INKLUSO · ACCESSIBILITY AUDIT

Website Accessibility Report

`https://www.example-shop.example`

2 September 2026

20 issue instances across 6 distinct rules, found on 3 scanned pages

D

51.1/100

CRITICAL — **Images without alternative text**
`image-alt · 4×`

CRITICAL — **Form fields without a label**
`label · 2×`

SERIOUS — **Insufficient colour contrast**
`color-contrast · 8×`

Legal framework: Directive (EU) 2019/882 (European Accessibility Act)
Supervisory authority: Product market surveillance per Arts. 19–22; services checked by the authorities responsible for compliance of services per Art. 23, each designated by the relevant EU/EEA member state
Standard: EN 301 549 v3.2.1 (WCAG 2.1 Level AA)
Test scope: WCAG 2.1 A/AA + WCAG 2.2 (target size) + best-practice checks

SUMMARY

The website https://www.example-shop.example received a grade of D (51.1/100). Across 3 scanned pages we found 20 accessibility violations (6 distinct rules), including 6 critical and 9 serious instances, plus 5 moderate and 0 minor. We recommend starting with the critical and serious issues.

SEVERITY OVERVIEW

SEVERITY	RULES	INSTANCES
CRITICAL	2	6
SERIOUS	2	9
MODERATE	2	5
MINOR	0	0

SCANNED PAGES

Fully tested: 3 of 3 pages in this audit. Colour contrast: 23 of 25 checks auto-verified, 2 remain for manual review.

#	URL	STATUS
1	https://www.example-shop.example/	OK
2	https://www.example-shop.example/contact	OK
3	https://www.example-shop.example/product/xr200	OK

KEYBOARD & FOCUS CHECK

Pages walked with the Tab key: 2 of 2. Items to review: 3.

These findings do not affect the score above and are not counted as automated violations; confirm them manually before reporting them as defects.

Page 1: https://www.example-shop.example/

On this page we made 41 Tab key presses across 27 interactive elements. This is a heuristic check, not an automated rule pass like the rules above; treat it as a starting point for manual testing, not a final verdict.

No visible focus indicator WCAG 2.4.7 Focus Visible

Related requirement: EN 301 549 9.2.4.7, EAA EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.2.4.7

What it is: A focused element showed no visible outline, box-shadow, or underline to indicate it currently has keyboard focus.

Who it affects: Keyboard-only users, who rely entirely on a visible focus indicator to know where they are on the page; without one, keyboard navigation becomes a guessing game.

Why it matters: WCAG 2.4.7 (Focus Visible) requires that any keyboard-operable interface has a mode where the focus indicator is visible.

> **How to fix:** Restore a visible focus style, most commonly lost to a CSS reset that removed the default outline (outline: none) without providing a replacement. A visible outline, box-shadow, or underline on :focus-visible is enough; it does not need to match the browser default.

Caveat: This check looks only for an outline, box-shadow, or underline change; a focus style implemented purely as a background-color or border-color change is currently invisible to it and may be misreported as missing. Confirm visually before treating as a defect.

Effort: CSS-only change

Focus got stuck on one element WCAG 2.1.2 No Keyboard Trap

Related requirement: EN 301 549 9.2.1.2, EAA EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.2.1.2

What it is: The same interactive element kept keyboard focus across multiple consecutive Tab presses instead of releasing it to the next element.

Who it affects: Anyone navigating by keyboard alone or with a screen reader; once focus cannot move past this element, the rest of the page becomes unreachable.

Why it matters: WCAG 2.1.2 (No Keyboard Trap) requires that focus can always be moved away from any component using the keyboard alone.

> **How to fix:** Check the element's own key event handlers for code that intercepts Tab or re-applies focus to itself. If it is a custom widget (menu, date picker, carousel), confirm it releases focus once its own internal navigation is exhausted.

Caveat: Some widgets legitimately hold focus within themselves while active, such as an open modal dialog or a listbox. Confirm manually whether this is a genuine trap or expected behavior before treating it as a defect.

Effort: one-line code change

Recorded focus order

#	ELEMENT	TEXT OR LABEL	VISIBLE FOCUS
1	link	Přeskočit na obsah	yes

#	ELEMENT	TEXT OR LABEL	VISIBLE FOCUS
2	link	Váš e-shop	yes
3	field	no label	yes
4	link	Elektronika	yes
5	link	Móda	yes
6	link	Domácnost	yes
7	link	Sport	yes
8	link	Zahrada	yes
9	link	Košík (0)	yes
10	link	Můj účet	yes
11	link	Bezdrátová sluchátka XR200	yes
12	button	Přidat do košíku	no
13	link	Chytré hodinky Pulse	yes
14	button	Přidat do košíku	yes
15	link	Reproduktor Boom Mini	yes
16	button	Přidat do košíku	yes
17	link	Powerbanka Volt 10000	yes
18	button	Přidat do košíku	yes
19	link	USB-C nabíječka 30W	yes
20	button	Přidat do košíku	yes
21	button	+	yes
22	button	+	yes
23	field	no label	yes
24	button	Odebírat novinky	yes
25	link	Obchodní podmínky	yes
26	link	Ochrana osobních údajů	yes
27	link	Kontakt	yes

Page 2: https://www.example-shop.example/product/xr200

On this page we made 31 Tab key presses across 19 interactive elements. This is a heuristic check, not an automated rule pass like the rules above; treat it as a starting point for manual testing, not a final verdict.

Focus loops without reaching most of the page WCAG 2.1.2 No Keyboard Trap

Related requirement: EN 301 549 9.2.1.2, EAA EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.2.1.2

What it is: Keyboard focus kept cycling through a small group of elements without ever reaching most of the page's other interactive elements.

Who it affects: Keyboard-only and screen-reader users trying to reach any content or control outside the cycling group, most commonly a stuck modal, menu, or widget.

Why it matters: WCAG 2.1.2 (No Keyboard Trap) requires that keyboard focus is never confined to a subset of the page.

How to fix: Identify the component the cycle stays within and check its focus-trap implementation, common in modals and menus: it should release focus back to the page, or close on Escape and return focus to its trigger, rather than looping indefinitely.

Caveat: An intentionally modal dialog that traps focus while open is correct behavior; the defect is failing to release that trap when the dialog closes, not having one in the first place.

Effort: template change

Recorded focus order

#	ELEMENT	TEXT OR LABEL	VISIBLE FOCUS
1	link	Přeskočit na obsah	yes
2	link	Váš e-shop	yes
3	link	Domů	yes
4	link	Elektronika	yes
5	button	Zvětšit obrázek produktu	yes
6	button	Zavřít	yes
7	button	Další	yes
8	button	Předchozí	yes
9	button	Zavřít	yes
10	button	Další	yes
11	button	Předchozí	yes
12	button	Zavřít	yes
13	button	Další	yes
14	button	Předchozí	yes
15	button	Zavřít	yes
16	button	Další	yes
17	button	Předchozí	yes
18	button	Zavřít	yes
19	button	Další	yes

DETAILED ISSUE LIST

Critical: 2 rules, 6 instances

RULE	WCAG	EN 301 549	LEGAL BASIS	INSTANCES	PAGE
image-alt Images without alternative text	1.1.1 Non-text Content	9.1.1.1	EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.1.1.1	4×	https://www.example-shop.example/
label Form fields without a label	4.1.2 Name, Role, Value	9.4.1.2	EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.4.1.2	2×	https://www.example-shop.example/contact

Serious: 2 rules, 9 instances

RULE	WCAG	EN 301 549	LEGAL BASIS	INSTANCES	PAGE
color-contrast Insufficient colour contrast	1.4.3 Contrast (Minimum)	9.1.4.3	EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.1.4.3	8×	https://www.example-shop.example/
html-has-lang Page language not declared	3.1.1 Language of Page	9.3.1.1	EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.3.1.1	1×	https://www.example-shop.example/

Moderate: 2 rules, 5 instances

RULE	WCAG	EN 301 549	LEGAL BASIS	INSTANCES	PAGE
target-size Touch targets smaller than 24×24 pixels	2.5.8 Target Size (Minimum): WCAG 2.2, not in EN 301 549 v3.2.1			3×	https://www.example-shop.example/

RULE	WCAG	EN 301 549	LEGAL BASIS	INSTANCES	PAGE
heading-order Heading levels skip	1.3.1 Info and Relationships			2×	`https://www.example-shop.example/contact`

DETAILED FINDINGS & FIXES

Automated first-line guidance, not a legal opinion and not a guarantee of conformity. Maps to WCAG 2.1 AA / EN 301 549; national-law references are pointers, not advice.

Images without alternative text image-alt · 4×

What it is: Informative images have no alt attribute describing their content or purpose.

Who it affects: Blind and low-vision screen-reader users hear only ‘graphic’ or the file name; the image content is lost.

Why it matters: Product photos, infographics, logos and image links become inaccessible. 1.1.1 is a WCAG Level-A baseline and therefore in scope under the EAA.

> **How to fix:** Give every informative image a short, purpose-describing alt (don’t start with ‘image of’). Decorative images get an EMPTY alt="" so the screen reader skips them. Images acting as a link/button describe the action, not the picture.

Caveat: alt="" on purely decorative images is CORRECT, not a failure. CSS background images need no alt, but then can’t convey information.

Effort: a few minutes per instance · WCAG 1.1.1 Non-text Content · EAA / EN 301 549 9.1.1.1: text alternatives for non-text content

Affected elements:

- `img:nth-of-type(1)` · <img class='c0'>…</img>
- `img:nth-of-type(2)` · <img class='c1'>…</img>
- `img:nth-of-type(3)` · <img class='c2'>…</img>
- `img:nth-of-type(4)` · <img class='c3'>…</img>

Form fields without a label label · 2×

What it is: Inputs, selects or textareas have no programmatically associated label.

Who it affects: Screen-reader users don't know what a field expects. A linked label also enlarges the click target, helpful for motor impairments.

Why it matters: Search, login, newsletter and especially checkout become unusable. A placeholder is not a label: it disappears on typing and is usually low-contrast.

> **How to fix:** Associate a <label for="id"> (visible is best). Where no visible text fits (icon search), use aria-label. Never rely on placeholder alone.

Caveat: A visible <label> is ideal; aria-label is acceptable when there is intentionally no visible text.

Effort: a few minutes per instance · WCAG 4.1.2 Name, Role, Value · EAA / EN 301 549 9.4.1.2: name, role and state programmatically determinable

Affected elements:

- `input:nth-of-type(1)` · <input class='c0'>…</input>
- `input:nth-of-type(2)` · <input class='c1'>…</input>

Insufficient colour contrast color-contrast · 8×

What it is: Text colour and background are too close; the ratio is below 4.5:1 (or 3:1 for large text).

Who it affects: People with low vision, older users, and anyone in poor conditions (sun, small screen, fatigue).

Why it matters: Light-grey text, thin type on colour, and placeholders are common causes. This is by far the most frequent finding on the web (WebAIM).

> **How to fix:** Raise contrast to ≥ 4.5:1 (≥ 3:1 for large text ≥ 24px, or ≥ 18.66px bold). Often a single design-token change fixes dozens of instances.

Caveat: Disabled controls and pure decoration are exempt. Large/bold text only needs 3:1. Logos are exempt.

Effort: CSS-only change · WCAG 1.4.3 Contrast (Minimum) · EAA / EN 301 549 9.1.4.3: minimum contrast of text

Affected elements:

- `p:nth-of-type(1)` · …
- `p:nth-of-type(2)` · …
- `p:nth-of-type(3)` · …
- `p:nth-of-type(4)` · …
- `p:nth-of-type(5)` · …
- … and further instances

Page language not declared html-has-lang · 1×

What it is: The <html> element has no (or an invalid) lang attribute.

Who it affects: Screen readers pick the pronunciation engine from lang. Missing or wrong, German text gets read with an English voice, barely intelligible.

Why it matters: It drives pronunciation and machine translation. A one-line attribute with outsized impact.

> **How to fix:** Set <html lang="de">. Mark sections in another language with lang on that element.

Caveat: lang MUST match the actual content language; don't serve lang="en" on a German page (a common international-template mistake).

Effort: one-line code change · WCAG 3.1.1 Language of Page · EAA / EN 301 549 9.3.1.1: language of the page

Affected elements:

- `html:nth-of-type(1)` · <html class='c0'>…</html>

Heading levels skip heading-order · 2×

What it is: Headings jump levels, e.g. an <h2> directly followed by an <h4>, breaking the document outline.

Who it affects: Screen-reader users navigating by headings lose the structure; the page reads as if content went missing.

Why it matters: A heading level that skips (h2 to h4) breaks the outline screen-reader users navigate by; the visual hierarchy should be mirrored by the heading levels themselves.

> **How to fix:** Renumber headings so each level nests under the previous one. Visual size comes from CSS, never from the heading level.

Caveat: Skipping DOWN when closing sections (h3 back to h2) is fine. Only skipping down INTO deeper levels mid-flow is the failure.

Effort: template change · WCAG 1.3.1 Info and Relationships · EN 301 549: skipped heading levels

Affected elements:

- `h4:nth-of-type(1)` · <h4 class='c0'>…</h4>
- `h4:nth-of-type(2)` · <h4 class='c1'>…</h4>

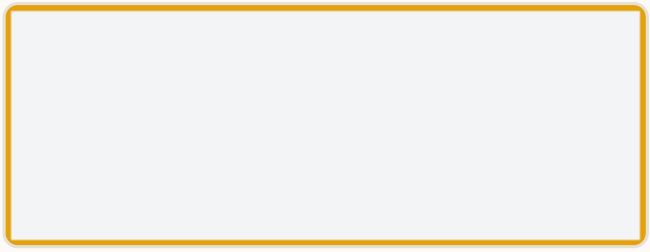
target-size target-size · 3×

Affected elements:

- `button:nth-of-type(1)` · <button class='c0'>…</button>
- `button:nth-of-type(2)` · <button class='c1'>…</button>
- `button:nth-of-type(3)` · <button class='c2'>…</button>

VISUAL EVIDENCE

image-alt on https://www.example-shop.example/ · element img:nth-of-type(1)



color-contrast on https://www.example-shop.example/ · element p:nth-of-type(1)

DRAFT ACCESSIBILITY STATEMENT

Accessibility statement (draft)

This is an automatically generated DRAFT accessibility statement for https://www.example-shop.example, based on an automated test. It does not constitute confirmation of full legal conformance.

Conformance status: Partially conforms to WCAG 2.1 AA / EN 301 549. The content listed below does not meet the requirements.

Non-accessible content:

- Images without alternative text, 4× (violates: EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.1.1.1).
- Form fields without a label, 2× (violates: EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.4.1.2).
- Insufficient colour contrast, 8× (violates: EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.1.4.3).
- Page language not declared, 1× (violates: EAA Annex I Section III(c) (Directive (EU) 2019/882); EN 301 549 §9.3.1.1).
- Touch targets smaller than 24×24 pixels, 3× (violates: -).
- Heading levels skip, 2× (violates: -).

Feedback and contact: Report accessibility issues with this website to [your organisation's email address].

Date prepared: 2 September 2026

Assessment method: Automated testing with axe-core. Note: automated tools detect roughly 30–40% of WCAG criteria; a full assessment requires a manual audit. Axe-core® is a trademark of Deque Systems, Inc. in the US and other countries.

DISCLAIMER & SCAN EVIDENCE

Disclaimer: This document is an automated technical record produced by axe-core; it is not legal advice or a guarantee of conformity. Automated testing typically detects 30–40% of WCAG requirements; a full conformity assessment requires a manual audit. Apply the

suggested fixes only after your own review. Never deploy them automatically. Axe-core® is a trademark of Deque Systems, Inc. in the US and other countries.

Generated	2026-09-02 12:20 UTC
Scan engine	axe-core® v4.12.1 (Deque Systems, Inc.)
Evidence hash (SHA-256)	`dcbe53e93106ce8e7e5f8cd88bcec0d74c917b4d9553a409c0fba39fe1fc0b22`

This report was generated on 2 September 2026. Automated evidence, not a legal opinion.
For an up-to-date accessibility status we recommend regular re-scans.
Inkluso (inkluso.eu)