin Fabian Alexander Kindler

Kernerstraße 13

71106 Magstadt

E-Mail: robin.kindler@iu-akademie.de / robin.kindler@gmail.com

Matrikelnummer: UPS10686611

Inhaltsverzeichnis

1 - Einleitung.....	1
2 - Hauptteil.....	2
2.1 Analyse der Aufgabenstellung.....	2
2.1.1 Umsetzungskonzept.....	2
2.1.2 Relevanz einer öffentlich zugänglichen Webseite.....	2
2.2 Wahl des Technologiestacks.....	3
2.2.1 Create React App.....	3
2.2.2 React.....	3
2.2.3 Typescript.....	4
2.2.4 Design und Layout mit Bootstrap und Flexbox, flag-icon-css.....	4
2.2.5 Lokale Übersetzungen mit i18next.....	4
2.2.6 CO2 Rohdaten im CSV-Format mit Papaparse.....	4
2.3 Entwurf der Benutzeroberfläche.....	5
2.3.1 Titel und Logo.....	5
2.3.2 Header und Footer.....	5
2.3.3 Impressum und Datenschutz.....	5
2.3.4 Landingpages.....	6
2.3.5 Emissionsdatenbank-Betrachter.....	6
2.4 Implementierung.....	7
2.4.1 Einrichtung des öffentlichen Code-Repositories.....	7
2.4.2 Umsetzung der Menüstruktur und Navigation.....	7
2.4.3 Implementierung der Seiteninhalte.....	8
2.4.4 Implementierung der Datentabelle.....	8
2.4.5 Lokalisierung.....	9
2.4.6 Implementierung der Suchfunktion.....	9
2.4.7 Sicherheitsaspekte: Schutz vor Code-Injektion und XSS.....	10
.....	10
2.5 Testing und Optimierung.....	11
2.5.1 Funktionstests.....	11
2.5.2 Responsivitätsprüfung.....	11
2.5.3 Sicherheitstests.....	11
3 - Schluss.....	12
3.1 Diskussion der Herausforderungen und Lösungsansätze.....	12
3.2 Ausblick und mögliche Weiterentwicklungen.....	12
I - Literaturverzeichnis.....	13
II - Abbildungsverzeichnis.....	14
III - Anhang A – Screenshots der Anwendung.....	15
IV - Anhang B – GitHub Repository zum Projekt.....	19

1 - Einleitung

In der vorliegenden Fallstudie wird die Entwicklung einer Webanwendung zur Visualisierung des CO₂-Fußabdrucks von Unternehmen und Ländern detailliert betrachtet. Ziel ist es, durch digitale Technologien und das Internet eine Plattform zu schaffen, die komplexe Daten zum Klimawandel auf eine benutzerfreundliche und intuitive Weise darstellt. Im Zentrum dieser Bemühungen steht die Implementierung einer Datenvisualisierung als CO₂-Emissionstabelle, um gegebene Datensätze transparent darzustellen und zugänglich zu machen.

Die Anwendung zielt darauf ab, das Verständnis für den Klimawandel zu fördern und Nutzenden handlungsorientierte Informationen zu bieten. Hierbei spielen Einfachheit in der Nutzung und die Minimierung technischer Barrieren eine entscheidende Rolle, um sicherzustellen, dass die Informationen eine breite Zielgruppe erreichen. Durch Verwendung moderner Webtechnologien wird eine Plattform geschaffen, die nicht nur informativ, sondern auch einfach erweiterbar und wartbar ist.

Diese Fallstudie dokumentiert den gesamten Entwicklungsprozess der Anwendung, von der Konzeption über die Auswahl des Technologiestacks bis hin zur Implementierung spezifischer Features wie Filter- und Sortierfunktionen, die eine gezielte Interaktion mit den dargestellten Daten ermöglichen. Besonderes Augenmerk wird auf die Benutzeroberfläche gelegt, um auf verschiedenen Endgeräten eine stets bestmögliche Erfahrung zu gewährleisten. Die Anwendung stellt ein praxisnahes Beispiel dar, wie Webanwendungen komplexe Informationen auf eine zugängliche und verständliche Weise darstellen können, indem Nutzerfreundlichkeit und aktuelle Webtechnologien wie React und HTML5 miteinander verbunden werden.

2 - Hauptteil

2.1 Analyse der Aufgabenstellung

Das Ziel der Webanwendung ist es, Transparenz über CO₂-Emissionen von Unternehmen und Ländern zu schaffen. Sie soll moderne Webtechnologien verwenden, ein benutzerfreundliches und ansprechendes Design bieten, das auf verschiedenen Geräten funktioniert, und gegen XSS-Angriffe gesichert sein. Zukünftige technische Weiterentwicklungen sollen durch den Einsatz entsprechender Frameworks und einer durchdachten Struktur leicht integrierbar sein.

2.1.1 Umsetzungskonzept

Eine gut navigierbare, öffentlich zugängliche Webanwendung fördert durch gezielte technische Entscheidungen deren Sichtbarkeit und Akzeptanz. Resultierende Entscheidungen übertreffen teilweise die Basisanforderungen, um mit wenig Aufwand maximale Wirkung zu erreichen.

- Entwicklung in einem öffentlichen Code-Repository.
- Einsatz eines modernen Technologie-Stacks.
- Globale und lokale Navigation durch Header und Seitenmenü.
- Footer mit Impressum, Datenschutz, Links und Projektkontext.
- Responsive Design mittels Kacheln.
- Eigener CSS-Style, passend zum Thema.
- Integration von sortierbaren Tabellen mit fiktiven CO₂-Emissionsdaten.
- Responsive Filterfunktion für Jahr, Land und Unternehmen.
- Schutz vor XSS und Sicherheitsgewährleistung.
- Seitensuchfunktion zur Verbesserung der User Experience.
- Mehrsprachigkeit (Englisch und Deutsch) zur Erweiterung der Zielgruppe.
- Verwendung des React-Routers für strukturierte URLs und eine 404-Seite zur Suchmaschinenoptimierung.

2.1.2 Relevanz einer öffentlich zugänglichen Webseite

Die Präsenz einer öffentlich zugänglichen Webseite ist für Unternehmen und Organisationen essentiell, um in der digitalen Ära sichtbar und relevant zu bleiben. Angesichts der Tatsache, dass das Internet die primäre Quelle für Informationen darstellt, ist eine qualitative Online-Präsenz unerlässlich, um Zielgruppen effektiv zu erreichen. Um dieses Ziel zu erreichen, gewährleistet die Implementierung eines responsiven Webdesigns, dass die Webseite auf einer Vielzahl von Geräten optimal dargestellt wird, wodurch eine breite Nutzerbasis angesprochen wird.

2.2 Wahl des Technologiestacks

Die Entscheidung für den letztendlichen Technologiestack folgte einer Untersuchung verschiedener technischer Lösungen. Anfangs kam ein traditioneller Ansatz zum Einsatz, bei dem statische HTML-Seiten, ergänzt durch das CSS-Framework Bootstrap für das Design und TypeScript für eine stärkere Typisierung, kompiliert mit dem TypeScript-Compiler tsc, verwendet wurden. Ein auf Express.js basierender lokaler Server diente der Auslieferung der Seiten. Diese Konfiguration bot zwar eine solide Basis für die Entwicklung, stieß jedoch bald an ihre Grenzen hinsichtlich Anpassbarkeit und Flexibilität. Diese Einschränkungen machten eine Neubewertung der technologischen Ausrichtung erforderlich. Die Wahl fiel schließlich auf die Nutzung von React und Create-React-App für ein clientseitiges, auf JavaScript basierendes dynamisches Rendering. Dieser Ansatz trug maßgeblich zur Optimierung der Benutzer- und Entwicklererfahrung bei (Bootstrap Dokumentation, 2024; TypeScript Dokumentation, 2024; TypeScript Compiler Dokumentation, 2024; Express.js Dokumentation, 2024; React Dokumentation, 2024; Create React App Dokumentation, 2024).

2.2.1 Create React App

Create-React-App ist ein offizielles Tool von Facebook. Es zielt darauf ab, die Erstellung neuer React-Anwendungen durch eine sofort einsatzbereite, moderne, getestete und robuste Build-Konfiguration zu vereinfachen. Es reduziert die Zeit des initialen Aufsetzens eines React-Projektes erheblich, die sonst in die Konfiguration der Entwicklungsumgebung inklusive lokaler Server investiert werden müsste, Entwicklerinnen und Entwickler können sich so in vielen Fällen direkt auf die Entwicklung zu konzentrieren oder auf der Grundkonfiguration eigene Konfigurationen aufbauen. Mit einem standardisierten, jedoch flexibel anpassbaren Setup unterstützt Create-React-App eine "Zero-Configuration"-Philosophie bietet jedoch auch eine "eject"-Option für Entwickler, die spezifische Anpassungen an der Build-Konfiguration vornehmen möchten, indem es ihnen erlaubt, die volle Kontrolle über die Projektabhängigkeiten zu übernehmen (Create React App Dokumentation, 2024) .

2.2.2 React

Im Rahmen des Projekts spielt React eine zentrale Rolle für die Entwicklung der Benutzeroberfläche. Als eine der führenden JavaScript-Bibliotheken für das Erstellen von User Interfaces ermöglicht React eine effiziente und flexible Entwicklung von Single Page Applications (SPAs). Insbesondere die Nutzung von Hooks und Functional Components innerhalb des React-Ökosystems erweist sich als besonders vorteilhaft. Hooks erlauben es, Zustandslogik in funktionalen Komponenten zu nutzen, ohne auf Klassenkomponenten zurückgreifen zu müssen. Dies führt zu einer Vereinfachung der Codebasis und verbessert die Lesbarkeit sowie Wartbarkeit

des Codes. Functional Components tragen weiterhin dazu bei, die Applikation modular und übersichtlich zu gestalten, indem sie die Verwendung von wiederverwendbaren, unabhängigen Komponenten ermöglichen (React Dokumentation, 2024).

2.2.3 Typescript

TypeScript, eine Erweiterung von JavaScript, wird im Projekt eingesetzt, um bessere Typsicherheit zu gewährleisten und den Code robuster zu machen. Durch die statische Typisierung hilft TypeScript dabei, Fehler bereits während der Entwicklungsphase zu erkennen und zu verhindern. Dies steigert nicht nur die Sicherheit des Codes, sondern trägt auch zu einer erhöhten Übersichtlichkeit bei, da Typen als eine Form der Dokumentation dienen können (TypeScript Dokumentation, 2024).

2.2.4 Design und Layout mit Bootstrap und Flexbox, flag-icon-css

Für das Design und Layout der Webanwendung kommt das CSS-Framework Bootstrap zum Einsatz, ergänzt durch Flexbox für flexibles Layout-Design und flag-icon-css zur Darstellung von Länderflaggen. Bootstrap bietet eine Vielzahl von vorgefertigten Komponenten wie Cards, Buttons und einen responsiven Header, die speziell für dieses Projekt angepasst und durch einen eigenen CSS-Stil, das „climate-scheme“, erweitert wurden. Die Flexibilität von Flexbox ermöglicht ein intuitives Anordnen von Layoutelementen, während flag-icon-css die visuelle Darstellung unterstützt. Die Anpassungsfähigkeit an verschiedene Gerätetypen wird durch Bootstrap's CSS-Hilfsklassen und zusätzliche media-Queries erreicht, was eine optimale Darstellung auf diversen Endgeräten sicherstellt. (Bootstrap Dokumentation, 2024; Flexbox Dokumentation, 2024; Flag-icon-css GitHub Repository, 2024).

2.2.5 Lokale Übersetzungen mit i18next

Das Projekt nutzt i18next, eine leistungsstarke Internationalisierungsbibliothek, um mehrsprachige Inhalte bereitzustellen. Die Texte der Webanwendung sind in den Dateien de.json für Deutsch und en.json für Englisch ausgelagert, was nicht nur die Übersetzungsarbeit vereinfacht, sondern auch die Grundlage für eine globale Suchfunktion bildet. Näheres hierzu im Punkt 2.4.6 – Implementierung der Suchfunktion. (i18next Dokumentation, 2024).

2.2.6 CO2 Rohdaten im CSV-Format mit Papaparse

Die CO2-Emissionsdaten (Rohdaten) werden im CSV-Format gehalten. Durch den Einsatz von Papaparse, einer schnellen und leistungsfähigen JavaScript-Bibliothek zum Parsen von CSV-Daten, wird die Verarbeitung dieser Daten ermöglicht. Die Wahl des CSV-Formats gewährleistet eine einfache Aktualisierbarkeit und Austauschbarkeit der Datenquelle. Nach dem Einlesen der

Daten ermöglichen speziell entwickelte Algorithmen die weitere Verarbeitung, wie Filterung und Sortierung der Emissionsdaten, unabhängig vom ursprünglichen CSV-Format, und ihre dynamische Integration in die Webanwendung (Papaparse Dokumentation, 2024).

2.3 Entwurf der Benutzeroberfläche

Dieses Kapitel beschreibt den Entwurf und die Gestaltung der Benutzeroberfläche.

2.3.1 Titel und Logo

Der Titel "EcoTrace - CO₂ Emissions Visualization Tool" spiegelt die Kernfunktion und die Intention der Webseite wider: eine Plattform zur Visualisierung von CO₂-Emissionsdaten zu bieten. Das dazu entworfene Logo, das ein Blatt darstellt, soll eine Verbindung zur Natur herstellen und die thematische Ausrichtung der Webseite unterstreichen. Das Logo wurde auf Grundlage eines freien Designs von Vecteezy.com frei gestaltet. Ein nötiger Quellenverweis findet sich im Impressum der Webseite und im Literaturverzeichnis (Grünes Blatt Logo Vektor, 2024).

2.3.2 Header und Footer

Der Header beinhaltet eine Suchfunktion, globale Links und ein Dropdown-Menü für die Sprachauswahl, was die Zugänglichkeit und Nutzerfreundlichkeit erhöht. Der Footer folgt dem gleichen Designprinzip und bietet neben essentiellen rechtlichen Links zu Datenschutz und Impressum auch den Projektkontext und eine weitere globale Linkstruktur. Beide Elemente sind vollständig responsiv gestaltet und integrieren sich nahtlos in das Gesamtdesign der Webseite.

Sowohl der Header als auch der Footer wurden mit einem eigenen CSS Style unter Verwendung von Bootstrap-Komponenten erstellt. Die Farbgestaltung der Webseite ist bewusst auf eine reduzierte, prägnante grün-schwarze Farbpalette abgestimmt, um eine visuelle Verbindung zum Thema Natur und Umweltschutz herzustellen.

2.3.3 Impressum und Datenschutz

Das Impressum und der Datenschutz sind essenzielle Bestandteile der Webseite, die nicht nur der Informationspflicht dienen, sondern auch die Rechtssicherheit der Online-Präsenz gewährleisten. Für die Erstellung dieser rechtlichen Dokumente wurde auf eRecht24 zurückgegriffen, eine Plattform, die spezialisiert ist auf die Generierung rechtssicherer Impressums- und Datenschutztexte für Webseiten in Deutschland (eRecht24, 2024). Diese Wahl stellt sicher, dass alle gesetzlichen Anforderungen erfüllt werden, insbesondere im Hinblick auf das Telemediengesetz (TMG) und die Datenschutz-Grundverordnung (DSGVO).

Diese rechtlichen Texte wurden in das Gesamtdesign der Webseite integriert und sind über den Footer leicht zugänglich gemacht worden. Da der Textabschnitt recht lang ist, wurde in den Datenschutzhinweisen eine lokale, responsive lokale Navigation integriert (Abbildung 8).

2.3.4 Landingpages

Die Webseite umfasst eine Reihe von Schlüsselseiten, die jeweils ein einzigartiges Element des Projekts hervorheben und dabei vollständig responsiv gestaltet sind.

- **Home:** Die Startseite dient als zentraler Anlaufpunkt, der Besucher über die Ziele der Plattform informiert (Abbildung 1).
- **Was ist CO2:** Diese Seite erklärt auf verständliche Weise, was CO2 ist, warum es wichtig ist und welche Rolle es beim Klimawandel spielt (Abbildung 7).
- **Wie kann ich zur Reduktion beitragen?:** Hier werden Handlungsempfehlungen und Tipps gegeben, wie Einzelpersonen ihren CO2-Fußabdruck verringern können (Abbildung 3).
- **Emissionsdatenbank-Betrachter:** Eine interaktive Datenbank, die es Benutzern ermöglicht, spezifische Emissionsdaten zu durchsuchen und zu analysieren (Abbildung 5).

Jede dieser Seiten wird durch kontextbezogene, KI-generierte Bilder bereichert und bietet ein klares, ansprechendes Design, das den Inhalt verstärkt und die Benutzererfahrung verbessert. Für die Bildgenerierung wurde DALL-E genutzt (DALL-E Dokumentation, 2024).

2.3.5 Emissionsdatenbank-Betrachter

Die Seite des Emissionsdatenbank-Betrachters (Abbildung 5, Abbildung 6) bietet eine benutzerfreundliche und interaktive Möglichkeit, Emissionsdaten zu durchsuchen und zu filtern:

- **Sidebar mit Presets und Filterfunktion:** Anwender können voreingestellte Ansichten wählen oder Daten nach Kriterien wie Jahr, Unternehmen und Länder filtern.
- **Datenanzeige als Tabelle:** Eine Tabelle zeigt den CO2-Ausstoß nach Unternehmen und Jahr, wobei Unternehmen einem bestimmten Land zugeordnet werden. Sortierbar durch Klick auf Header-Zeile.
- **Datenanzeige für Länder:** Eine separate Tabelle listet den CO2-Ausstoß nach Land und Jahr auf.
- **Visualisierung der Länder:** Die Integration von flag-icon-css ermöglicht es, jedes Land zusätzlich durch seine Flagge darzustellen, was die visuelle Verständlichkeit der Daten verbessert und die Benutzerinteraktion fördert.

Das Layout, die Farbpalette und die Typografie der Seite sind darauf ausgerichtet, die Lesbarkeit zu optimieren und gleichzeitig ein ansprechendes visuelles Erlebnis zu bieten.

2.4 Implementierung

Dieses Kapitel beschreibt die Implementierung auf technischer Ebene.

2.4.1 Einrichtung des öffentlichen Code-Repositories

Im Rahmen der Fallstudie war der erste Schritt, das Projekt mit dem Befehl `npx create-react-app` zu initialisieren, welches ein vorstrukturiertes React-Anwendungsgerüst bereitstellt. Nach der erfolgreichen Initialisierung wurde das lokale Git-Repository mit `git init` erstellt, um die Versionskontrolle zu ermöglichen.

Der nächste Schritt bestand darin, die erstellten Dateien zum lokalen Repository hinzuzufügen (`git add .`) und den ersten Commit mit einer aussagekräftigen Nachricht zu versehen (`git commit -m "Initial commit"`), um die anfängliche Version des Projekts festzuhalten.

Anschließend wurde das Repository auf GitHub hochgeladen, eine Plattform, die sich ideal für den Code-Austausch und die Kollaboration eignet, da sie umfangreiche Tools für die Versionskontrolle, das Issue Tracking und die Pull Requests bietet, was die gemeinsame Arbeit und die Transparenz in Open-Source-Projekten erheblich verbessert.

Für das Hochladen wurde ein neues Repository auf GitHub erstellt und dann das lokale Repository mit diesem verknüpft. Schließlich wurden die lokalen Commits mit `git push` in das neu eingerichtete öffentliche GitHub-Repository übertragen. Alle weiteren Entwicklungsarbeiten wurden in dieses Repository committet, wodurch eine kontinuierliche und nachvollziehbare Entwicklungsgeschichte der Anwendung gewährleistet wurde (Git Dokumentation, 2024; Kindler, 2024).

2.4.2 Umsetzung der Menüstruktur und Navigation

Es wurde eine möglichst effiziente Nutzernavigation mit besonderem Fokus auf Funktionalität und Erweiterbarkeit realisiert. Die Anwendung ist strukturiert in drei Hauptkomponenten:

- **AppHeader:** Hier findet sich eine programmatisch implementierte Navigationsleiste, die über die Suchfunktion, Sprachauswahl und Verlinkungen zu den Hauptsektionen der Anwendung verfügt. Technisch ermöglicht der Einsatz des React Routers eine Navigation ohne Seitenneuladungen, was Ladezeiten reduziert und das Nutzererlebnis verbessert.
- **AppMain:** Diese zentrale Komponente verwaltet den Hauptinhalt der Anwendung, wobei sie auf Änderungen in der Navigation reagiert und Inhalte dynamisch nachlädt und anzeigt.
- **AppFooter:** Technisch stellt der Footer eine konsistente und jederzeit zugängliche Informationsquelle dar, indem er über Verlinkungen zu weiterführenden Informationen und rechtlichen Texten verfügt.

Das technische Design der Anwendung gewährleistet volle Responsivität durch eine flexible Codebasis, die eine reibungslose Darstellung auf verschiedenen Geräten und Bildschirmgrößen ermöglicht. Bei der Entwicklung wurde darauf geachtet, dass Code und Struktur der Anwendung eine leichte Wartbarkeit und Skalierbarkeit für künftige Erweiterungen zulassen.

2.4.3 Implementierung der Seiteninhalte

Die technische Umsetzung der Seiteninhalte fokussiert sich auf eine modulare Struktur, welche die Bereitstellung von Inhalten in einem responsiven Format ermöglicht. Kernstück dieser Struktur sind Bootstrap-Komponenten.

Jede Seite ist in Kacheln aufgeteilt, die mittels Bootstrap "Cards" implementiert sind. Diese Card-Komponenten bieten einen flexiblen Container für Inhalte. In Kombination mit Bootstrap's Grid-System und Flexbox können diese Kacheln so konfiguriert werden, dass sie sich an verschiedene Bildschirmgrößen anpassen, was eine optimale Darstellung auf Geräten vom Smartphone bis zum Desktop-Monitor gewährleistet.

Für die Anordnung der Kacheln wird das Grid-System von Bootstrap genutzt, das eine reihenbasierte Flexbox-Struktur zur Verfügung stellt. Mit den entsprechenden Klassen wie `col-md-6` für mittelgroße Bildschirme oder `col-lg-4` für größere Displays wird die Inhaltsanzeige automatisch an die Bildschirmbreite des Endgeräts angepasst.

Die technische Umsetzung berücksichtigt auch Leistungsaspekte. Dazu gehört das Lazy Loading von Bildern, um die initiale Ladezeit zu minimieren, und die sorgfältige Verwaltung des Applikationszustands, um unnötige Rendervorgänge zu vermeiden. Auf diese Weise gewährleistet die Anwendung eine hohe Performance und verbessert das Nutzererlebnis.

2.4.4 Implementierung der Datentabelle

Die Implementierung der Datentabelle mit Filter- und Sortierfunktionen stützt sich auf die Verarbeitung und Darstellung von Daten aus CSV-Dateien. Ein Schlüsselaspekt hierbei ist die Integration eines CSV-Parsers, der es ermöglicht, Rohdaten effizient zu lesen und für die Anwendung aufzubereiten.

CSV Parser: Die Anwendung nutzt Papa Parse, eine leistungsstarke Bibliothek, um CSV-Daten zu parsen. Durch die Konvertierung der CSV-Daten in ein durch Javascript nutzbares Format kann die Anwendung auf diese Daten zugreifen und sie filtern und sortieren.

Datentabelle: Zwei Haupttabellen, `CompanyDataTable` und `CountryDataTable`, präsentieren die gefilterten Daten. Sie sind so konzipiert, dass die Flexibilität der Darstellung durch die Anpassungsfähigkeit an diverse Filter- und Sortieranforderungen gewährleistet wird.

Sortierfunktion: Nutzende haben die Möglichkeit, Daten basierend auf verschiedenen Kriterien wie dem Namen des Unternehmens, dem Land oder den CO2-Emissionswerten zu sortieren. Die Sortierung erfolgt durch eine benutzerdefinierte Hook `useSortData`, die es ermöglicht, die Sortierreihenfolge dynamisch zu ändern und die Daten entsprechend neu anzuordnen.

Filterfunktion: Die `FilterSidebar` ermöglicht es den Nutzern, spezifische Kriterien für die Datenanzeige festzulegen, wie Zeitraum, Unternehmen oder Länder. Diese Filterlogik wird durch die Funktion `useFilterCompanyData` unterstützt, die die Gesamtdatenmenge basierend auf den vom Nutzer gewählten Kriterien einschränkt.

2.4.5 Lokalisierung

Für die Mehrsprachigkeit wurde die Bibliothek `i18next` integriert, um Inhalte in Deutsch und Englisch über `de.json` und `en.json` bereitzustellen. Diese JSON-Dateien enthalten die für die Anwendung relevanten Texte, strukturiert in Schlüssel-Wert-Paaren, die eine einfache Pflege und Erweiterung ermöglichen. Mittels `react-i18next` und dem Hook `useTranslation` werden die lokalisierten Texte in die React-Komponenten eingebunden. Nutzer können die Sprache über ein Dropdown-Menü ändern, wobei die Anwendung die Inhalte dynamisch und ohne Neuladen der Seite aktualisiert.

2.4.6 Implementierung der Suchfunktion

Die Implementierung der Suchfunktion, eng verknüpft mit der Lokalisierung der Webanwendung, nutzt `i18next` für die Bereitstellung von mehrsprachigen Inhalten aus den Dateien `de.json` und `en.json`. Diese Konfiguration ermöglicht es, Suchanfragen nicht nur auf der Oberfläche der Anwendung, sondern auch in den lokalisierten Textinhalten zu berücksichtigen.

Zur Realisierung der Suchfunktion wird zunächst eine rekursive Hilfsfunktion `searchInTranslations` eingesetzt, die durch die lokalisierten Texte navigiert. Diese Funktion identifiziert alle Textelemente, die die Suchanfrage enthalten, und sammelt sie, unabhängig davon, in welcher Tiefe des Objekts sie sich befinden. Dieser Ansatz gewährleistet eine umfassende Durchsuchung der Übersetzungsressourcen.

Anschließend konsolidiert die Funktion `calculateSearchResults` die Suchergebnisse, indem sie die gefundenen Treffer den entsprechenden Routen der Anwendung zuordnet. Diese Zuordnung basiert auf den Schlüsseln der obersten Ebene in den Übersetzungsdateien, wie `home` oder `co2info`, die den einzelnen Seiten der Anwendung entsprechen. Die Ergebnisse werden dann so zurückgegeben, dass sowohl die Route als auch die zugehörigen Suchtreffer umfasst werden.

Innerhalb der Komponente `SearchResults` wird auf diese Funktionalität zurückgegriffen, um die Suchergebnisse basierend auf der aktuellen Spracheinstellung und der eingegebenen

Suchanfrage zu ermitteln und anzuzeigen. Ein wichtiger Aspekt hierbei ist die dynamische Anpassung der Suchergebnisse an die Spracheinstellung des Nutzers, was durch die Abhängigkeit der `useEffect-Hook` vom aktuellen Sprachkontext (`i18n.language`) sichergestellt wird.

Für die Darstellung der Suchergebnisse werden Bootstrap-Komponenten verwendet, die eine konsistente und ansprechende visuelle Präsentation ermöglichen. Jedes Suchergebnis wird in einer card-Komponente angezeigt, die Links zu den jeweiligen Seiten der Anwendung enthält und eine intuitive Navigation durch die Suchergebnisse bietet (Abbildung 4).

2.4.7 Sicherheitsaspekte: Schutz vor Code-Injektion und XSS

Beim Parsen und Filtern der Eingaben für die Such- und Filterfunktionen wird besonderer Wert auf die Sicherheitsaspekte gelegt. Hierzu wird beispielsweise die Funktion `escapeStringRegexp` eingesetzt, die sicherstellt, dass potenziell gefährliche Zeichen oder Zeichenketten, die Teil eines Angriffsvektors sein könnten, neutralisiert werden, bevor sie weiterverarbeitet oder in die Anwendung eingespeist werden. Diese Vorgehensweise verhindert effektiv, dass schädlicher Code als Teil der Nutzereingaben interpretiert und ausgeführt wird.

Des Weiteren trägt die sorgfältige Behandlung von externen Datenquellen und deren Integration in die Anwendung, beispielsweise durch den Einsatz von Papa Parse für das Einlesen von CSV-Dateien, zur Sicherheit bei. Durch die Validierung und Aufbereitung der Daten vor ihrer Darstellung schützt Papaparse die Anwendung vor möglichen XSS-Angriffen, die durch das Einbetten von Schadcode in die Datenquellen erfolgen könnten (Papaparse Dokumentation, 2024).

2.5 Testing und Optimierung

In diesem Prozessschritt werden sowohl die Funktionalität als auch die Benutzerfreundlichkeit der Anwendung überprüft, um sicherzustellen, dass sie den Erwartungen gerecht wird.

2.5.1 Funktionstests

Im Rahmen dieser Fachstudie wurden die Funktionstests manuell durchgeführt. Dabei wurden sämtliche Funktionalitäten der Anwendung durchgespielt, um sicherzustellen, dass alle Komponenten wie erwartet arbeiten. Perspektivisch bietet sich die Integration eines automatisierten Testframeworks an, um beispielsweise Unit Tests systematisch durchführen zu können. Aufgrund der Integration von i18next muss hierfür auch Mocking zum Einsatz kommen. Die meisten Komponenten sind durch Verwendung von Dependency Injection als Design Pattern bereits für Testing vorbereitet. Automatisierte Tests ermöglichen eine umfassende Überprüfung der Anwendung, insbesondere bei zukünftigen Erweiterungen oder Anpassungen des Codes.

2.5.2 Responsivitätsprüfung

Die Überprüfung der Responsivität wurde mithilfe des Browser-Web-Inspectors durchgeführt, der es erlaubt, die Anwendung in verschiedenen Bildschirmgrößen und Auflösungen zu testen. Diese umfangreichen Tests stellten sicher, dass die Webseite auf einer Vielzahl von Geräten korrekt dargestellt wird. Zusätzlich wurde die Seite auf Mobiltelefonen in Hoch- und Querformat betrachtet, um nicht nur die visuelle Darstellung, sondern auch das Touchverhalten zu prüfen. Durch diese Tests konnte die Benutzererfahrung auf mobilen Geräten optimiert werden, was angesichts des zunehmenden mobilen Internetverkehrs von großer Bedeutung ist.

2.5.3 Sicherheitstests

Im Rahmen der Sicherheitstests wurden speziell manuelle Überprüfungen aller Eingabefelder der Webanwendung durchgeführt, um die Anfälligkeit für Cross-Site Scripting (XSS) zu evaluieren. Ein Beispiel ist die Eingabe von `<script>alert("XSS")</script>` in die verfügbaren Eingabefelder.

Das primäre Ziel dieser Tests war es, das ordnungsgemäße Escaping von Benutzereingaben zu überprüfen. Escaping ist ein wichtiger Sicherheitsmechanismus, der sicherstellt, dass Eingaben als reiner Text behandelt und nicht als Teil des ausführbaren Codes der Webseite interpretiert werden. Die korrekte Anwendung dieses Verfahrens verhindert, dass eingegebener Code als Skript ausgeführt wird, das potenziell dazu genutzt werden könnte, sensible Informationen zu stehlen oder Benutzerumleitungen zu schädlichen Websites zu verursachen.

Die Tatsache, dass keine der Testeingaben zur Ausführung von JavaScript führte, bestätigt die Wirksamkeit der implementierten Escaping-Strategien gegen XSS-Angriffe.

3 - Schluss

Im Zuge dieses Projekts wurde erfolgreich eine responsive und funktionsfähige Webanwendung mit mehreren Landingpages und einer Datenbankansicht entwickelt, die den festgelegten Anforderungen gerecht wird. Die Anwendung dient als Plattform, um Bewusstsein für CO2-Emissionen zu schaffen und bietet Nutzern die Möglichkeit, sich über das Thema zu informieren und aktiv an der Reduzierung ihres CO2-Fußabdrucks zu arbeiten.

3.1 Diskussion der Herausforderungen und Lösungsansätze

Die Entwicklung der Anwendung stellte mit der Findung des Technologieansatzes zunächst eine Herausforderung dar. Der letztendliche Einsatz von Create React App lieferte eine solide Grundlage, die eine schnelle und effiziente Implementierung ermöglichte. Die Entscheidung für den Einsatz weiterer Technologiebausteine wie CSV für die Datenspeicherung, Typescript für höhere Codequalität, i18next für Mehrsprachigkeit und React für dynamisches Rendering wurde mit Blick auf eine zukünftige Skalierbarkeit und Erweiterbarkeit getroffen. Das Design stellte eine besondere Herausforderung dar, da eine Balance zwischen Ästhetik und Funktionalität gefunden werden musste. Mithilfe von Bootstrap und einem iterativen Gestaltungsprozess konnte eine benutzerfreundliche und zugleich ansprechende Benutzeroberfläche geschaffen werden, die auf verschiedenen Endgeräten einwandfrei funktioniert.

3.2 Ausblick und mögliche Weiterentwicklungen

Für die Zukunft könnten mehrere potentielle Erweiterungen in die Webanwendung integriert werden. Um Nutzende regelmäßig über Neuigkeiten und Entwicklungen zu informieren, könnte eine Integration von Mailchimp für Newsletter-Funktionalität realisiert werden. Eine Profilerstellung, die es Nutzern ermöglicht, eigene Presets für häufig genutzte Abfragen zu speichern, würde den Komfort der Anwendung weiter steigern. Grafische Darstellungen der Daten könnten die Interaktion und das Verständnis der Informationen verbessern. Eine Umstellung auf eine echte Datenbasis oder ein Backend würde die Aktualität und die Präzision der Daten erhöhen. Die Nutzung eines Content Management Systems wie Storyblok könnte die Inhaltsverwaltung erleichtern und dynamisieren. Abschließend würde eine SEO-Optimierung der Seite ihre Auffindbarkeit verbessern, was eventuell eine Auftrennung zwischen den informativen Landingpages und der interaktiven Datenbank-App erfordern könnte oder durch den Einsatz von serverseitigem Rendering oder modernen Webframeworks wie Astro erreicht werden könnte.

Zusammenfassend hat dieses Projekt eine robuste Basis geschaffen, auf der zukünftige Erweiterungen und Verbesserungen aufbauen können, um die Anwendung noch effektiver in der Aufklärung und im Engagement gegen den Klimawandel zu machen.

I - Literaturverzeichnis

- Bootstrap Dokumentation. "Bootstrap Documentation." Abgerufen am 22.04.24, <https://getbootstrap.com/docs/>.
- Create React App Dokumentation. "Create React App Documentation." Abgerufen am 22.04.24, <https://create-react-app.dev/docs/getting-started/>.
- React Dokumentation. "React Documentation." Abgerufen am 22.04.24, <https://reactjs.org/docs/getting-started.html>.
- I18next Dokumentation. "i18next Documentation." Abgerufen am 22.04.24, <https://www.i18next.com/>.
- Flag-icon-css GitHub Repository. "flag-icon-css GitHub Repository." Abgerufen am 22.04.24, <https://github.com/lipis/flag-icon-css>.
- Papaparse Dokumentation. "Papa Parse Documentation." Abgerufen am 22.04.24, <https://www.papaparse.com/>.
- TypeScript Dokumentation. "TypeScript Documentation." Abgerufen am 22.04.24, <https://www.typescriptlang.org/docs/>.
- TypeScript Compiler Dokumentation. "TypeScript Compiler Documentation." Abgerufen am 22.04.24, <https://www.typescriptlang.org/docs/handbook/compiler-options.html>.
- Flexbox Dokumentation. "CSS Flexible Box Layout - MDN Web Docs." Abgerufen am 22.04.24, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout.
- Express.js Dokumentation. "Express.js Documentation." Abgerufen am 22.04.24, <https://expressjs.com/>.
- Git Dokumentation, "Git Documentation." Abgerufen am 22.04.24, <https://git-scm.com/doc>.
- eRecht24. "eRecht24 – Rechtssichere Websites nach deutschem Recht." Abgerufen am 22.04.24, <https://www.e-recht24.de>
- DALL-E Dokumentation. „DALL-E: A Generative Model for Image Creation.“ OpenAI. Abgerufen am 22.04.24, <https://openai.com/dall-e>
- Kindler, R. F. A. (2024). *Climate Impact Dashboard*. GitHub. Abgerufen am 22. April 2024, von <https://github.com/Rondidon/climate-impact-dashboard/tree/main>
- Grünes Blatt Logo Vektor. „Vecteezy“ Abgerufen am 22.04.24. <https://de.vecteezy.com/vektorkunst/626835-grunes-blatt-logo-vektor>

II - Abbildungsverzeichnis

Abbildung 1: Startseite der Anwendung (große Bildschirme).....	15
Abbildung 2: Lokale Seitennavigation (mobile Endgeräte).....	15
Abbildung 3: Ansicht und Menüstruktur (mobilen Endgeräte).....	16
Abbildung 4: Präsentation der Suchergebnisse (mobile Endgeräte).....	16
Abbildung 5: Darstellung der CO2-Emissionsdaten inklusive Filterfunktion (große Bildschirme)....	17
Abbildung 6: Darstellung der CO2-Emissionsdaten inklusive Filterfunktion (mobile Endgeräte).....	17
Abbildung 7: Lokalisierung inklusive Auswahlmenü im Headerbereich (große Bildschirme).....	18
Abbildung 8: Lokale Seitennavigation (große Bildschirme).....	18

III - Anhang A – Screenshots der Anwendung

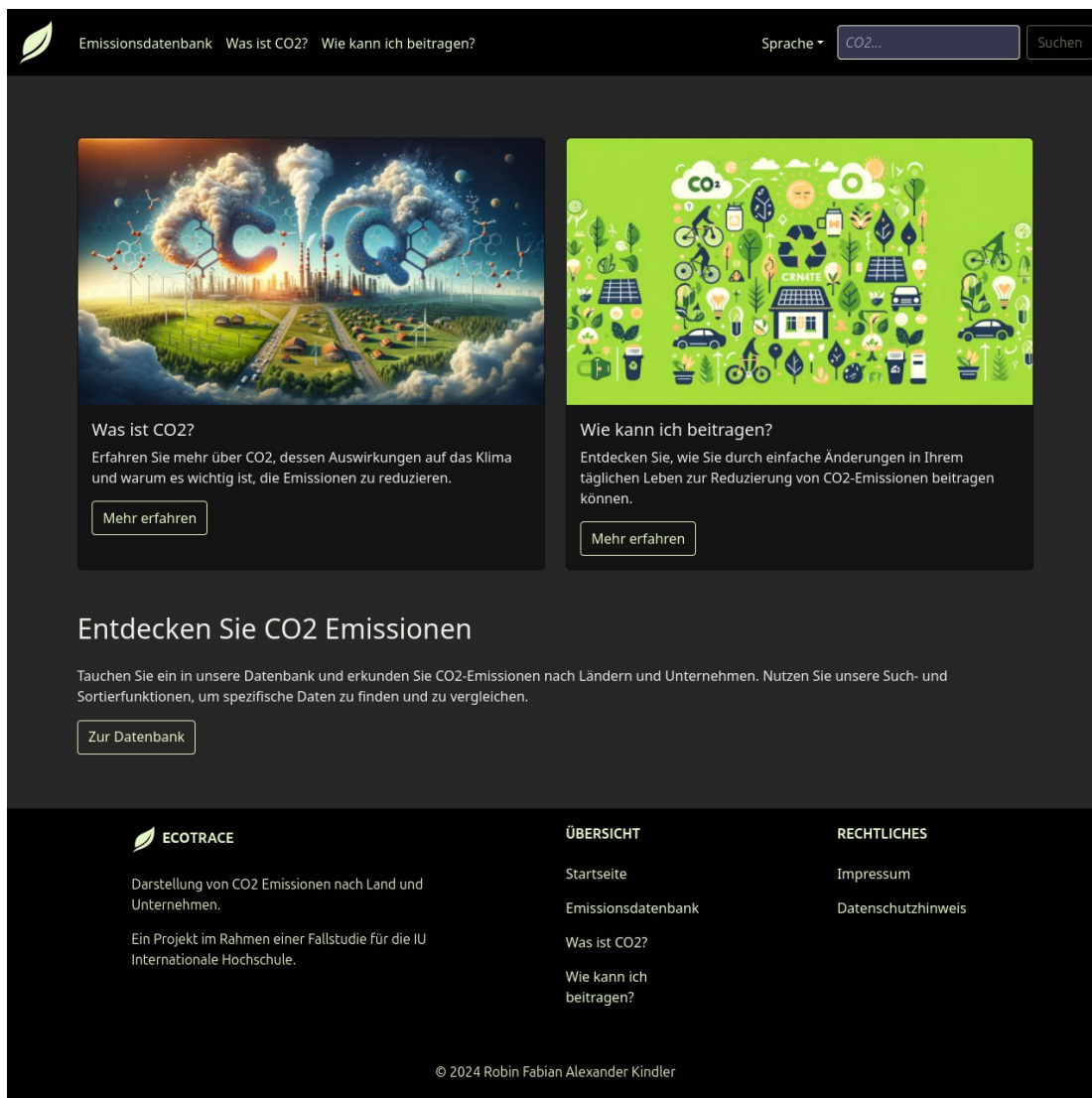


Abbildung 1: Startseite der Anwendung (große Bildschirme).

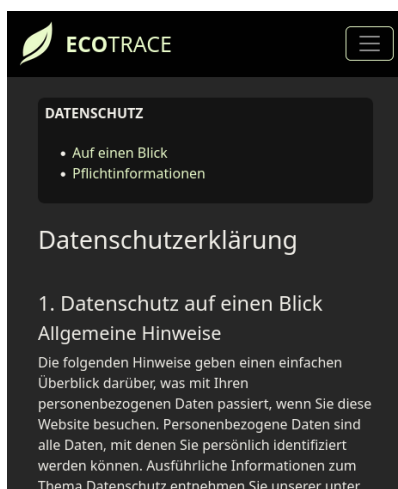


Abbildung 2: Lokale Seitennavigation (mobile Endgeräte).

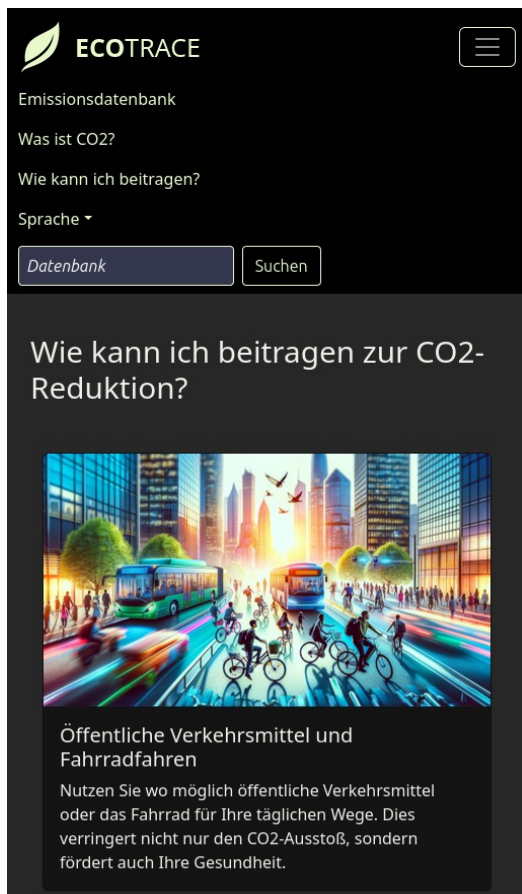


Abbildung 3: Ansicht und Menüstruktur (mobilen Endgeräte).

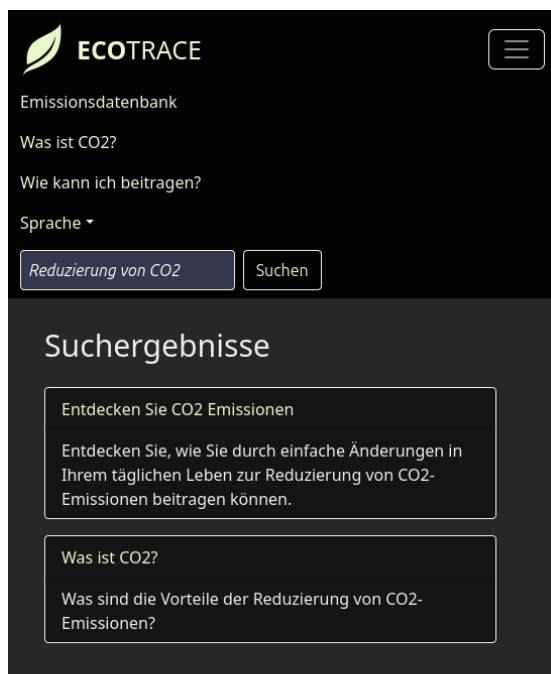


Abbildung 4: Präsentation der Suchergebnisse (mobile Endgeräte)

Emissionsdatenbank
Was ist CO2?
Wie kann ich beitragen?

Sprache
Datenbank
Suchen

CO2 Emissionsdatenbank

Die hier aufgeführten Daten zeigen die jährlichen CO2-Emissionen, gemessen in Kilotonnen. Sie können die Emissionen jedes Unternehmens nachverfolgen und vergleichen, um Muster, Trends und Fortschritte in Richtung umweltfreundlicherer Geschäftspraktiken zu erkennen. Mit der Sortier- und Filterfunktion können Sie die Informationen anpassen, um spezifische Datenpunkte herauszufiltern, wie beispielsweise Emissionen in einem bestimmten Jahr oder die Performance von Unternehmen in bestimmten Ländern.

Voreinstellungen

Alle Daten anzeigen
Dieses Jahr
Letzte drei Jahre

Filter

Jahr (von):
2021

Jahr (bis):
2021

☒ Unternehmenstabelle anzeigen
☒ Ländertabelle anzeigen

Unternehmen

Commerce

Ländercodes

CN,DE,US

Abschicken

Unternehmen

CO2-Emissionen in Kilotonnen nach Unternehmen.

Unternehmen	Land	2021
Oil	US	332
Battery	CN	330
Space	US	297
Spy	CN	290
Metal	DE	262
Control	US	263
Silicon	US	242
Commerce	CN	239

Länder

CO2-Emissionen in Kilotonnen nach Land.

Land	Ländername	2021
DE	Germany	758
US	United States	4655
CN	China	9489

Abbildung 5: Darstellung der CO2-Emissionsdaten inklusive Filterfunktion (große Bildschirme)

Voreinstellungen

Alle Daten anzeigen
Dieses Jahr
Letzte drei Jahre

Filter

Jahr (von):
2021

Jahr (bis):
2021

☒ Unternehmenstabelle anzeigen
☒ Ländertabelle anzeigen

Unternehmen

Commerce

Ländercodes

CN

Abschicken

Unternehmen

CO2-Emissionen in Kilotonnen nach Unternehmen.

Unternehmen	Land	2021
Battery	CN	330
Spy	CN	290
Commerce	CN	239

Länder

CO2-Emissionen in Kilotonnen nach Land.

Land	Ländername	2021
CN	China	9489

Abbildung 6: Darstellung der CO2-Emissionsdaten inklusive Filterfunktion (mobile Endgeräte)

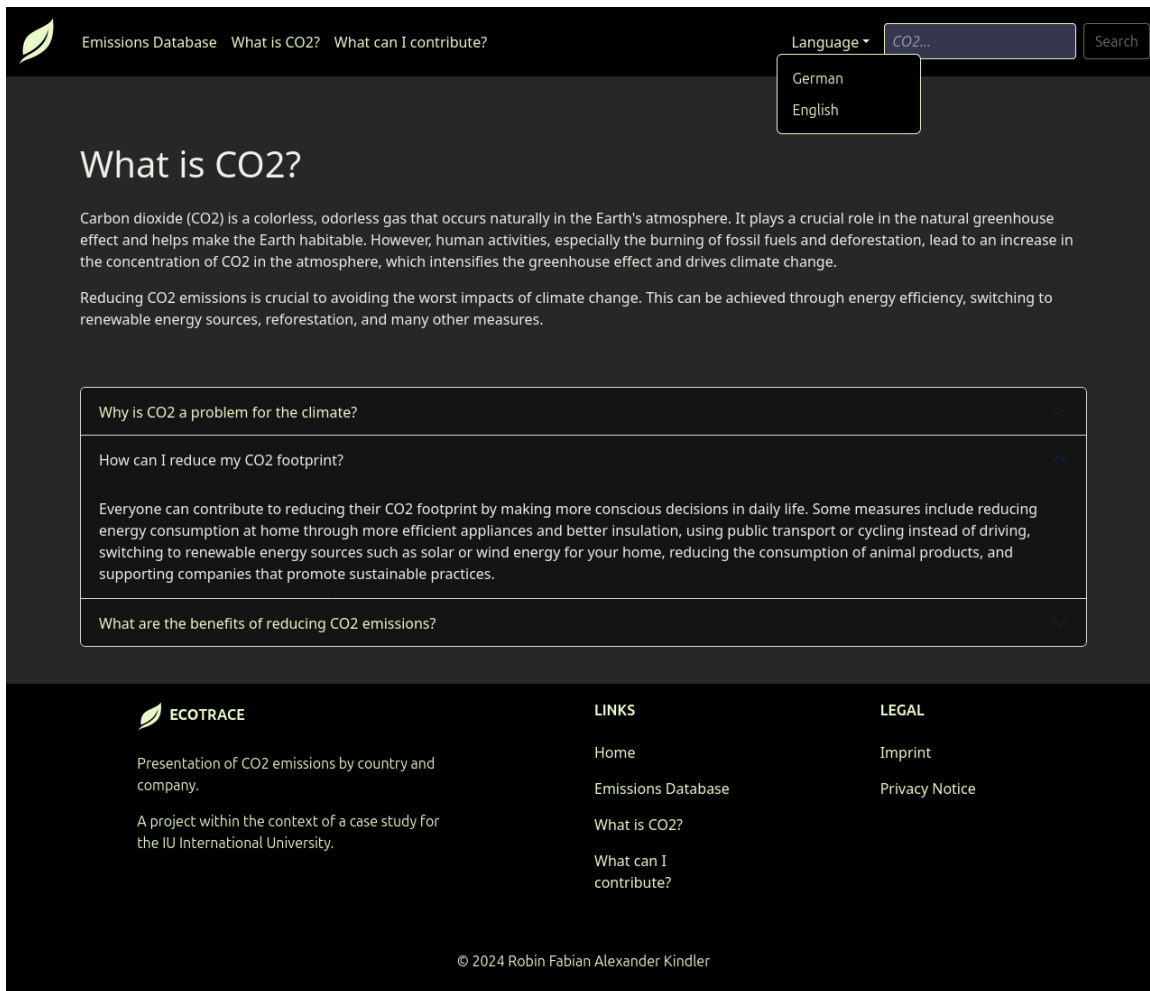


Abbildung 7: Lokalisierung inklusive Auswahlmenü im Headerbereich (große Bildschirme).

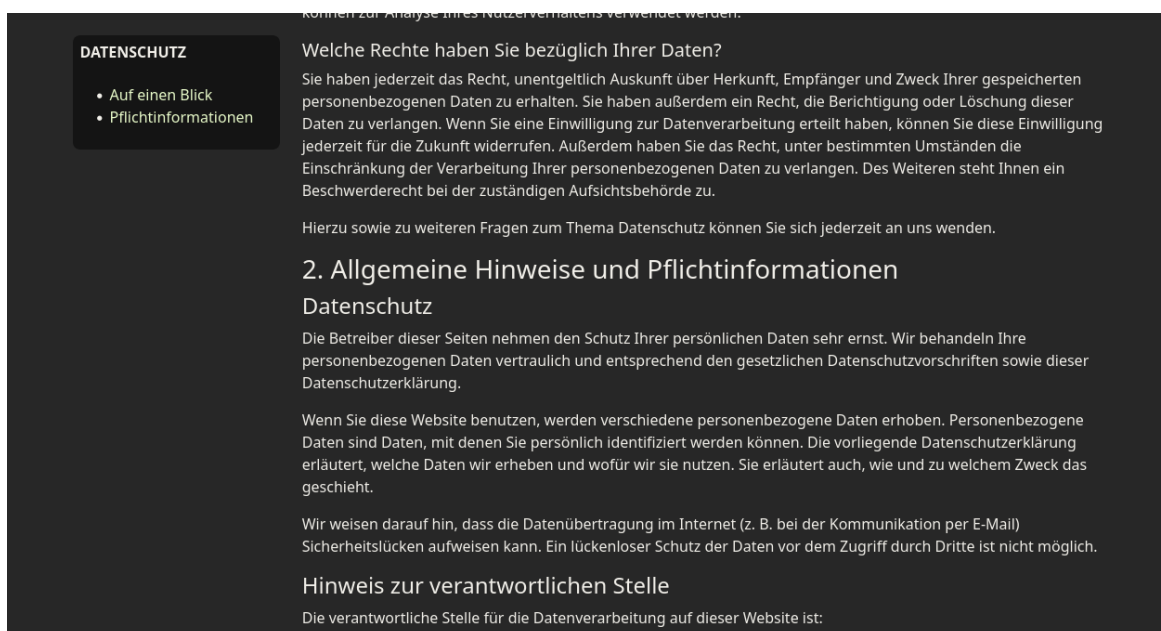


Abbildung 8: Lokale Seitennavigation (große Bildschirme).

IV - Anhang B – GitHub Repository zum Projekt

GitHub Repository: <https://github.com/Rondidon/climate-impact-dashboard/tree/main>

Nach dem Klonen des Repositorys kann im Projektverzeichnis folgender Befehl ausgeführt werden: *npm start*. Dies startet die App im Entwicklungsmodus. Öffnen Sie <http://localhost:3000> in Ihrem Browser, um sie anzusehen.

Weitere Anweisungen finden Sie in der README.md, die sich im Projektverzeichnis befindet, oder auf der GitHub-Seite des Repositories.