



ACCESSIBILITY MONITORING

Accessibility Monitoring Report

UK garden & outdoor retailer (anonymised)

CLIENT	SITE	SCAN DATE
UK garden & outdoor retailer (anonymised)	client-site.example	31 July 2026
PAGES SCANNED	STANDARD ASSESSED	REPORT
25	WCAG 2.2 Level AA	Automated monitoring & manual review scope

Monitoring evidence prepared by Letrix Labs for UK garden & outdoor retailer (anonymised). Automated findings in Section A, manual review scope in Section B, methodology and audit trail in Section C.

What we checked, and what we found

A plain-English summary of the automated accessibility monitoring scan of client-site.example, carried out on 31 July 2026.

25 pages across **client-site.example** were scanned, running the automated rules that map to WCAG 2.2 Level AA. The scan recorded **505 instances** of **12 issue types** where an automated check failed, after de-duplicating repeated components across pages. A further **1,196 elements** could not be decided automatically and are flagged for human review.

PAGES SCANNED	ISSUE TYPES FOUND	TOTAL INSTANCES	FLAGGED FOR HUMAN REVIEW
25 1 homepage, 1 cart, 1 search, 9 collections, 5 products, 8 content	12 across 7 success criteria	505 208 distinct elements	1,196 6 check types undecided

FAILED INSTANCES BY SEVERITY

Critical		84
Serious		421

TOP 3 ISSUE CATEGORIES

WHAT IT IS, IN PLAIN ENGLISH	INSTANCES
1 Low colour contrast Text or interface elements that do not stand out enough from what is behind them. Anyone reading in bright light, on an older screen, or with reduced vision has to work harder to read them — and some will not be able to read them at all.	219
2 Accessibility markup used incorrectly ARIA attributes — extra code that tells assistive technology what a component is — used in a way that is invalid or contradictory. Incorrect ARIA is often worse than none, because it actively misdescribes the page.	81
3 Keyboard focus problems Parts of the page where keyboard focus goes somewhere it should not — into panels that are meant to be hidden, or into regions that cannot be operated without a mouse. Someone tabbing through the page lands on things they cannot see, and a screen reader announces nothing when they get there.	67

WHAT THIS MEANS

Most of what the scan found is repetition rather than variety: 12 distinct problems account for all 505 instances, which usually means a handful of shared templates or theme components are responsible. Fixing them at component level tends to clear large numbers of instances at once — the single highest-priority item, *text is too low in contrast against its background*, accounts for 219 of them on its own.

The practical next step is to work down Section A in the order given — it is sorted so that the highest-impact, highest-volume issues come first — and to plan the manual review set out in Section B. The two are complementary: the automated findings tell you what is definitely wrong in the markup, and the manual review covers the criteria that depend on human judgment, which is the larger share of WCAG.

SCOPE OF AUTOMATED TESTING

Automated testing can detect only a subset of accessibility barriers. Criteria requiring human judgment are listed in the Human Review Checklist. This report documents monitoring evidence; it is not a certification of conformance.

Automated monitoring evidence

Findings grouped by issue type rather than by page, so a single theme component that repeats across templates is one entry to fix, not fifty.

12 issue types recorded 505 failing instances across 208 distinct elements. Entries are ordered by **priority score** — severity weight × number of instances — so the items that affect the most people in the most serious way appear first. The weights are documented in Section C.

INSTANCES BY ISSUE TYPE (TOP 8)

Text is too low in contrast against its background		219 serious
Hidden content can still be reached by keyboard		63 serious
Images have no alternative text		30 critical
Links have no readable text		36 serious
ARIA attributes are not allowed on this element's role		26 critical
Lists contain elements that are not list items		34 serious
ARIA components are missing their required parent element		25 critical
Tap targets are too small or too close together		31 serious

FINDINGS IN DETAIL — 8 HIGHEST-PRIORITY OF 12

1

Text is too low in contrast against its background

● SERIOUS

Rule: color-contrast

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
1.4.3 Contrast (Minimum) (Level AA)	219	25 of 25 pages	657

WHO THIS AFFECTS

Anyone with reduced vision, colour vision deficiency, or simply reading on a phone in daylight may not be able to make out the text at all.

EXAMPLE ELEMENTS (3 OF 99 DISTINCT)

.cart-items_wrapper > p > a · client-site.example/cart

Log in

Element has insufficient color contrast of 3.11 (font size: 10.5pt (14px), font weight: normal). Expected contrast ratio of 4.5:1

.text-block--AeWtaV1lTb001RXZCR__text_UFxiYb > .subHeading.h5 > h5 · client-site.example

```
<h5>For Life on The Go</h5>
```

Element has insufficient color contrast of 3.11 (font size: 13.5pt (18px), font weight: bold). Expected contrast ratio of 4.5:1

.text-block--AdGxyZXQ3eVZWZ1MrM__text_VDAcRa > p · client-site.example

```
<p>At term-3 term-4, we're passionate about...</p>
```

Element has insufficient color contrast of 3.11 (font size: 10.5pt (14px), font weight: normal). Expected contrast ratio of 4.5:1

RECOMMENDED FIX

Darken the text or lighten the background until the contrast ratio reaches at least 4.5:1 for body text, or 3:1 for text at 18pt/14pt-bold and larger. Fix this in the theme's colour tokens rather than element by element, so it holds across the whole site.

BEFORE — FROM THIS SCAN

```
/* .cart-items__wrapper > p > a */  
/* measured 3.11:1 at 10.5pt (14px), normal  
weight */  
/* below the 4.5:1 minimum */
```

AFTER

```
/* Darken the background or lighten the text  
until this  
same pair measures at least 4.5:1.  
Change it in the theme's colour tokens, not  
on this  
element, so every use of the pair moves  
together. */
```

2 Hidden content can still be reached by keyboard

● SERIOUS

Rule: aria-hidden-focus

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
4.1.2 Name, Role, Value (Level A)	63	18 of 25 pages	189

WHO THIS AFFECTS

An element is hidden from screen readers but still in the tab order, so a keyboard user tabs into something they cannot see and a screen reader user tabs to silence — typically an off-canvas menu that was never fully closed.

EXAMPLE ELEMENTS (3 OF 19 DISTINCT)

.resource-list__slide:nth-child(5) · client-site.example/cart

```
<slideshow-slide ref="slides[]" aria-hidden="true" style="view-timeline-name: --slide-4; --  
product-media-fit: cover;" class="resource-list__slide">
```

Focusable content should be disabled or be removed from the DOM · Focusable content should have tabindex="-1" or be removed from the DOM

#section--0007 > .section-resource-list.section--page-width.section > .term-4-full-width.resource-list[data-testid="resource-list-grid"] > ... · client-site.example

```
<slideshow-slide ref="slides[]" aria-hidden="true" style="view-timeline-name: --slide-4; --  
product-media-fit: cover;" class="resource-list__slide">
```

Focusable content should be disabled or be removed from the DOM · Focusable content should have tabindex="-1" or be removed from the DOM

```
#section--0007 > .section-resource-list.section--page-width.section > .term-4-full-width.resource-list[data-testid="resource-list-grid"] > ... · client-site.example
```

```
<slideshow-slide ref="slides[]" aria-hidden="true" style="view-timeline-name: --slide-5; --product-media-fit: cover;" class="resource-list__slide">
```

Focusable content should be disabled or be removed from the DOM · Focusable content should have `tabindex="-1"` or be removed from the DOM

RECOMMENDED FIX

When hiding a region with `aria-hidden="true"`, also remove its contents from the tab order — use `inert` on the container, or `display: none` / `visibility: hidden`, rather than hiding it visually with opacity or off-screen positioning.

BEFORE — FROM THIS SCAN

```
<slideshow-slide ref="slides[]"
aria-hidden="true" style="view-timeline-name:
--slide-4; --product-media-fit: cover;"
class="resource-list__slide">
```

AFTER

```
<!-- inert takes the whole subtree out of the
tab order -->
<slideshow-slide ref="slides[]"
aria-hidden="true" style="view-timeline-name:
--slide-4; --product-media-fit: cover;"
class="resource-list__slide" inert>
```

3 Images have no alternative text

● CRITICAL

Rule: image-alt

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
1.1.1 Non-text Content (Level A)	30	5 of 25 pages	120

WHO THIS AFFECTS

Screen reader users hear only a file name, or nothing at all, in place of the picture — so any product, promotion or instruction carried by that image is invisible to them.

EXAMPLE ELEMENTS (3 OF 30 DISTINCT)

```
#-spin-studio-large-thumb · client-site.example/products/product-1
```

```

```

Element does not have an alt attribute · aria-label attribute does not exist or is empty (+3 more)

```
.reel-cache > img:nth-child(1) · client-site.example/products/product-1
```

```

```

Element does not have an alt attribute · aria-label attribute does not exist or is empty (+3 more)

```
.reel-cache > img:nth-child(2) · client-site.example/products/product-1
```

```

```

Element does not have an alt attribute · aria-label attribute does not exist or is empty (+3 more)

RECOMMENDED FIX

Add an `alt` attribute describing the image's purpose. If the image is purely decorative and repeats adjacent text, use `alt=""` so screen readers skip it — an empty alt is a decision, a missing alt is a defect.

BEFORE — FROM THIS SCAN

```

```

AFTER

```
<!-- or alt="" if the image is decorative -->

```

4 Links have no readable text

● SERIOUS

Rule: link-name

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
2.4.4 Link Purpose (In Context) (Level A)	36	25 of 25 pages	108
4.1.2 Name, Role, Value (Level A)			

WHO THIS AFFECTS

Screen reader users hear only "link" with no destination, and voice-control users have no name to speak in order to activate it — most often this is an icon-only or image-only link.

EXAMPLE ELEMENTS (3 OF 7 DISTINCT)

.size-style · client-site.example/search?q=test

```
<a href="/" class="size-style spacing-s..." style="--size-style-wi...">
```

Element is in tab order and does not have accessible text · Element does not have text that is visible to screen readers (+3 more)

.header-logo · client-site.example

```
<a href="/" class="size-style spacing-s..." style="--size-style-wi...">
```

Element is in tab order and does not have accessible text · Element does not have text that is visible to screen readers (+3 more)

.group-block__link[href\$="shipping-policy"] · client-site.example/products/product-1

```
<a href="/policies/shipping-policy" class="group-block__link"></a>
```

Element is in tab order and does not have accessible text · Element does not have text that is visible to screen readers (+3 more)

RECOMMENDED FIX

Give the link visible text, or — where the design calls for an icon alone — an `aria-label`. If the link wraps an image, the image's `alt` text becomes the link's name, so describe the destination rather than the picture.

BEFORE — FROM THIS SCAN

```
<a href="/" class="size-style spacing-s..."
style="--size-style-wi...">
```

AFTER

```
<a href="/" class="size-style spacing-s..."
style="--size-style-wi..."
aria-label="Describe where this link goes">
```

5 ARIA attributes are not allowed on this element's role

● CRITICAL

Rule: aria-allowed-attr

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
4.1.2 Name, Role, Value (Level A)	26	6 of 25 pages	104

WHO THIS AFFECTS

Assistive technology ignores or misreports the attribute, so the component behaves differently from how it looks — on a slideshow, the visitor is never told which slide is currently showing.

EXAMPLE ELEMENTS (3 OF 16 DISTINCT)

button[aria-label="Slide 1 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-1"
aria-label="Slide 1 of 8" on:click="/select/0" ref="dots[]" aria-selected="true"> 1 </button>
```

ARIA attribute is not allowed: aria-selected="true"

button[aria-label="Slide 2 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-2"
aria-label="Slide 2 of 8" on:click="/select/1" ref="dots[]" aria-selected="false"> 2 </button>
```

ARIA attribute is not allowed: aria-selected="false"

button[aria-label="Slide 3 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-3"
aria-label="Slide 3 of 8" on:click="/select/2" ref="dots[]" aria-selected="false"> 3 </button>
```

ARIA attribute is not allowed: aria-selected="false"

RECOMMENDED FIX

The usual cause is `aria-selected` on slideshow or carousel dot buttons. That attribute is only valid on a `tab`, `option`, `row` or `gridcell`, and a plain `<button>` is none of those, so it is discarded and the current slide is never announced. Either give the dots the role the attribute belongs to — `role="tab"` on each dot, inside a `role="tablist"` container — or, where the dots are not a real tab set, drop `aria-selected` in favour of `aria-current`, which is valid on any element and needs no surrounding structure. The second option is usually the smaller change.

■ BEFORE — FROM THIS SCAN

```
<button class="slideshow-control button
button-unstyled" style="animation-timeline: --
slide-1" aria-label="Slide 1 of 8"
on:click="/select/0" ref="dots[]"
aria-selected="true"> 1 </button>
```

■ AFTER

```
<!-- aria-current is valid here; aria-selected
needs role="tab" -->
<button class="slideshow-control button
button-unstyled" style="animation-timeline: --
slide-1" aria-label="Slide 1 of 8"
on:click="/select/0" ref="dots[]"
aria-current="true"> 1 </button>
```

6 Lists contain elements that are not list items

● SERIOUS

Rule: list

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
1.3.1 Info and Relationships (Level A)	34	25 of 25 pages	102

WHO THIS AFFECTS

Screen readers announce "list, 6 items" based on this structure. When other elements are mixed in, the count and navigation are wrong, so users lose their place.

EXAMPLE ELEMENTS (3 OF 3 DISTINCT)

.overflow-menu ul[part="list"] · client-site.example/search?q=test

```
<ul part="list" style="height: 28.4062px; counter-reset: overflow-count 0;">
```

List element has direct children that are not allowed: [role=presentation]

overflow-list ul[part="list"] · client-site.example

```
<ul part="list" style="height: 28.4062px; counter-reset: overflow-count 0;">
```

List element has direct children that are not allowed: [role=presentation]

.facets__overflow-list ul[part="list"] · client-site.example/collections/collection-1

```
<ul part="list" style="height: 44px; counter-reset: overflow-count 0;">
```

List element has direct children that are not allowed: accordion-custom

RECOMMENDED FIX

Allow only `<li>` (plus `<script>` / `<template>`) as direct children of `<ul>` and `<ol>`. Wrap stray `<div>`s inside the `<li>` instead.

BEFORE — FROM THIS SCAN

```
<ul part="list" style="height: 28.4062px;
counter-reset: overflow-count 0;">
```

AFTER

```
<!-- non-list-item children moved inside an
<li> -->
<ul part="list" style="height: 28.4062px;
counter-reset: overflow-count 0;">
  <li>
```

7 ARIA components are missing their required parent element

● CRITICAL

Rule: aria-required-parent

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
1.3.1 Info and Relationships (Level A)	25	25 of 25 pages	100

WHO THIS AFFECTS

A component such as a menu item or tab is announced out of context — the screen reader cannot tell the visitor it is "item 2 of 5 in a menu", because as far as the markup is concerned there is no menu.

EXAMPLE ELEMENT (1 OF 1 DISTINCT)

button[role="menuitem"] · client-site.example

```
<button role="menuitem" class="button menu-list__link button-unstyled"> <span
class="menu-list__link-title">More</span> </button>
```

Required ARIA parents role not present: menu, menubar, group

RECOMMENDED FIX

Give each ARIA role its required parent role, with no intervening element that changes the structure. For example, `role="listitem"` needs a `role="list"` parent; `role="tab"` needs `role="tablist"`.

BEFORE — FROM THIS SCAN

```
<button role="menuitem" class="button
menu-list__link button-unstyled"> <span
class="menu-list__link-title">More</span>
</button>
```

AFTER

```
<!-- wrapped in the required role="menu"
parent -->
<div role="menu">
  <button role="menuitem" class="button
menu-list__link button-unstyled"> <span
class="menu-list__link-title">More</span>
</button>
</div>
```

8 Tap targets are too small or too close together

● SERIOUS

Rule: target-size

WCAG CRITERION	INSTANCES	SPREAD	PRIORITY SCORE
2.5.8 Target Size (Minimum) (Level AA)	31	7 of 25 pages	93

WHO THIS AFFECTS

People with hand tremor, limited dexterity, or anyone tapping on a moving train will hit the wrong control — most often in tightly packed footer links and quantity steppers.

EXAMPLE ELEMENTS (3 OF 19 DISTINCT)

button[aria-label="Slide 1 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-1"
aria-label="Slide 1 of 8" on:click="/select/0" ref="dots[]" aria-selected="true"> 1 </button>
```

Target has insufficient size (16px by 21.3px, should be at least 24px by 24px) · Target has insufficient space to its closest neighbors. Safe clickable space has a diameter of 11.2px instead of at least 24px.

button[aria-label="Slide 2 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-2"
aria-label="Slide 2 of 8" on:click="/select/1" ref="dots[]" aria-selected="false"> 2 </button>
```

Target has insufficient size (16px by 21.3px, should be at least 24px by 24px) · Target has insufficient space to its closest neighbors. Safe clickable space has a diameter of 11.2px instead of at least 24px.

button[aria-label="Slide 3 of 8"] · client-site.example/products/product-1

```
<button class="slideshow-control button button-unstyled" style="animation-timeline: --slide-3"
aria-label="Slide 3 of 8" on:click="/select/2" ref="dots[]" aria-selected="false"> 3 </button>
```

Target has insufficient size (16px by 21.3px, should be at least 24px by 24px) · Target has insufficient space to its closest neighbors. Safe clickable space has a diameter of 11.2px instead of at least 24px.

RECOMMENDED FIX

Make interactive controls at least 24×24 CSS pixels, or leave enough spacing that a 24px circle centred on each target does not overlap its neighbour. Padding counts towards the target size; visual size does not have to change.

BEFORE — FROM THIS SCAN

```
/* button[aria-label="Slide 1 of 8"] */  
/* smaller than 24×24 CSS px, or overlapping a  
neighbour */
```

AFTER

```
/* Pad the control to a 24×24 px target.  
Padding counts,  
so the visible size does not have to  
change. */  
.control { padding: 8px; box-sizing:  
content-box; }
```

REMAINING ISSUE TYPES (4)

These scored below the 8 detailed above on the same severity × instance measure. They are listed in full so nothing found by the scan is omitted from this report, each with the fix that applies to it.

ISSUE	WCAG	SEVERITY	INST.	PAGES
ARIA attributes are used where they are not permitted aria-prohibited-attr Remove the prohibited attribute, or give the element a role that supports a name (or use a native element that does).	4.1.2	● SERIOUS	30	7
Buttons have no readable name button-name Add visible button text, or <code>aria-label</code> for icon-only buttons, and hide the decorative icon from assistive technology.	4.1.2	● CRITICAL	3	2
Embedded frames have no title frame-title Add a <code>title</code> attribute describing the frame's content.	4.1.2	● SERIOUS	4	4
Scrollable areas cannot be scrolled by keyboard scrollable-region-focusable Give the scrollable container <code>tabindex="0"</code> so it can receive focus and be scrolled with the arrow keys, plus an accessible name describing what it contains.	2.1.1	● SERIOUS	4	4

FLAGGED FOR HUMAN REVIEW

These are the scan's *undecided* results: checks the tooling started but could not decide, usually because the answer depends on something it cannot see — a background image behind text, a value set at runtime, or an element it could not measure reliably. They are neither failures nor passes. Each one needs a person to look at it and reach a verdict, and they form part of the manual review scope in Section B.

CHECK	WCAG	SEVERITY	ELEM.	PAGES
Text is too low in contrast against its background color-contrast Contrast could not be computed — typically text over a background image, gradient or semi-transparent overlay.	1.4.3	● SERIOUS	982	25
The same id is used more than once by ARIA references duplicate-id-aria Duplicate ids were detected in content that may be generated dynamically.	4.1.2	● CRITICAL	143	25

CHECK	WCAG	SEVERITY	ELEM.	PAGES
ARIA attributes have invalid values aria-valid-attr-value An ARIA reference could not be resolved at scan time — it may be populated by script after load.	4.1.2	● CRITICAL	59	25
ARIA attributes are used where they are not permitted aria-prohibited-attr Whether the element's accessible name is permitted could not be determined automatically.	4.1.2	● SERIOUS	10	10
Video has no captions video-caption The tool could not determine whether a caption track is present and correct.	1.2.2	● CRITICAL	1	1
Form fields have more than one label form-field-multiple-labels Ensure form field does not have multiple label elements	3.3.2	● MODERATE	1	1

Human review checklist

The manual review scope for client-site.example — 48 checks across 43 WCAG 2.2 Level AA success criteria that no automated tool can decide.

Everything in Section A was found by a machine reading the page source. The criteria below cannot be assessed that way: they depend on whether something *works* when a person uses it, or on whether text is *meaningful* rather than merely present. This is the defined scope of work for a manual review of this site — each item is a discrete, assignable check with a stated method, so it can be planned, delivered and evidenced alongside the monitoring in Section A.

REVIEW SCOPE AT A GLANCE

REVIEW THEME	HOW IT IS REVIEWED	CHECKS
Keyboard & focus	Keyboard-only pass on every page template — mouse unplugged	6
Media	Review of every video and audio asset published on the site	5
Meaningful alt text quality	Content review of images across each template, with a screen reader	6
Forms & error handling	Deliberate error-path testing on every form, including checkout	9
Gestures & motion	Touch-device and pointer testing across mobile templates	7
Visual presentation & zoom	Zoom, reflow and orientation testing at 320 px and 400%	5
Content & consistency	Editorial and cross-template review	10
Total		48

Keyboard & focus

6 CHECKS

Method: Keyboard-only pass on every page template — mouse unplugged

Automated tools can see that an element is focusable. They cannot judge whether the page can actually be operated, in a sensible order, by someone who never touches a mouse.

1	☐	2.1.1 Keyboard Level A	Operate every control — menus, carousels, filters, quantity steppers, modals, checkout — using only Tab, Shift+Tab, Enter, Space and the arrow keys, and confirm nothing requires a mouse.
2	☐	2.1.2 No Keyboard Trap Level A	Tab into and back out of every overlay, embedded video, date picker and third-party widget, confirming focus is never trapped with no keyboard way out.
3	☐	2.4.3 Focus Order Level A	Tab through each template and confirm focus moves in an order that matches the visual reading order, including after a modal opens and after it closes.
4	☐	2.4.7 Focus Visible Level AA	Confirm the focused element is always clearly visible, on every background it appears against, and that no stylesheet removes the focus indicator.
5	☐	2.4.11 Focus Not Obscured (Minimum) Level AA	Check that sticky headers, cookie banners and chat widgets never completely cover the element that currently has focus.
6	☐	2.1.4 Character Key Shortcuts Level A	Check that any single-character keyboard shortcut — a "/" that jumps to search, say — can be switched off or remapped, so speech-input users cannot trigger it by accident.

Method: Review of every video and audio asset published on the site

Automated tools can detect that a caption track is absent. They cannot tell whether captions are accurate, synchronised, or complete — and they cannot judge whether a description conveys what is shown on screen.

7	☐	1.2.1 Audio-only and Video-only (Prerecorded) Level A	For audio-only and video-only content, confirm an equivalent alternative exists — a transcript for audio, a description or transcript for silent video.
8	☐	1.2.2 Captions (Prerecorded) Level A	Watch each prerecorded video with sound off and confirm captions are present, accurate, correctly synchronised, and identify who is speaking.
9	☐	1.2.3 Audio Description or Media Alternative (Prerecorded) Level A	Confirm that visual information not conveyed in the soundtrack — on-screen text, demonstrations, product detail — is available as audio description or a full text alternative.
10	☐	1.2.4 Captions (Live) Level AA	For any live streamed content, confirm real-time captions are provided.
11	☐	1.2.5 Audio Description (Prerecorded) Level AA	Confirm prerecorded video has an audio description track where visual detail is essential to understanding it.

Method: Content review of images across each template, with a screen reader

Automated testing can only confirm that an `alt` attribute exists. Whether the text is *useful* — accurate, appropriately brief, and describing purpose rather than appearance — is a judgment only a person can make. Section A lists images with no alternative text at all; this theme covers the quality of the ones that have it.

12	☐	1.1.1 Non-text Content Level A	INFORMATIVE IMAGES Read each alt text alongside the image and confirm it conveys the same information the image does — not the filename, not "image", not the product SKU.
13	☐	1.1.1 Non-text Content Level A	DECORATIVE IMAGES Confirm purely decorative images use <code>alt=""</code> so they are skipped, rather than being described redundantly alongside adjacent text.
14	☐	1.1.1 Non-text Content Level A	FUNCTIONAL IMAGES For images that are links or buttons, confirm the alt text describes the destination or action, not the picture.
15	☐	1.1.1 Non-text Content Level A	COMPLEX IMAGES For charts, size guides, infographics and diagrams, confirm a full text equivalent exists nearby — a short alt alone cannot carry them.
16	☐	1.1.1 Non-text Content Level A	TEXT IN IMAGES Confirm any words baked into an image — promotional banners especially — are repeated in the alt text or in adjacent real text.
17	☐	1.1.1 Non-text Content Level A	ICON-ONLY CONTROLS Confirm each icon button's accessible name describes what it does ("Search", "Open menu") rather than what it looks like ("magnifier").

Method: Deliberate error-path testing on every form, including checkout

Automated tools can see whether a field has a label. They cannot submit a form incorrectly and judge whether the resulting error message actually helps somebody fix the problem.

18	☐	3.3.1 Error Identification Level A	Submit each form with errors and confirm the failing fields are identified in text — not by a red border alone — and that the error is announced to screen reader users.
19	☐	3.3.2 Labels or Instructions Level A	Confirm every field has a visible label plus any instructions needed before entry — required fields, accepted formats, character limits.
20	☐	3.3.3 Error Suggestion Level AA	Confirm error messages suggest how to correct the problem ("Enter a postcode such as SW1A 1AA"), not just that one exists ("Invalid input").
21	☐	3.3.4 Error Prevention (Legal, Financial, Data) Level AA	For orders, payments and anything legally binding or deletable, confirm the visitor can reverse, check or confirm the action before it becomes final.
22	☐	3.3.8 Accessible Authentication (Minimum) Level AA	Confirm login and checkout do not require solving a puzzle, transcribing an image, or recalling information from memory, and that password managers can paste into every field.
23	☐	2.5.3 Label in Name Level A	Confirm each control's accessible name begins with the words printed on it, so someone using voice control can say what they see — a button reading "Add to basket" must not be named "Submit".
24	☐	3.2.1 On Focus Level A	Tab through each form and confirm nothing changes context on focus alone — no dropdown that submits, no modal that opens, no page that navigates simply because a control received focus.
25	☐	3.2.2 On Input Level A	Change every select, checkbox and radio button and confirm none of them submits the form or navigates away without warning the visitor first.
26	☐	4.1.3 Status Messages Level AA	Confirm messages that appear without moving focus — "Added to basket", validation counts, loading states — are announced to screen reader users via a live region.

Gestures & motion

7 CHECKS

Method: Touch-device and pointer testing across mobile templates

These criteria concern how a control responds to a pointer or to physical movement of the device. No static analysis of the page source can determine that.

27	☐	2.5.1 Pointer Gestures Level A	Confirm anything driven by a swipe, pinch or multi-finger gesture — carousels, image zoom, maps — also works with a single tap or an equivalent button.
28	☐	2.5.2 Pointer Cancellation Level A	Confirm actions fire on pointer release rather than press, and that sliding off a control before releasing cancels it.
29	☐	2.5.4 Motion Actuation Level A	Confirm any feature triggered by shaking or tilting the device also has a conventional control, and can be switched off.
30	☐	2.5.7 Dragging Movements Level AA	Confirm anything requiring a drag — sliders, reorderable lists, drag-to-cart — has a single-pointer alternative such as buttons or a text input.
31	☐	2.2.1 Timing Adjustable Level A	Confirm any time limit — a checkout hold, a session timeout, a countdown offer — can be switched off, adjusted, or extended before it expires.
32	☐	2.2.2 Pause, Stop, Hide Level A	Confirm every carousel, auto-advancing slideshow, video and animation that starts on its own and runs for more than five seconds can be paused, stopped or hidden.
33	☐	2.3.1 Three Flashes or Below Threshold Level A	Confirm nothing flashes more than three times per second, including video content and animated promotional banners.

Visual presentation & zoom

5 CHECKS

Method: Zoom, reflow and orientation testing at 320 px and 400%

The scan measures the page once, at one fixed desktop viewport. Whether the site still works when somebody zooms to 400%, rotates their phone, or applies their own text spacing is something only a person resizing a real browser can determine.

34	☐	1.3.4 Orientation Level AA	Rotate a phone and a tablet through both orientations and confirm the site works in each — nothing locked to portrait or landscape unless that orientation is essential.
35	☐	1.4.5 Images of Text Level AA	Confirm text is real text rather than a picture of text, logos aside — promotional banners and campaign headlines are the usual offenders.
36	☐	1.4.10 Reflow Level AA	View each template at 320 CSS pixels wide (or 400% zoom at 1280 px) and confirm content reflows to a single column with no horizontal scrolling.
37	☐	1.4.11 Non-text Contrast Level AA	Confirm button edges, input borders, focus indicators, icons and chart colours all reach 3:1 contrast against whatever sits behind them.
38	☐	1.4.13 Content on Hover or Focus Level AA	Confirm tooltips and hover menus can be dismissed without moving the pointer, stay visible while the pointer moves onto them, and remain until dismissed.

Method: Editorial and cross-template review

These criteria are about meaning and consistency across the site. A tool can compare markup, but only a person can judge whether a heading is descriptive or whether two components that look alike are named alike.

39	☐	1.3.3 Sensory Characteristics Level A	Confirm no instruction relies on shape, size, position or sound alone — "the button on the right", "the round icon" — without also naming the element.
40	☐	1.4.1 Use of Color Level A	USE OF COLOR — BEYOND CONTRAST Confirm colour is never the only way information is conveyed: stock status, form errors, sale prices, chart series and required fields all need a text or shape cue as well. (Automated testing covers only the link-in-text-block case.)
41	☐	2.4.6 Headings and Labels Level AA	Read every heading and label out of context and confirm each one describes the content or purpose it introduces.
42	☐	3.2.3 Consistent Navigation Level AA	Confirm navigation appears in the same relative order on every page it occurs on.
43	☐	3.2.4 Consistent Identification Level AA	Confirm components with the same function are named and iconified identically everywhere — the basket is not "Cart" on one template and "Bag" on another.
44	☐	3.2.6 Consistent Help Level A	Confirm help mechanisms — contact details, live chat, help links — appear in the same relative place on every page that offers them.
45	☐	3.3.7 Redundant Entry Level A	Confirm multi-step processes never ask for the same information twice without auto-filling it or offering it for selection — billing and delivery address especially.
46	☐	1.3.2 Meaningful Sequence Level A	Read each template with a screen reader, or with stylesheets disabled, and confirm the reading order still makes sense — multi-column layouts and product grids are where it usually breaks.
47	☐	2.4.5 Multiple Ways Level AA	Confirm there is more than one way to reach any given page — the navigation plus at least one of search, a sitemap, or category listings.
48	☐	3.1.2 Language of Parts Level AA	Confirm any passage written in another language carries its own <code>lang</code> attribute, so a screen reader switches pronunciation for it.

Checks are drawn from WCAG 2.2 Level AA. Where a criterion is partly testable by automation — 1.1.1 and 1.4.1 in particular — the automated portion is reported in Section A and only the human-judgment portion appears here.

Audit trail & methodology

The record of what was tested and when: the pages covered, the viewport they were rendered at, the ruleset applied, and the standard assessed against.

SCAN RECORD

Scan date and time	31 July 2026 at 22:40 UTC
Site	https://client-site.example
Pages scanned	25
Testing method	Headless Chromium, automated rule evaluation against the ruleset below
Ruleset	wcag2a, wcag2aa, wcag21a, wcag21aa, wcag22aa
Viewport	1,440 × 900 px, landscape-primary
User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Page load strategy	networkidle, load+settle
Standards baseline	WCAG 2.2 Level AA
Report generated	1 August 2026 at 16:39 UTC

STANDARDS BASELINE

STANDARDS NOTE

This report evaluates against WCAG 2.2 Level AA. The current EU harmonised standard (EN 301 549 v3.2.1) references WCAG 2.1 Level AA, which is a subset of the criteria assessed here. An updated harmonised standard incorporating WCAG 2.2 is expected.

WCAG 2.2 adds six success criteria at Level A and AA over WCAG 2.1 — 2.4.11, 2.5.7, 2.5.8, 3.2.6, 3.3.7 and 3.3.8. Assessing against 2.2 covers every 2.1 criterion that remains current; 4.1.1 Parsing was removed in 2.2 and is obsolete.

SCOPE OF AUTOMATED TESTING

LIMITATIONS

Automated testing can detect only a subset of accessibility barriers. Criteria requiring human judgment are listed in the Human Review Checklist. This report documents monitoring evidence; it is not a certification of conformance.

Individual automated checks are recorded as passed or failed against the specific elements tested. That result does not extend to the page, the template, or the site: a rule that records no failure means only that the condition it tests was not detected on the elements it examined, at the moment of the scan, at the viewport used. On this scan, 43 rules ran: 26,912 element checks passed and 505 failed.

CHECKS THAT RECORDED NO FAILURES

43 rules ran against this sample. **30** recorded no failure anywhere and are listed below, across 14,419 element checks. **12** recorded at least one failure and are reported in Section A. **1** returned only undecided results and appears under Flagged for human review (video-caption). $30 + 12 + 1 = 43$. Recording no failure is not the same as settling every element examined: 3 of the rules below also returned undecided elements, counted in the Undecided column and detailed in Section A.

CHECK	WHAT IT VERIFIES	ELEMENTS	UNDECIDED	RESULT
Inline text spacing must be adjustable with custom stylesheets avoid-inline-spacing	Ensures that text spacing set through style attributes can be adjusted with custom stylesheets	3,732	—	No issues detected
ARIA attributes must be used as specified for the element's role aria-conditional-attr	Ensures ARIA attributes are used as described in the specification of the element's role	1,442	—	No issues detected
ARIA attributes must conform to valid names aria-valid-attr	Ensures attributes that begin with aria- are valid ARIA attributes	1,442	—	No issues detected
ARIA attributes must conform to valid values aria-valid-attr-value	Ensures all ARIA attributes have valid values	1,383	59	No failures; 59 undecided
Interactive controls must not be nested nested-interactive	Ensures interactive controls are not nested as they are not always announced by screen readers or can cause focus problems for assistive technologies	1,382	—	No issues detected
 elements must be contained in a or listitem	Ensures elements are used semantically	1,227	—	No issues detected
IDs used in ARIA and labels must be unique duplicate-id-aria	Ensures every id attribute value used in ARIA and in labels is unique	908	143	No failures; 143 undecided
Deprecated ARIA roles must not be used aria-deprecated-role	Ensures elements do not use deprecated roles	654	—	No issues detected
Required ARIA attributes must be provided aria-required-attr	Ensures elements with ARIA roles have all required ARIA attributes	654	—	No issues detected
ARIA roles used must conform to valid values aria-roles	Ensures all elements with a role attribute use a valid value	654	—	No issues detected
Summary elements must have discernible text summary-name	Ensures summary elements have discernible text	209	—	No issues detected

CHECK	WHAT IT VERIFIES	ELEMENTS	UNDECIDED	RESULT
Form elements must have labels label	Ensures every form element has a label	161	—	No issues detected
Form field must not have multiple label elements form-field-multiple-labels	Ensures form field does not have multiple label elements	160	1	No failures; 1 undecided
ARIA commands must have an accessible name aria-command-name	Ensures every ARIA button, link and menuitem has an accessible name	66	—	No issues detected
<svg> elements with an img role must have alternative text svg-img-alt	Ensures <svg> elements with an img, graphics-document or graphics-symbol role have accessible text	56	—	No issues detected
<html> element must have a valid value for the lang attribute html-lang-valid	Ensures the lang attribute of the <html> element has a valid value	54	—	No issues detected
Links must be distinguishable without relying on color link-in-text-block	Ensures links are distinguished from surrounding text in a way that does not rely on color	50	—	No issues detected
Frames must have a unique title attribute frame-title-unique	Ensures <iframe> and <frame> elements contain a unique title attribute	27	—	No issues detected
aria-hidden="true" must not be present on the document body aria-hidden-body	Ensures aria-hidden="true" is not present on the document body	25	—	No issues detected
Page must have means to bypass repeated blocks bypass	Ensures each page has at least one mechanism for a user to bypass navigation and jump straight to the content	25	—	No issues detected
Documents must have <title> element to aid in navigation document-title	Ensures each HTML document contains a non-empty <title> element	25	—	No issues detected
<html> element must have a lang attribute html-has-lang	Ensures every HTML document has a lang attribute	25	—	No issues detected
Zooming and scaling must not be disabled meta-viewport	Ensures <meta name="viewport"> does not disable text scaling and zooming	25	—	No issues detected
Certain ARIA roles must contain particular children aria-required-children	Ensures elements with an ARIA role that require child roles contain them	14	—	No issues detected
ARIA input fields must have an accessible name aria-input-field-name	Ensures every ARIA input field has an accessible name	9	—	No issues detected
[role="img"] elements must have alternative text role-img-alt	Ensures [role="img"] elements have alternative text	6	—	No issues detected

CHECK	WHAT IT VERIFIES	ELEMENTS	UNDECIDED	RESULT
autocomplete attribute must be used correctly autocomplete-valid	Ensures the autocomplete attribute is correct and suitable for the form field	1	—	No issues detected
<video> or <audio> elements must not play automatically no-autoplay-audio	Ensures <video> or <audio> elements do not autoplay audio for more than 3 seconds without a control mechanism to stop or mute the audio	1	—	No issues detected
Table cell headers attributes must refer to other <th> elements in the same table td-headers-attr	Ensures that each cell in a table that uses the headers attribute refers only to other <th> elements in that table	1	—	No issues detected
Table headers in a data table must refer to data cells th-has-data-cells	Ensures that <th> elements and elements with role=columnheader/rowheader have data cells they describe	1	—	No issues detected

HOW FINDINGS WERE PRIORITISED

Section A is ordered by a priority score of **severity weight × instance count**. Severity uses the four-level scale the rule engine assigns to each finding — a widely used convention, not a WCAG grading and not devised for this report: critical = 4, serious = 3, moderate = 2, minor = 1. Ties are broken by severity, then by instance count. Instances are counted per element occurrence; where the same component recurs across pages it is reported once as an issue type, with the number of pages affected shown alongside.

PAGES SCANNED

URL	TEMPLATE	HTTP	ISSUES	REVIEW
client-site.example	homepage	200	29	85
client-site.example/cart	cart	200	13	61
client-site.example/search?q=test	search	200	8	13
client-site.example/collections/collection-1	collections	200	13	135
client-site.example/collections/collection-2	collections	200	11	60
client-site.example/collections/collection-3	collections	200	11	50
client-site.example/products/product-1	products	200	73	13
client-site.example/products/product-2	products	200	34	13
client-site.example/products/product-3	products	200	43	16
client-site.example/products/product-4	products	200	41	17
client-site.example/products/product-5	products	200	38	16
client-site.example/pages/page-1	content	200	9	25
client-site.example/blogs/blog/article-1	content	200	38	40
client-site.example/blogs/blog/article-2	content	200	24	21
client-site.example/collections/collection-4	collections	200	11	40
client-site.example/pages/page-2	content	200	7	6

URL	TEMPLATE	HTTP	ISSUES	REVIEW
client-site.example/collections/collection-5	collections	200	12	136
client-site.example/pages/page-3	content	200	13	20
client-site.example/collections/collection-6	collections	200	11	66
client-site.example/pages/page-4	content	200	12	5
client-site.example/collections/collection-7	collections	200	12	79
client-site.example/pages/page-5	content	200	9	5
client-site.example/collections/collection-8	collections	200	14	172
client-site.example/pages/page-6	content	200	7	5
client-site.example/collections/collection-9	collections	200	12	97

KNOWN LIMITS OF THIS SCAN

- Pages were tested at a single desktop viewport (1,440 × 900 px); mobile layouts, reflow at 320 px and zoom to 400% were not assessed automatically.
- Each page was tested in one state — as it loads. Content behind interaction (menus, modals, accordions, validation states, logged-in areas) was not exercised.
- Only pages listed above were scanned; they are a template-representative sample of the 169 URLs discovered in the sitemap, not the whole site.
- Third-party embeds and content injected after the scan window may not be represented.
- Results reflect the site as it was on 31 July 2026 and will drift as the site changes.

Report produced by Letrix Labs for UK garden & outdoor retailer (anonymised) · client-site.example · Scan of 31 July 2026 · Generated 1 August 2026 at 16:39 UTC

Criterion coverage map

Where each WCAG 2.2 Level A and AA success criterion is dealt with in this report, and whether a machine or a person establishes it.

Every WCAG 2.2 A/AA success criterion appears exactly once in this table. 12 are established by automated checks alone, 5 partly by automation and partly by human review, and 38 by human review only — which is the ratio that makes the Human Review Checklist the larger part of the work. *Section A* means a failure was recorded against that criterion; *Section C* means the rules for it ran and recorded no issue; *Section B item #n* is the numbered check in the Human Review Checklist.

SC	NAME	LVL	METHOD	HANDLED IN	SC	NAME	LVL	METHOD	HANDLED IN
1.1.1	Non-text Content	A	Hybrid	Section A + Section B items #12–17	2.4.3	Focus Order	A	Human review	Section B item #3
1.2.1	Audio-only and Video-only (Prerecorded)	A	Human review	Section B item #7	2.4.4	Link Purpose (In Context)	A	Automated	Section A
1.2.2	Captions (Prerecorded)	A	Hybrid	Section C + Section B item #8	2.4.5	Multiple Ways	AA	Human review	Section B item #47
1.2.3	Audio Description or Media Alternative (Prerecorded)	A	Human review	Section B item #9	2.4.6	Headings and Labels	AA	Human review	Section B item #41
1.2.4	Captions (Live)	AA	Human review	Section B item #10	2.4.7	Focus Visible	AA	Human review	Section B item #4
1.2.5	Audio Description (Prerecorded)	AA	Human review	Section B item #11	2.4.11	Focus Not Obscured (Minimum)	AA	Human review	Section B item #5
1.3.1	Info and Relationships	A	Automated	Section A	2.5.1	Pointer Gestures	A	Human review	Section B item #27
1.3.2	Meaningful Sequence	A	Human review	Section B item #46	2.5.2	Pointer Cancellation	A	Human review	Section B item #28
1.3.3	Sensory Characteristics	A	Human review	Section B item #39	2.5.3	Label in Name	A	Human review	Section B item #23
1.3.4	Orientation	AA	Human review	Section B item #34	2.5.4	Motion Actuation	A	Human review	Section B item #29
1.3.5	Identify Input Purpose	AA	Automated	Section C	2.5.7	Dragging Movements	AA	Human review	Section B item #30
1.4.1	Use of Color	A	Hybrid	Section C + Section B item #40	2.5.8	Target Size (Minimum)	AA	Automated	Section A
1.4.2	Audio Control	A	Automated	Section C	3.1.1	Language of Page	A	Automated	Section C
1.4.3	Contrast (Minimum)	AA	Automated	Section A	3.1.2	Language of Parts	AA	Human review	Section B item #48
1.4.4	Resize Text	AA	Automated	Section C	3.2.1	On Focus	A	Human review	Section B item #24
1.4.5	Images of Text	AA	Human review	Section B item #35	3.2.2	On Input	A	Human review	Section B item #25
1.4.10	Reflow	AA	Human review	Section B item #36	3.2.3	Consistent Navigation	AA	Human review	Section B item #42
1.4.11	Non-text Contrast	AA	Human review	Section B item #37	3.2.4	Consistent Identification	AA	Human review	Section B item #43
1.4.12	Text Spacing	AA	Automated	Section C	3.2.6	Consistent Help	A	Human review	Section B item #44
1.4.13	Content on Hover or Focus	AA	Human review	Section B item #38	3.3.1	Error Identification	A	Human review	Section B item #18
2.1.1	Keyboard	A	Hybrid	Section A + Section B item #1	3.3.2	Labels or Instructions	A	Hybrid	Section C + Section B item #19
2.1.2	No Keyboard Trap	A	Human review	Section B item #2	3.3.3	Error Suggestion	AA	Human review	Section B item #20
2.1.4	Character Key Shortcuts	A	Human review	Section B item #6	3.3.4	Error Prevention (Legal, Financial, Data)	AA	Human review	Section B item #21
2.2.1	Timing Adjustable	A	Human review	Section B item #31	3.3.7	Redundant Entry	A	Human review	Section B item #45
2.2.2	Pause, Stop, Hide	A	Human review	Section B item #32	3.3.8	Accessible Authentication (Minimum)	AA	Human review	Section B item #22
2.3.1	Three Flashes or Below Threshold	A	Human review	Section B item #33	4.1.2	Name, Role, Value	A	Automated	Section A
2.4.1	Bypass Blocks	A	Automated	Section C	4.1.3	Status Messages	AA	Human review	Section B item #26
2.4.2	Page Titled	A	Automated	Section C					

Coverage is derived from the scan and the checklist rather than listed by hand, so this table cannot fall out of step with Sections A and B. WCAG 2.2 has 86 success criteria in total; the 55 here are the Level A and AA criteria, which are the scope of this report. Success Criterion 4.1.1 Parsing is absent because it was removed in WCAG 2.2.