

IU Internationale Hochschule

Programmierung von Web-Anwendungsoberflächen (IPWA01-01)

Alexander Christopher Bock

FSS 2025

31.01.2025

Fallstudie

Die Höhe der unternehmerischen CO₂-Emissionen mittels Emission Omission entdecken

Evsin Rahmiev

Wirtschaftsinformatik B. Sc.

Akademiestr. 6

68159 Mannheim

Tel. +49 176 43474925

evsin.rahmiev@iu-study.org

Matrikeln. 32105477

GitHub Repository: <https://github.com/evsinevsin/evsin.git>

Inhaltsverzeichnis

I. Abkürzungsverzeichnis.....	II
II. Abbildungsverzeichnis.....	II
1. Einleitung	4
2. Header, Titel und Logo	5
3. Globale Navigation, Content-Bereich und Footer.....	6
4. Sidebar-Menu.....	9
5. Die Anwendung responsiv machen	11
6. Interaktive Emissionstabelle	14
7. Absicherung gegen Cross-Site-Scripting (XSS).....	16
8. Schluss.....	18
IV. Literaturverzeichnis	19

I. Abkürzungsverzeichnis

1. u. a. – unter anderem	12
3. bzw. - beziehungsweise	4
4. z. B. – zum Beispiel	9
6. bspw. - beispielsweise	9
7. sog. – so genannt	6

II. Abbildungsverzeichnis

Abb. 1: Header mit Titel und Logo	5
Abb. 2: Globale-Navigation.....	6
Abb. 3: FAQ-Frage als Card-Header	7
Abb. 4: Antwort als Collapse.....	8
Abb. 5: Footer.....	8
Abb. 6: Sidebar-Navigation.....	9
Abb. 7 Positionierungsbutton	10
Abb. 8: Eigene CSS-Klasse.....	10
Abb. 9: JavaScript Event-Handler Funktion	11
Abb. 10: Responsiver Grid-Container	11
Abb. 12. Ausblenden des Dropdown-Buttons.....	13
Abb. 13: Media Query zum Dropdown Navigations-Button	13
Abb. 14: Zugriff mittels ID und Anlegen eines Zeilenarrays.....	14
Abb. 15: Alphabetische Sortierung der Zelleninhalte.....	14
Abb. 16: Anhängen der Zellen an das Tabellenende	15
Abb. 17: Zugriff auf die Tabelle und ihre Zeilen	15
Abb. 18: Überprüfung der Zeileninhalte und Ein- bzw. Ausblenden der Zeilen	16
Abb. 19: Funktion zur Input-Ersetzung.....	17

1. Einleitung

Die Non-Profit Organisation „Emission Omission“ (aus dem Englischen übersetzt: Emissionsunterlassung) beschäftigt sich damit, Menschen über den negativen Beitrag diverser Länder und Unternehmen zum Klimawandel zu informieren. Unternehmen werden von immer mehr Stakeholdern dazu aufgefordert, Nachhaltigkeitsberichte als eine Art „Treibhausgas-Bilanzierung“ bereitzustellen, damit sie der Öffentlichkeit Aufschluss über ihren CO₂-Fußabdruck geben (Glaser, 2022, S. 30). Da allerdings nicht alle Unternehmen dieser Aufforderung nachkommen, engagiert sich Emission Omission dafür, die konkreten Werte zugänglich zu machen. Zu diesem Zweck hat die Non-Profit Organisation die Zahlen auf einer Webanwendung übertragen, weil Internetseiten einfach zu bedienen sind und für eine Vielzahl an unterschiedlichen Endgeräten zugänglich sind.

Der Gegenstand dieser schriftlichen Ausarbeitung ist es, zu erläutern, wie mithilfe moderner Webentwicklungstechnologien eine Webanwendung programmiert werden kann, die im konkreten Fallbeispiel die CO₂-Emissionen verschiedener Länder und deren Unternehmen auflistet. Um die Idee über eine solche Webanwendung umzusetzen, werden HTML, CSS und JavaScript und das Frontend-Framework CSS Bootstrap eingesetzt. Im konkreten Umsetzungsablauf werden im ersten Schritt der Titel und das Logo vorgestellt, die auf der Homepage ersichtlich sind. Danach werden der Header und der Footer, die auf jeder Seite der Anwendung zu finden sind, zusammen mit dem kurzen Content-Bereich auf der Startseite vorgestellt. Der nächste Schritt präsentiert die Sidebar-Navigation, die per Click entweder auf der linken oder auf der rechten Seite des Bildschirms platziert werden kann. Im darauffolgenden Schritt wird übermittelt, wie die Webanwendung responsiv programmiert wurde, so dass sie auf unterschiedliche Bildschirmauflösungen bzw. Endgeräten optimal dargestellt werden kann. Im vorletzten Schritt wird die Emissionstabelle präsentiert, die die wichtigste Eigenschaft der Web-Anwendung ist. Im letzten Schritt wird erklärt, wie die Eingabefelder auf der Webseite gegen Cross-Site-Scripting Angriffe abgesichert werden. Abschließend wird die Umsetzung des Projekts zusammengefasst. Die Anwendung wurde in der Entwicklungsumgebung „Visual Studio Code“ entwickelt. Zur Veranschaulichung bzw. einfacheren Nachvollziehbarkeit der Implementierungsschritte werden Screenshots in dieser schriftlichen Ausarbeitung integriert. Darüber hinaus wird eine Einsicht in den Programmcode ermöglicht, der in einer öffentlichen Repository auf der Internetplattform zur Versionskontrolle „GitHub“ abgelegt wird. Die Anwendung ist eine reine Front-End-Anwendung und wird somit nur clientseitig ausgeführt – auf ein Backend zum Speichern der Daten und Ausführung von serverseitigen Prozessen (Donvir, Saraswathi & Jain, 2024, S. 32) wird zu diesem Zeitpunkt verzichtet.

2. Header, Titel und Logo

Der Titel und das Logo der Emission Omission Webseite werden zusammen in dem semantischen HTML-Tag `<header>` abgelegt, da diese eine Art Dokumentenkopf darstellen und somit der Tag zur inhaltlichen Beschreibung der Webanwendung dienen (Bühler, Schlaich & Sinner, 2023, S. 14). Der Titel hat die Aufgabe, den Zweck der Webseite in einem Satz zu verfassen und wird als eine Überschrift erster Ordnung (`<h1>`) angelegt:

Abb. 1: Header mit Titel und Logo

```
<header class="text-center py-5">
  <div class="container">
    <h1>
      Verstehen Sie CO2-Emissionen: Ein entscheidender Schritt in Richtung
      Nachhaltigkeit.
    </h1>
    <p class="mt-3">
      Diese Seite bietet Ihnen umfassende Informationen über die
      CO2-Emissionen verschiedener Unternehmen – aufgeschlüsselt nach
      Branchen und einzelnen Firmen. Unser Ziel ist es, Transparenz zu
      schaffen und Einblicke in die Auswirkungen verschiedener
      Wirtschaftszweige auf das Klima zu geben.
    </p>
    <a href="tabelle.html" class="link-success text-decoration-none">
      >Überblick Emissionen</a>
    >
  </div>
  
</header>
```

Quelle: Eigene Darstellung

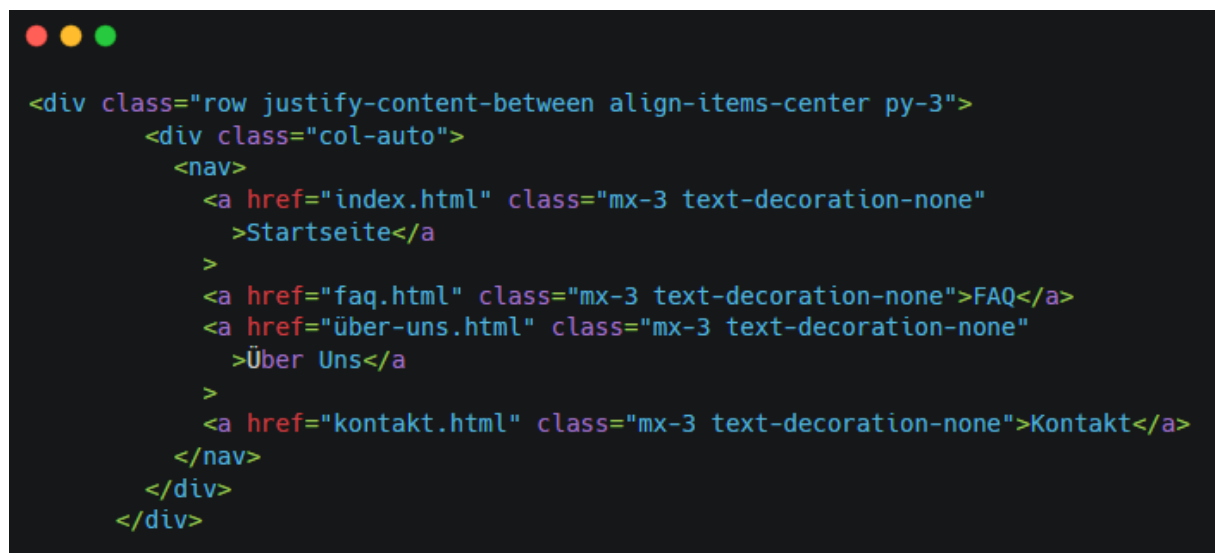
Ein weiterer Bestandteil des Headers ist der Hyperlink mit der Bezeichnung *Überblick Emissionen*, der als eine Art *Call-to-Action* (CTA) funktioniert, weil der Link die Besucher dazu bewegt, eine Handlung auszuführen (Becker, 2024). Bootstrap CSS wird im Header verwendet, um den Text mittels der Bootstrap Utility-Klasse *text-center* zu zentrieren, dem gesamten Header einen Innenabstand (*Padding*) mittels *py-5*, der Beschreibung unterhalb des Titels und dem Logo-Bild einen oberen Außenabstand (*Margin-Top*) mittels der Bootstrap Utility-Klasse *mt-3* bzw. *mt-4* zu vergeben (Matsinopoulos, 2020, S. 333, S. 293). Darüber hinaus wird dem Logo-Bild, also dem ``-Element, die Helfer-Klasse *img-fluid* zugewiesen,

damit die Höhe des Bilds sich automatisch an seine Breite anpasst (Matsinopoulos, 2020, S. 325).

3. Globale Navigation, Content-Bereich und Footer

Die globale Navigation ist eins der Seitenelemente, die auf jeder Seite der Emission Omission Webanwendung vorhanden ist. Strukturell ist sie von zwei Containern umgeben: der äußere besitzt die Klasse *row* (Krause, 2020, S. 89), weil er ein Grid-Container ist und die Elemente des Containers horizontal als eine Reihe platziert werden sollen, was in Standard-CSS der *Grid-Row*-Eigenschaft entspricht. Der innere Container hat die Klasse *col-auto*, die ermöglicht, dass die Kind Elemente des Containers sich entsprechend der Größe deren Inhalte dehnen (Krause, 2020, S. 88). Im Kern der globalen Navigation ist ein semantisches Navigations-Element (*<nav>*), dass die als Anker-Elemente angelegte Auswahl-Optionen der Navigation enthält. Den Anker-Elementen werden die Bootstrap-Klassen *mx-3* und *text-decoration-none* zugewiesen, damit sie einen Außenabstand nach links und rechts bekommen und die Unterstriche der Standard-Hyperlinks entfallen:

Abb. 2: Globale-Navigation

A screenshot of a code editor with a dark background and light-colored text. The code is HTML for a global navigation bar. It starts with a `<div class="row justify-content-between align-items-center py-3">` tag. Inside is a `<div class="col-auto">` tag, which contains a `<nav>` tag. The `<nav>` tag contains four anchor tags: `Startseite`, `FAQ`, `Über Uns`, and `Kontakt`. The code is properly closed with `</nav>`, `</div>`, and `</div>` tags. The code is color-coded: `<div` is blue, `class` is green, `href` is green, `class` is green, `>` is red, `` is red, `</div>` is blue, and `</div>` is blue. The text of the links is in a light gray color.

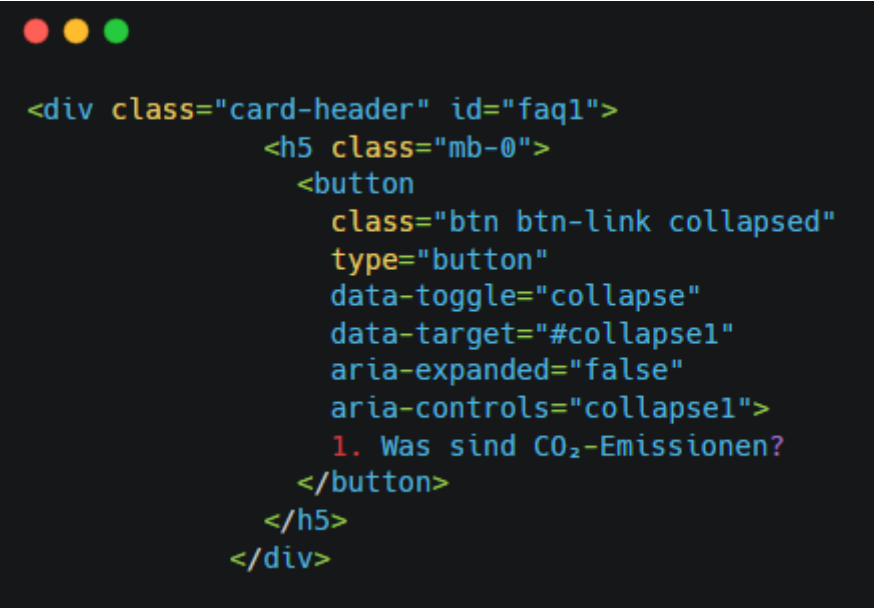
```
<div class="row justify-content-between align-items-center py-3">
  <div class="col-auto">
    <nav>
      <a href="index.html" class="mx-3 text-decoration-none"
      >Startseite</a>
      <a href="faq.html" class="mx-3 text-decoration-none">FAQ</a>
      <a href="über-uns.html" class="mx-3 text-decoration-none"
      >Über Uns</a>
      <a href="kontakt.html" class="mx-3 text-decoration-none">Kontakt</a>
    </nav>
  </div>
</div>
```

Quelle: Eigene Darstellung

Da die Webseite von Emission Omission eine Mehrseitige Webanwendung ist, hat jede Seite auf der Anwendung einen eigenen Content-Bereich. Zur Veranschaulichung in diesem wird der Content-Bereich der FAQ-Seite als Beispiel herangezogen. Die FAQ-Seite enthält eine

Auflistung von acht häufig gestellten Fragen zu Thema CO₂-Emissionen und die Antworten zu diesen Fragen. Die Fragen und dazugehörigen Antworten sind als sog. *Cards* mithilfe der Helfer-Klasse *card* (Matsinopoulos, 2020, S. 219) innerhalb eines Containers, der die Klasse *accordion* besitzt (Krause, 2020, S. 317). Der Aufbau jeder Card besteht aus jeweils einem *Card-Header*, der die eigentliche Frage darstellt und dem *Collapse*, der einen aufklappbaren Inhaltsbereich darstellt und die Aufgabe hat, nur temporär einen bestimmten Inhalt darzustellen und im Nachhinein den wertvollen Platz auf der Seite wieder freizustellen (Krause, 2020, S. 315). Die Frage, die sich innerhalb des Elements mit der Card-Header-Klasse befindet, ist als ein Button angelegt, der bei dem ersten Anklicken den *Collapse* aufklappt und bei dem zweiten diesen wieder zuklappt. Der Frage-Button wiederum befindet sich in einer Überschrift fünfter Ordnung (<h5>), die die Helfer-Klasse *mb-0* beinhaltet, damit kein Außenabstand zwischen der Frage und der Antwort vorhanden ist (Matsinopoulos, 2020, S. 198):

Abb. 3: FAQ-Frage als Card-Header



```
<div class="card-header" id="faq1">
  <h5 class="mb-0">
    <button
      class="btn btn-link collapsed"
      type="button"
      data-toggle="collapse"
      data-target="#collapse1"
      aria-expanded="false"
      aria-controls="collapse1">
      1. Was sind CO2-Emissionen?
    </button>
  </h5>
</div>
```

Quelle: Eigene Darstellung

Alle Buttons besitzen die Klasse *btn-link*, damit sie als Links erscheinen und die Klasse *collapsed*, damit der Default-Zustand von jedem Antwort-Bereich zugeklappt ist (Matsinopoulos, 2020, S. 166). Der Antwort-Bereich, der innerhalb jedes Collapse-Containers zu finden ist, ist in einem eigenen Container mit der Klasse *card-body* zu finden (Krause, 2020, S. 257):

Abb. 4: Antwort als Collapse


```

<div
    id="collapse1"
    class="collapse"
    aria-labelledby="faq1"
    data-parent="#faqAccordion">
    <div class="card-body">
        CO2-Emissionen sind Freisetzungen von Kohlendioxid (CO2) in die
        Atmosphäre. Sie entstehen vor allem durch menschliche
        Aktivitäten wie die Verbrennung fossiler Brennstoffe (z. B.
        Kohle, Öl und Gas), Industrieprozesse und Abholzung. CO2 trägt
        wesentlich zum Treibhauseffekt und zur globalen Erwärmung bei.
    </div>
</div>

```

Quelle: Eigene Darstellung

Der Footer ist von dem semantischen HTML-Element `<footer>` umgewickelt und besitzt auch einige Bootstrap-Klassen, die das Erscheinungsbild des Elements modifizieren. Diese sind *bg-dark*, was dem Footer einen dunklen Hintergrund verleiht, *text-white*, damit die Auswahl-Optionen des Footers eine weiße Schrift haben (Krause, 2020, S. 189), *py-3* und *mt-5*, damit der Footer über Innen- und Außenabstände (*Margins* und *Paddings*) verfügt. Die Anker-Elemente haben ähnlich zu den Anker-Elementen in der globalen Navigation die gleichen Klassen für Margin-Angabe und Text-Dekoration und zusätzlich die Klasse für weiße Schriftfarbe:

Abb. 5: Footer

```

<footer class="bg-dark text-white py-3 mt-5">
    <div class="container text-center">
        <a href="impressum.html" class="mx-3 text-decoration-none text-white">
            >Impressum</a>
        >
        <a
            href="datenschutz.html"
            class="mx-3 text-decoration-none text-white"
            >Datenschutz</a>
        >
    </div>
</footer>

```

Quelle: Eigene Darstellung.

4. Sidebar-Menu

Ein weiteres Component, das auf jeder Seite der Emission Omission Webanwendung vorhanden ist, ist die Sidebar-Navigation. Diese stellt Links zu Inhalten, die über die globale Navigation nicht zugänglich sind, wie z. B. die Emissionstabelle und die Emissionsgrafiken. Der äußere Container der Sidebar ist die semantische `<nav>`, welche die Bootstrap-Klasse `navbar` und die individuell eingerichtete Klasse `sidebar-navigation` zugewiesen bekommt, weil Bootstrap nur die Grundlage der Navigation bereitstellt und spezifische Einstellungen, wie bspw. die Position der Navigation, händisch eingerichtet werden müssen (Krause, 2020, S. 222). Innerhalb des Containers ist die Navigation als eine ungeordnete Liste (``) und jeder Menüpunkt der Navigation als Anker-Element in einem Listenelement (``) angelegt. Die Listenelemente tragen die Helfer-Klasse `nav-item` und die Anker-Elemente die Klasse `nav-link`.

Abb. 6: Sidebar-Navigation




```
<nav class="navbar sidebar-navigation">
  <div class="container-fluid">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="tabelle.html">Tabelle</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="faq.html">FAQ</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="grafiken.html">Grafiken</a>
      </li>
    </ul>
  </div>
</nav>
```

Quelle: Eigene Darstellung

Der zu der Sidebar-Navigation gehörende Button betitelt „LTR/RTL“ ermöglicht den Benutzern eine Interaktion mit der Webanwendung. Durch Klicken des Buttons können die Besucher die Sidebar-Navigation entweder rechts oder links auf dem Bildschirm positioniert werden, je nachdem ob in ihrer Kultur die Leserichtung von rechts nach links (RTL) oder links nach rechts (LTR) ist:

Abb. 7 Positionierungsbutton



```
<div class="sidebar-steuerung">
  <button class="toggle">LTR/RTL</button>
</div>
```

Quelle: Eigene Darstellung

Die Positionierung des Buttons ist dank der individuell eingerichteten CSS-Klasse *sidebar-steuerung* möglich, welche die Position des Buttons fixiert und die Navigation vertikal ausrichtet, da der Container des Buttons ein Flex-Container ist:

Abb. 8: Eigene CSS-Klasse



```
.sidebar-steuerung {
  display: flex;
  flex-direction: column;
  margin-top: 200px;
  position: fixed;
}
```

Quelle: Eigene Darstellung

Der Seitenwechsel der Sidebar-Navigation wird mittels einer JavaScript Event-Handler Funktion implementiert (Bewersdorff, 2018, S. 292), welche per Manipulation der DOM-Struktur (Bewersdorff, 2018, S. 313), auf den Container des Positionsbuttons und auf den Button selbst zugreift. Die Funktion kontrolliert anhand einer if-Abzweigung ob die Sidebar-Navigation und der dazugehörige Button die CSS-Klasse *right* enthalten, welche dafür sorgt, dass die Navigation und der Button auf der rechten Bildschirmhälfte platziert werden. Falls die beiden Components die Klasse nicht enthalten, dann bekommen sie durch einen Mausklick die Klasse zugewiesen, damit sie entsprechend rechts platziert werden. Falls sie die Klasse bereits enthalten, was nach einem initialen Mausklick der Fall ist, dann bekommen die beiden Components nach erneutem Mausklick die Klasse entnommen, damit sie entsprechend wieder links platziert werden:

Abb. 9: JavaScript Event-Handler Funktion



```
document.querySelector(".toggle").addEventListener("click", () => {
  const sidebar = document.querySelector(".sidebar-navigation");
  const toggler = document.querySelector(".toggle");

  if (sidebar.classList.contains("right")) {
    sidebar.classList.remove("right");
  } else {
    sidebar.classList.add("right");
  }

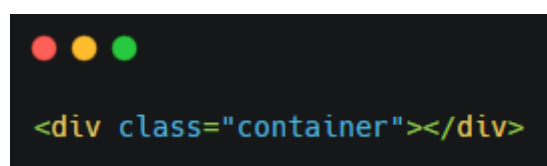
  if (toggler.classList.contains("right")) {
    toggler.classList.remove("right");
  } else {
    toggler.classList.add("right");
  }
});
```

Quelle: Eigene Darstellung

5. Die Anwendung responsiv machen

Der häufige Zugriff auf (Web)Anwendungen von unterschiedlichen Geräten aus wird bei der Webanwendung von Emission Omission berücksichtigt, die responsiv umgesetzt wurde, damit sie nach Breite und Höhe des Ansichtsfensters und Ausrichtung (Hoch- oder Querformat) optimal dargestellt wird (Bewersdorff, 2018, S. 57). Da die Anwendung von Emission Omission hauptsächlich mit CSS-Bootstrap entwickelt wurde, wurde somit ein großer Teil der Components bereits responsiv mitentwickelt. Dies kommt durch die Verwendung der Bootstrap Helfer-Klasse *container* bei jedem Container zustande, die die Container in zentrierten und responsiven Grid-Containern umwandelt, deren Breite sich an den Viewport anpasst (Krause, 2020, S. 74):

Abb. 10: Responsiver Grid-Container



```
<div class="container"></div>
```

Quelle: Eigene Darstellung

Zu den wenigen Components, die nicht komplett über Bootstrap responsiv gemacht wurden, zählen die Steuerungsbutton der Sidebar-Navigation und ihr Steuerungsbutton. Zur Sicherstellung der responsiven Darstellung der beiden Komponenten wurde eine Kombination aus Media Query und Bootstrap eingesetzt, bei der sie unter einer Bildschirmbreite von 1655px als ein Button auftauchen, durch dessen Klicken ein Dropdown-Menü ausgelöst werden kann. Dafür wurde dem Container die Utility-Klasse *dropdown* vergeben und als Inhalt der eigentliche Button, der das Dropdown-Menü aufklappt, und die Auswahloptionen angelegt. Ähnlich zu anderen Buttons auf der Anwendung besitzt auch dieser Button, u. a., die Klasse *btn-success*, die ihm eine dunkel-grüne Farbe verleiht (Matsinopoulos, 2020, S. 166). Die Links in der Sidebar-Navigation machen zusammen mit dem Button zum Positionswechsel den Inhalt des Dropdown-Menüs aus und sind deswegen als Listenelemente einer ungeordneten Liste angelegt. Jedes Listenelement enthält die Helfer-Klasse *dropdown-item*. Der Container des Dropdown-Buttons enthält zusätzlich die eingerichtete Klasse *sidebar-menu*, die ähnlich zu der Standard Sidebar-Navigation die Anfangsposition des Buttons festlegt. Der Steuerungsbutton wiederum besitzt analog zu dem Standard Steuerungsbutton die Klasse *toggle-sidebar*, welche die Toggle-Funktion über einen Event-Handler ermöglicht:

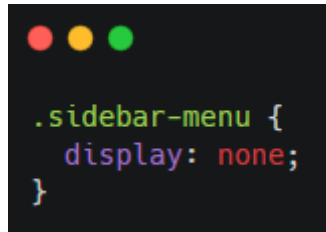
Abb. 11: Responsives Sidebar-Menü

```
<div class="dropdown sidebar-menu">
  <button
    type="button"
    class="btn btn-success dropdown-toggle mt-5"
    data-bs-toggle="dropdown">
    Menü
  </button>
  <ul class="dropdown-menu">
    <li>
      <a href="tabelle.html" class="dropdown-item text-center">Tabelle</a>
    </li>
    <li><a href="faq.html" class="dropdown-item text-center">FAQ</a></li>
    <li>
      <a href="grafiken.html" class="dropdown-item text-center"
        >Grafiken</a>
    >
    </li>
    <li class="dropdown-item sidebar-menu">
      <button class="toggle-sidebar">LTR/RTL</button>
    </li>
  </ul>
</div>
```

Quelle: Eigene Darstellung

Damit der Dropdown-Button nicht auftaucht, wenn die maximale Bildschirmbreite in der Bedingung seiner Media Query nicht erfüllt wird, wird er über CSS ausgeblendet:

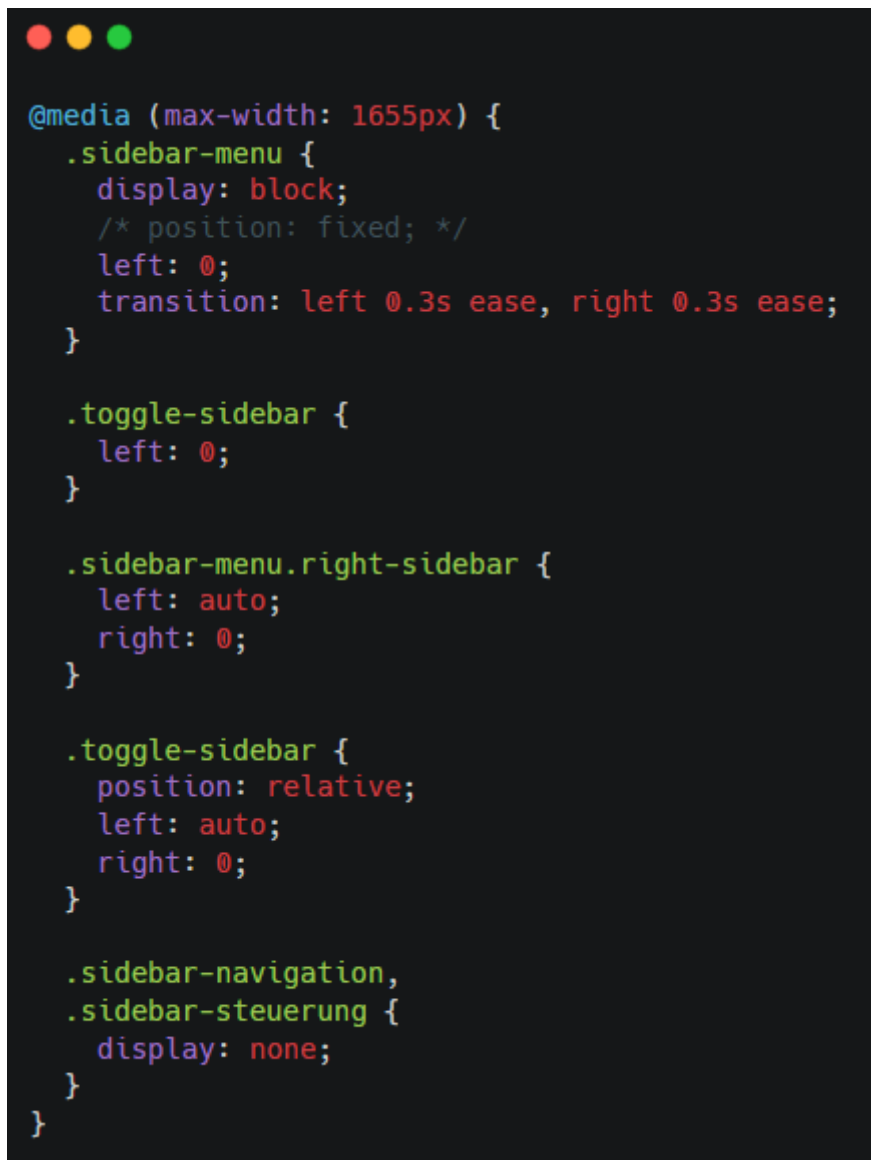
Abb. 12. Ausblenden des Dropdown-Buttons



```
.sidebar-menu {  
  display: none;  
}
```

Quelle: Eigene Darstellung

Abb. 13: Media Query zum Dropdown Navigations-Button



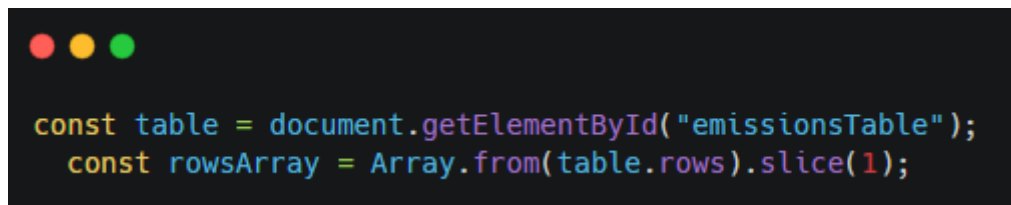
```
@media (max-width: 1655px) {  
  .sidebar-menu {  
    display: block;  
    /* position: fixed; */  
    left: 0;  
    transition: left 0.3s ease, right 0.3s ease;  
  }  
  
  .toggle-sidebar {  
    left: 0;  
  }  
  
  .sidebar-menu.right-sidebar {  
    left: auto;  
    right: 0;  
  }  
  
  .toggle-sidebar {  
    position: relative;  
    left: auto;  
    right: 0;  
  }  
  
  .sidebar-navigation,  
  .sidebar-steuerung {  
    display: none;  
  }  
}
```

Quelle: Eigene Darstellung

6. Interaktive Emissionstabelle

Die Emissionstabelle ist ein weiteres Element der Emission Omission Anwendung, bei der JavaScript Funktionen bzw. Event-Handler gewisse Interaktionen zwischen den Nutzern und der Anwendung ermöglichen. Die erste Funktion, `sortTable()`, sorgt dafür, dass die Daten in der Emissionstabelle per Mausklick sortiert werden können. Sie greift auf die Tabelle anhand ihres ID's „`emissionsTable`“ zu und erstellt ein Array mit den Tabellenzeilen:

Abb. 14: Zugriff mittels ID und Anlegen eines Zeilenarrays

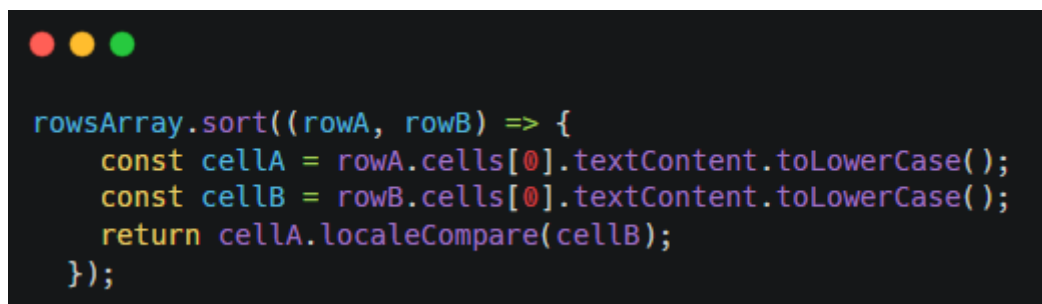


```
const table = document.getElementById("emissionsTable");
const rowsArray = Array.from(table.rows).slice(1);
```

Quelle: Eigene Darstellung auf der Basis von W3Schools (2025).

Im weiteren Ablauf werden die Zeilen anhand der `sort()`-Methode sortiert. Es wird auf den Text der ersten Zelle der Zeilen `rowA` und `rowB` zugegriffen und mittels der `toLowerCase()`-Methode sichergestellt, dass im Vergleich nicht zwischen Groß- und Kleinschreibung unterschieden wird (W3Schools, 2025). Die Strings in den Tabellenzellen werden mittels `localeCompare` verglichen, und zwar in Form einer lexikografischen (alphabetischen) Sortierung:

Abb. 15: Alphabetische Sortierung der Zelleninhalte

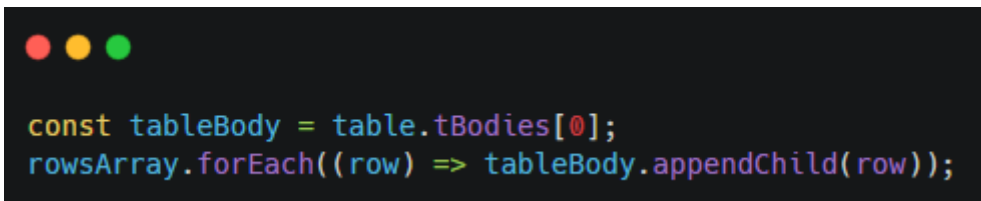


```
rowsArray.sort((rowA, rowB) => {
  const cellA = rowA.cells[0].textContent.toLowerCase();
  const cellB = rowB.cells[0].textContent.toLowerCase();
  return cellA.localeCompare(cellB);
});
```

Quelle: Eigene Darstellung auf der Basis von W3Schools (2025).

Im letzten Schritt der Funktion wird der erste (und einzige) *tbody*-Tag der Tabelle geholt und jede sortierte Zeile wird wieder in den Tabellenkörper (*tbody*) der Tabelle eingefügt, was dafür sorgt, dass die Zeilen in der neuen Reihenfolge angezeigt werden, da *appendChild* die Zeile an das Ende des *tbody* anhängt (W3Schools, 2025):

Abb. 16: Anhängen der Zellen an das Tabellenende

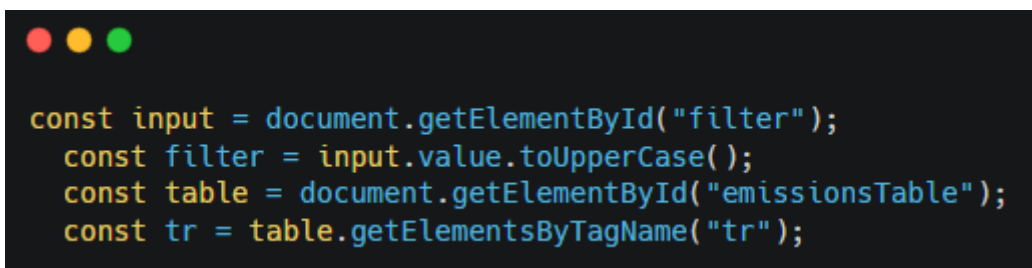


```
const tableBody = table.tBodies[0];
rowsArray.forEach((row) => tableBody.appendChild(row));
```

Quelle: Eigene Darstellung auf der Basis von W3Schools (2025).

filterFunction(), die zweite Funktion hinter der Emissionstabelle, ermöglicht, dass die Tabelle entweder nach Land oder Unternehmen sortiert werden kann. Wie zu Anfang des Kapitels erwähnt, werden hier Event-Handler eingesetzt und mit deren Einsatz wird auf das Eingabefeld oberhalb der Tabelle, die Tabelle selbst, den Filterwert aus dem Eingabefeld und die Tabellenzeilen zugegriffen (W3Schools, 2025):

Abb. 17: Zugriff auf die Tabelle und ihre Zeilen



```
const input = document.getElementById("filter");
const filter = input.value.toUpperCase();
const table = document.getElementById("emissionsTable");
const tr = table.getElementsByTagName("tr");
```

Quelle: Eigene Darstellung auf der Basis von W3Schools (2025).

Der Kern der Funktion besteht aus den zwei *for-Schleifen*, die ineinander verschachtelt sind. Die äußere Schleife iteriert durch jede Tabellenzeile, greift auf die Inhalte der Zeilen und weist der Variable *isMatch* einen initialen Wert von „false“ zu. Die innere *for-Schleife* iteriert durch die Tabelleninhalte, prüft mittels einer *if-Abzweigung* ob der textuelle Inhalt der Zeilen

der gleiche ist wie dieser, der in das Eingabefeld mit der ID „filter“ eingetragen wurde und weist bei einer Übereinstimmung der Variable *isMatch* einen neuen Wert von „true“ zu (W3Schools, 2025). Ein ternärer Operator (Bewersdorff, 2018, S. 101) außerhalb der inneren for-Schleife aber innerhalb der äußeren prüft, ob *isMatch* wahr ist und wenn ja, bleibt die Zeile sichtbar, andernfalls wird sie ausgeblendet:

Abb. 18: Überprüfung der Zeileninhalte und Ein- bzw. Ausblenden der Zeilen

A screenshot of a code editor with a dark background and light-colored text. The code is a JavaScript function that iterates through the rows of a table. For each row, it gets the table data cells (tds) and checks if any cell's text content (converted to uppercase) includes a specific filter value. If it does, the row's display style is set to 'block' (making it visible); otherwise, it is set to 'none' (hiding it).

```
for (let i = 0; i < tr.length; i++) {  
  const tds = tr[i].getElementsByTagName("td");  
  let isMatch = false;  
  
  for (let j = 0; j < tds.length; j++) {  
    const td = tds[j];  
    if (td && td.textContent.toUpperCase().includes(filter)) {  
      isMatch = true;  
      break;  
    }  
  }  
  
  tr[i].style.display = isMatch ? "" : "none";  
}
```

Quelle: Eigene Darstellung auf der Basis von W3Schools (2025).

7. Absicherung gegen Cross-Site-Scripting (XSS)

Bei der Entwicklung einer Webanwendung ist es wichtig, entsprechende Sicherheitsmechanismen anzuwenden, um gängige digitale Angriffe zu verhindern. Cross-Site-Scripting (XSS) stellt eine Art DOM-basierte Schwachstelle, die es ermöglicht, in Eingabefeldern einer Webanwendung böartigen z. B. JavaScript- oder HTML-Code zu injizieren, um die Anwendung unerlaubt zu manipulieren (Grossman, S. 150). Im Fall der Anwendung von Emission Omission wird z. B. das Kontaktformular auf der Kontaktseite mittels einer Event-Handler Funktion abgesichert, die auf das Kontaktformular zugreift und bei einer Eingabe in einem Feld, das sog. *input*-Event, die Ersetzungsfunktion ausführt. Wenn Text in ein einfaches Eingabe- oder mehrzeiliges Eingabefeld (*Textarea*) (Bühler, Schlaich & Sinner, 2023, S. 29) eingegeben wird, prüft dann ein regulärer Ausdruck (Bewersdorff, 2018, S. 86) in der Funktion, ob die Eingabe ein Sonderzeichen wie öffnende oder schließende Tags oder Klammer enthält.

Wenn ja, werden diese mit einer vordefinierten Zeichenkette ersetzt (Taha & Karabatak, 2018, S. 3):

Abb. 19: Funktion zur Input-Ersetzung

```
const formular = document.getElementsByClassName("kontaktformular");
for (let element of formular) {
  element.addEventListener("input", (event) => {
    if (
      event.target.tagName === "INPUT" ||
      event.target.tagName === "TEXTAREA"
    ) {
      const formInput = event.target.value;
      const sanitizedFormInput = formInput.replace(
        /[\<>()#]/g,
        function (match) {
          switch (match) {
            case "<":
              return "&lt;";
            case ">":
              return "&gt;";
            case "(":
              return "&#40;";
            case ")":
              return "&#41;";
            case "#":
              return "&#35;";
            default:
              return match;
          }
        }
      );
      event.target.value = sanitizedFormInput;
    }
  });
}
```

Quelle: Eigene Darstellung auf der Basis von Taha & Karabatak (2018).

Im letzten Schritt der Funktion wird der sanitisierte, also der bereinigte Wert (mit den ersetzten Zeichen) wieder dem Eingabefeld zugewiesen. Eine ähnliche Funktion wurde für das Eingabefeld oberhalb der Emissionstabelle geschrieben, über welches die Tabelle sortiert

werden kann. Auf diese Funktion wird allerdings nicht näher eingegangen, weil sie von der Umsetzung her

8. Schluss

Der Einsatz der Webentwicklungstechnologien HTML, CSS, JavaScript und das Framework Bootstrap hat es Emission Omission ermöglicht, eine praktische, übersichtliche und benutzerfreundliche Webanwendung zu entwickeln, die die Besuchenden über die CO₂-Emissionen einiger Länder und deren Unternehmen informiert. Die Umsetzung hat sich an einem Faden gehalten, bei dem die Komponenten der Anwendung zuerst als HTML-Elemente angelegt wurden, die dann später mittels CSS und Bootstrap Einstellungen wie Schriftgröße, Innen- und Außenabstände, Positionierung und Farbe bekommen haben. Zudem wurden die Komponenten responsiv gemacht, sodass sie unabhängig von Bildschirm-auflösung immer noch gut sicht- und lesbar sind. Mithilfe von JavaScript wurden die Sidebar-Navigation und die Emissionstabelle mit Funktionen ausgestattet, die einen Positionswechsel der Sidebar und entsprechend eine Bearbeitung der Tabellendaten ermöglichen. Damit die Anwendung auch vor Malware, die über die Eingabefelder injiziert werden kann, geschützt wird, wurden die Eingabefelder mit entsprechender Funktion zum Ersetzen von einigen gängigen Codezeichen abgesichert. Weiterentwicklungen an der Anwendung können vor allem die Anbindung eines Backends abzielen, damit Daten, die während der Benutzung generiert werden, auch gespeichert werden können. Weitere Module bzw. Komponenten, die Nützlichkeit der Emission Omission Webanwendung erhöhen können, sind z. B. ein Chat-Modul, Kommentare oder auch die Möglichkeit, Dokumente wie eine CSV- oder PDF-Datei mit den Emissionsdaten aus der Tabelle zu generieren.

IV. Literaturverzeichnis

- Bewersdorff, J. (2018): *Objektorientierte Programmierung mit JavaScript. Direktstart für Einsteiger*. (2. Auflage). Springer Verlag.
- Bühler, P., Schlaich, P. & Sinner, D. (2023): *HTML und CSS. Semantik. Design. Responsive Layouts*. (2. Auflage). Springer Verlag.
- Donvir, A., Saraswathi, P. K. & Jain, A.. (2024): *End-to-End Application and Backend State Management: A Comprehensive Review*. In: *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*.
- Frain, B. (2020): *Responsive Web Design with HTML5 and CSS: Develop Future-Proof Responsive Websites Using the Latest HTML5 and CSS Techniques*. Packt Publishing, Limited.
- Glaser, C. (2022): Carbon Accounting: *Erfassung von CO₂-Emissionen bei Objektfinanzierungen*. In M. Knapp (Hrsg.), *Zeitschrift für das gesamte Kreditwesen: Pflichtblatt der Frankfurter Wertpapierbörse*. (71. Auflage, S. 30 -35). Koesel Verlag.
- Grossman et al. (2017). *XSS Attacks: Cross Site Scripting Exploits and Defense*. Syngress.
- Krause, J. (2020): *Introducing Bootstrap 4. Create Powerful Web Applications Using Bootstrap 4.5*. (2. Auflage). Apress.
- Matsinopoulos, P. (2020): *Practical Bootstrap. Learn to Develop Responsively with One of the Most Popular CSS Frameworks*. Apress.
- Taha, T. W., Karabatak, M. (2018): *A proposed approach for preventing Cross-Site Scripting*. IEEE.
- W3Schools. (2025). *How To Sort a Table. Learn how to sort an HTML table using JavaScript*. https://www.w3schools.com/howto/tryit.asp?filename=tryhow_js_sort_table
- W3Schools. (2025): *How To Filter/Search Table. Learn how to create a filter table with JavaScript*. https://www.w3schools.com/howto/howto_js_filter_table.asp