Automated Accessibility Scan Report

www.w3.org · November 27, 2026 · Standard: WCAG 2.1 Level A and AA (automated checks)
Pages tested (desktop 1366×900 and iPhone 13 size): `https://www.w3.org/WAI/demos/bad/before/home.html`, `https://www.w3.org/WAI/demos/bad/before/news.html`

LIGHTHOUSE ACCESSIBILITY SCORE (MOBILE)

50/100

Automated tools find only part of the issues, so even 100 does not mean fully accessible.

Other pages: /WAI/demos/bad/before/news.html 57

PROBLEMS FOUND

Severity	Problems	Elements
CRITICAL	4	74
SERIOUS	5	15
MODERATE	4	30
MINOR	0	0
Total	**13**	**119**

Includes 4 best-practice items that are not a strict WCAG requirement.

Fix these first

1. CRITICAL **Images have no text alternative** (39 elements, WCAG 1.1.1)
   Screen readers cannot describe a picture without alt text. Blind visitors hear only "image" or the file name, so product photos and image buttons mean nothing to them.
2. CRITICAL **Images have no text alternative** (33 elements, WCAG 1.1.1)
   Screen readers cannot describe a picture without alt text. Blind visitors hear only "image" or the file name, so product photos and image buttons mean nothing to them.
3. CRITICAL **Drop-down menus have no label** (1 element, WCAG 4.1.2)
   Screen reader users hear "combo box" with no hint of what they are choosing (size, color, country).
4. CRITICAL **Drop-down menus have no label** (1 element, WCAG 4.1.2)
   Screen reader users hear "combo box" with no hint of what they are choosing (size, color, country).
5. SERIOUS **Links have no readable name** (7 elements, WCAG 2.4.4)
   Links made only of an icon or an image without alt text are announced as just "link". Screen reader users cannot tell where the link goes.

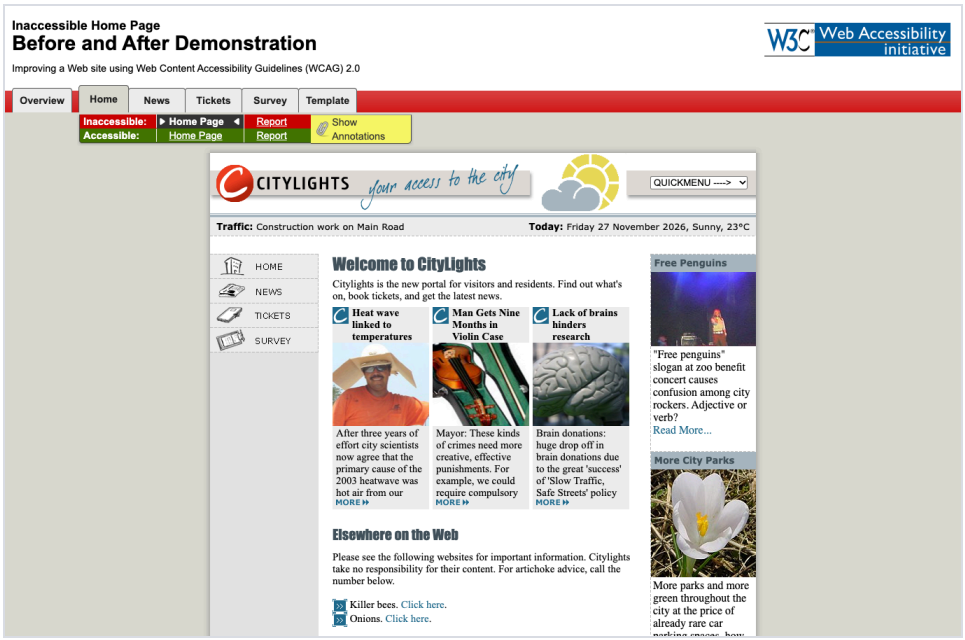
Why this matters

More than 1 in 4 U.S. adults have some type of disability (CDC), and many more shop with tired eyes, a small screen, or one hand. When text is hard to read, buttons have no name, or forms cannot be filled in with a keyboard or screen reader, those visitors leave and buy somewhere else. Most fixes are small changes to the theme that make the site easier for everyone. On the legal side, plaintiffs filed 3,117 website accessibility lawsuits in U.S. federal court in 2025, 27% more than in 2024 (Seyfarth Shaw, Mar 2026). This report does not predict or prevent legal claims.

This is an automated scan report. It is not a certification, a full audit, or legal advice. See the full notice at the end.

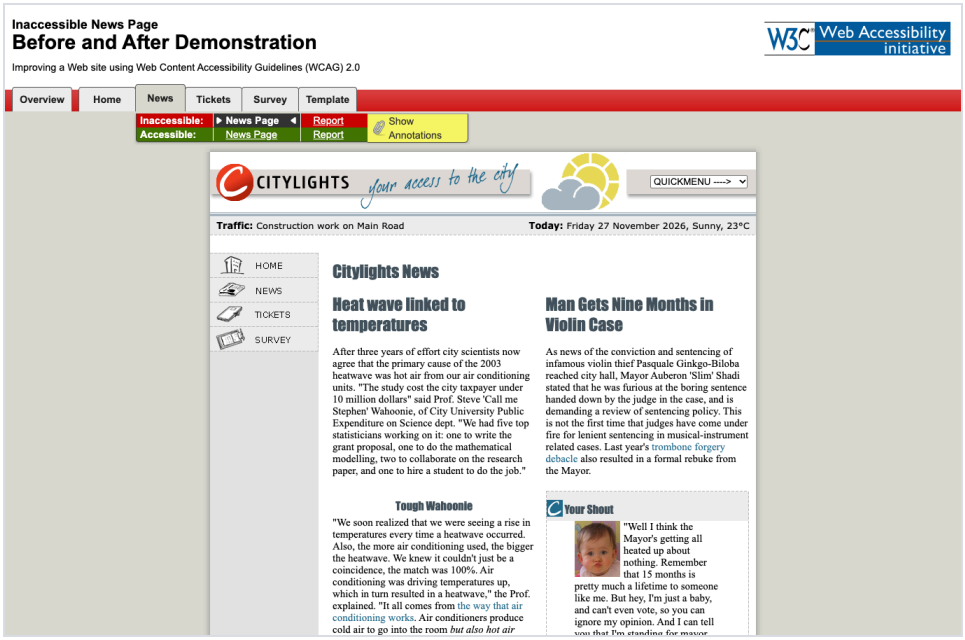
Scope and method

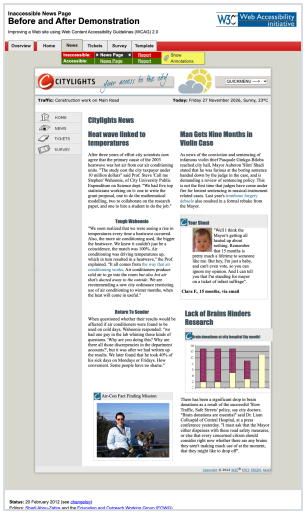
- Each page was loaded in Google Chrome 158.0.8127.88 at desktop size (1366×900) and at iPhone 13 size, and checked with axe-core 4.14.0 against WCAG 2.1 A/AA rules plus best practices. Before each check the page was scrolled once and animations were reduced, so lazy-loaded and animated content was tested in its final state.
- The overall score comes from Lighthouse 13.6.0 (accessibility category, mobile).
- Only public pages were tested, with a few requests per page and the site's robots.txt respected. No forms were submitted and no purchases were made.
- The problems themselves are found by fixed, rule-based tests (axe-core and Lighthouse), not by AI. The suggested fixes marked "AI-drafted" were written with an AI model (Claude, by Anthropic) from the details of each finding and a short snippet of the page's public HTML, which is sent to the provider to write the suggestion. AI suggestions can be wrong or may not fit your theme. Each one was checked by automatic rules against the finding it refers to; where a suggestion did not pass, the standard guidance is shown instead. Test every change before you publish it.
- Severity follows axe-core: **Critical**: blocks some visitors from using the page. **Serious**: makes the page very hard to use for some visitors. **Moderate**: causes friction or confusion. **Minor**: small annoyance; fix when convenient.





`https://www.w3.org/WAI/demos/bad/before/home.html` · “Welcome to CityLights! [Inaccessible Home Page]”





`https://www.w3.org/WAI/demos/bad/before/news.html` · “Welcome to CityLights! [Inaccessible News Page]”

Detailed findings by WCAG criterion

1.1.1 Non-text Content (Level A) Understanding 1.1.1

CRITICAL Images have no text alternative

39 elements · Desktop and mobile · axe rule `image-alt` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html`

Why it matters: Screen readers cannot describe a picture without alt text. Blind visitors hear only "image" or the file name, so product photos and image buttons mean nothing to them.

How to fix: Add an alt attribute that says what the image shows or does (alt="Red leather tote bag, front view"). For purely decorative images, use an empty alt="" so screen readers skip them.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> These border images (`border_left_top.gif`, `border_top.gif`, `border_right_top.gif` and similar) look purely decorative, so add an empty `alt=""` to each, e.g. `<img src="./img/border_top.gif" height="10px" alt="">`. This affects 39 images in total. Check the theme's layout/theme.liquid, sections/*.liquid or snippets/*.liquid for these tags. Better still, replace the images with CSS borders in assets/*.css. If any image carries meaning, give it alt text describing that.

Reference: https://dequeuniversity.com/rules/axe/4.14/image-alt?application=playwright

What the scanner found (all elements):

Fix any of the following:
Element does not have an alt attribute
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has no title attribute
Element's default semantics were not overridden with role="none" or role="presentation"

Where: `img[src$="border_left_top.gif"]` · Desktop and mobile

```
<img src="./img/border_left_top.gif" width="10px" height="10px">
```



Where: `img[src$="border_top.gif"]` · Desktop and mobile



```
<img src="./img/border_top.gif" height="10px">
```

Where: `img[src$="border_right_top.gif"]` · Desktop and mobile



```
<img src="./img/border_right_top.gif" width="10px" height="10px">
```

Where: `img[src$="border_left.gif"]` · Desktop and mobile

```
<img src="./img/border_left.gif" width="10px">
```

Where: `img[src$="top_logo_next_end.gif"]` · Desktop and mobile

```
<img src="./img/top_logo_next_end.gif" width="24px" height="86px">
```

Where: `img[src$="top_weather.gif"]` · Desktop and mobile

```
<img src="./img/top_weather.gif" width="128px" height="86px">
```

Where: `img[src$="top_logo_next_start.gif"]` · Desktop and mobile

```
<img src="./img/top_logo_next_start.gif" width="22px" height="86px">
```

Where: `img[src$="mark.gif"]` · Desktop and mobile

```
<img src="./img/mark.gif" width="158px" height="7px">
```

…and 31 more elements with the same problem (full list in raw.json).

CRITICAL Images have no text alternative

33 elements · Desktop and mobile · axe rule `image-alt` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Screen readers cannot describe a picture without alt text. Blind visitors hear only "image" or the file name, so product photos and image buttons mean nothing to them.

How to fix: Add an alt attribute that says what the image shows or does (alt="Red leather tote bag, front view"). For purely decorative images, use an empty alt="" so screen readers skip them.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> These images look like decorative border pieces (for example `border_left_top.gif`), so give each an empty `alt=""`: `<img src="./img/border_left_top.gif" alt="" width="10px" height="10px">`. This applies to all 33 affected images, including `border_top.gif` and `border_right_top.gif`. Look in layout/theme.liquid, sections/*.liquid or snippets/*.liquid. Better still, move the borders into CSS (assets/*.css) so no image tags are needed. If any image carries meaning, describe it in the alt text instead.

Reference: https://dequeuniversity.com/rules/axe/4.14/image-alt?application=playwright

What the scanner found (all elements):

Fix any of the following:
Element does not have an alt attribute
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has no title attribute
Element's default semantics were not overridden with role="none" or role="presentation"

Where: `img[src$="border_left_top.gif"]` · Desktop and mobile



```
<img src="./img/border_left_top.gif" width="10px" height="10px">
```

Where: `img[src$="border_top.gif"]` · Desktop and mobile



```
<img src="./img/border_top.gif" height="10px">
```

Where: `img[src$="border_right_top.gif"]` · Desktop and mobile



```
<img src="./img/border_right_top.gif" width="10px" height="10px">
```

Where: `img[src$="border_left.gif"]` · Desktop and mobile

```
<img src="./img/border_left.gif" width="10px">
```

Where: `img[src$="top_logo_next_end.gif"]` · Desktop and mobile

```
<img src="./img/top_logo_next_end.gif" width="24px" height="86px">
```

Where: `img[src$="top_weather.gif"]` · Desktop and mobile

```
<img src="./img/top_weather.gif" width="128px" height="86px">
```

Where: `img[src$="top_logo_next_start.gif"]` · Desktop and mobile

```
<img src="./img/top_logo_next_start.gif" width="22px" height="86px">
```

Where: `img[src$="mark.gif"]` · Desktop and mobile

```
<img src="./img/mark.gif" width="158px" height="7px">
```

…and 25 more elements with the same problem (full list in raw.json).

1.4.3 Contrast (Minimum) (Level AA) Understanding 1.4.3

SERIOUS Text is hard to read against its background

2 elements · Desktop and mobile · axe rule `color-contrast` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Light gray text, or text on a busy or similar-colored background, is hard to read for people with low vision, older visitors, and anyone on a phone in bright sunlight.

How to fix: Darken the text or lighten the background until the contrast ratio is at least 4.5:1 (3:1 for large text: 24px, or 18.66px bold). Check colors with a contrast checker and update the theme colors, not each element.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> Two labels, such as "Free Penguins", use dark grey-blue text (#41545D) on a light blue-grey cell (`bgcolor="#A9B8BF"`). Contrast is 3.88:1 and needs 4.5:1. Darken the `font color` (for example toward near-black) or lighten the cell background. Both cells use the same colors, so change them together. Look in sections/*.liquid or snippets/*.liquid for the table markup, then recheck with a contrast checker.

Reference: https://dequeuniversity.com/rules/axe/4.14/color-contrast?application=playwright

What the scanner found (all elements):

Fix any of the following:
Element has insufficient color contrast of 3.88 (foreground color: #41545d, background color: #a9b8bf, font size: 9.8pt (13px), font weight: bold). Expected contrast ratio of 4.5:1

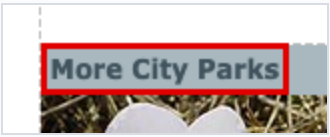
Where: `tr[height="25px"]:nth-child(2) > td[bgcolor="#A9B8BF"][width="150px"] > font[color="#41545D"][size="2"] > b` · Desktop and mobile

```
<b>Free Penguins</b>
```



Where: `tr[height="25px"]:nth-child(7) > td[bgcolor="#A9B8BF"][width="150px"] > font[color="#41545D"][size="2"] > b` · Desktop and mobile

```
<b>More City Parks</b>
```



2.4.4 Link Purpose (In Context) (Level A) Understanding 2.4.4

SERIOUS Links have no readable name

7 elements · Desktop and mobile · axe rule `link-name` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html` · Also relates to WCAG 4.1.2

Why it matters: Links made only of an icon or an image without alt text are announced as just "link". Screen reader users cannot tell where the link goes.

How to fix: Give every link visible text, or add an aria-label (aria-label="Open cart") or alt text on the image inside the link.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> These 7 navigation links wrap only an image with no `alt` text, so screen readers have nothing to announce. Add a short `alt` to each `img` that names the destination, e.g. for `nav_home.gif` the alt should convey "Home". Also replace `href="javascript:location.href='home.html';"` with a plain link to the page, and remove `onfocus="blur();"`, which stops keyboard users from seeing where focus is. Check layout/theme.liquid or sections/*.liquid for this navigation.

Reference: https://dequeuniversity.com/rules/axe/4.14/link-name?application=playwright

What the scanner found (all elements):

Fix all of the following:
Element is in tab order and does not have accessible text

Fix any of the following:
Element does not have text that is visible to screen readers
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has…

Where: `#home > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='home.html';" onfocus="blur();"><img name="nav_home" src="./img/nav_home.gif" width="88" height="27" hspace="15" border="0px"></a>
```

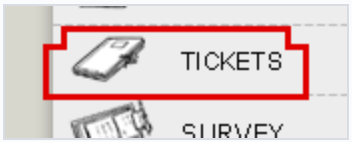


Where: `#news > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='news.html';" onfocus="blur();"><img src="./img/nav_news.gif" name="nav_news" width="90" height="21" hspace="12" border="0px"></a>
```



Where: `#tickets > a[onfocus="blur();"]` · Desktop and mobile




```
<a href="javascript:location.href='tickets.html';" onfocus="blur();"><im
g name="nav_facts" src="./img/nav_facts.gif" width="105" height="23" hsp
ace="9" border="0px"></a>
```

Where: `#survey > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='survey.html';" onfocus="blur();"><img src="./img/nav_survey.gif"
name="nav_survey" width="107" height="32" hspace="8" border="0px"></a>
```

Where: `.story:nth-child(1) > span > a[href$="news.html"][onfocus="blur();"]` · Desktop and mobile

```
<a href="news.html" onfocus="blur();"><img src="./img/morearrow.gif" width="48" height="10" alt="" b
order="0" onmouseover="this.src='./img/morearrow_a.gif'" onmouseout="this.src='./img/morearrow.gif'"
style="vertical-align: bottom"></a>
```

Where: `.story:nth-child(2) > span > a[href$="news.html"][onfocus="blur();"]` · Desktop and mobile

```
<a href="news.html" onfocus="blur();"><img src="./img/morearrow.gif" width="48" height="10" alt="" b
order="0" onmouseover="this.src='./img/morearrow_a.gif'" onmouseout="this.src='./img/morearrow.gif'"
style="vertical-align: bottom"></a>
```

Where: `.story:nth-child(3) > span > a[href$="news.html"][onfocus="blur();"]` · Desktop and mobile

```
<a href="news.html" onfocus="blur();"><img src="./img/morearrow.gif" width="48" height="10" alt="" b
order="0" onmouseover="this.src='./img/morearrow_a.gif'" onmouseout="this.src='./img/morearrow.gif'"
style="vertical-align: bottom"></a>
```

SERIOUS Links have no readable name

4 elements · Desktop and mobile · axe rule `link-name` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html` · Also relates to WCAG 4.1.2

Why it matters: Links made only of an icon or an image without alt text are announced as just "link". Screen reader users cannot tell where the link goes.

How to fix: Give every link visible text, or add an aria-label (aria-label="Open cart") or alt text on the image inside the link.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> These 4 image links have no `alt` text, so screen readers cannot say where they go. Add an `alt` to each `<img>` that names the destination, such as the page the link opens (home, news, tickets). Also remove `onfocus="blur();"`, which stops keyboard users from seeing focus. Replace `href="javascript:location.href=..."` with a normal link to the page. Check sections/*.liquid or snippets/*.liquid for this navigation.

Reference: https://dequeuniversity.com/rules/axe/4.14/link-name?application=playwright

What the scanner found (all elements):

Fix all of the following:
Element is in tab order and does not have accessible text

Fix any of the following:
Element does not have text that is visible to screen readers
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has…

Where: `#home > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='home.html';" onfocus="blur();"><img na
me="nav_home" src="./img/nav_home.gif" width="88" height="27" hspace="15"
border="0px"></a>
```



Where: `#news > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='news.html';" onfocus="blur();"><img src
="./img/nav_news.gif" name="nav_news" width="90" height="21" hspace="12" b
order="0px"></a>
```



Where: `#tickets > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='tickets.html';" onfocus="blur();"><im
g name="nav_facts" src="./img/nav_facts.gif" width="105" height="23" hsp
ace="9" border="0px"></a>
```



Where: `#survey > a[onfocus="blur();"]` · Desktop and mobile

```
<a href="javascript:location.href='survey.html';" onfocus="blur();"><img src="./img/nav_survey.gif"
name="nav_survey" width="107" height="32" hspace="8" border="0px"></a>
```

3.1.1 Language of Page (Level A) Understanding 3.1.1

SERIOUS The page does not say what language it is in

1 element · Desktop and mobile · axe rule `html-has-lang` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Screen readers use the page language to pick the right voice and pronunciation. Without it, English text may be read with the wrong accent and be hard to understand.

How to fix: Add the language to the <html> tag: <html lang="en">.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> The `<html>` tag has no `lang` attribute, so screen readers can't tell which language to read the page in. Add one that matches the store's main language, for example `<html lang="en">` for English. In Shopify this is usually in layout/theme.liquid. Check that the tag there is set up this way, and that the language code matches the page's actual content. This affects 1 element.

Reference: https://dequeuniversity.com/rules/axe/4.14/html-has-lang?application=playwright

What the scanner found (all elements):

Fix any of the following:
The <html> element does not have a lang attribute

Where: `html` · Desktop and mobile

```
<html>
```

SERIOUS The page does not say what language it is in

1 element · Desktop and mobile · axe rule `html-has-lang` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html`

Why it matters: Screen readers use the page language to pick the right voice and pronunciation. Without it, English text may be read with the wrong accent and be hard to understand.

How to fix: Add the language to the <html> tag: <html lang="en">.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> The `<html>` tag has no `lang` attribute, so screen readers can't tell which language to read the page in. Add one that matches your store's main language, for example `<html lang="en">` for English. In Shopify this tag is usually in layout/theme.liquid. Check there. Use the store's language setting if the theme offers one, so the value stays correct if the language changes. This affects 1 element.

Reference: https://dequeuniversity.com/rules/axe/4.14/html-has-lang?application=playwright

What the scanner found (all elements):

Fix any of the following:
The <html> element does not have a lang attribute

Where: `html` · Desktop and mobile

```
<html>
```

4.1.2 Name, Role, Value (Level A) Understanding 4.1.2

CRITICAL Drop-down menus have no label

1 element · Desktop and mobile · axe rule `select-name` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Screen reader users hear "combo box" with no hint of what they are choosing (size, color, country).

How to fix: Connect a <label> to the <select> with for/id, or add aria-label to the <select>.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> The `select` drop-down has no accessible name, so screen reader users won't know what it does. Add an `aria-label` to `<select onchange="location.href = this.value;">`, or connect a visible `<label>` using `for` and a matching `id`. The text should say what the menu chooses, such as the page or option it navigates to. Look in sections/*.liquid or snippets/*.liquid for this markup.

Reference: https://dequeuniversity.com/rules/axe/4.14/select-name?application=playwright

What the scanner found (all elements):

Fix any of the following:
Element does not have an implicit (wrapped) <label>
Element does not have an explicit <label>
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has no title attribute
Element's default semantics we…

Where: `select` · Desktop and mobile

```
<select onchange="location.href = this.value;">
```



CRITICAL Drop-down menus have no label

1 element · Desktop and mobile · axe rule `select-name` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html`

Why it matters: Screen reader users hear "combo box" with no hint of what they are choosing (size, color, country).

How to fix: Connect a <label> to the <select> with for/id, or add aria-label to the <select>.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES

> One `select` drop-down has no accessible name, so screen reader users can't tell what it does. Add an `aria-label` to `<select onchange="location.href = this.value;">` that says what the choice does (for example, what the menu navigates to). Alternatively, add a visible `<label>` linked with `for` and a matching `id` on the `select`. Look in sections/*.liquid or snippets/*.liquid for this markup.

Reference: https://dequeuniversity.com/rules/axe/4.14/select-name?application=playwright

What the scanner found (all elements):

Fix any of the following:
Element does not have an implicit (wrapped) <label>
Element does not have an explicit <label>
aria-label attribute does not exist or is empty
aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
Element has no title attribute
Element's default semantics we…

Where: `select` · Desktop and mobile

```
<select onchange="location.href = this.value;">
```



Best practice (not a specific WCAG requirement)

MODERATE Some content is outside the page landmarks

22 elements · Desktop and mobile · axe rule `region` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Screen reader users jump between landmarks (header, navigation, main, footer). Content outside them is easy to miss.

How to fix: Put all visible content inside landmark elements: <header>, <nav>, <main>, <footer> (or matching ARIA roles).

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> 22 elements sit outside any landmark. For `#logos`, wrap the logo links in a `<header>` (or `role="banner"`) together with the `h1` and the `.subline` paragraph. Make sure the main page content is inside one `<main>` and any closing content is inside `<footer>`. In a Shopify theme, check layout/theme.liquid and sections/*.liquid for content placed outside these wrappers.

Reference: https://dequeuniversity.com/rules/axe/4.14/region?application=playwright

What the scanner found (all elements):

Fix any of the following:
Some page content is not contained by landmarks

Where: `#logos` · Desktop and mobile

```
<p id="logos"><a href="https://www.w3.org/" title="W3
C Home"><img alt="W3C logo" src="../img/w3c.png" heig
ht="48" width="72"></a><a href="https://www.w3.org/WA
I/" title="WAI Home"><img alt="Web Accessibility Init
iative (WAI) logo" src="../img/wai.png" height="48">
</a></p>
```



Where: `h1` · Desktop and mobile

```
<h1><span class="subhead">Inaccessible Home Page</spa
n><span class="hidden"> –</span> Before and After Dem
onstration</h1>
```



Where: `.subline` · Desktop and mobile



```
<p class="subline">Improving a Web site using Web Con
tent Accessibility Guidelines (WCAG) 2.0</p>
```

Where: `#mnav` · Desktop and mobile

```
<div id="mnav" class="inaccessible">
```

Where: `#startcontent` · Desktop and mobile

```
<p class="skip" id="startcontent">Demo starts here</p>
```

Where: `tr[height="10px"]:nth-child(1)` · Desktop and mobile

```
<tr height="10px">
```

Where: `td[background="./img/border_left.gif"]` · Desktop and mobile

```
<td width="10px" background="./img/border_left.gif"><img src="./img/border_left.gif" width="10px"></
td>
```

Where: `table[height="144px"]` · Desktop and mobile

```
<table width="780px" height="144px" border="0px" cellspacing="0px" cellpadding="0px" bgcolor="WHIT
E">
```

…and 14 more elements with the same problem (full list in raw.json).

MODERATE Some content is outside the page landmarks

6 elements · Desktop and mobile · axe rule `region` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html`

Why it matters: Screen reader users jump between landmarks (header, navigation, main, footer). Content outside them is easy to miss.

How to fix: Put all visible content inside landmark elements: <header>, <nav>, <main>, <footer> (or matching ARIA roles).

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> 6 elements sit outside any landmark, including `#logos`, the `h1` and `.subline`. Wrap the logo links in a `<header>`, and put the heading and subline inside `<header>` or `<main>`. Keep the main page content in a single `<main>`, with navigation in `<nav>` and the page bottom in `<footer>`. In a Shopify theme, check layout/theme.liquid and the sections/*.liquid files that output this content.

Reference: https://dequeuniversity.com/rules/axe/4.14/region?application=playwright

What the scanner found (all elements):

Fix any of the following:
Some page content is not contained by landmarks

Where: `#logos` · Desktop and mobile

```
<p id="logos"><a href="https://www.w3.org/" title="W3C Home"><img alt="W3C logo" src="../img/w3c.png" height="48" width="72"></a><a href="https://www.w3.org/WAI/" title="WAI Home"><img alt="Web Accessibility Initiative (WAI) logo" src="../img/wai.png" height="48"></a></p>
```



Where: `h1` · Desktop and mobile

```
<h1><span class="subhead">Inaccessible News Page</span><span class="hidden"> –</span> Before and After Demonstration</h1>
```



Where: `.subline` · Desktop and mobile

```
<p class="subline">Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0</p>
```



Where: `#mnav` · Desktop and mobile

```
<div id="mnav" class="inaccessible">
```

Where: `#page` · Desktop and mobile

```
<div id="page">
```

Where: `#meta-footer` · Desktop and mobile

```
<div id="meta-footer" class="meta">
```

MODERATE The page has no main landmark

1 element · Desktop and mobile · axe rule `landmark-one-main` · Page: `https://www.w3.org/WAI/demos/bad/before/home.html`

Why it matters: Without a <main> element, keyboard and screen reader users cannot jump straight to the main content.

How to fix: Wrap the main content of each page in a single <main> element.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> The page has no main landmark, so screen reader users can't jump straight to the page content. Check `layout/theme.liquid` and add one <main> element around the `{{ content_for_layout }}` area, between the header and footer. Don't put the header or footer inside it. Make sure no other <main> exists in `sections/*.liquid` or `snippets/*.liquid`, since each page should have only one.

Reference: https://dequeuniversity.com/rules/axe/4.14/landmark-one-main?application=playwright

What the scanner found (all elements):

Fix all of the following:
Document does not have a main landmark

Where: `html` · Desktop and mobile

```
<html>
```

MODERATE The page has no main landmark

1 element · Desktop and mobile · axe rule `landmark-one-main` · Page: `https://www.w3.org/WAI/demos/bad/before/news.html`

Why it matters: Without a <main> element, keyboard and screen reader users cannot jump straight to the main content.

How to fix: Wrap the main content of each page in a single <main> element.

> **Suggested fix for this page** AI-DRAFTED, CHECKED BY AUTOMATIC RULES
>
> The page has no main landmark, so screen reader users can't jump straight to the page's primary content. Add one `<main>` element around the central content, between the header and footer. In Shopify this is usually in layout/theme.liquid, wrapped around `{{ content_for_layout }}`. Check that no other `<main>` exists and that `role="main"` isn't already used incorrectly.

Reference: https://dequeuniversity.com/rules/axe/4.14/landmark-one-main?application=playwright

What the scanner found (all elements):

Fix all of the following:
Document does not have a main landmark

Where: `html` · Desktop and mobile

```
<html>
```

What an automated scan cannot check

Automated tools find only part of the accessibility barriers on a site. These need a person:

Manual review: pending. The checks in the table below have not been done yet.

Check	What is tested
Keyboard-only use	Every menu, button, form, pop-up, and the checkout can be reached and used with Tab, Enter, Space, and the arrow keys, with no keyboard traps.
Visible focus and focus order	You can always see where the keyboard focus is, and it moves in a logical order.
Screen reader (VoiceOver)	Pages, products, forms, errors, and cart updates are announced clearly and in a sensible order.
Zoom and reflow	Text zoomed to 200% and a 320px-wide screen still show all content without sideways scrolling.
Quality of text alternatives	Alt text exists, but does it describe the product or the purpose of the image? A scan can only check that it is there.
Video and audio	Captions are accurate, and transcripts or audio descriptions exist where needed.
Forms and error messages	Errors are explained in text, tell people how to fix them, and are announced to screen readers.
Motion, timing, and pop-ups	Carousels and animations can be paused, time limits can be extended, and pop-ups can be closed with the keyboard.
Consistency	Navigation and repeated components look and work the same way across pages.

The scan also flagged 2 checks it could not decide automatically ("needs review"); they are listed in raw.json.

Important notice

This report lists issues found by automated accessibility testing (axe-core and Lighthouse) on the pages and the date listed above. It is not a full accessibility audit, not a certification, and not a statement that the site meets or fails the ADA, WCAG, or any other law or standard. It is not legal advice. Automated tools cannot detect every accessibility barrier, and websites change over time. Following the recommendations will make the site easier to use, but no audit can guarantee that a business will not face a complaint or lawsuit. Fix suggestions written with AI can be wrong; test every change before you publish it. For legal questions, consult a qualified attorney.