

Website-Qualitätsbericht

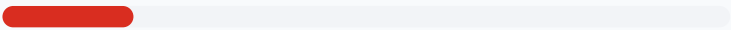
deutsche-rentenversicherung.de

Technischer Website-Check mit Fokus auf Accessibility, SEO und Performance

GESAMTSCORE

18

Kritischer technischer Zustand



WCAG-VORKOMMEN

184

93 kritisch/hoch

MODULE · 0-100, HÖHER IST BESSER



Barrierefreiheit

MANAGEMENT-SUMMARY

Management-Sicht

Gesamtstatus, Hauptrisiken, wichtigste Maßnahmen und Hebel auf einen Blick.

WCAG-Vorkommen	Kritisch	Hoch	Mittel
184	0	93	90

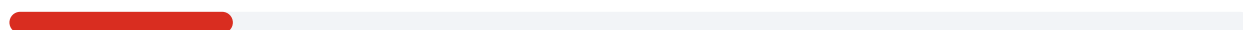
Gesamturteil

<https://www.deutsche-rentenversicherung.de/> erreicht nur 18/100 im Accessibility-Audit. Akuter Handlungsbedarf: Wesentliche Inhalte und Funktionen sind für einen Teil der Nutzer nicht zugänglich.

Wo sich Optimierung lohnt

Barrierefreiheit

18



Kritisch — wesentliche Anforderungen an die Barrierefreiheit werden nicht erfüllt.

Die 5 wichtigsten Risiken

- | | |
|-------------------------------------|--|
| ● 1. Nutzbarkeit | Screenreader und Tastaturnavigation an kritischen Stellen blockiert oder beeinträchtigt. |
| ● 2. Rechtliche Konformität (BFSG) | BFSG-relevante Verstöße (WCAG Level A/AA) gefunden — Behebung empfohlen. |
| ● 3. Conversion & Business-Risiko | Hohes Risiko von Prozessabbrüchen und Conversion-Verlusten. |
| ● 4. SEO & KI-Sichtbarkeit | Sehr gute Auffindbarkeit und strukturierte Schema-Daten. |
| ● 5. Ladezeit & Mobilfreundlichkeit | Schnelle Ladezeit und optimale responsive Darstellung. |

Die 5 wichtigsten Maßnahmen

- 1. Fehlende Name/Rolle bei Bedienelementen: Alle interaktiven Elemente mit einem verständlichen, zugänglichen Namen versehen. Native HTML-Elemente bevorzugen. Bei Icon-Buttons: aria-label verwenden.
- 2. Fehlende Alternativtexte bei Bildern: Für informative Bilder einen beschreibenden Alt-Text hinterlegen, der den Bildinhalt oder -zweck vermittelt. Rein dekorative Bilder mit einem leeren Alt-Attribut markieren (alt="").
- 3. Tastaturfokus auf unsichtbaren Elementen: Wenn ein Element vollständig versteckt sein soll, display:none oder visibility:hidden verwenden. Soll es nur vor Screenreadern versteckt sein, muss es unfokussierbar sein (tabindex="-1").
- 4. Ungültige oder verbotene ARIA-Attribute: Prüfe die ARIA-Spezifikation für die jeweilige Rolle. Verwende ARIA-Attribute nur auf passenden Elementen und Rollen.
- 5. ARIA-Rollen-Konflikt in Navigation: Website-Navigation ohne ARIA-Menubar-Muster aufbauen: `<nav><ul><li><a>` reicht für Screenreader aus. `role="menuitem"` und `role="menubar"` sind für Desktop-App-Menüs reserviert, nicht für Seitennavigation. `role="none"` nur auf Elementen verwenden, die wirklich keine Kinder mit Eigenrollen haben.

Inhalt

Management-Sicht	2
Inhalt	4
Befunde nach Ursache	5
Ursachenanalyse	5
Maßnahmenplan	7
Technische Details für Entwickler	10
Systemdiagnose	10
Klassifizierung der Befunde	11
Systemische Template- & Komponentenfehler (WCAG A/AA)	11
Einzelfälle & Redaktionelle Befunde (WCAG A/AA)	20
Tastatur-Accessibility-Journey	31
Screenreader-Lesereihenfolge	33
Qualitäts-Analysen	35
Dark Mode	36
KI & Vertrauen	38
Anhang & Methodik	45
Prüfumfang	45
Vollständige Fundstellen	49

01

TEIL 1 VON 3

Befunde nach Ursache

Welche Ursachen die meisten Befunde erklären – und der priorisierte Plan, sie zu beheben.

Zielgruppe: Inhaber, Entscheider und Entwickler. Dieser Teil erklärt, was die Befunde verursacht und was als Nächstes zu tun ist.

URSACHEN

Ursachenanalyse

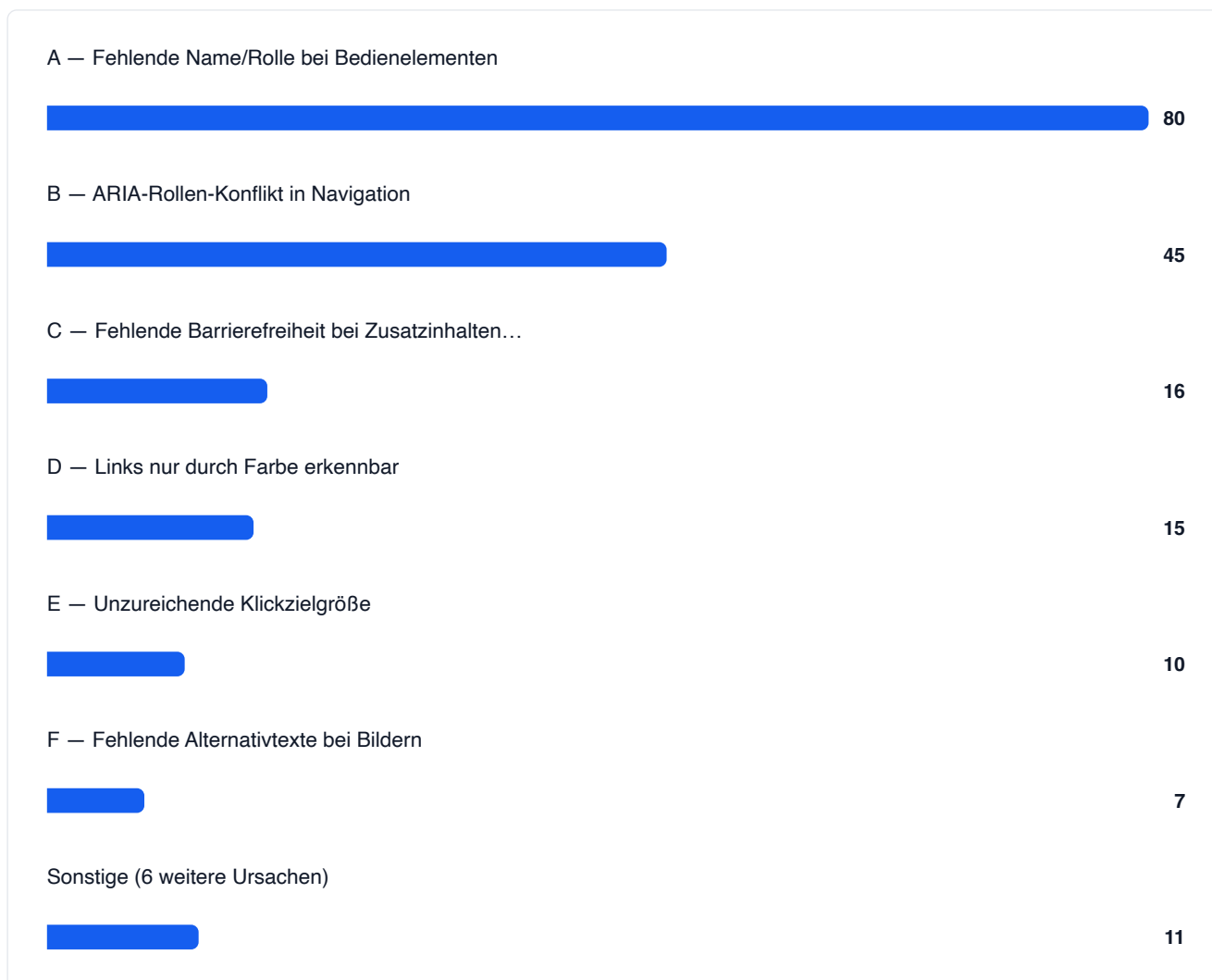
Viele Einzelbefunde gehen auf wenige wiederkehrende Ursachen zurück – deren Behebung wirkt gebündelt.

Das bedeutet konkret: Ein Fix an der richtigen Stelle – etwa einer wiederverwendeten Komponente oder einem Template – behebt oft mehrere Befunde gleichzeitig, statt jeden einzeln reparieren zu müssen.

Erkannte Kernursachen

- Ursache A: 80 Vorkommen — Fehlende Name/Rolle bei Bedienelementen
- Ursache B: 45 Vorkommen — ARIA-Rollen-Konflikt in Navigation
- Ursache C: 16 Vorkommen — Fehlende Barrierefreiheit bei Zusatzinhalten (Tooltips)
- Ursache D: 15 Vorkommen — Links nur durch Farbe erkennbar
- Ursache E: 10 Vorkommen — Unzureichende Klickzielgröße
- Ursache F: 7 Vorkommen — Fehlende Alternativtexte bei Bildern

Verteilung der Vorkommen nach Ursache



Verteilung der Mängel nach Ursache

Ursache	Vorkommen	Anteil
A — Fehlende Name/Rolle bei Bedienelementen	80	43 %
B — ARIA-Rollen-Konflikt in Navigation	45	24 %
C — Fehlende Barrierefreiheit bei Zusatzinhalten...	16	9 %
D — Links nur durch Farbe erkennbar	15	8 %
E — Unzureichende Klickzielgröße	10	5 %
F — Fehlende Alternativtexte bei Bildern	7	4 %
Sonstige (6 weitere Ursachen)	11	6 %

MASSNAHMEN

Maßnahmenplan

Empfohlene Maßnahmen, gruppiert nach Ebene des Problems — inklusive ergänzender SEO- und Qualitätsempfehlungen ohne Rechtsbezug.

Systemische Maßnahmen

Einmal in Vorlage oder Komponente beheben — wirkt auf allen betroffenen Seiten zugleich.

Interaktive Elemente (Buttons, Links) verständlich benennen

Weitverbreitet: Kann Nutzer ausschließen und rechtliches Risiko erhöhen.

→ Ursache A, 80 Vorkommen (43 %)

Seitenstruktur mit Orientierungspunkten auszeichnen

Weitverbreitet: Beeinträchtigt Qualität und Nutzererlebnis der Website.

→ Ursache B, 45 Vorkommen (24 %)

Verwende eine sichtbare Beschriftung oder moderne, barrierefreie Tooltips (schließbar mit Escape, verknüpft über aria-describedby und hoverbar).

Häufig: Beeinträchtigt Lesbarkeit für Nutzer mit Sehschwäche.

→ Ursache C, 16 Vorkommen (9 %)

Links im Fließtext mit einem nicht-farblichen Merkmal kennzeichnen: Unterstrich (Standard und empfohlen), Fettschrift oder ein kleines Icon. Der Unterstrich ist die stärkste Konvention.

Häufig: Beeinträchtigt Lesbarkeit für Nutzer mit Sehschwäche.

→ Ursache D, 15 Vorkommen (8 %)

Klickflächen auf mindestens 24x24 CSS-Pixel vergrößern, z. B. durch Padding, oder ausreichend Abstand zu benachbarten Zielen einhalten.

Häufig: Beeinträchtigt Qualität und Nutzererlebnis der Website.

→ Ursache E, 10 Vorkommen (5 %)

Lokale Maßnahmen & Einzelfälle

Punktuelle Korrekturen an einzelnen Seiten, Bildern oder redaktionellen Inhalten.

Bilder mit beschreibendem Alternativtext versehen

Häufig: Kann Nutzer ausschließen und rechtliches Risiko erhöhen.

→ Ursache F, 7 Vorkommen (4 %)

Wenn ein Element vollständig versteckt sein soll, `display:none` oder `visibility:hidden` verwenden. Soll es nur vor Screenreadern versteckt sein, muss es unfokussierbar sein (`tabindex="-1"`).

Häufig: Kann Nutzer ausschließen und rechtliches Risiko erhöhen.

Prüfe die ARIA-Spezifikation für die jeweilige Rolle. Verwende ARIA-Attribute nur auf passenden Elementen und Rollen.

Kann Nutzer ausschließen und rechtliches Risiko erhöhen.

Formularfelder eindeutig beschriften

Beeinträchtigt Qualität und Nutzererlebnis der Website.

Schriftgrößen in relativen Einheiten (rem/em) angeben. Container so gestalten, dass sie bei vergrößertem Text mitwachsen. Browser-Zoom auf 200% testen.

Beeinträchtigt Lesbarkeit für Nutzer mit Sehschwäche.

Tastaturnavigation und Fokus-Reihenfolge sicherstellen

Beeinträchtigt Qualität und Nutzererlebnis der Website.

Hinweise zur Aussagekraft: Automatisierte Befunde sind belastbar für erkennbare Muster, aber kontextabhängige WCAG-Kriterien, semantische Inhaltsqualität und Nutzerwege benötigen weiterhin manuelle Prüfung. Indikator-Module wie KI-Sichtbarkeit, Content Visibility und UX sind Hinweismodule, keine Garantien.

Audit-Hinweise**● Unvollständiger Messumfang**

Eine oder mehrere angeforderte Prüfungen konnten nicht abgeschlossen werden. Die Scores beschreiben nur den erfolgreich gemessenen Umfang.

02

TEIL 2 VON 3

Technische Details für Entwickler

Detaillierte WCAG-Verstöße, HTML-Evidenzen und Code-Beispiele, gruppiert nach systemischen Komponenten vs. Einzelfällen.

Zielgruppe: Entwickler und QA-Teams. Dieser Teil liefert konkrete Code-Befunde zur Behebung aller Barrieren.

Modulübersicht

● Barrierefreiheit 18/100

Systemdiagnose

Geklusterte Probleme: Probleme konzentrieren sich in zusammenhängenden Bereichen. Die Behebung eines Clusters reduziert mehrere Findings gleichzeitig.

Kategorie-Übersicht

Kategorie	Vorkommen	Schwerster Schweregrad
Accessibility	184	Hoch

Problemcluster

Cluster	Befunde	Vorkommen	Max. Schweregrad
Barrierefreiheits-Barrieren (17 Befunde)	17	184	Hoch

BEFUNDE

Klassifizierung der Befunde

Alle technischen Befunde, getrennt nach systemischen Komponentenfehlern und Einzelfällen.

Systemisch (Pflicht)	Systemisch (Optimierung)	Lokal (Pflicht)	Lokal (Optimierung)
5	0	7	0

SYSTEMISCHE PFLICHT

Systemische Template- & Komponentenfehler (WCAG A/AA)

Diese Befunde betreffen wiederkehrende Template- oder Komponentenfehler. Eine zentrale Behebung in der Vorlage behebt die Fehler auf allen betroffenen Seiten gleichzeitig.

[Hoch] Fehlende Name/Rolle bei Bedienelementen — WCAG 4.1.2 (A)

Interaktive Elemente (Buttons, Links, Formularfelder) haben keinen zugänglichen Namen oder keine erkennbare Rolle. Assistive Technologien können nicht vermitteln, worum es sich handelt. Screenreader-Nutzer hören z. B. nur 'Button' ohne Beschreibung der Funktion, oder ein klickbares Element wird gar nicht als interaktiv erkannt.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Hoch	Entwicklung	Geringer Aufwand	Hoch

PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz (Barrierefreiheitsstärkungsgesetz / European Accessibility Act)	Hoch
Elemente	80
Vorkommen	80
Referenz	https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html

MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 80 Vorkommen; erwartete Auswirkung auf den Score: hoch; WCAG-Level: A.
Komplexitätsgrund	80 Vorkommen deuten auf ein Komponenten- oder Template-Problem hin.

Element-Typen 2x button

Empfehlung

Alle interaktiven Elemente mit einem verständlichen, zugänglichen Namen versehen. Native HTML-Elemente bevorzugen. Bei Icon-Buttons: aria-label verwenden.

× Falsch

```
<button><svg> ... </svg></button>
```

✓ Richtig

```
<button aria-label="Open menu"><svg  
aria-hidden="true"> ... </svg></button>
```

Repräsentative Fundstellen

Fundstelle	Hinweis
a#_mokvsao5s [href: #]	Button is missing accessible text
button#slider-_2ix2axxlv-prev-0	Element's accessible name and description are identical: 'Vorherige Seite'
button.backLink	Element's accessible name and description are identical: 'Keine vorherige Seite'

Fundstelle: a#_mokvsao5s [href: #]

Node **8294**

Hinweis **Button is missing accessible text**

HTML **<a class="js-simple-toggle-opener inactive-control" href="#" id="_mokvsao5s" role="button" tabindex="0" ari...**

DOM-Pfad **a#_mokvsao5s**

Vorgeschlagene Code-Korrektur

```
<a class="js-simple-toggle-opener inactive-control" href="#" id="_mokvsao5s" role="button"  
tabindex="0" aria-expanded="false" aria-controls="_r70ny0pjw" aria-label="Button"> <sp...
```

Fundstelle: button#slider-_2ix2axxlv-prev-0

Node 7623

Hinweis Element's accessible name and description are identical: 'Vorherige Seite'

HTML `<button class="c-page-header-slider__prev prev js-slideshow__prev" title="Vorherige Seite" id="slider-_2ix2axxlv-prev-0">`

DOM-Pfad `button#slider-_2ix2axxlv-prev-0`

Vorgeschlagene Code-Korrektur

`<button class="c-page-header-slider__prev prev js-slideshow__prev" title="Vorherige Seite" id="slider-_2ix2axxlv-prev-0" aria-label="Vorherige Seite" disabled=""> `

Fundstelle: button.backLink

Node 8032

Hinweis Element's accessible name and description are identical: 'Keine vorherige Seite'

HTML `<button class="backLink" title="Keine vorherige Seite" disabled="disabled"> <img alt="Keine vorherige Seite" />`

DOM-Pfad `div.navigation:nth-of-type(1) > div.prev:nth-of-type(1) > button.backLink`

Figure 1: Element auf der Seite (Desktop-Ansicht)

Vorgeschlagene Code-Korrektur

```
<button class="backLink" title="Keine vorherige Seite" disabled="disabled" aria-label="Button"> https://www.w3.org/WAI/WCAG21/Techniques/failures/F92</a> |

### MASSNAHMENBEWERTUNG

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| Erwartete Wirkung | Behebt 45 Vorkommen; erwartete Auswirkung auf den Score: mittel; WCAG-Level: A. |
| Komplexitätsgrund | 45 Vorkommen deuten auf ein Komponenten- oder Template-Problem hin.             |

### Empfehlung

Website-Navigation ohne ARIA-Menubar-Muster aufbauen: `<nav><ul><li><a>` reicht für Screenreader aus. `role="menuitem"` und `role="menubar"` sind für Desktop-App-Menüs reserviert, nicht für Seitennavigation. `role="none"` nur auf Elementen verwenden, die wirklich keine Kinder mit Eigenrollen haben.

#### ✗ Falsch

```
<li role="none"><a href="/home"
role="menuitem">Home
```

#### ✓ Richtig

```
Home
```

**Root Cause**

Root Cause: 1 Komponentenproblem erzeugt 45 Vorkommen. Wahrscheinlich ein gemeinsam genutztes Template oder eine Komponente — ein einmaliger Fix behebt alle Vorkommen gleichzeitig.

## [Mittel] Fehlende Barrierefreiheit bei Zusatzinhalten (Tooltips) — WCAG 1.4.13 (AA)

Zusatzinhalte, die erst bei Hover (Mauszeiger) oder Fokus (Tastatur) erscheinen (wie Tooltips oder native title-Attribute), sind nicht barrierefrei zugänglich oder unvollständig verknüpft. Nutzer von Screenreadern hören diese Inhalte gar nicht, Tastaturnutzer können sie oft nicht schließen, und auf Touchgeräten (Mobilgeräten) sind sie in der Regel unbrauchbar.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Entwicklung	Mittlerer Aufwand	Hoch

**PRÜFMETADATEN**

Erkennungssicherheit	Mittel
Falschpositiv-Risiko	Mittel
BFSG-/EAA-Relevanz	Mittel
Elemente	16
Vorkommen	16
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html">https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html</a>

**MASSNAHMENBEWERTUNG**

Erwartete Wirkung	Behebt 16 Vorkommen; erwartete Auswirkung auf den Score: mittel; WCAG-Level: AA.
Komplexitätsgrund	16 Vorkommen deuten auf ein Komponenten- oder Template-Problem hin.
Prüfung	Manuelle Prüfung empfohlen

**Empfehlung**

Verwende eine sichtbare Beschriftung oder moderne, barrierefreie Tooltips (schließbar mit Escape, verknüpft über aria-describedby und hoverbar).



## × Falsch

```
<button title="Mehr Info">...</button>
<div role="tooltip">Hilfetext</div>
```

## ✓ Richtig

```
<button aria-describedby="tip1">Mehr
Info</button>
<div id="tip1" role="tooltip">Hilfe-
text</div>
```

**Root Cause**

Root Cause: 1 Komponentenproblem erzeugt 16 Vorkommen. Wahrscheinlich ein gemeinsam genutztes Template oder eine Komponente — ein einmaliger Fix behebt alle Vorkommen gleichzeitig.

**[Mittel] Links nur durch Farbe erkennbar — WCAG 1.4.1 (A)**

Links im Fließtext lassen sich ausschließlich durch ihre Farbe von normalem Text unterscheiden. Für Menschen mit Farbsehschwäche sind diese Links nicht als klickbar erkennbar. Nutzer mit Rot-Grün-Schwäche oder anderen Farbsehschwächen können Links im Text nicht erkennen und verpassen so wichtige Navigationsmöglichkeiten.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Design / UX	Geringer Aufwand	Hoch

**PRÜFMETADATEN**

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Mittel
Elemente	15
Vorkommen	15
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/use-of-color.html">https://www.w3.org/WAI/WCAG21/Understanding/use-of-color.html</a>

**MASSNAHMENBEWERTUNG**

Erwartete Wirkung	Behebt 15 Vorkommen; erwartete Auswirkung auf den Score: mittel; WCAG-Level: A.
Komplexitätsgrund	15 Vorkommen deuten auf ein Komponenten- oder Template-Problem hin.

## Empfehlung

Links im Fließtext mit einem nicht-farblichen Merkmal kennzeichnen: Unterstrich (Standard und empfohlen), Fettschrift oder ein kleines Icon. Der Unterstrich ist die stärkste Konvention.

### × Falsch

```
a { color: #0057b8; text-decoration: none; }
```

### ✓ Richtig

```
a { color: #0057b8; text-decoration: underline; }
```

## Root Cause

Root Cause: 1 Komponentenproblem erzeugt 15 Vorkommen. Wahrscheinlich ein gemeinsam genutztes Template oder eine Komponente — ein einmaliger Fix behebt alle Vorkommen gleichzeitig.

## [Mittel] Unzureichende Klickzielgröße — WCAG 2.5.8 (AA)

Bedienelemente wie Buttons, Links oder Icons haben eine Klickfläche von weniger als 24x24 CSS-Pixeln. Nutzer mit motorischen Einschränkungen oder auf Touchscreens treffen kleine Zielflächen leicht daneben und lösen versehentlich falsche Aktionen aus.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Design / UX	Geringer Aufwand	Hoch

### PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Niedrig
Elemente	10
Vorkommen	10
Referenz	<a href="https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html">https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html</a>

### MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 10 Vorkommen; erwartete Auswirkung auf den Score: mittel; WCAG-Level: AA.
Komplexitätsgrund	10 Vorkommen deuten auf ein Komponenten- oder Template-Problem hin.

## Empfehlung

Klickflächen auf mindestens 24x24 CSS-Pixel vergrößern, z. B. durch Padding, oder ausreichend Abstand zu benachbarten Zielen einhalten.

### × Falsch

```
<button style="width: 16px; height: 16px; padding: 0;">x</button>
```

### ✓ Richtig

```
<button style="width: 24px; height: 24px; padding: 4px;">x</button>
```

## Root Cause

Root Cause: 1 Komponentenproblem erzeugt 10 Vorkommen. Wahrscheinlich ein gemeinsam genutztes Template oder eine Komponente — ein einmaliger Fix behebt alle Vorkommen gleichzeitig.

## Einzelfälle & Redaktionelle Befunde (WCAG A/AA)

Punktuelle Barrieren, die nur einzelne Seiten, spezifische Bilder oder redaktionelle Texte betreffen und meist individuell behoben werden müssen.

### [Hoch] Fehlende Alternativtexte bei Bildern — WCAG 1.1.1 (A)

Bilder auf der Website haben keinen beschreibenden Alternativtext. Dadurch können Screenreader den Bildinhalt nicht an blinde oder sehbeeinträchtigte Nutzer vermitteln. Menschen mit Sehbeeinträchtigung erhalten an diesen Stellen keine oder nur unvollständige Information. Der Bildinhalt geht für sie vollständig verloren.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Hoch	Redaktion	Geringer Aufwand	Hoch

#### PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Hoch
Elemente	7
Vorkommen	7
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/non-text-content.html">https://www.w3.org/WAI/WCAG21/Understanding/non-text-content.html</a>

#### MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 7 Vorkommen; erwartete Auswirkung auf den Score: hoch; WCAG-Level: A.
Komplexitätsgrund	7 Vorkommen erfordern einheitliche Anpassungen in Inhalten oder Templates.

Element-Typen 1× 19 · 1× 20 · 1× 23

#### Empfehlung

Für informative Bilder einen beschreibenden Alt-Text hinterlegen, der den Bildinhalt oder -zweck vermittelt. Rein dekorative Bilder mit einem leeren Alt-Attribut markieren (alt="").

## Repräsentative Fundstellen

Fundstelle	Hinweis
svg [viewBox=0 0 13 23]	Image is missing alternative text
svg [viewBox=0 0 25 19]	Image is missing alternative text
svg [viewBox=0 0 20 20]	Image is missing alternative text

### Fundstelle: svg [viewBox=0 0 13 23]

Node **8319**

Hinweis **Image is missing alternative text**

HTML `<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 13 23"><def...`

DOM-Pfad `li.c-sticky-nav__item:nth-of-type(3) > a.js-to-top > svg`

#### Vorgeschlagene Code-Korrektur

`<!-- Decorative (icon next to visible text): --> <svg aria-hidden="true">...</svg> <!-- Informative (standalone image without text context): --> <svg role="img" aria-label="Desc...`

### Fundstelle: svg [viewBox=0 0 25 19]

Node **6937**

Hinweis **Image is missing alternative text**

HTML `<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 25 19"><path fill="currentColor" d="M0,19V16H25v3Zm0-8...`

DOM-Pfad `button.js-mobile-nav__toggle > div.c-mobile-nav__toggle-button-image:nth-of-type(1) > svg`

**Vorgeschlagene Code-Korrektur**

<!-- Decorative (icon next to visible text): --> <svg aria-hidden="true">...</svg> <!-- Informative (standalone image without text context): --> <svg role="img" aria-label="Desc...

**Fundstelle: svg [viewBox=0 0 20 20]**

Node 7614

Hinweis Image is missing alternative text

HTML <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" viewBox="0 0 20 20"> <path fill="#FFFFFF" fi...

DOM-Pfad span.formField:nth-of-type(2) &gt; button &gt; svg

**Vorgeschlagene Code-Korrektur**

<!-- Decorative (icon next to visible text): --> <svg aria-hidden="true">...</svg> <!-- Informative (standalone image without text context): --> <svg role="img" aria-label="Desc...

**Häufige Muster**

Muster	Vorkommen
svg [viewBox=0 0 13 23]	5

Diese Tabelle gruppiert Vorkommen nach struktureller Ähnlichkeit (gleiches Element- und Auswahlmuster) über die gesamte Seite hinweg, nicht nach exakt identischem Wert — die Zahl kann daher von der Vorkommenszahl eines einzelnen Befunds abweichen.

**Strukturelle Ursache**

Dieses Problem tritt bei 7 Elementen auf — möglicherweise eine gemeinsam genutzte Komponente oder ein Template.

## [Hoch] Tastaturfokus auf unsichtbaren Elementen — WCAG aria-hidden-focus (A)

Ein interaktives Element ist über `aria-hidden="true"` vor Screenreadern versteckt, kann aber dennoch mit der Tastatur (Tabulator) fokussiert werden. Tastaturnutzer steuern auf unsichtbare Elemente (Fokus-Stolperfallen), während Screenreader-Nutzer keinerlei Rückmeldung über dieses Element erhalten.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Hoch	Entwicklung	Geringer Aufwand	Hoch

### PRÜFMETADATEN

Erkennungssicherheit	Hoch
Falschpositiv-Risiko	Niedrig
BFSG-/EAA-Relevanz	Niedrig
Elemente	6
Vorkommen	6
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html">https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html</a>

### MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 6 Vorkommen; erwartete Auswirkung auf den Score: hoch; WCAG-Level: A.
Komplexitätsgrund	Die Behebung ist technisch, betrifft aber nur wenige Muster.

### Empfehlung

Wenn ein Element vollständig versteckt sein soll, `display:none` oder `visibility:hidden` verwenden. Soll es nur vor Screenreadern versteckt sein, muss es unfokussierbar sein (`tabindex="-1"`).

#### ✗ Falsch

```
<button aria-hidden="true">Menu</button>
```

#### ✓ Richtig

```
<button aria-hidden="true"
tabindex="-1">Menu</button>
```

### Strukturelle Ursache

Dieses Problem tritt bei 6 Elementen auf — möglicherweise eine gemeinsam genutzte Komponente oder ein Template.

## [Hoch] Ungültige oder verbotene ARIA-Attribute — WCAG aria-prohibited-attr (A)

Für bestimmte HTML-Elemente oder ARIA-Rollen werden Attribute verwendet, die für diese Rolle nicht zulässig oder sinnvoll sind. Screenreader erhalten widersprüchliche oder fehlerhafte Signale, was die Bedienbarkeit der Website beeinträchtigen kann.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Hoch	Entwicklung	Geringer Aufwand	Hoch

### PRÜFMETADATEN

Erkennungssicherheit	Hoch
Falschpositiv-Risiko	Niedrig
BFSG-/EAA-Relevanz	Niedrig
Elemente	1
Vorkommen	1
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html">https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html</a>

### MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 1 Vorkommen; erwartete Auswirkung auf den Score: mittel; WCAG-Level: A.
Komplexitätsgrund	Die Behebung ist technisch, betrifft aber nur wenige Muster.

#### Empfehlung

Prüfe die ARIA-Spezifikation für die jeweilige Rolle. Verwende ARIA-Attribute nur auf passenden Elementen und Rollen.

#### × Falsch

```
...
```

#### ✓ Richtig

```
<button aria-label="Menu">...</button>
```

## [Mittel] Fehlende Beschriftungen und Anleitungen bei Formularen — WCAG 3.3.2 (A)



Formularfelder haben keine sichtbare Beschriftung oder Anleitung. Nutzer wissen nicht, welche Eingaben erwartet werden. Alle Nutzer, besonders Menschen mit kognitiven Einschränkungen und Screenreader-Nutzer, können Formulare nicht korrekt ausfüllen.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Entwicklung	Geringer Aufwand	Quick Win

## PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Mittel
Elemente	1
Vorkommen	1
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/labels-or-instructions.html">https://www.w3.org/WAI/WCAG21/Understanding/labels-or-instructions.html</a>

## MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 1 Vorkommen; erwartete Auswirkung auf den Score: niedrig; WCAG-Level: A.
Komplexitätsgrund	Wenige Vorkommen und ein klar abgegrenzter Fix.

Element-Typen	1× fieldset
---------------	-------------

### Empfehlung

Jedes Formularfeld mit einem sichtbaren, per `<label>` verknüpften Label versehen. Pflichtfelder kennzeichnen. Bei speziellen Formaten (z. B. Datum) das erwartete Format angeben.

#### × Falsch

```
<input type="text" placeholder="Name">
```

#### ✓ Richtig

```
<label for="name">Name *</label>
<input type="text" id="name" required
aria-required="true">
```

## Repräsentative Fundstellen

Fundstelle	Hinweis
div.column > form.c-searchbar__form > fieldset	Form group has no legend or label

## Fundstelle: div.column > form.c-searchbar\_\_form > fieldset

Node 7604

Hinweis Form group has no legend or label

HTML `<fieldset> <p> <span id="searchLabel"> <span class="formLabel aural"> <label for="f6a0983b3ca84ce5471c544fe..."`

DOM-Pfad `div.column > form.c-searchbar__form > fieldset`

Figure 2: Element auf der Seite (Desktop-Ansicht)

### Häufige Muster

Muster	Vorkommen
div.column > form.c-searchbar__form > fieldset	1

Diese Tabelle gruppiert Vorkommen nach struktureller Ähnlichkeit (gleiches Element- und Auswahlmuster) über die gesamte Seite hinweg, nicht nach exakt identischem Wert — die Zahl kann daher von der Vorkommenszahl eines einzelnen Befunds abweichen.

### [Mittel] Sichtbarer Text stimmt nicht mit dem zugänglichen Namen überein — WCAG 2.5.3 (A)

Der sichtbare Text eines Bedienelements stimmt nicht mit dem Namen überein, den assistive Technologien vorlesen. Sprachsteuerungsnutzer können das Element nicht ansprechen. Nutzer, die Sprachsteuerung verwenden, können Elemente nicht per Sprachbefehl aktivieren, weil der vorgelesene Name nicht dem sichtbaren Text entspricht.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Entwicklung	Geringer Aufwand	Quick Win

## PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Mittel
Elemente	1
Vorkommen	1
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/label-in-name.html">https://www.w3.org/WAI/WCAG21/Understanding/label-in-name.html</a>

## MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 1 Vorkommen; erwartete Auswirkung auf den Score: niedrig; WCAG-Level: A.
Komplexitätsgrund	Wenige Vorkommen und ein klar abgegrenzter Fix.

**Empfehlung**

Sicherstellen, dass der zugängliche Name (aria-label) den sichtbaren Text enthält oder identisch ist.

## × Falsch

```
<button aria-label="Close menu">X</button>
```

## ✓ Richtig

```
<button aria-label="Close">Close</button>
```

**[Mittel] Text nicht ausreichend skalierbar — WCAG 1.4.4 (AA)**

Texte auf der Website können nicht ohne Verlust von Inhalt oder Funktion auf 200% vergrößert werden. Nutzer, die auf größere Schrift angewiesen sind, verlieren dadurch Inhalte. Menschen mit Sehbeeinträchtigung, die die Schriftgröße im Browser vergrößern, stoßen auf abgeschnittene Texte, überlagerte Elemente oder nicht scrollbare Bereiche.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Mittel	Entwicklung	Mittlerer Aufwand	Quick Win

## PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Mittel
BFSG-/EAA-Relevanz	Mittel
Elemente	1
Vorkommen	1
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/resize-text.html">https://www.w3.org/WAI/WCAG21/Understanding/resize-text.html</a>

## MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 1 Vorkommen; erwartete Auswirkung auf den Score: niedrig; WCAG-Level: AA.
Komplexitätsgrund	Wenige Vorkommen und ein klar abgegrenzter Fix.
Prüfung	Manuelle Prüfung empfohlen

### Empfehlung

Schriftgrößen in relativen Einheiten (rem/em) angeben. Container so gestalten, dass sie bei vergrößertem Text mitwachsen. Browser-Zoom auf 200% testen.

#### × Falsch

```
font-size: 12px; height: 40px; overflow: hidden;
```

#### ✓ Richtig

```
font-size: 0.75rem; min-height: 2.5rem; overflow: auto;
```

## [Gering] Inhalte nicht per Tastatur bedienbar — WCAG 2.1.1 (A)

Bestimmte Funktionen der Website können nur mit der Maus bedient werden. Nutzer, die auf die Tastatur angewiesen sind, können diese Inhalte nicht erreichen oder nutzen. Menschen mit motorischen Einschränkungen, die keine Maus verwenden können, sowie Screenreader-Nutzer sind von diesen Funktionen ausgeschlossen.

Einstufung	Zuständig	Aufwand	Umsetzungspriorität
Niedrig	Entwicklung	Mittlerer Aufwand	Normal

## PRÜFMETADATEN

Erkennungssicherheit	Sehr hoch
Falschpositiv-Risiko	Sehr niedrig
BFSG-/EAA-Relevanz	Mittel
Elemente	1
Vorkommen	1
Referenz	<a href="https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html">https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html</a>

## MASSNAHMENBEWERTUNG

Erwartete Wirkung	Behebt 1 Vorkommen; erwartete Auswirkung auf den Score: niedrig; WCAG-Level: A.
Komplexitätsgrund	Die Behebung ist technisch, betrifft aber nur wenige Muster.

Element-Typen 1× body

### Empfehlung

Alle interaktiven Elemente mit nativen HTML-Elementen (button, a, input) umsetzen. Bei Custom-Widgets Tastaturnavigation (Tab, Enter, Escape, Pfeiltasten) implementieren.

#### × Falsch

```
<div onclick="doAction()">Klick mich</div>
```

#### ✓ Richtig

```
<button type="button" onclick="doAction()">Klick mich</button>
```

## Repräsentative Fundstellen

Fundstelle	Hinweis
body#Start	Focusable element without interactive role

### Fundstelle: body#Start

Node 6870

Hinweis Focusable element without interactive role

HTML `<body id="Start" tabindex="-1" class="gsb startseite lang-de fixed js-on" data-nn="6a097dd8ca84ce5471c199fb...`

Figure 3: Element auf der Seite (Desktop-Ansicht)

**Vorgeschlagene Code-Korrektur**

```
<!-- Add tabindex="0" and a keydown handler to make this element keyboard-operable --> <body id="Start" tabindex="-1" class="gsb startseite lang-de fixed js-on" data-nn="6a097dd...
```

**Häufige Muster**

Muster	Vorkommen
body#Start	1

Diese Tabelle gruppiert Vorkommen nach struktureller Ähnlichkeit (gleiches Element- und Auswahlmuster) über die gesamte Seite hinweg, nicht nach exakt identischem Wert — die Zahl kann daher von der Vorkommenszahl eines einzelnen Befunds abweichen.

# Tastatur-Accessibility-Journey

6 interaktive Befunde mit hoher Schwere gefunden.

## MEDIUM Tab-Reihenfolge

Die Tab-Reihenfolge weicht von der DOM-Reihenfolge ab: 2 rückwärtige Sprünge beobachtet. Zuerst betroffene Elemente: `div > div > div > div:nth-of-type(1) > div:nth-of-type(2) > button`, `div > div > div > ul > li:nth-of-type(2) > a`. Tastaturnutzer können dem Lesefluss möglicherweise nicht folgen. — Negative oder hohe `tabindex`-Werte vermeiden. Lese-/DOM-Reihenfolge an die visuelle Reihenfolge angleichen.

## HIGH Zustandswechsel

Der Disclosure-Button wurde geklickt, aber `aria-expanded` ändert sich nicht. Der Zustandswechsel wird Screenreadern nicht angekündigt. — `aria-expanded="truefalse"` im Klick-Handler des Buttons umschalten.

## HIGH Tab-Navigation

Die Pfeiltasten-Navigation verschiebt die Auswahl nicht zwischen den Tabs. Tastaturnutzer können die Tab-Liste nicht bedienen. — Das Roving-Tabindex-Muster implementieren: Pfeil-rechts verschiebt Fokus und `aria-selected` zum nächsten Tab.

## HIGH Tab-Navigation

Die Pfeiltasten-Navigation verschiebt die Auswahl nicht zwischen den Tabs. Tastaturnutzer können die Tab-Liste nicht bedienen. — Das Roving-Tabindex-Muster implementieren: Pfeil-rechts verschiebt Fokus und `aria-selected` zum nächsten Tab.

## HIGH Tab-Navigation

Die Pfeiltasten-Navigation verschiebt die Auswahl nicht zwischen den Tabs. Tastaturnutzer können die Tab-Liste nicht bedienen. — Das Roving-Tabindex-Muster implementieren: Pfeil-rechts verschiebt Fokus und aria-selected zum nächsten Tab.

### HIGH Tab-Navigation

Die Pfeiltasten-Navigation verschiebt die Auswahl nicht zwischen den Tabs. Tastaturnutzer können die Tab-Liste nicht bedienen. — Das Roving-Tabindex-Muster implementieren: Pfeil-rechts verschiebt Fokus und aria-selected zum nächsten Tab.

### HIGH Zustandswechsel

Der Disclosure-Button wurde geklickt, aber aria-expanded ändert sich nicht. Der Zustandswechsel wird Screenreadern nicht angekündigt. — `aria-expanded="truefalse"` im Klick-Handler des Buttons umschalten.

## Geprüfte Journey-Sequenzen

<code>tab_walk</code>	26 Schritte
<code>disclosure_0</code>	4 Schritte
<code>disclosure_1</code>	2 Schritte
<code>tabs_0</code>	3 Schritte
<code>tabs_1</code>	3 Schritte
<code>tabs_2</code>	3 Schritte
<code>tabs_3</code>	3 Schritte
<code>disclosure_2</code>	4 Schritte
<code>disclosure_3</code>	2 Schritte

Diese Tests prüfen, ob Browser, DOM, Fokus und Accessibility Tree eine robuste Grundlage für Screenreader-Nutzung liefern. Sie simulieren nicht den exakten Output von NVDA, JAWS oder VoiceOver.



# Screenreader-Lesereihenfolge

Wie die Seite Screenreader-Nutzern angekündigt wird: Lesereihenfolge, Landmark- und Heading-Qualität sowie zugängliche Namen. Strukturelle Prüfungen am Accessibility Tree, keine Simulation von NVDA, JAWS oder VoiceOver.

Heading-Qualität	Landmark-Qualität	Namens-Qualität	Angekündigte Knoten	Tab-Stopps
100 / 100	100 / 100	98 / 100	740	101

Die drei Qualitätswerte nutzen eine Skala von 0–100 (höher ist besser) und verdichten erkannte Überschriftenhierarchie, Landmark-Struktur und zugängliche Namen. Angekündigte Knoten und Tab-Stopps sind reine Anzahlen, keine Qualitätswerte.

BFSG-Prüfung: Verstöße im geprüften Umfang festgestellt (siehe Befunde unten).

## Screenreader-Befunde

Problem	Schweregrad
Langer Abschnitt ab Sequenzposition 120 ohne Landmark, Überschrift oder Fokusziel (18 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 162 ohne Landmark, Überschrift oder Fokusziel (23 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 188 ohne Landmark, Überschrift oder Fokusziel (23 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 223 ohne Landmark, Überschrift oder Fokusziel (22 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 254 ohne Landmark, Überschrift oder Fokusziel (20 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 333 ohne Landmark, Überschrift oder Fokusziel (29 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 408 ohne Landmark, Überschrift oder Fokusziel (16 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 474 ohne Landmark, Überschrift oder Fokusziel (25 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 502 ohne Landmark, Überschrift oder Fokusziel (16 Einträge).	Niedrig
Langer Abschnitt ab Sequenzposition 632 ohne Landmark, Überschrift oder Fokusziel (19 Einträge).	Niedrig

Problem	Schweregrad
101 Tab-Stops ohne erkennbaren Skip-Link erschweren Tastatur- und Screenreader-Navigation.	Mittel
Interaktive Elemente ohne zugänglichen Namen werden im Screenreader nicht verständlich angekündigt.	Hoch

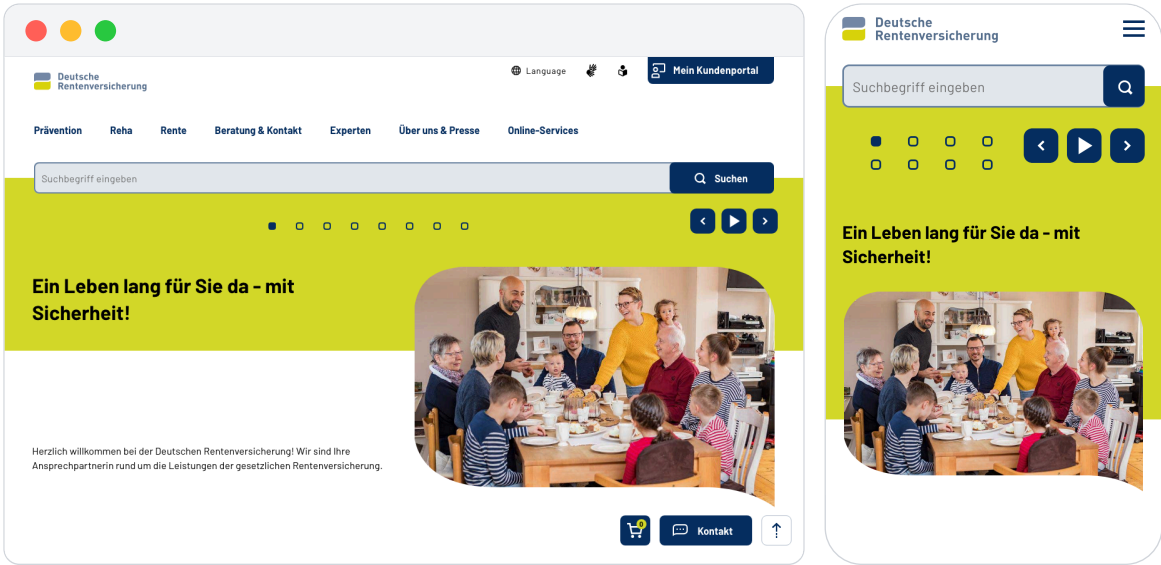
# 03

TEIL 3 VON 3

## Qualitäts-Analysen

Suchmaschinenoptimierung, Ladezeiten, mobiles Layout und Sicherheits-Metriken.

Zielgruppe: Marketing, SEO-Spezialisten und Product Owner. Dieser Teil ergänzt die Barrierefreiheit um Qualitäts-Indikatoren.



### Dual-Viewport-Zusammenfassung

Barrierefreiheit - Desktop	Barrierefreiheit - Mobile	Barrierefreiheit - Gesamt
18 / 100 Kritischer Bereich	18 / 100 Kritischer Bereich	18 / 100 70/30 gewichtet

Jeder Barrierefreiheits-Score bewertet Schwere und Vielfalt der automatisiert erkannten WCAG-Befunde seiner Ansicht; 100 bedeutet, dass im automatisierten Prüfumfang kein Problem erkannt wurde. Niedrigere Werte bedeuten höheren Handlungsdruck. WCAG-Vorkommen vor der ansichtsübergreifenden Zusammenführung: 112 auf Desktop und 119 auf Mobile. Die normalisierte gemeinsame Befundliste enthält 184 Vorkommen; diese Anzahl dient Evidenz und Maßnahmenplanung, nicht als dritter Score. Der ausgewiesene Barrierefreiheits-Gesamtwert wird mit 70 % Mobile und 30 % Desktop berechnet.

# Dark Mode

Die Seite unterstützt ein dunkles Farbschema.

Gesamtwertung · 0–100

91



Sehr gut

Status	Methoden	CSS Variablen	Print	Forced Colors
Unterstützt	1	214	Ja	Ja

## Dark Mode Übersicht

Unterstützung	Unterstützt
Implementierungsmethoden	@media (prefers-color-scheme: dark)
color-scheme CSS	Ja
CSS Custom Properties (Farben)	214
Print-Stylesheet	Ja
Interaktive Seitenelemente im Print verborgen	Nein
Potenziell geclippte Print-Elemente	0
Forced Colors	Ja
Emulation aktiv	Ja
forced-color-adjust Elemente	0
Fokus/Controls sichtbar	Ja
Kontrast-Violations im Dark Mode	0

## Farbsehschwäche-Emulation

Modus	Kontrastprobleme	Neue Probleme	Nur-Farbe-Probleme
protanopia	0	0	8
deutanopia	0	0	8
tritanopia	0	0	8
achromatopsia	0	0	8

**Dark Mode Hinweis**

Es fehlt ein `<meta name="color-scheme">`-Tag. Ohne dieses Tag kann der Browser das Rendering-Verhalten nicht optimieren, bevor das CSSOM aufgebaut ist.

**Dark Mode Hinweis**

Ein Print-Stylesheet ist vorhanden, das Druck-Layout wirkt jedoch riskant: interaktive Bedienelemente ausgeblendet = nein, potenziell abgeschnittene Elemente = 0.

## VERTRAUEN

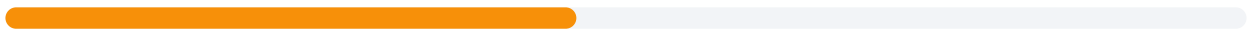
# KI & Vertrauen

Quellenqualität, KI-Lesbarkeit und Content-Sichtbarkeit — Vertrauens- und Auffindbarkeits-Indikatoren.

## | Quellenqualität

Gesamtwertung · 0–100

46



Ausbaufähig

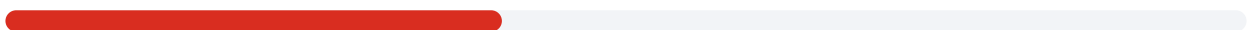
Quellqualität — Überblick: Diese Bewertung basiert ausschließlich auf technischen Signalen (Struktur, Semantik, Metadaten, Sicherheit). Sie beurteilt nicht, ob die dargestellten Inhalte inhaltlich korrekt, vollständig oder aktuell sind.

Der Score zeigt in diesem Bereich eine deutliche Schwäche. Quellenqualität zeigt, ob Inhalte substanziell, konsistent und vertrauenswürdig genug wirken, um Entscheidungen zu stützen.

## | Substanz

Schwach · 0–100

40



Ausbaufähig

Signal	Status	Detail
Bildbeschreibungen	×	7 Bilder ohne Alternativtext
Semantische Struktur	✓	Korrekte Landmark-Regionen

## Konsistenz

Uneinheitlich · 0–100

50



Ausbaufähig

Signal	Status	Detail
Benannte Bedienelemente	×	87 Elemente ohne zugänglichen Namen
Keine kritischen Fehler	✓	Keine kritischen Accessibility-Verstöße

## Autorität

Uneinheitlich · 0–100

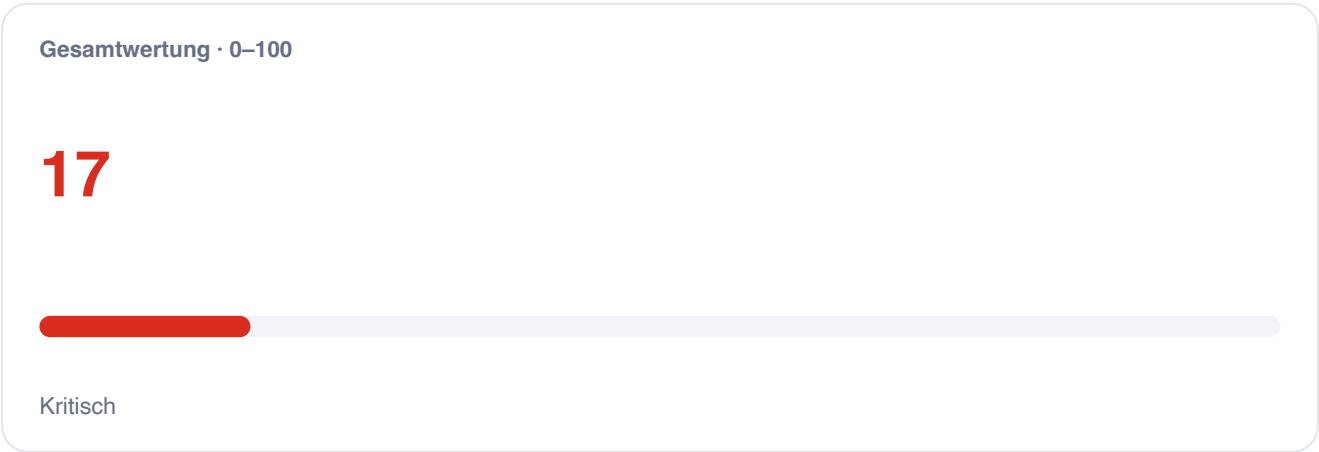
50



Ausbaufähig

Signal	Status	Detail
HTTPS	✓	Verschlüsselte Verbindung
Barrierefreiheit	×	Accessibility-Score: 16 (niedrig)

## AI-Sichtbarkeit (Indikator)



Indikator-basierte Metriken: Experimentelle Heuristiken — keine standardisierte Metrik. Die Indikatoren sind heuristische Schätzwerte auf Basis messbarer Signale, kein direktes Ranking-Signal. Sie zeigen Optimierungspotenzial, beweisen aber kein konkretes LLM-Verhalten.

KI-Sichtbarkeit — Überblick: Diese Bewertung basiert auf heuristischen Signalen zur AI-Sichtbarkeit. Sie bewertet, wie gut Inhalte für die maschinelle Extraktion und Zitierung durch KI-Systeme aufbereitet sind — nicht deren inhaltliche Qualität.

Der Score zeigt in diesem Bereich eine deutliche Schwäche. KI-Sichtbarkeit zeigt, ob Inhalte von KI-gestützten Systemen gelesen, gegliedert, zugeordnet und zitiert werden können.

## KI-Lesbarkeit

Heuristische Schätzung auf Basis struktureller Signale

Basis

~0 / 100 — Schwach

Kritisch

Signal	Status	Detail
Überschriftenstruktur	×	Keine Überschriften — LLMs können keine Abschnitte erkennen
Inhaltsumfang	×	0 Wörter — zu wenig für gehaltvolle LLM-Extraktion (empfohlen ≥ 300)
Absatzstruktur	×	Weniger als 3 Absätze — kaum Gliederung für Chunk-Extraktion



Signal	Status	Detail
Listen / Aufzählungen	×	Keine Listen — Aufzählungen verbessern LLM-Extrahierbarkeit
Tabellen	×	Keine Tabellen (nicht immer nötig)

5 weitere Signale im detaillierten Anhang.

## Zitierbarkeit

Heuristische Schätzung auf Basis struktureller Signale  
Basis

~23 / 100 — Schwach  
Kritisch

Signal	Status	Detail
Verschlüsselung	✓	HTTPS — Vertrauenssignal für Zitierwürdigkeit
Herausgeber-Identität	×	Kein Herausgeber-Markup — Autorität nicht maschinell prüfbar
Artikelstruktur	×	Kein Artikel-Schema — Content-Typ nicht maschinenlesbar
Publikationsdatum	×	Kein Publikationsdatum — Aktualität nicht einschätzbar
Kanonische URL	×	Keine Canonical-URL — Duplikate möglich

6 weitere Signale im detaillierten Anhang.

## Technische KI-Lesbarkeit

Heuristische Schätzung auf Basis struktureller Signale  
Basis

~15 / 100 — Schwach  
Kritisch

Signal	Status	Detail
Abschnittszahl	×	Nur 1 Abschnitte — zu wenig für granulare Chunk-Bildung
Heuristik: Abschnittslänge	×	Heuristik: 0 von 1 Abschnitten liegen im Bereich 100–800 Wörter (0%).

Signal	Status	Detail
		Richtwert, keine standardisierte Metrik.
Keine Übergroßen Abschnitte	✓	Kein Abschnitt über 800 Wörter — gut für Token-Limits
Wenig Fragmente	×	1 Kurzabschnitte — viele Fragmente können Kontext verlieren
Hierarchische Gliederung	×	Flache Heading-Struktur — nur sequenzielles Chunking möglich

3 weitere Signale im detaillierten Anhang.

## Strukturierte Daten

Heuristische Schätzung auf Basis struktureller Signale  
Basis

~0 / 100 — Schwach  
Kritisch

Signal	Status	Detail
Entitäten erkannt	×	0 Entitäten extrahiert — wenig maschinenlesbare Entitäten
Schema.org-Entitäten	×	0 Entitäten aus Schema.org — wenig strukturierte Entitäten
Beziehungen	×	0 Beziehungen zwischen Entitäten — isolierte Entitäten, kein Netz
Typen-Vielfalt	×	0 verschiedene Entitätstypen — wenig Typen-Diversität
Breadcrumb-Hierarchie	×	Kein Breadcrumb — thematische Einordnung fehlt

2 weitere Signale im detaillierten Anhang.

## AI-Policy

Heuristische Schätzung auf Basis struktureller Signale  
Basis

~55 / 100 — Uneinheitlich  
Ausbaufähig

Signal	Status	Detail
robots.txt erreichbar	×	robots.txt nicht erreichbar — AI-Crawler haben keine Policy-Info
Kein Wildcard-Block	✓	Kein globaler Block — Crawler haben grundsätzlich Zugang
KI-Suche erreichbar	✓	KI-Suchbots nicht blockiert — Inhalte für KI-Antworten verfügbar
Explizite AI-Policy	×	Keine robots.txt — keine AI-Policy definierbar
Sitemap-Verweis	×	Keine Sitemap in robots.txt — AI-Crawler müssen Seiten selbst entdecken

1 weitere Signale im detaillierten Anhang.

## Content-Abschnitte

### Abschnitte

Abschnitt	Level	Wörter
Gesamter Inhalt	H0	0

#### Empfehlung

Zu wenig Gliederung für effektives Chunking. Zusätzliche Zwischenüberschriften würden die Extrahierbarkeit verbessern.

**Erkannte Strukturmuster****● Semantische Hauptnavigation**

Vorhanden: 8 navigation landmarks found, but 1 have no accessible name. Screen reader users cannot distinguish them.

**● Disclosure-Menü**

Vorhanden: 2 disclosure triggers with aria-expanded (0 with aria-controls). Controls relationship strengthens screen-reader announcements.

**● Tab-Liste**

Stark: 4 tablists with 32 tabs; 32 have aria-selected, 4 declare aria-controls.

**● Accordion**

Vorhanden: 2 accordion triggers; 0 with aria-controls; 0 non-button triggers.

**● WCAG-Signal**

Vorhanden: <main> landmark detected — assistive technology users can navigate directly to main content

# Anhang & Methodik

Prüfumfang, Methodik, WCAG-Coverage und die vollständige Fundstellenliste.

## Prinzip-Abdeckung (bestandene Kriterien)

Wahrnehmbar	13/18 (72 %)
Bedienbar	23/26 (88 %)
Verständlich	8/9 (89 %)
Robust	2/3 (67 %)

## Prüfumfang

Dieses Audit prüft 36 von ca. 50 WCAG-2.1-AA-Kriterien automatisch. Die unten aufgeführten Kriterien benötigen manuelle Prüfung.

### Automatisch geprüft (36)

WCAG 1.1.1 (A) WCAG 1.2.1 (A) WCAG 1.2.2 (A) WCAG 1.2.8 (AAA) WCAG 1.3.1 (A)

WCAG 1.3.2 (A) WCAG 1.3.4 (AA) WCAG 1.3.5 (AA) WCAG 1.3.6 (AAA) WCAG 1.4.1 (A)

WCAG 1.4.3 (AA) WCAG 1.4.4 (AA) WCAG 1.4.7 (AAA) WCAG 1.4.8 (AAA) WCAG 1.4.10 (AA)

WCAG 1.4.11 (AA) WCAG 1.4.12 (AA) WCAG 1.4.13 (AA) WCAG 2.1.1 (A) WCAG 2.1.2 (A)

WCAG 2.2.1 (A) WCAG 2.2.2 (A) WCAG 2.2.3 (AAA) WCAG 2.2.4 (AAA) WCAG 2.2.5 (AAA)

WCAG 2.2.6 (AAA) WCAG 2.3.3 (AAA) WCAG 2.4.1 (A) WCAG 2.4.2 (A) WCAG 2.4.3 (A)

WCAG 2.4.4 (A) WCAG 2.4.6 (AA) WCAG 2.4.7 (AA) WCAG 2.4.8 (AAA) WCAG 2.4.9 (AAA)

WCAG 2.4.10 (AAA) WCAG 2.4.11 (AA) WCAG 2.4.12 (AAA) WCAG 2.5.1 (A) WCAG 2.5.2 (A)

WCAG 2.5.3 (A) WCAG 2.5.4 (A) WCAG 2.5.5 (AAA) WCAG 2.5.8 (AA) WCAG 3.1.1 (A)

WCAG 3.1.3 (AAA) WCAG 3.1.4 (AAA) WCAG 3.2.1 (A) WCAG 3.2.2 (A) WCAG 3.3.1 (A)

WCAG 3.3.2 (A) WCAG 3.3.5 (AAA) WCAG 3.3.7 (A) WCAG 4.1.1 (A) WCAG 4.1.2 (A)

WCAG 4.1.3 (AA)

### Manuelle Prüfung erforderlich (10)

1.2.3 (A) – Audio Description or Media Alternative 1.2.5 (AA) – Audio Description (Prerecorded)

1.4.2 (A) – Audio Control 1.4.5 (AA) – Images of Text 2.1.4 (A) – Character Key Shortcuts

2.3.1 (A) – Three Flashes or Below Threshold 3.2.3 (AA) – Consistent Navigation

## 3.2.4 (AA) – Consistent Identification

## 3.3.3 (AA) – Error Suggestion

## 3.3.4 (AA) – Error Prevention (Legal, Financial, Data)

**So testen Sie manuell**

- |                                        |                                                                                                                           |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| ● <b>Tastaturnavigation</b>            | Komplette Seite per Tab navigieren. Kein Fokus verloren, kein Keyboard-Trap, jedes interaktive Element erreichbar.        |
| <hr/>                                  |                                                                                                                           |
| ● <b>Screenreader</b>                  | Test mit NVDA/JAWS (Windows) oder VoiceOver (Mac/iOS) — Landmark-Navigation und Formular-Interaktion.                     |
| <hr/>                                  |                                                                                                                           |
| ● <b>400% Zoom</b>                     | Bei 400% Browser-Zoom: Seite ohne horizontales Scrollen bedienbar, kein Inhalt verloren.                                  |
| <hr/>                                  |                                                                                                                           |
| ● <b>Reduced Motion</b>                | Betriebssystem-Einstellung „Bewegung reduzieren“ aktivieren und prüfen, ob Animationen deaktiviert oder reduziert werden. |
| <hr/>                                  |                                                                                                                           |
| ● <b>Modal- / Dropdown-Interaktion</b> | Vollständige Tastaturbedienung: Tab, Enter, Space, Escape, Pfeiltasten. Fokus kehrt nach Schließen zum Trigger zurück.    |
| <hr/>                                  |                                                                                                                           |
| ● <b>Farbenblindheit</b>               | Mit einem Werkzeug wie „Color Oracle“ prüfen, ob Informationen nicht ausschließlich über Farbe vermittelt werden.         |

## Manuelle Prüfpunkte und heuristische Hinweise

- |                            |                                                                                                                                                                                                                                        |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ● 2.1.2 · Manuelle Prüfung | Sicherstellen, dass der Fokus jedes Element mit Tab und Shift+Tab verlassen kann. Modale Dialoge mit Escape schließbar machen und den Fokus danach korrekt zurücksetzen.                                                               |
| ● 2.2.1 · Manuelle Prüfung | Provide a mechanism to disable, adjust (at least 10x), or extend any time limit before it expires, with at least 20 seconds to respond.                                                                                                |
| ● 2.5.1 · Manuelle Prüfung | For every multipoint or path-based gesture, provide an equivalent single-pointer alternative (e.g. a button).                                                                                                                          |
| ● 2.5.2 · Manuelle Prüfung | Trigger actions on the up-event (mouseup/touchend/click) rather than the down-event (mousedown/touchstart), unless the down-event is essential to the function.                                                                        |
| ● 2.5.4 · Manuelle Prüfung | For any device motion interaction, provide equivalent button controls and allow users to disable motion actuation.                                                                                                                     |
| ● 2.1.1 · Hinweis          | Add tabindex="0" to make the element focusable, or use the native HTML element (e.g. <button>, <a>).                                                                                                                                   |
| ● 2.1.1 · Hinweis          | Add tabindex="0" to make the element focusable, or use the native HTML element (e.g. <button>, <a>).                                                                                                                                   |
| ● 2.1.1 · Hinweis          | Add tabindex="0" to make the element focusable, or use the native HTML element (e.g. <button>, <a>).                                                                                                                                   |
| ● 2.1.1 · Hinweis          | Add tabindex="0" to make the element focusable, or use the native HTML element (e.g. <button>, <a>).                                                                                                                                   |
| ● 1.3.2 · Hinweis          | If the order carries meaning, reorder the elements in the DOM/source instead of using CSS `order`, or add `aria-owns` on the container listing the children in their visual order so assistive technology announces the same sequence. |
| ● 1.3.4 · Hinweis          | Remove orientation-specific CSS that hides or rotates content. Let the OS and browser handle orientation changes. Only lock orientation if it is essential to the content (e.g. a piano keyboard).                                     |

Interaktive Abdeckung: 9 von 9 versuchten Journeys abgeschlossen; 0 fehlgeschlagen, 0 übersprungen.

---

Grenzen automatisierter Tests: Automatisierte Tests können ca. 30–40% aller Barrierefreiheitsprobleme erkennen. Komplexe Aspekte wie korrekte Tab-Reihenfolge, sinnvolle Alt-Texte oder verständliche Sprache erfordern zusätzlich manuelle Prüfung. Dieser Lauf ist unvollständig: Die Scores beschreiben nur den erfolgreich gemessenen Umfang.

Hinweis: Dieser Report stellt eine automatisierte technische Analyse dar. Er ersetzt keine vollständige Konformitätsbewertung nach WCAG 2.1. Für eine rechtsverbindliche Aussage zur Barrierefreiheit ist eine umfassende manuelle Prüfung durch Experten erforderlich.



## ANHANG

# Vollständige Fundstellen

Rohdaten des Audits und die vollständige Liste aller erkannten Verstöße.

## Audit-Daten

Bereich	Signal	Wert
Audit	Barrierefreiheits-Score	18 / 100 — 70 % Mobile (18) + 30 % Desktop (18)
Audit	Gesamtscore	18 / 100 — beitragende Gewichtung: Barrierefreiheit 100 % Separat ausgewiesene Indikator-Module: Dark Mode, KI-Sichtbarkeit, Quellenqualität.
Audit	Zählweise	17 unterschiedliche WCAG-Befundgruppen, 184 WCAG-Vorkommen; 17 Befundzeilen und 184 Vorkommen über alle Kategorien
Audit	WCAG-Level	AA
Audit	Geprüfte Knoten	11774
Audit	Laufzeit	8.1 s
Audit	Aktive Module	accessibility, accessibility_journey, ai_visibility, dark_mode, source_quality
Audit	Fehlgeschlagene Module	Keine
Audit	Audit-Qualität	Partial
Audit	Audit-Hinweise	1
Audit	Vorschau	Desktop und Mobile erfasst
Modul	Barrierefreiheit	18 / 100 — Kritisch — wesentliche Anforderungen an die Barrierefreiheit werden nicht erfüllt.. 8 Problemgruppe(n) mit hoher Auswirkung
Finding	a11y.name_role.missing (4.1.2)	80 Vorkommen — Screenreader-Nutzer hören z. B. nur 'Button' ohne Beschreibung der Funktion, oder ein klickbares Element wird gar nicht als interaktiv erkannt.
Finding	a11y.alt_text.missing (1.1.1)	7 Vorkommen — Menschen mit Sehbeeinträchtigung erhalten an diesen Stellen keine oder nur unvollständige Information.
Finding	a11y.aria_hidden_focus.invalid (aria-hidden-focus)	6 Vorkommen — Tastaturnutzer steuern auf unsichtbare Elemente (Fokus-Stolperfallen), während Screenreader-Nutzer keinerlei Rückmeldung über dieses Element erhalten.
Finding	a11y.aria_prohibited_attr.invalid (aria-prohibited-attr)	1 Vorkommen — Screenreader erhalten widersprüchliche oder fehlerhafte Signale, was die Bedienbarkeit der Website beeinträchtigen kann.
Finding	a11y.presentation_semantic_children_invalid (1.3.1)	45 Vorkommen — Screenreader können Navigationsmenüs nicht korrekt interpretieren.

### **Rohdaten & Weiterverarbeitung**

Der begleitende JSON-Report enthält die vollständige maschinenlesbare Fehlerliste mit Selektoren, Vorkommen und allen Detaildaten für automatisierte Weiterverarbeitung.

## Alle Verstöße (aggregiert nach Regel)

- **4.1.2 — Fehlende Name/Rolle bei Bedienelementen** Button is missing accessible text

---

- **1.3.1 — Semantische Kinder in presentational Container** Element with role="presentation"/"none" contains semantic child tab (button#slider-\_2ix2axxlv-dots-0)

---

- **4.1.2 — ARIA-Attribut für diese Rolle nicht zulässig** ARIA attribute 'aria-labelledby--' is not allowed on role 'tabpanel'

---

- **4.1.2 — Ungültiger ARIA-Attribut-name** Element has invalid ARIA attribute: 'aria-labelledby--'

---

- **1.4.13 — Unzugängliche Hover- oder Fokus-Inhalte (Tooltips)** Interactive element uses the native `title` attribute as its only descriptive text (a). Tooltips from `title` are not keyboard-accessible and unreliable on touch devices.

---

- **1.1.1 — Fehlende Alternativtexte bei Bildern** Image is missing alternative text

---

- **1.4.1 — Link nur durch Farbe unterscheidbar** Inline link distinguishable from surrounding text by color alone (a). Users with low vision or color blindness may not recognize it as a link.

---

- **4.1.2 — Ungültiger Wert eines ARIA-Attributs** 'aria-describedby' has invalid value "f6a0983b3ca84ce5471c544fed6a0983beca84ce5471c54be2-error ", expected: existing DOM id: f6a0983b3ca84ce5471c544fed6a0983beca84ce5471c54be2-error not found

---

- **aria-hidden-focus — Fokussierbares Element versteckt vor Screenreadern** Focusable element 'button#\_xuqp1gksg' is inside an aria-hidden subtree — keyboard users reach it but screen readers cannot announce it

---

- **2.5.8 — Unzureichende Klickzielgröße** Interactive target 'Ein Leben lang für Sie da - mit Sicherheit!' is 12x12 CSS pixels, below the 24x24 minimum.

---

- **4.1.2 — Fehlender Name bei Befehlselement** Element with role="button" has no accessible name

---

- **aria-prohibited-attr — Verbotene ARIA-Attribute** ARIA attributes 'aria-labelledby' cannot be used on a generic element without a valid role