

# Projektinformationen

---

Projekt für das Modul Accessibility

Erstelllt von:

- Dean Braband

## Projekt starten

Das Projekt verwendet Node.js und einen Express.js-Server für das Backend.

Das Backend is in javascript geschrieben, und verwendet eine SQLite-Datenbank.

Um das Projekt zu builden und den Server zu starten muss folgender Befehl in die Kommandozeile eingegeben werden: `npm run test`

Falls npm nicht gefunden wurde müssen die Binärdateien von Node.js der Umgebung bekannt gemacht werden. Dies könnte Beispielsweise wie folgt aussehen: `export PATH="$HOME/Desktop/Accessibility.git/node-v24.11.0-linux-x64/bin:$PATH"`

## Genutzte Tools

### Linting

- eslint
- lesshint

### Bundling

- esbuild
  - `esbuild ./webapp/src/js/Main.mjs --log-level=warning --bundle --outfile=./webapp/dist/bundle.js`

### Minify und Obfuscation

- lessc
- terser

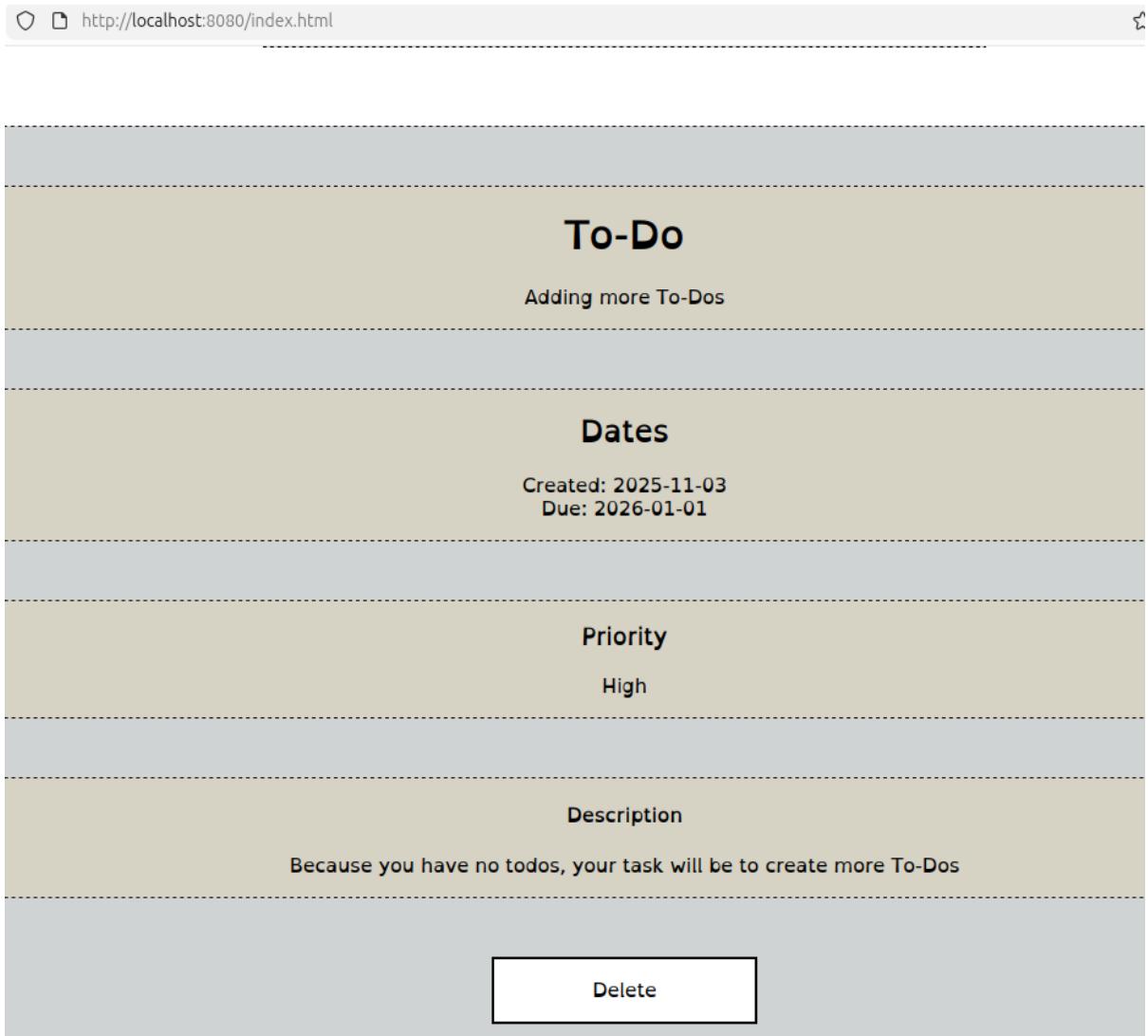
## Gewählte umgesetzte WCAG\_2.2-Kriterien für das Projekt

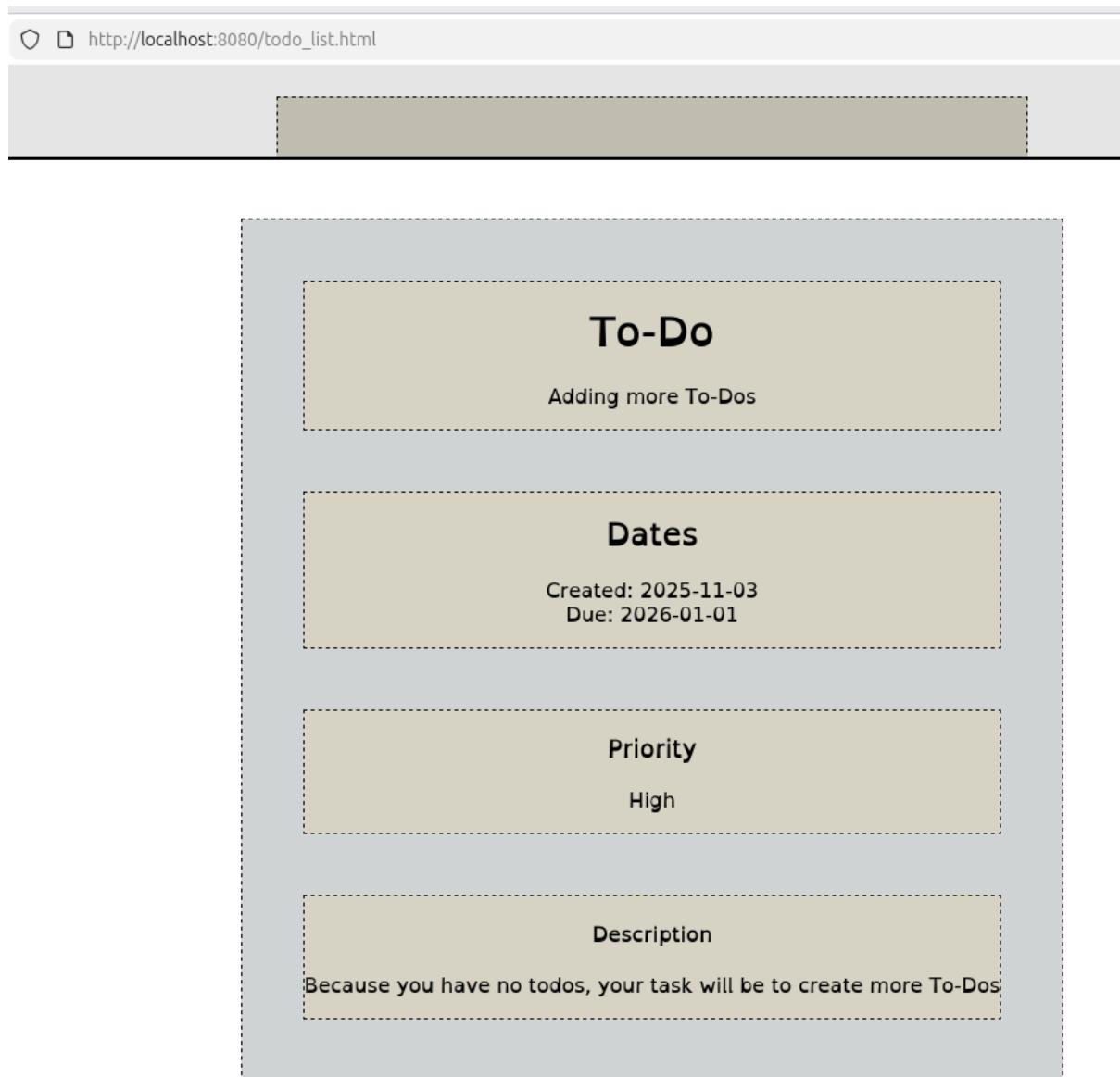
### AA

#### 1. 3.2.4 Consistent Identification

Components that have the same functionality within a set of web pages are identified consistently.

- Knöpfe haben das Gleiche Design, und ähnliche Komponenten sind konsistent in deren Aufbau.  
Z.B. sind die Inhalte und Reihenfolge der To-Do-Objekt-Abschnitte auf allen Seiten konsistent gehalten.





Write the name of your To-Do here:

Select when your To-Do is due:

mm / dd / yyyy

Choose a priority for your To-Do:

Low

Describe your To-Do in detail:

## 2. 3.2.3 Consistent Navigation

Navigational mechanisms that are repeated on multiple web pages within a set of web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.

- Die Navigation auf jeder Seite ist identisch geregelt, der einzige Unterschied ist, dass die momentane Seite auf der der Nutzer sich befindet, gehighlightet wird



### 3. 1.4.10 Reflow

Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions for:

Vertical scrolling content at a width equivalent to 320 CSS pixels.

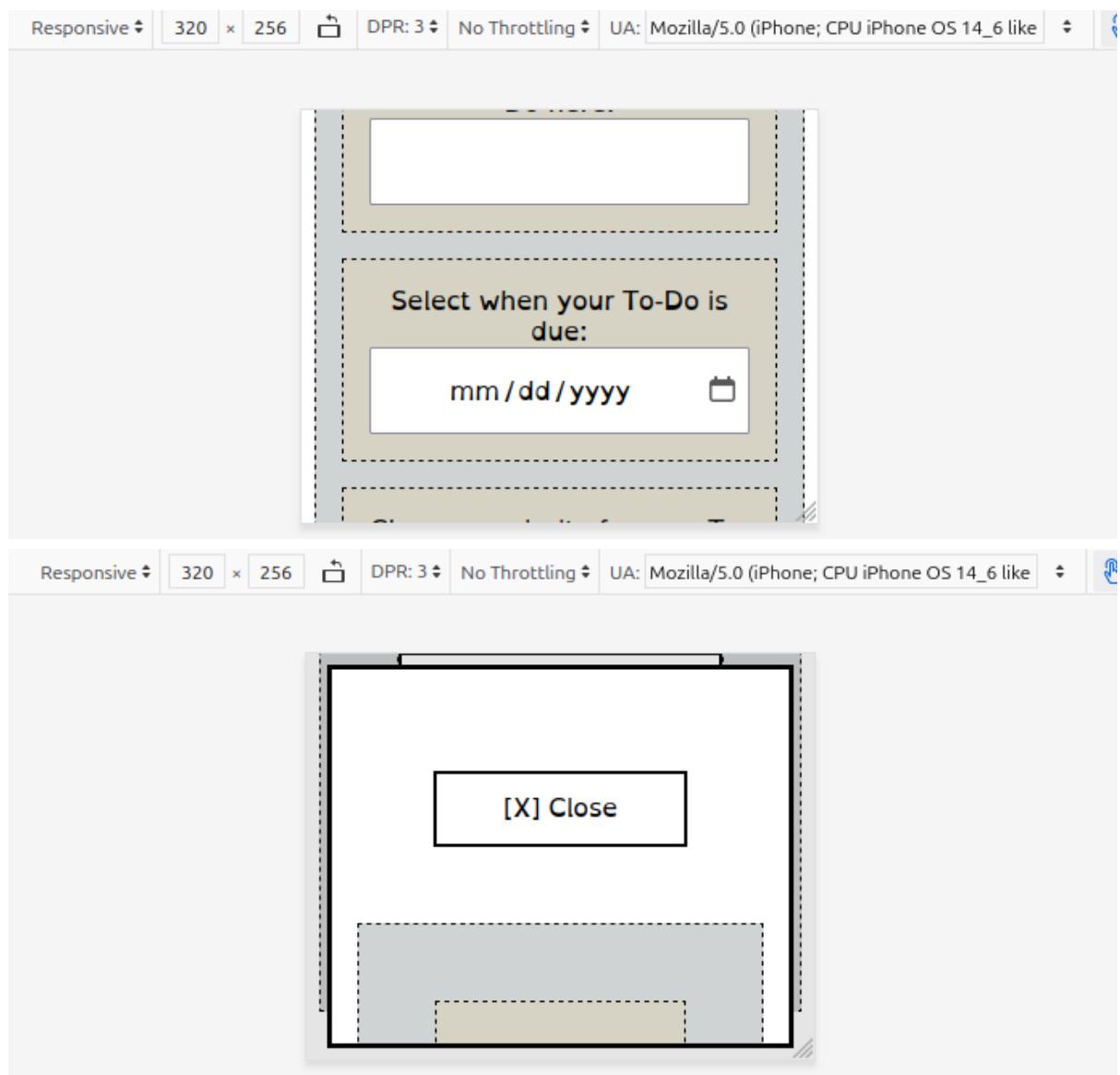
Horizontal scrolling content at a height equivalent to 256 CSS pixels.

- Ich habe überall keine Scrollbars verwendet und darauf geachtet, dass mein Design bei 320 x 256 keine Overflows besitzt.

Meine Seite ist nur vertikal scrollbar.

Anschließend habe ich die Bildschirmgröße im Browser auf 320 x 256 gestellt und getestet.

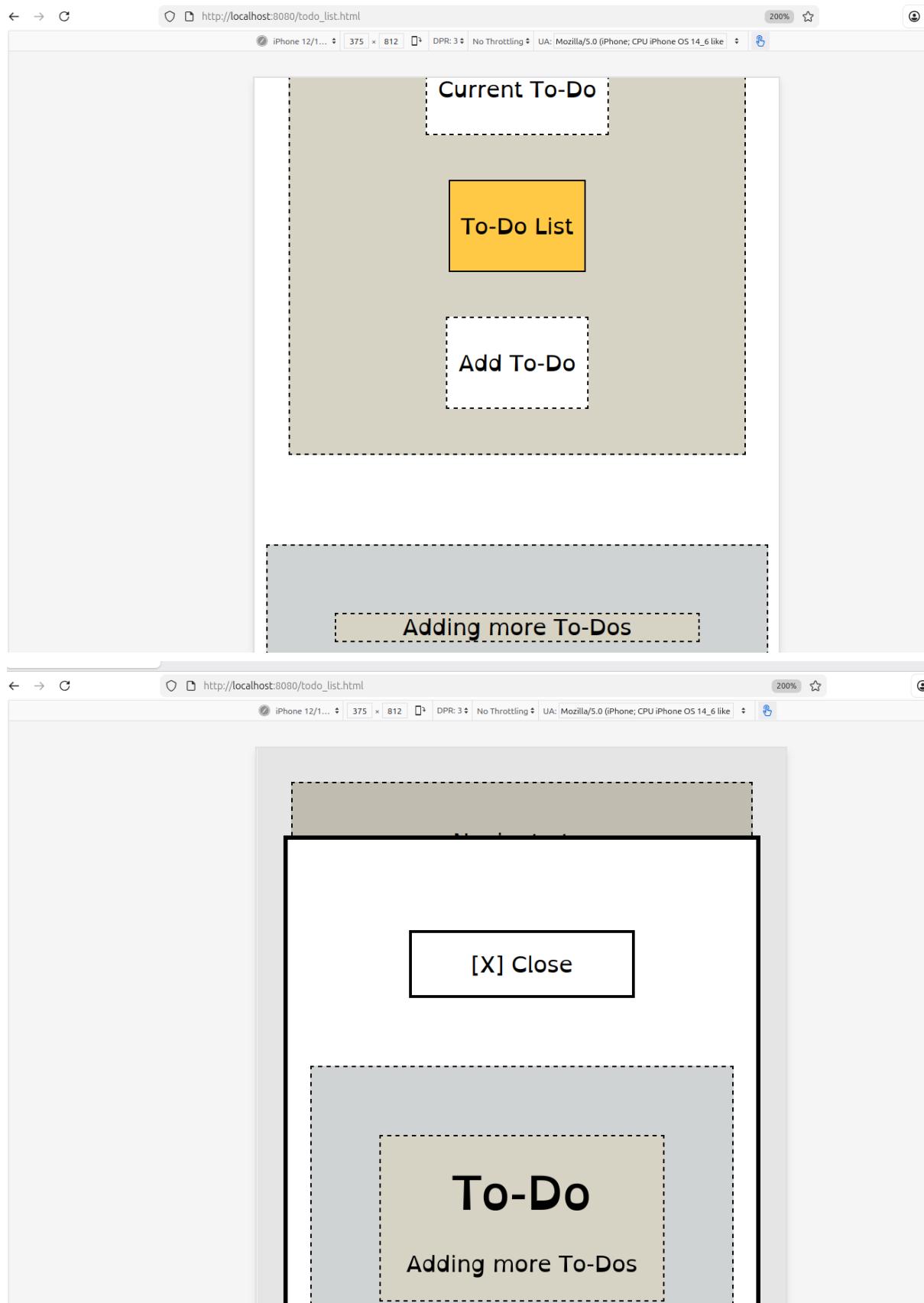
The screenshot shows a browser's responsive design mode. At the top, there are settings for 'Responsive' (set to '320 x 256'), 'DPR: 3', 'No Throttling', and 'UA: Mozilla/5.0 (iPhone; CPU iPhone OS 14\_6'. Below these settings is a preview area. In the preview area, the 'Navigate to:' section is visible, but the 'Current To-Do' button has been moved to a new position below the 'To-Do List' button, illustrating the concept of reflow where content is rearranged within the available space.

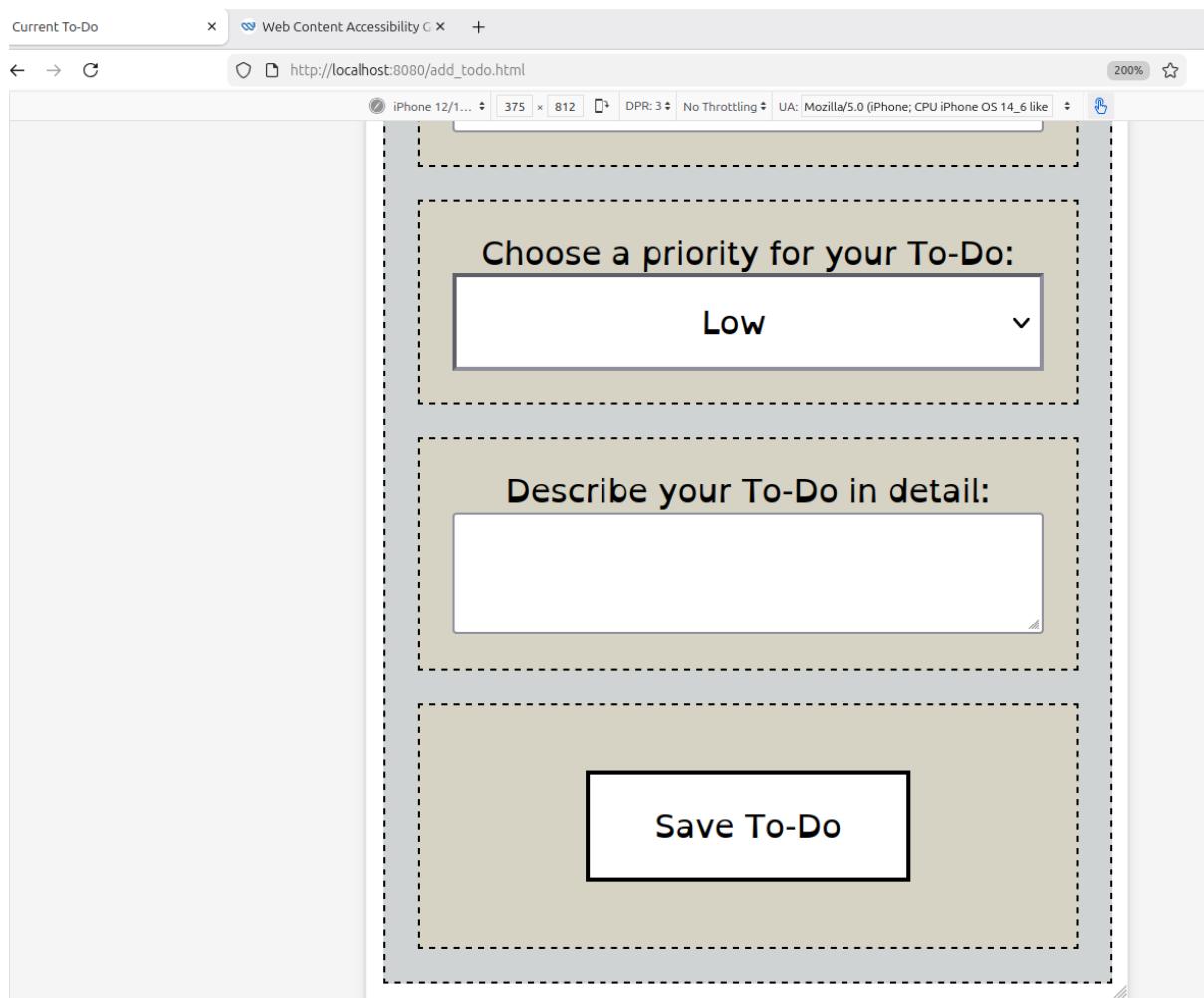


#### 4. 1.4.4 Resize Text

Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.

- Ich habe dies umgesetzt indem ich einfach die Elemente solange angepasst hatte, bis dies der Fall war. Getestet habe ich dies, indem ich die Webseiten Größe auf 200% im Browser gesetzt habe.





##### 5. 1.3.5 Identify Input Purpose

The purpose of each input field collecting information about the user can be programmatically determined when:

The input field serves a purpose identified in the Input Purposes for user interface components section.

The content is implemented using technologies with support for identifying the expected meaning for form input data.

- Ich habe die offiziellen HTML-Tags verwendet, um die DOM-Objekte direkt semantisch korrekt zu definieren.

Zudem habe ich Labels mit Beschreibungen für die Elemente verwendet.

Aria-Lables beschreiben die Input-Elemente desweiteren textuell.

```
function add_title_input(form) {
    const s_title = document.createElement("section");
    s_title.className = "title";
    s_title.setAttribute('aria-label', "Your To-Do, or a title for your To-Do");
    s_title.tabIndex = 0;

    const label = document.createElement('label');
    label.setAttribute('for', 'title-input');
    label.textContent = 'Write the name of your To-Do here:';

    const titleInput = document.createElement("input");
    titleInput.type = "text";
    titleInput.id = "title-input";

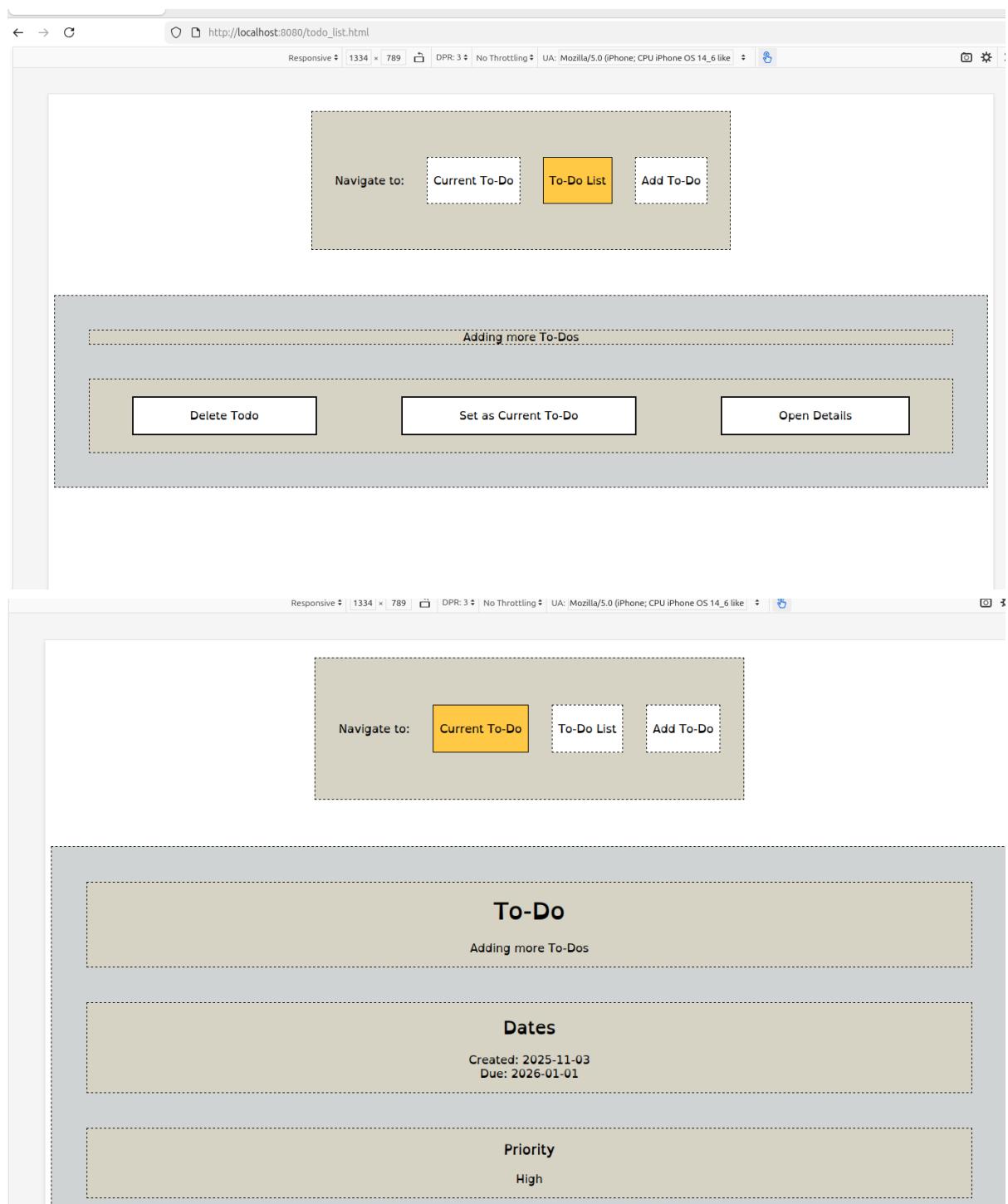
    s_title.appendChild(label);
    s_title.appendChild(titleInput);
    form.appendChild(s_title);
```

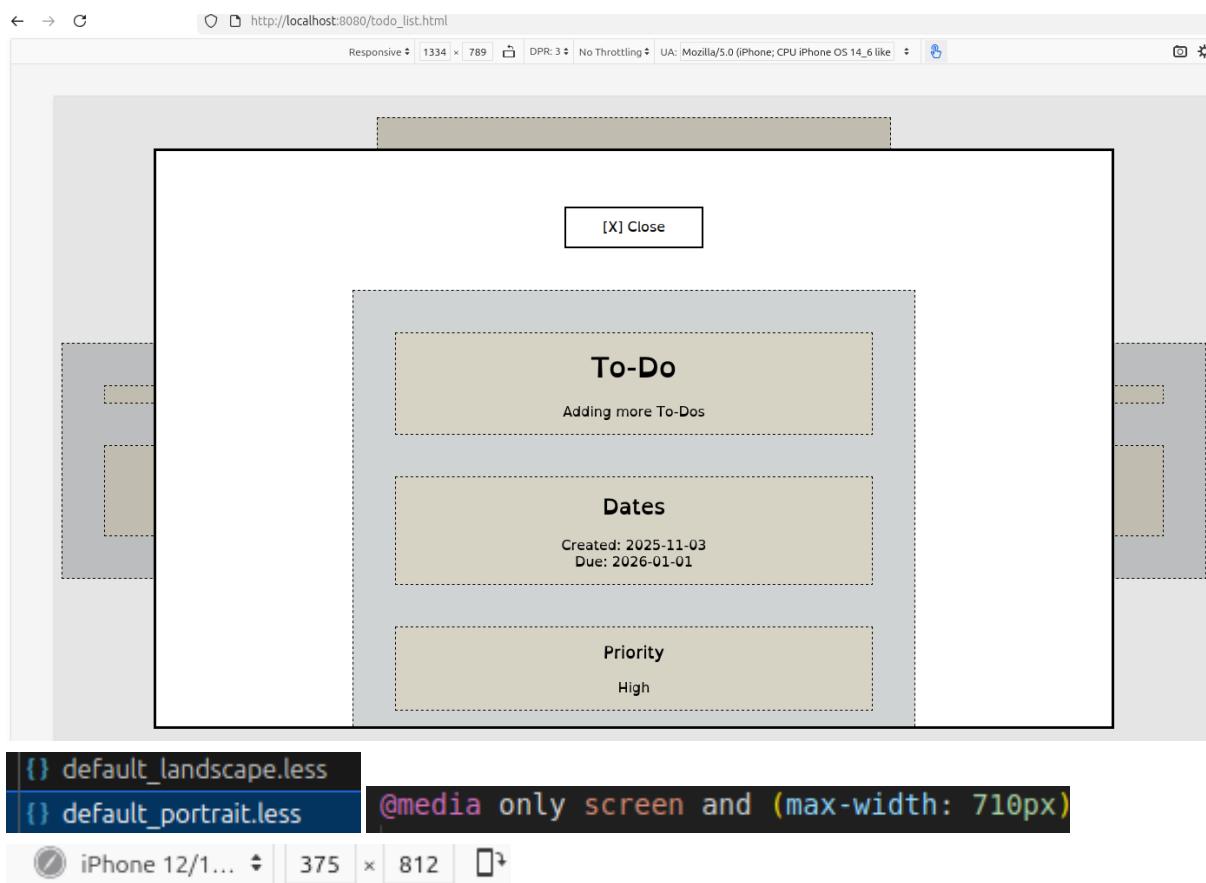
#### 6. 1.3.4 Orientation

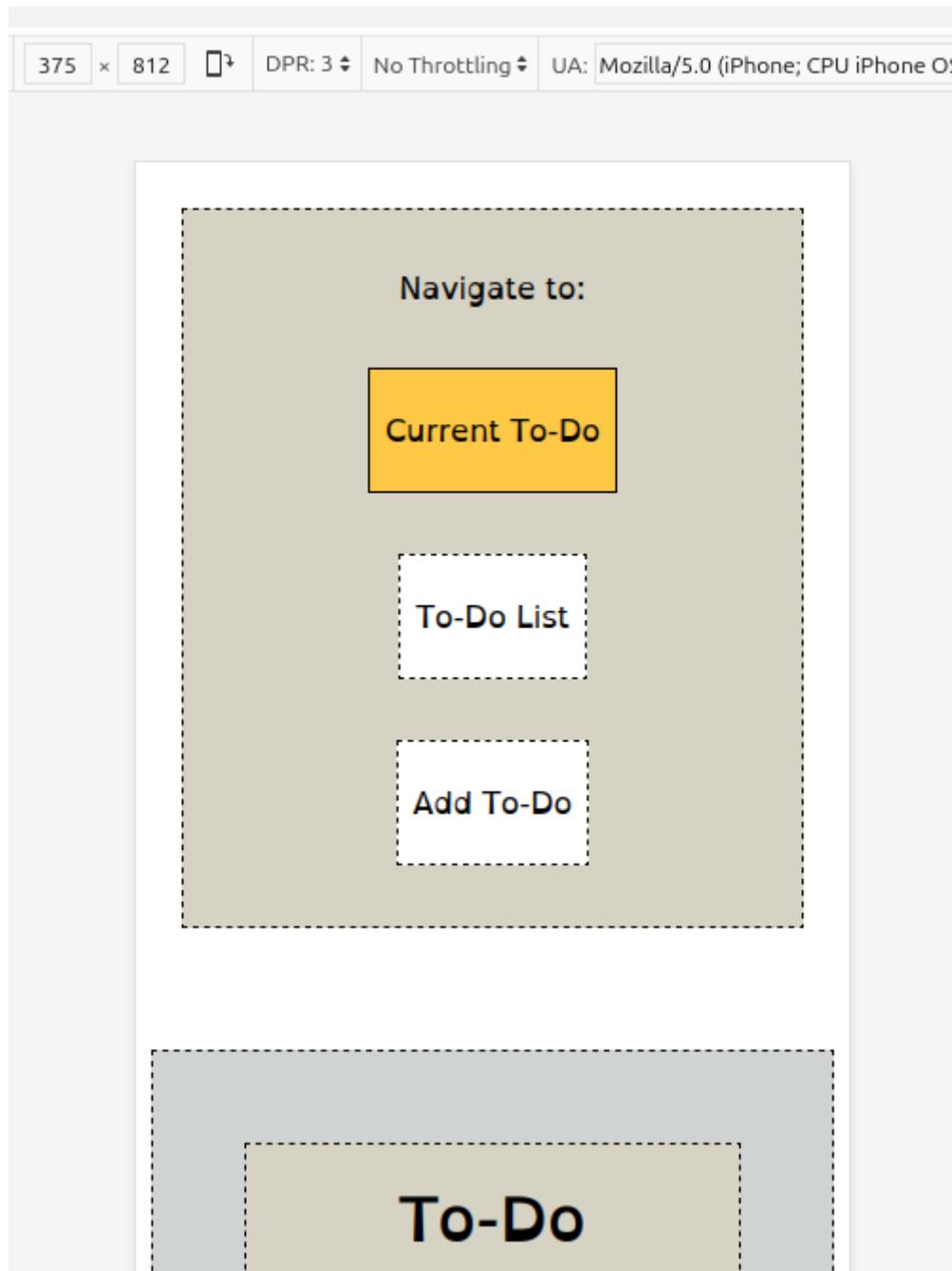
Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.

- Ich habe diese Anforderung umgesetzt, indem ich eine css-Datei für beide Ansichten erstellt habe und über eine Media-Query die Änderungen dynamisch an die Bildschirmgröße angepasst.

Getestet habe ich die Ansichten mit Firefox. Das kleinste Gerät für welches ich eine gute Ansicht garantiere ist das iPhone12/13 mini. Diese Entscheidung habe ich getroffen, da die Schrift auf der Seite ansonsten verkleinert werden müsste, und dann nicht mehr gut lesbar wäre.







The diagram illustrates a user interface for adding a new To-Do item. It features a central vertical column of input fields, each enclosed in a dashed rectangular border. At the top is a yellow button labeled "Add To-Do". Below it is a text input field with the placeholder "Write the name of your To-Do here:". The next section, "Select when your To-Do is due:", contains a date input field with the placeholder "mm / dd / yyyy" and a small calendar icon. The third section, "Choose a priority for your To-Do:", includes a dropdown menu set to "Low". The final section, "Describe your To-Do in detail:", is represented by a large, empty text input area.

Add To-Do

Write the name of your To-Do here:

Select when your To-Do is due:

mm / dd / yyyy

Choose a priority for your To-Do:

Low

Describe your To-Do in detail:



#### 7. 2.4.11 Focus Not Obscure

When a user interface component receives keyboard focus, the component is not entirely hidden due to author-created content.

Dieser Fall trifft auf meiner Seite auf die Detail-Ansicht von To-Dos auf der To-Do-List-Seite zu. Dort wird ein Overlay erzeugt welches andere Elemente der Seite verdeckt. Damit die verdeckten Elemente nicht mehr über die Tastatur angewählt werden können, verwende ich das Dialog-Element, welches automatisch beim öffnen dafür sorgt, dass alles, was nicht Teil des Dialog-Modals ist, nicht mehr über die Tastatur ansteuerbar ist. Auf dem Rest der Seiten existieren sonst keine Interaktionen, die etwas

verdecken könnten.

```
function create_modal(todoData) {
  const dialog = document.createElement("dialog");
  dialog.className = "todo-modal";

  // Create close button
  const closeButton = document.createElement("button");
  closeButton.type = "button";
  closeButton.className = "close-modal";
  closeButton.textContent = "[X] Close";
  closeButton.setAttribute("aria-label", "Close details pop-up");
  closeButton.onclick = () => dialog.close();

  dialog.appendChild(closeButton);

  // Use your existing function to create the todo content
  create_dom_todo_without_buttons(dialog, todoData);

  // Close on click outside
  dialog.onclick = (e) => {
    if (e.target === dialog) {
      dialog.close();
    }
  };

  // Remove from DOM when closed
  dialog.addEventListener("close", () => {
    dialog.remove();
  });
}
```

AAA

### 1. 2.5.5 Target Size

The size of the target for pointer inputs is at least 44 by 44 CSS pixels

- Meine Font-Größe ist 16px, basierend darauf habe ich jedem anklickbaren Element eine Mindestgröße von mindestens 3rem (3 \* Font-Größe) zugewiesen.

```
html,
body,
input,
textarea,
button,
select {
  font-size: 16px;

  font-family: "Open-Dyslexic", Verdana, Tahoma, "Segoe UI", "Helvetica Neue", Arial, sans-serif;
}

.min_size {
  min-height: 3rem;
  min-width: 3rem;
}
```

## 2. 2.4.13 Focus Appearance

When the keyboard focus indicator is visible, an area of the focus indicator meets all the following:

... is at least as large as the area of a 2 CSS pixel thick perimeter of the unfocused component or sub-component

... has a contrast ratio of at least 3:1 between the same pixels in the focused and unfocused states.

- Fokussierte Elemente haben eine Border mit der Strichstärke 0.25rem (0.25 \* 16px = 4px).

Die Farben für die Kontraste habe ich von der folgenden Seite

[https://www.sussex.ac.uk/tel/resource/tel\\_website/accessiblecontrast](https://www.sussex.ac.uk/tel/resource/tel_website/accessiblecontrast)

Die Farben habe ich dann nochmal mit einem Kontrast-Checker für die schwarze Farbe von den Rahmen geprüft.

```
Variables
*/
@MARGIN: 3rem;
@BG_COLOUR_1: #ffc845;
@BG_COLOUR_2: #d6d2c4;
@BG_COLOUR_3: #d0d3d4;
```



## Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker

**Foreground**

Hex Value:   
 Color Picker:   
 Alpha:   
 Lightness:

**Background**

Hex Value:   
 Color Picker:   
 Lightness:

**Contrast Ratio**

**13.87:1**

[permalink](#)

**Related Resources**

- [Contrast and Color Accessibility](#)
- [Quick Reference: Testing Web Content for Accessibility](#)
- [WebAIM Auditing & Evaluation Services](#)
- [Web Accessibility for Designers](#)
- [Link Contrast Checker](#)
- [Contrast Checker Bookmarklet](#)

The screenshot shows the WebAIM Contrast Checker interface. It displays two color swatches: a black foreground color (#000000) and a light gray background color (#D0D3D4). Below each color is a hex value input field and a color picker. A lightness slider is also present under each color. The central result area shows a large green-bordered box containing the text "Contrast Ratio" above "13.95:1". To the right, a sidebar titled "Related Resources" lists several links related to web accessibility.

This screenshot is identical to the one above, showing the same interface and results for a black foreground color and a yellow-orange background color (#FFC845). The central result area shows a large green-bordered box containing the text "Contrast Ratio" above "13.59:1". The sidebar on the right also contains the same list of related resources.

```
.focused {  
    border: 2px solid black;  
    border-style: solid;  
    border-width: 1rem * 0.25;  
}
```

3. 2.4.9 Link Purpose A mechanism is available to allow the purpose of each link to be identified from link text alone, except where the purpose of the link would be ambiguous to users in general.

- Die Links die auf der Seite existieren (Navigationszeile) wurden durch die Namen der Seiten auf welche diese führen dargestellt.

```
// Label
const label = document.createElement('label');
label.setAttribute('for', 'page-select');
label.textContent = 'Navigate to:';

nav.id = 'page-select';
nav.appendChild(label);

// Pages
const pages = [
  { value: './index.html', text: 'Current To-Do' },
  { value: './todo_list.html', text: 'To-Do List' },
  { value: './add_todo.html', text: 'Add To-Do' }
];

pages.forEach(pageData => {
  const page_link = document.createElement('a');
  page_link.href = pageData.value;
  page_link.textContent = pageData.text;

  // Aktuelle Seite markieren
  if (pageData.value === current_page) {
    page_link.setAttribute('aria-current', 'page');
    page_link.id = 'current-page';
  }
})
```

4. 1.4.6 Contrast

The visual presentation of text and images of text has a contrast ratio of at least 7:1, except for the following

- Der Text erfüllt überall den Kontrast.

The screenshot shows the WebAIM Contrast Checker interface. It displays two examples of color contrast ratios:

- Example 1 (Top):** Foreground color is black (#000000) and background color is orange (#FFC845). The contrast ratio is 13.59:1.
- Example 2 (Bottom):** Foreground color is black (#000000) and background color is white (#FFFFFF). The contrast ratio is 21:1.

The interface includes input fields for Hex Value, Color Picker, Alpha, and Lightness for both foreground and background colors. A "Contrast Ratio" box highlights the calculated ratios. A "Related Resources" sidebar on the right lists various web accessibility links.

### 5. 2.1.3 Keyboard Navigation

All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes.

- Die gesamte Seite ist navigierbar mit der Tastatur über Tab.

Ich habe zudem für die Abschnitte der To-Dos Tabindices eingebunden, um die Navigation besser zu strukturieren.

## 6. 2.4.8 Location

Information about the user's location within a set of web pages is available.

- Die Navigationszeile im Header highlighted die momentan ausgewählte Seite, und weißt ihr das Aria-Attribut "aria-current" zu, für den Kontext "page".

```
// Aktuelle Seite markieren
if (pageData.value === current_page) {
    page_link.setAttribute('aria-current', 'page');
    page_link.id = 'current-page';
}

function create_nav(header, current_page) {
    const nav = document.createElement('nav');

    // Label
    const label = document.createElement('label');
    label.setAttribute('for', 'page-select');
    label.textContent = 'Navigate to:';

    nav.id = 'page-select';
    nav.appendChild(label);

    // Pages
    const pages = [
        { value: './index.html', text: 'Current To-Do' },
        { value: './todo_list.html', text: 'To-Do List' },
        { value: './add_todo.html', text: 'Add To-Do' }
    ];

    pages.forEach(pageData => {
        const page_link = document.createElement('a');
        page_link.href = pageData.value;
        page_link.textContent = pageData.text;

        // Aktuelle Seite markieren
        if (pageData.value === current_page) {
            page_link.setAttribute('aria-current', 'page');
            page_link.id = 'current-page';
        }

        nav.appendChild(page_link);
    });
}
```