TECHNICAL ACCESSIBILITY REPORT

Your Agency

Interactive E-Commerce Funnel Audit · EN 301 549 v3.2.1 & WCAG 2.2 AA

AUDIT CONTROL REF
aud_6116f2740e2f4eab
CLIENT ENGINEERING DELIVERABLE
SYNTHETIC DOM VERIFICATION

AUDITED DOMAIN	AUDIT ID	DATE & TIME (UTC)
`https://tienda-de-pruebas.test/`	`aud_6116f2740e2f4eab`	**2026-11-17 04:09:59 UTC** Scan execution: 1397 ms
EVALUATED STANDARD	**DETECTED CMS**	**PREPARED BY**
EN 301 549 v3.2.1 · WCAG 2.2 Level AA EU Directive 2019/882 Annex I	**WooCommerce / WordPress** Automated DOM Signature	**Your Agency**

01 PLAIN-LANGUAGE SUMMARY

4 of 4 stages of the purchase journey were audited.

The barriers that most affect a purchase:

1. **The keyboard gets stuck inside a pop-up: shoppers who do not use a mouse cannot continue.** — FIX EFFORT: HIGH
   Where: Cart · 1 finding
2. **Some fields or drop-downs have no label: a screen reader cannot say what is being asked.** — FIX EFFORT: LOW
   Where: Checkout · 1 finding
3. **Form errors are not announced: the shopper cannot tell which field to correct.** — FIX EFFORT: MEDIUM
   Where: Checkout · 1 finding

Estimated effort: low = one-off change of text, attribute or colour; medium = template or script adjustment; high = rebuilding how a component behaves.

02 TECHNICAL INDEX & SEVERITY OF FINDINGS

TECHNICAL INDEX	CRITICAL	SERIOUS	MODERATE	MINOR
17 / 100 WCAG 2.2 AA BARRIERS DETECTED	**4** Flow Blockers	**6** WCAG Barriers	**0** AT Friction	**0** Advisories

How it is calculated: starts at 100; each distinct type of barrier subtracts once according to its severity (critical > serious > moderate > minor), however many pages or elements repeat it. Based on 4 of 4 stages audited. It is a technical indicator, not a legal rating.

03 PURCHASE JOURNEY: WHAT WAS AUDITED AT EACH STAGE

STAGE 01	STAGE 02	STAGE 03	STAGE 04
BARRIERS FOUND	BARRIERS FOUND	BARRIERS FOUND	BARRIERS FOUND
01 Landing `/` Home page scanned	**02 Product Page** `/product/atlantic-wool-coat` Variant change is not announced (aria-live)	**03 Cart** `/product/atlantic-wool-coat` Mini-cart by keyboard: focus trapped, Escape does not close	**04 Checkout** `/checkout` Form errors are not announced (aria-invalid)
Critical 1	**Critical** 1	**Critical** 1	**Critical** 1
Serious 1	**Serious** 1	**Serious** 2	**Serious** 2
Moderate 0	Moderate 0	Moderate 0	Moderate 0
Minor 0	Minor 0	Minor 0	Minor 0

METHODOLOGY & REGULATORY SCOPE NOTICE (EN 301 549 · WCAG 2.2 AA)

Automated synthetic accessibility regression testing and assisted code remediation aligned with WCAG 2.2 AA & EN 301 549. This report reflects automated browser verification of interactive e-commerce flows and does not constitute legal advice, official certification, or a guarantee of legal immunity under the European Accessibility Act (Directive (EU) 2019/882) or the ADA.

C5 Technical Scope: Automated synthetic DOM & keyboard verification aligned with EN 301 549 v3.2.1 & WCAG 2.2 AA. Not legal advice or official EAA certification.

⚠ **TECHNICAL OVERLAY ALERT · ACCESSIBILITY TREE REMAINS UNMODIFIED**

Detected Signature(s): accessiBe [acsbapp.com / acsb.js]

Client-side overlay scripts mutate superficial visual styles at runtime without repairing the underlying DOM accessibility tree. Screen readers and keyboard users still encounter unlabelled controls, missing aria-live variant state regions, modal focus traps, and unbound checkout validation errors evaluated under EN 301 549 v3.2.1. The European Disability Forum (EDF) and the IAAP state that overlays do not make a website accessible.

04 ENGINEERING FINDINGS & SOURCE REMEDIATION SPECIFICATIONS

F-01 CRITICAL — STAGE 01 · LANDING

Images must have alternative text

WCAG: 1.1.1 Non-text Content (Level A) · Standard: EN 301 549 Clause 9.1.1.1 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#catalog-coat-missing-alt`

PAGE: https://tienda-de-pruebas.test/

Ensure <img> elements have alternative text or a role of none or presentation

Remediation (PHP): Populate non-empty WooCommerce product image alt attributes from the parent product title when missing.

```
add_filter('wp_get_attachment_image_attributes', function (array $attr, WP_Post $attachment):
array {
    if (empty(trim($attr['alt'] ?? ''))) {
        $parent_title = $attachment->post_parent ? get_the_title($attachment->post_parent) : '';
        $attr['alt'] = $parent_title ?: get_the_title($attachment->ID);
    }
    return $attr;
}, 10, 2);
```

F-02 CRITICAL — STAGE 02 · PRODUCT PAGE

Images must have alternative text

WCAG: 1.1.1 Non-text Content (Level A) · Standard: EN 301 549 Clause 9.1.1.1 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#pdp-coat-unlabeled`

PAGE: https://tienda-de-pruebas.test/product/atlantic-wool-coat

Ensure <img> elements have alternative text or a role of none or presentation

Remediation (PHP): Populate non-empty WooCommerce product image alt attributes from the parent product title when missing.

```
add_filter('wp_get_attachment_image_attributes', function (array $attr, WP_Post $attachment):
array {
    if (empty(trim($attr['alt'] ?? ''))) {
        $parent_title = $attachment->post_parent ? get_the_title($attachment->post_parent) : '';
        $attr['alt'] = $parent_title ?: get_the_title($attachment->ID);
    }
    return $attr;
}, 10, 2);
```


C5 Technical Scope: Automated synthetic DOM & keyboard verification aligned with EN 301 549 v3.2.1 & WCAG 2.2 AA. Not legal advice or official EAA certification.

F-03 CRITICAL — STAGE 03 · CART

Keyboard Focus Trap detected in Mini-Cart drawer (Tab looped & Escape ignored)

WCAG: 2.1.2 No Keyboard Trap (Level A) · Standard: EN 301 549 Clause 9.2.1.2 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#mini-cart`

PAGE: https://tienda-de-pruebas.test/product/atlantic-wool-coat

Keyboard focus is locked in a repeating Tab loop inside the cart drawer and pressing Escape fails to dismiss it, trapping keyboard and switch users.

Remediation (JAVASCRIPT): Apply WAI-ARIA dialog semantics (role="dialog", aria-modal="true"), support Escape dismissal, and restore trigger focus.

```
function enableAccessibleCartDrawer(drawerEl, triggerBtn, closeBtn) {
  drawerEl.setAttribute('role', 'dialog');
  drawerEl.setAttribute('aria-modal', 'true');
  const onKeyDown = (e) => { if (e.key === 'Escape') closeDrawer(); };
  function closeDrawer() {
    drawerEl.hidden = true;
    document.removeEventListener('keydown', onKeyDown);
    triggerBtn?.focus();
  }
  document.addEventListener('keydown', onKeyDown);
  closeBtn?.addEventListener('click', closeDrawer);
}
```

F-04 CRITICAL — STAGE 04 · CHECKOUT

Checkout input field lacks programmatic <label> or aria-label

WCAG: 3.3.2 Labels or Instructions (Level A) · Standard: EN 301 549 Clause 9.3.3.2 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#billing_email`

PAGE: https://tienda-de-pruebas.test/checkout

Shoppers using screen readers or voice control cannot identify the purpose of the required checkout input.

Remediation (HTML): Associate an explicit <label for="billing_email"> with the <input> field (#billing_email).

```
<label for="billing_email">Billing email address</label>
<input id="billing_email" type="email" name="billing_email" autocomplete="email">
```

F-05 SERIOUS — STAGE 01 · LANDING

Third-party overlay widget detected (accessiBe)

WCAG: 4.1.2 Name, Role, Value (Level A) · Standard: EN 301 549 Clause 9.4.1.2 & Clause 5.2 (EAA Overlay Risk Advisory)

SELECTOR `script[data-overlay="accessibe"]`

PAGE: https://tienda-de-pruebas.test/

accessiBe (acsbapp.com / acsb.js) injects runtime DOM overrides that frequently conflict with native screen readers and do not remediate source template barriers under EN 301 549.

Remediation (HTML): Remove third-party JavaScript overlay scripts that mutate ARIA at runtime and remediate source templates directly.

```
<!-- Remove third-party overlay script (acsbapp / userway / equalweb) and ship native WCAG 2.2 AA
markup -->
```


C5 Technical Scope: Automated synthetic DOM & keyboard verification aligned with EN 301 549 v3.2.1 & WCAG 2.2 AA. Not legal advice or official EAA certification.

F-06 SERIOUS — STAGE 02 · PRODUCT PAGE

Dynamic product variant price/stock updates are not announced via aria-live

WCAG: 4.1.3 Status Messages (Level AA) · Standard: EN 301 549 Clause 9.4.1.3 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#pa_size`

PAGE: https://tienda-de-pruebas.test/product/atlantic-wool-coat

Changing product variants updates price or stock text visually without an aria-live="polite" or role="status" container.

Remediation (JAVASCRIPT): Announce the chosen variation price and stock in an aria-live="polite" region that already exists on the page (WooCommerce found_variation event).

```
// The live region must exist BEFORE the change; then only its text is updated.
jQuery(function ($) {
  var $form = $('form.variations_form');
  var $live = $('<div class="screen-reader-text" role="status" aria-live="polite" aria-
atomic="true"></div>').appendTo($form);
  $form.on('found_variation', function (event, variation) {
    var price = $('<div>').html(variation.price_html || '').text();
    var stock = $('<div>').html(variation.availability_html || '').text();
    $live.text($.trim(price + ' ' + stock));
  });
  $form.on('reset_data', function () { $live.text(''); });
});
```

F-07 SERIOUS — STAGE 03 · CART

Cart drawer/modal lacks role="dialog", aria-modal="true", or accessible label

WCAG: 4.1.2 Name, Role, Value (Level A) · Standard: EN 301 549 Clause 9.4.1.2 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#mini-cart`

PAGE: https://tienda-de-pruebas.test/product/atlantic-wool-coat

Assistive technologies cannot identify modal boundaries without role="dialog", aria-modal="true", and aria-label/aria-labelledby.

Remediation (JAVASCRIPT): Apply WAI-ARIA dialog semantics (role="dialog", aria-modal="true"), support Escape dismissal, and restore trigger focus.

```
function enableAccessibleCartDrawer(drawerEl, triggerBtn, closeBtn) {
  drawerEl.setAttribute('role', 'dialog');
  drawerEl.setAttribute('aria-modal', 'true');
  const onKeyDown = (e) => { if (e.key === 'Escape') closeDrawer(); };
  function closeDrawer() {
    drawerEl.hidden = true;
    document.removeEventListener('keydown', onKeyDown);
    triggerBtn?.focus();
  }
  document.addEventListener('keydown', onKeyDown);
  closeBtn?.addEventListener('click', closeDrawer);
}
```


C5 Technical Scope: Automated synthetic DOM & keyboard verification aligned with EN 301 549 v3.2.1 & WCAG 2.2 AA. Not legal advice or official EAA certification.

F-08 SERIOUS STAGE 03 · CART

Interactive elements inside mini-cart lack visible keyboard focus indicator

WCAG: 2.4.7 Focus Visible (Level AA) · Standard: EN 301 549 Clause 9.2.4.7 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#cart-checkout-cta`

PAGE: https://tienda-de-pruebas.test/product/atlantic-wool-coat

Focused controls suppress CSS outline/box-shadow, preventing sighted keyboard users from tracking focus.

Remediation (CSS): Provide a high-contrast :focus-visible outline (minimum 3:1 contrast ratio) on all interactive controls.

```
:where(a, button, input, select, textarea, [tabindex]):focus-visible {
  outline: 3px solid #0f62fe !important;
  outline-offset: 2px !important;
  box-shadow: 0 0 0 4px rgba(15, 98, 254, 0.25) !important;
}
```

F-09 SERIOUS STAGE 04 · CHECKOUT

All touch targets must be 24px large, or leave sufficient space

WCAG: 2.5.8 Target Size (Minimum) (Level AA) · Standard: EN 301 549 Clause 9.2.5.8 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#billing_email`

PAGE: https://tienda-de-pruebas.test/checkout · 2 elements with this barrier on the page

Ensure touch targets have sufficient size and space

Remediation (CSS): Expand interactive pointer target dimensions on #billing_email to at least 24x24 CSS pixels (WCAG 2.5.8).

```
#billing_email {
  min-width: 24px;
  min-height: 24px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
}
```

F-10 SERIOUS STAGE 04 · CHECKOUT

Checkout validation error lacks aria-invalid="true" and programmatic alert association

WCAG: 3.3.1 Error Identification (Level A) · Standard: EN 301 549 Clause 9.3.3.1 (EAA Directive (EU) 2019/882 Annex I)

SELECTOR `#billing_email`

PAGE: https://tienda-de-pruebas.test/checkout

Submitting an empty required checkout field does not expose aria-invalid="true" paired with aria-describedby or role="alert".

Remediation (HTML): Expose validation errors programmatically via aria-invalid="true", aria-describedby, and role="alert".

```
<div class="checkout-field">
  <label for="billing_email">Billing email address</label>
  <input id="billing_email" name="billing_email" type="email" required aria-invalid="true" aria-
describedby="billing_email-error">
  <p id="billing_email-error" role="alert">Please enter a valid value to continue.</p>
</div>
```


C5 Technical Scope: Automated synthetic DOM & keyboard verification aligned with EN 301 549 v3.2.1 & WCAG 2.2 AA. Not legal advice or official EAA certification.