

Accessibility audit: demo.accessaudit.dev

20 October 2026 · prepared for a sample · 4 page(s) checked

4 of 4 pages could be read. All of them carry at least one problem from our fixed list: 14 different problem(s) in total, and 0 pages with error-level findings.

This is an automated audit, not a certificate. Every finding below is a fact about the HTML that a machine can prove, and each one quotes the markup it is about. A machine cannot judge whether alt text is *useful*, cannot see whether two colours have enough contrast, and cannot use the page with a screen reader. The list of what this audit did not check is on page 2, and it matters as much as the findings.

What was checked

Page	Result	Findings
https://demo.accessaudit.dev/	read	9
https://demo.accessaudit.dev/services.html	read	6
https://demo.accessaudit.dev/about.html	read	3
https://demo.accessaudit.dev/contact.html	read	5

What we found

Grouped by problem, most severe and most widespread first. The markup quoted is taken from the pages named, so every line can be checked by hand.

ERROR Image without alt text

On all 4 pages (/ , /about.html, /contact.html, /services.html).

Why it matters. A screen reader announces the file name instead of the picture, so a reader who cannot see the image learns nothing from it.

How to fix it. Add an alt attribute describing the image in context. If the image is purely decorative, use an empty alt (alt="") so it is skipped on purpose.

```

```

```

```

```

```

ERROR Embedded frame without a title

On 1 page of 4 pages (/).

Why it matters. An embedded frame is announced as 'frame' with no title, so the reader cannot tell what it contains or whether to enter it.

How to fix it. Add a title attribute describing the embedded content.

```
<iframe src="https://example.com/map" width="100%" height="240"></iframe>
```

ERROR Link built out of JavaScript

On 1 page of 4 pages (/).

Why it matters. A link that is really a script breaks middle-click, 'open in new tab' and the browser's link list, and it can be unreachable by keyboard.

How to fix it. Use a real URL, or use a <button> if it triggers an action.

```
<a href="javascript:window.print()">Print this page</a>
```

ERROR Link with no accessible name

On 1 page of 4 pages (/services.html).

Why it matters. A link with no readable text is announced as 'link' with nothing to go on, so it is a dead end.

How to fix it. Give the link text, or an aria-label, or a nested image with meaningful alt text.

```
<a href="contact.html" target="_blank"></a>
```

ERROR Positive tabindex

On 1 page of 4 pages (/services.html).

Why it matters. A positive tabindex pulls keyboard focus out of document order, so tabbing through the page jumps around in a way that does not match what is on screen.

How to fix it. Use tabindex="0" or nothing at all, and put elements in the right order in the markup instead.

```
<p tabindex="3"><a href="contact.html">Ask us about a job</a></p>
```

ERROR Button with no accessible name

On 1 page of 4 pages (/contact.html).

Why it matters. A button with no accessible name is announced as 'button'; a screen reader user cannot tell what pressing it does.

How to fix it. Add visible text or an aria-label to the button.

```
<button type="submit"></button>
```

ERROR Form field without a label

On 1 page of 4 pages (/contact.html).

Why it matters. A form field with no label is announced as 'edit text' with no clue what belongs in it, so the form cannot be filled in reliably.

How to fix it. Connect a `<label for="...">` to the field id, or wrap the field in a label, or add aria-label.

```
<input type="email" name="email" placeholder="Email address">
```

ERROR Label points at an id that does not exist

On 1 page of 4 pages (/contact.html).

Why it matters. An aria-labelledby or aria-describedby points at an id that does not exist, so the name it was meant to supply is silently missing.

How to fix it. Fix the spelling of the id, or add the missing element.

```
<span aria-labelledby="opening-hours">Open 7am to 9pm, seven days.</span>
```

WARNING Zoom is blocked

On all 4 pages (/ , /about.html, /contact.html, /services.html).

Why it matters. The page forbids zooming, so anyone who needs to enlarge the text cannot - on a phone this is often the difference between readable and not.

How to fix it. Remove user-scalable=no and maximum-scale from the viewport meta tag.

```
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
```

WARNING Heading levels skip

On 3 pages of 4 pages (/ , /about.html, /services.html).

Why it matters. Headings that jump levels make the page's outline wrong, and that outline is how screen reader users navigate a page by sections.

How to fix it. Use headings in order (h1, h2, h3...). If you need smaller text, use CSS, not a lower heading level.

```
<h3>Boiler servicing</h3>
```

```
<h3>Servicing and safety</h3>
```

```
<h3>The team</h3>
```

WARNING Link opens a new window without warning

On 2 pages of 4 pages (/, /services.html).

Why it matters. The link opens in a new window or tab with no warning, and the reader has no way back except the browser's own back button - which does not return them.

How to fix it. Say so in the link text ("opens in a new tab") or remove target="_blank".

```
<a href="https://example.com/terms" target="_blank">Terms and conditions</a>
```

```
<a href="contact.html" target="_blank"></a>
```

WARNING Link text does not say where it goes

On 1 page of 4 pages (/).

Why it matters. Link text like 'read more' is meaningless out of context, and screen reader users often browse a page by its list of links alone.

How to fix it. Describe the destination, for example "read the pricing guide" instead of "click here".

```
<a href="services.html">Read more</a>
```

WARNING Table without header cells

On 1 page of 4 pages (/).

Why it matters. Without header cells, a screen reader cannot say which column a value belongs to, so a data table reads as a list of loose numbers.

How to fix it. Use <th scope="col"> or scope="row" for the header cells of the table.

```
<table> <tr><td>Callout, weekday</td><td>£65</td><td>First hour included</td></tr> <tr>
<td>Callout, weekend</td><td>£95</td><td>First hour included</td></tr> <tr><td>Boiler
service</td><td>£85</td><td>
```

WARNING Duplicate id attributes

On 1 page of 4 pages (/).

Why it matters. Two elements share an id, so a label or a skip link pointing at it may reach the wrong place - and duplicate ids can break scripts and in-page links.

How to fix it. Make every id unique on the page.

```
<p id="quote">"Tidy, punctual, and the price was the price." – another customer.</p>
```

What this audit did not check

- **Colour contrast.** It needs the rendered colours, not the markup; a stylesheet can override anything we can see.
- **Whether text is understandable.** Alt text can be present and useless; link text can be grammatical and meaningless in context.
- **Keyboard use.** We can spot focus traps by their markup, not by tabbing through the page.
- **Screen reader output,** including reading order, which depends on how a browser builds the accessibility tree.
- **Anything behind a login,** a paywall, a country block, or a form submission.
- **PDFs and other documents** linked from the pages.
- **Content that only appears after interaction:** menus, modals, carousels, cookie banners.
- **Whether the pages we chose are representative.** We checked a handful of pages, not the site.

An automated audit can find and prove mechanical failures quickly, and that is all it can do. Any statement that a site is compliant, or that it meets WCAG, requires a person to test it and a named organisation to take responsibility for the claim. We do neither.

Suggested order of work

- 1. Image without alt text** – 4 pages. Add an alt attribute describing the image in context. If the image is purely decorative, use an empty alt (alt="") so it is skipped on purpose.
- 2. Zoom is blocked** – 4 pages. Remove user-scalable=no and maximum-scale from the viewport meta tag.
- 3. Heading levels skip** – 3 pages. Use headings in order (h1, h2, h3...). If you need smaller text, use CSS, not a lower heading level.
- 4. Link opens a new window without warning** – 2 pages. Say so in the link text ("opens in a new tab") or remove target="_blank".

Fix in this order because the first items affect the most pages, not because they are the most serious in principle. An accessibility practitioner may disagree, and on a real remediation plan they should be asked.

Method

Each page was fetched once, as a browser would fetch it, with the crawler obeying the site's robots.txt. The same checking engine as our free checker (version 0.2.0) examined the HTML returned, and nothing on the page was executed or interacted with. Pages were chosen to be varied rather than random: the home page, then pages listed in the site's sitemap and links found on the home page, skipping utility and account pages. Findings are

mechanically determined: the same page checked twice by the same engine version gives the same findings.

Markproof Audit by Stochastic Technologies, George Town, Cayman Islands · hello@accessaudit.dev · accessaudit.dev/audit.html

Prepared 20 October 2026 at 14:46 UTC. Engine 0.2.0. Each finding links to an explanation with the same markup and the same fix at accessaudit.dev/checks/. You may give this document to clients, contractors and auditors in full. Please do not edit it: every claim in it is tied to the pages checked here, and an edited copy would no longer be true.