



Fallstudie

Internationale Hochschule

Studiengang: Softwareentwickler

IPWA01-01 – Programmierung von Webanwendungsoberflächen

Dr. Petra Lakatos

Geb.: 20.04.1981

Matrikelnummer: IU14090917

Projekt:

Aufgabenstellung 1: CO2-Footprint

Tutor: Alexander Christopher Bock

Abgabedatum: 05.10.2025

Inhalt

1. Zusammenfassung	4
2. HTML – Aufbau & Semantik	4
3. CSS – Layout, Design & Responsivität.....	7
3.1. <i>Insgesamt gibt es in der CSS-Datei drei verschiedene Flex-Container:.....</i>	7
3.2. <i>Responsibilität.....</i>	7
3.3. <i>Benutzerfreundlichkeit:</i>	8
3.4. <i>Fokuszustände: sichtbare Umrandung bei Tastaturnavigation.</i>	9
4. JavaScript – Funktionalität.....	10
4.1. <i>Suche in der Tabelle</i>	10
4.2. <i>Klickbare Tabellenköpfe</i>	11
4.3. <i>Panel ein-/ausblenden</i>	12
4.4. <i>Robustheit: Ereignislistener prüfen auf vorhandene DOM-Elemente.</i>	13
4.5. <i>Sicherheit.....</i>	13
5. Daten & Annahmen	13
6. Testing & Qualitätssicherung.....	13
7. Deployment / Repository	13


Abbildungen

Abb. 1 Überblick der Webseite (Teil)	4
Abb. 2 Html – Code von header	4
Abb. 3 Html – Code von aside	5
Abb. 4 HTML-Codeteil der Tabelle (die Spalten und der erste Datensatz).	5
Abb. 5 HTML-Code von Footer.	5
Abb. 6 Html- Code von Impressum (A) und Datenschutz (B).....	6
Abb. 7 Verknüpfung zur CSS-Datei.....	6
Abb. 8 Verknüpfung zur JavaScript-Datei.	7
Abb. 9 Desktop-Ansicht: Die Sidebar erscheint auf der linken Seite, daneben die Tabelle.....	7
Abb. 10 Mobile Ansicht: Durch die Media Query wird die Sidebar nach oben verschoben, darunter erscheint die Tabelle.....	8
Abb. 11 In der Sidebar ist ein Menüpunkt erst inaktiv und dann aktiv und hervorgehoben dargestellt.	8
Abb. 12 CSS-Code für die inaktiven/aktiven Knöpfe.....	8
Abb. 13 Bei der Tastaturnavigation ist das Eingabefeld mit der hellblauen Umrandung (Outline) markiert.	9
Abb. 14 Codeteil des Suchfunktion.....	10
Abb. 15 Codeteil zur Sortierung mit Pfeilfen.	11
Abb. 16 In der Tabelle können die sortierbaren Spaltenköpfe mit Pfeilen aufsteigende oder absteigende Werte anzeigen.	11
Abb. 17 Codierung zum Ein- und Ausblenden von Panels.	12
Abb. 18 Filterpanel Aus/Eingeblendet.	12
Abb. 19 Robustheit in dem CSS-Datei.....	13
Abb. 20 Cleaning von Eingabedaten.	13

1. Zusammenfassung

Die Website „CO₂-Footprint“ wurde im Rahmen des Moduls IPWA01-01 von Petra Lakatos erstellt.

Ziel ist es, fiktive CO₂-Emissionsdaten übersichtlich und responsiv Webseite darzustellen. Nutzer:innen können die Tabelle sortieren und über mehrere Suchfelder (Unternehmen, Anlage, Land, tCO₂e) filtern.



The screenshot shows the 'CO₂-Footprint' website. It has a dark teal header with a star icon and the title. A search bar is in the top right. On the left, there's a sidebar with 'Filter & Suche' and 'Tabelle'. The main content area displays a table with four columns: 'Unternehmen', 'Anlage', 'Land', and 'tCO₂e_2024'. The table contains three rows of data.

Unternehmen	Anlage	Land	tCO ₂ e_2024
EnergieUnternehmen 075 GmbH	Kraftwerk 075	DE	22696080
EnergieUnternehmen 001 GmbH	Kraftwerk 001	DE	22628711
EnergieUnternehmen 055 GmbH	Kraftwerk 055	DE	20529012

Abb. 1 Überblick der Webseite (Teil)

Hinweis: Die Daten sind zufällig generiert und nicht real.

2. HTML – Aufbau & Semantik

Struktur:

<body>

l--<header> (Logo + Seitentitel + Suchfeld für Unternehmen)

```
<header>
  <a href="/" class="logo-wrap">
    
  </a>
  <h1 class="site-title">CO2-Footprint</h1>
  <form class="header-search" onsubmit="return false;">
    <input type="text" id="header-company" placeholder="Unternehmen suchen">
  </form>
</header>
```

Abb. 2 Html – Code von header.

l--<main>

l--<aside> (Das lokale Menü auf der linken Seite ist immer sichtbar. Es enthält Suchfelder, eine Tabelle und ein Panel für detaillierte Suchen, das zu Beginn versteckt ist.)

```

<aside>
  <!--DAS IST IMMER SICHTBAR-->
  <nav aria-label="Lokales Menü">
    <ul>
      <li><a id="link-filter" href="#toggle-filter" aria-controls="filterPanel" aria-expanded="false">Filter & Suche</a></li>
      <li><a id="link-table" href="#toggle-table" aria-controls="table" aria-expanded="false">Tabelle</a></li>
    </ul>
  </nav>

  <!--DIES IST EIN ROLLENDE PANEL: ZU BEGINN VERSTECKT-->
  <div id="filterPanel" class="filter-panel" hidden>
    <form onsubmit="return false;">
      <input type="text" name="company" placeholder="Unternehmen suchen">
      <input type="text" name="installation" placeholder="Anlage suchen">
      <input type="text" name="country" placeholder="Land suchen">
      <input type="text" name="emissions" placeholder="tCO2e suchen">
    </form>
  </div>
</aside>

```

Abb. 3 Html – Code von aside.

I--<section> (Das Hauptkontent enthält die Tabelle, die aus vier verschiedenen Spalten besteht: „Unternehmen“, „Anlage“, „Land“ und „CO₂-Emissionsdaten“. Außerdem enthält sie 100 zufällig generierte fiktive Datensätze.)

```

<!--DIE TABELLE-->
<section id="table">
  <table>
    <thead>
      <tr>
        <th scope="col">Unternehmen</th>
        <th scope="col">Anlage</th>
        <th scope="col">Land</th>
        <th scope="col">tCO2e_2024</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>EnergieUnternehmen 075 GmbH</td>
        <td>Kraftwerk 075</td>
        <td>DE</td>
        <td>22696080</td>
      </tr>
    </tbody>
  </table>

```

Abb. 4 HTML-Codeteil der Tabelle (die Spalten und der erste Datensatz).

I--<footer> (Der Footer enthält Links zu den HTML-Seiten mit dem Impressum und den Datenschutzbestimmungen.)

```

<!--FOOTER MIT DEN LINKS-->
<footer>
  <nav>
    <ul>
      <a href="./Datenschutz.html">Datenschutz</a>
      <a href="./Impressum.html">Impressum</a>
    </ul>
  </nav>
</footer>

```

Abb. 5 HTML-Code von Footer.

```

<!doctype html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Impressum</title>
  <link rel="stylesheet" href="main.css">
</head>
<body>
<header>
  <h1>Impressum</h1>
</header>
<main>
  <p><strong>Betreiber der Website:</strong></p>
  <p>Beispiel GmbH<br>
  Musterstraße 1<br>
  91000 Erlangen<br>
  Deutschland</p>

  <p><strong>Kontakt:</strong><br>
  Telefon: +49 (0)123 456789<br>
  E-Mail: info@beispiel.de</p>

  <p><strong>Geschäftsführen:</strong> Max Mustermann</p>
  <p><strong>USt-IdNr.:</strong> DE123456789</p>
</main>
</body>
</html>

```

A

```

<!doctype html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Datenschutzerklärung</title>
  <link rel="stylesheet" href="main.css">
</head>
<body>
<header>
  <h1>Datenschutzerklärung</h1>
</header>
<main>
  <h2>1. Verantwortlicher</h2>
  <p>Beispiel GmbH<br>
  Musterstraße 1, 91000 Erlangen<br>
  E-Mail: datenschutz@beispiel.de</p>

  <h2>3. Zweck der Datenverarbeitung</h2>
  <p>Die Website wurde im Rahmen des Moduls IPWA01-01 von Petra Lakatos erstellt. Die dargestellten Daten sind zufällig generiert und nicht real.</p>
</main>
</body>
</html>

```

B

Abb. 6 Html- Code von Impressum (A) und Datenschutz (B).

Zusätzliche Codeteile:

1. Stylesheet-Verknüpfung

```

<head>
  <meta charset="UTF-8">
  <title>C02-Footprint</title>
  <link rel="stylesheet" href="main.css">
</head>

```

Abb. 7 Verknüpfung zur CSS-Datei.

2. Der Link auf die JavaScript-Datei, welche die im Projekt geschriebenen Funktionen (z. B. Suche, Tabellensortierung, Toggle) lädt.

```

</footer>
<script src="js/app.js"></script>
</body>
</html>

```

Abb. 8 Verknüpfung zur JavaScript-Datei.

3. CSS – Layout, Design & Responsivität

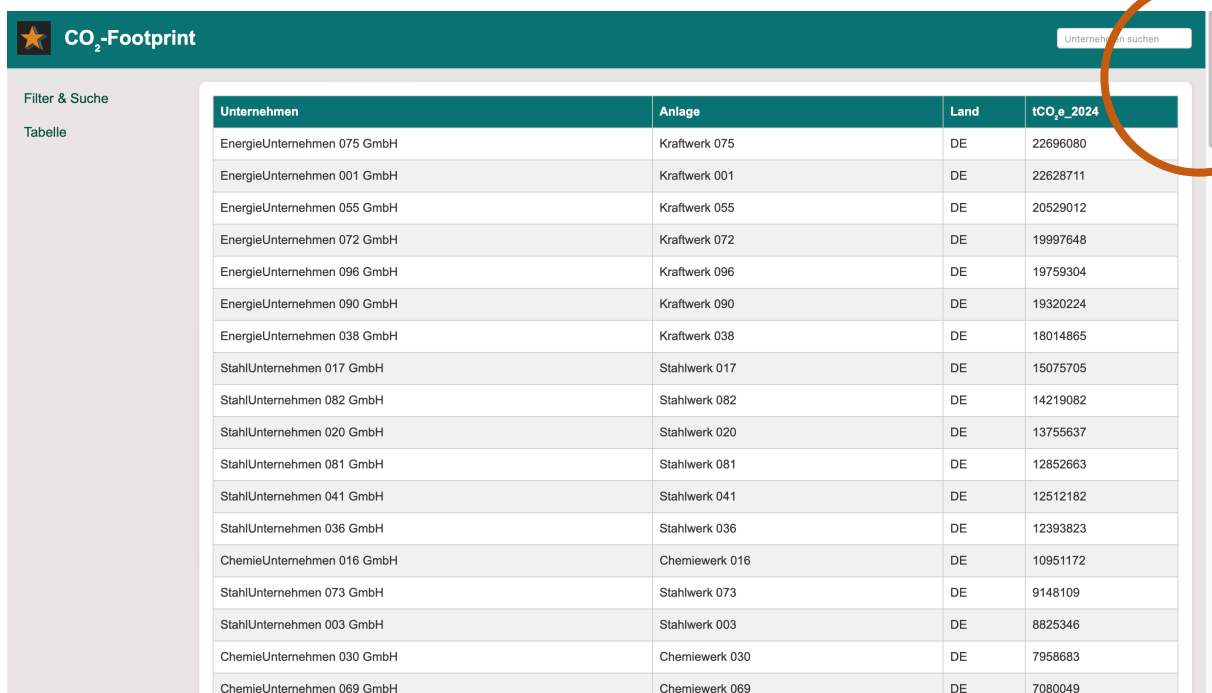
Neben den allgemeinen Formatierungselementen enthält die CSS-Datei die folgenden wichtigen Elemente:

3.1. Insgesamt gibt es in der CSS-Datei drei verschiedene Flex-Container:

- der Header (header) → header { display: flex; }
- die Navigationsliste im Header (header nav ul) → header nav ul { display: flex; }
- das Hauptcontent-Layout (main) → main { display: flex; } – (aside) + (section)

3.2. Responsivität

Die Responsivität der Webseite wird über eine Media Query bei **@media (max-width: 768px)** gesteuert. Der Hauptbereich (main) wechselt von einer horizontalen zu einer vertikalen Anordnung. Die Sidebar (aside) erhält die volle Breite und wird über order: -1 oberhalb des Inhalts angezeigt. Fokuseffekte für Links, Buttons und Eingabefelder sind speziell für die Tastaturnavigation gestaltet. Der Header und die Sidebar sind mit position: sticky fixiert und bleiben beim Scrollen sichtbar. Zusätzlich sorgt die Gestaltung der .filter-panel-Elemente für eine gute Bedienbarkeit auf mobilen Geräten.



Unternehmen	Anlage	Land	tCO ₂ e_2024
EnergieUnternehmen 075 GmbH	Kraftwerk 075	DE	22696080
EnergieUnternehmen 001 GmbH	Kraftwerk 001	DE	22628711
EnergieUnternehmen 055 GmbH	Kraftwerk 055	DE	20529012
EnergieUnternehmen 072 GmbH	Kraftwerk 072	DE	19997648
EnergieUnternehmen 096 GmbH	Kraftwerk 096	DE	19759304
EnergieUnternehmen 090 GmbH	Kraftwerk 090	DE	19320224
EnergieUnternehmen 038 GmbH	Kraftwerk 038	DE	18014865
StahlUnternehmen 017 GmbH	Stahlwerk 017	DE	15075705
StahlUnternehmen 082 GmbH	Stahlwerk 082	DE	14219082
StahlUnternehmen 020 GmbH	Stahlwerk 020	DE	13755637
StahlUnternehmen 081 GmbH	Stahlwerk 081	DE	12852663
StahlUnternehmen 041 GmbH	Stahlwerk 041	DE	12512182
StahlUnternehmen 036 GmbH	Stahlwerk 036	DE	12393823
ChemieUnternehmen 016 GmbH	Chemiewerk 016	DE	10951172
StahlUnternehmen 073 GmbH	Stahlwerk 073	DE	9148109
StahlUnternehmen 003 GmbH	Stahlwerk 003	DE	8825346
ChemieUnternehmen 030 GmbH	Chemiewerk 030	DE	7958683
ChemieUnternehmen 069 GmbH	Chemiewerk 069	DE	7080049

Abb. 9 Desktop-Ansicht: Die Sidebar erscheint auf der linken Seite, daneben die Tabelle.

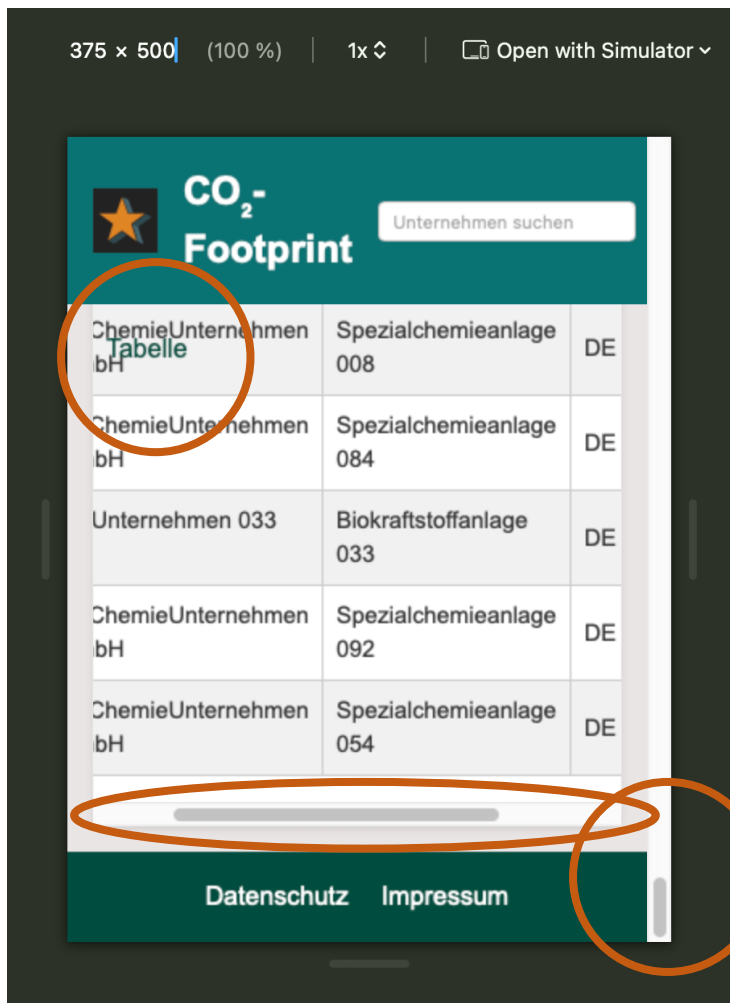


Abb. 10 Mobile Ansicht: Durch die Media Query wird die Sidebar nach oben verschoben, darunter erscheint die Tabelle.

3.3. Benutzerfreundlichkeit:

- Hervorhebung aktiver Menüpunkte: Aktive Menüpunkte in der Sidebar (z. B. Tabelle oder Filter & Suche) werden durch Unterstreichung bzw. Fettschrift hervorgehoben.

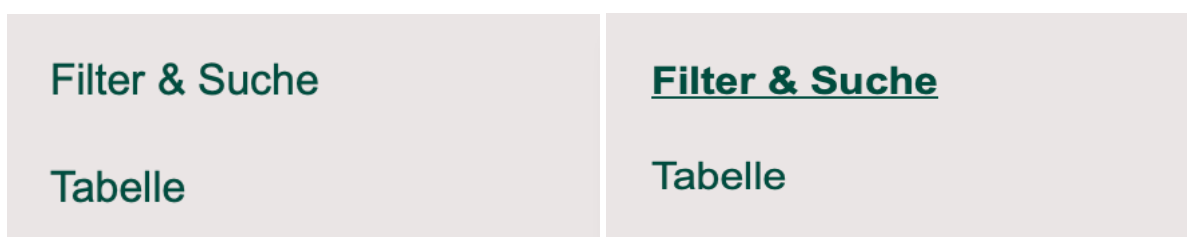


Abb. 11 In der Sidebar ist ein Menüpunkt erst inaktiv und dann aktiv und hervorgehoben dargestellt.

```

aside a.is-active {
  font-weight: 600;
  text-decoration: underline;
}

```

Abb. 12 CSS-Code für die inaktiven/aktiven Knöpfe.

Vorteilen:

Usability: Zeigt jederzeit, wo man sich in der Anwendung befindet („Where am I?“-Problem).
Reduziert die kognitive Belastung und verbessert die Navigationsgeschwindigkeit.

Barrierefreiheit: Kontrastreiche, eindeutige Zustände (z. B. Unterstreichungen/Fettschrift) unterstützen Menschen mit Aufmerksamkeits- oder Wahrnehmungseinschränkungen; der aktive Zustand kann auch per Tastaturfokus erkennbar sein.

3.4. Fokuszustände: sichtbare Umrandung bei Tastaturnavigation.

Bei der Tastaturnavigation (z. B. mit der Tabulatortaste) erhalten anklickbare Elemente wie Links, Buttons oder Eingabefelder eine sichtbare Umrandung.



Abb. 13 Bei der Tastaturnavigation ist das Eingabefeld mit der hellblauen Umrandung (Outline) markiert.

Vorteilen:

Usability: Alle interaktiven Elemente sind auch ohne Maus erreichbar.

Barrierefreiheit: Sichtbare Fokusrahmen (outline) stellen die Orientierung im Interface für Screenreader- und Tastaturnutzer:innen sicher.

4. JavaScript – Funktionalität

4.1. Suche in der Tabelle

Die JavaScript-Datei ermöglicht eine mehrfache Suche innerhalb der Tabelle.

Die Benutzer:innen können in verschiedenen Feldern suchen: nach Unternehmen, Anlage, Land oder den Emissionswerten. Zusätzlich gibt es im Header ein Schnellsuchfeld, das gezielt nur nach Unternehmensnamen filtert.

Die Suche funktioniert in Echtzeit: Sobald ein:e Benutzer:in tippt, überprüft das Skript jede Tabellenzeile und blendet diejenigen Zeilen aus, die nicht den Suchkriterien entsprechen. Auf diese Weise bleiben nur die relevanten Einträge sichtbar.

Bevor die Eingaben mit den Tabellendaten verglichen werden, bereinigt das Skript sie von potenziell problematischen Sonderzeichen (z. B. < > " '). Dadurch wird verhindert, dass unerwünschter oder schädlicher Code ausgeführt wird (Schutz vor XSS-Angriffen).

```
// ---- SUCHE (4 FELDER, KOMBINIERTES FILTERUNG) ----
const companyInput = document.querySelector( selectors: '#filterPanel input[name="company"]');
const installationInput = document.querySelector( selectors: '#filterPanel input[name="installation"]');
const countryInput = document.querySelector( selectors: '#filterPanel input[name="country"]');
const emissionsInput = document.querySelector( selectors: '#filterPanel input[name="emissions"]');

const table = document.querySelector( selectors: "table");
const tbody = table?.querySelector( selectors: "tbody");
const headers = table?.querySelectorAll( selectors: "th");

const clean = s => (s || "").replace( searchValue: /[<>"']/g, replaceValue: "");

function filterRows() {
  if (!tbody) return;
  const qCompany = clean(companyInput?.value).trim().toLowerCase();
  const qInst = clean(installationInput?.value).trim().toLowerCase();
  const qCountry = clean(countryInput?.value).trim().toLowerCase();
  const qEmiss = clean(emissionsInput?.value).trim().toLowerCase();

  const qHeaderCompany = clean(headerCompanyInput?.value).trim().toLowerCase();

  tbody.querySelectorAll("tr").forEach( callbackfn: row => {
    const c = row.children;
    const company = c[0].textContent.toLowerCase();
    const inst = c[1].textContent.toLowerCase();
    const country = c[2].textContent.toLowerCase();
    const emissions = c[3].textContent.toLowerCase();

    const ok =
      (!qCompany || company.includes(qCompany)) &&
      (!qInst || inst.includes(qInst)) &&
      (!qCountry || country.includes(qCountry)) &&
      (!qEmiss || emissions.includes(qEmiss)) &&
      (!qHeaderCompany || company.includes(qHeaderCompany));

    row.style.display = ok ? "" : "none";
  });
}
```

Abb. 14 Codeteil des Suchfunktion.

4.2. Klickbare Tabellenköpfe

Die Spaltenüberschriften sind anklickbar. Dadurch kann die Tabelle nach den jeweiligen Werten auf- oder absteigend sortiert werden (sichtbar durch ein Pfeilsymbol ▲/▼ im Tabellenkopf).

```
// ---- SORTIEREN MIT PFEILEN ----
let sortColumnIndex = null;
let sortAsc = true;

function attachSorting() {
  if (!headers || !tbody) return;
  headers.forEach( (header, index : number) => {
    header.style.cursor = "pointer";
    header.addEventListener("click", () => {
      const rows = Array.from(tbody.querySelectorAll("tr"))
        .filter(r => r.style.display !== "none");

      if (sortColumnIndex === index) {
        sortAsc = !sortAsc;
      } else {
        sortAsc = true;
        sortColumnIndex = index;
      }

      rows.sort( compareFn: (a : HTMLTableRowElement, b : HTMLTableRowElement) => {
        const aText = a.children[index].textContent.trim();
        const bText = b.children[index].textContent.trim();
        const aVal = isNaN(aText) ? aText.toLowerCase() : parseFloat(aText);
        const bVal = isNaN(bText) ? bText.toLowerCase() : parseFloat(bText);
        if (aVal < bVal) return sortAsc ? -1 : 1;
        if (aVal > bVal) return sortAsc ? 1 : -1;
        return 0;
      });

      rows.forEach(r => tbody.appendChild(r));
    });
  });
  //---- UPDATE VON PFEILEN ----
  headers.forEach( (header) => {
    header.textContent = header.textContent.replace( searchValue: / ▲| ▼/g, replaceValue: "");
    header.textContent += sortAsc ? " ▲" : " ▼";
  });
}
```

Abb. 15 Codeteil zur Sortierung mit Pfeilen.

tCO ₂ e_2024 ▲	tCO ₂ e_2024 ▼
296729	22696080
307755	22628711
358666	20529012
438029	19997648

Abb. 16 In der Tabelle können die sortierbaren Spaltenköpfe mit Pfeilen aufsteigende oder absteigende Werte anzeigen.

Vorteilen:

Usability: Nutzer:innen finden schneller relevante Informationen (z. B. höchste Emissionen oben).

Barrierefreiheit: Die Interaktion ist konsistent und vorhersagbar; die aktuelle Sortierrichtung wird visuell (▲/▼) kenntlich gemacht und kann zusätzlich per Tastatur erreicht werden.

4.3. Panel ein-/ausblenden

Im Sidebar befindet sich unter dem Menüpunkt „Filter & Suche“ ein detailliertes Such-Panel. Dieses Panel ist standardmäßig ausgeblendet, kann jedoch per Klick geöffnet oder wieder geschlossen werden.

Das JavaScript überwacht hier die Klicks und ändert den Zustand des Panels entsprechend (offen/geschlossen).

Außerdem werden die **ARIA-Attribute** (aria-expanded) aktualisiert, sodass auch Screenreader-Nutzer:innen klar erkennen können, ob das Panel geöffnet ist.

```
function toggle(panelEl, linkEl) {
  const willShow = panelEl.hidden;
  panelEl.hidden = !willShow;
  linkEl.setAttribute( 'qualifiedName: "aria-expanded"', String(willShow));
  linkEl.classList.toggle( token: "is-active", willShow);
}

linkFilter?.addEventListener( type: "click", listener: (e : MouseEvent) => {
  e.preventDefault();
  toggle(filterPanel, linkFilter);
});
```

Abb. 17 Codierung zum Ein- und Ausblenden von Panels.



Abb. 18 Filterpanel Aus/Eingeblendet.

4.4. Robustheit: Ereignislistener prüfen auf vorhandene DOM-Elemente.

Durch die Verwendung von optional chaining (`table?.querySelector("tbody")`) und Sicherheitsabfragen (`if (!headers || !tbody) return;`) wird sichergestellt, dass Funktionen nur ausgeführt werden, wenn die benötigten DOM-Elemente tatsächlich vorhanden sind. Dies erhöht die Robustheit des Skripts und verhindert Laufzeitfehler.

```
const table = document.querySelector(selectors: "table");
const tbody = table?.querySelector(selectors: "tbody");
const headers = table?.querySelectorAll(selectors: "th");

function attachSorting() {
  if (!headers || !tbody) return;
```

Abb. 19 Robustheit in dem CSS-Datei.

4.5. Sicherheit

Es werden keine externen Eingaben verarbeitet. Alle Suchfilter arbeiten ausschließlich mit Textvergleichen. Zusätzlich werden potenziell gefährliche Zeichen (<, >, ", ') aus den Eingaben entfernt, um Code-Injektionen zu verhindern.

```
const clean = s => (s || "").replace(searchValue: /[<>"']/g, replaceValue: "");
```

Abb. 20 Cleaning von Eingabedaten.

5. Daten & Annahmen

Die in der Tabelle dargestellten Daten sind keine realen Werte, sondern fiktive Daten, die mit einem Python-Skript zufällig generiert wurden.

Dazu wurden zunächst verschiedene Industriesektoren (z. B. Energie, Stahl, Chemie) sowie realistische Emissionsbereiche (tCO_2e) definiert.

Das Programm erzeugte 100 Unternehmen und Anlagen und wies jedem einen zufälligen Emissionswert zu.

Die so erstellten Daten wurden mit der Bibliothek pandas in eine Excel-Datei exportiert und anschließend in die HTML-Tabelle übernommen.

Einschränkung: Die Suche basiert auf einfacher Teilstring-Erkennung. Numerische Filter sind nur Basisvergleiche (keine Operatoren wie > 1000000).

6. Testing & Qualitätssicherung

Responsives Design: getestet mit Safari („Responsives Design-Modus“) und Chrome/Firefox DevTools.

Funktionstests: Sortierung in allen vier Spalten, Filter mit Einzel- und Mehrfeldeingaben, Test mit leeren Ergebnissen.

Zugänglichkeit: Fokussteuerung per Tastatur, Nutzung von aria-Attributen.

7. Deployment / Repository

Repository: https://github.com/AStarkPoetal/IPWA01-01_2025_PL

Ausführung: Statische Seite; index.html im Browser öffnen.