auditzo

PUBLIC SAMPLE / HUMAN-REVIEWED ACCESSIBILITY

Manual WCAG 2.2 AA Technical Accessibility Evidence Review

Representative report format, methodology and illustrative findings

EXAMPLE DELIVERABLE • NO CLIENT WEBSITE OR LIVE EVIDENCE

WHAT THIS SAMPLE SHOWS

Realistic deliverable structure, without real client data.

This controlled demonstration illustrates how Auditzo documents human-reviewed WCAG-oriented findings, connects observations to user impact, and supports remediation and retesting.

> Transparency: All example findings, UI diagrams, evidence identifiers and sample statuses in this document are illustrative. No client website was tested for this PDF; no original client screenshots or evidence artifacts are reproduced.

Scope-based service • Evidence-first • Human judgment • No legal certification

01 / DECISION GUIDE

What makes a manual review different

An automated tool can surface useful candidates. A human reviewer investigates the user experience and determines what the available evidence supports.

Automated screening	Human-reviewed accessibility assessment
Rule-based issues and diagnostics	Manual validation of selected candidate issues
Captured page state	Defined pages, journeys and interaction states
Automated focus and geometry indicators	Keyboard behavior and screen-reader-oriented checks
General correction guidance	Finding-level user impact, implementation direction and retest criteria
Technology-generated output	Curated, traceable technical evidence package

Our position

Auditzo does not equate a scanner result with a confirmed accessibility barrier. Final technical findings are selected after human review, evidence reconciliation and contextual WCAG mapping.

The deliverable is intended for

Business and compliance teams

Understand the main access barriers, scope, uncertainty and priorities.

Developers and agencies

Receive reproducible findings and specific implementation/retesting guidance.

Counsel and advisors

Review dated technical observations, without treating the report as a legal conclusion.

Product and QA teams

Track shared-component issues and verify that selected fixes work.

02 / ILLUSTRATIVE SCOPE

How a real engagement is scoped

The actual pages, states, assistive technologies and deliverables are agreed before testing. The following is an example, not a completed customer engagement.

Journey or component	Representative testing focus
Homepage and global navigation	Navigation names, keyboard sequence, focus management
Search or listing experience	Card links, controls, embedded widgets, keyboard actions
Contact or inquiry form	Labels, errors, validation, required fields
Dialog or third-party component	Accessible names, focus containment and dismissal
Narrow-viewport experience	320 CSS px reflow, text clipping, content visibility

Scope boundaries that should be recorded

- Representative URLs and defined states, not an unexplained claim of full-site coverage.
- Browsers, viewport settings and assistive technologies actually used.
- Safe form validation parameters; no unapproved submission of real personal data.
- Third-party integration constraints, exclusions and known test limitations.

The number of affected pages or findings shown in an actual report depends on the tested website and the agreed scope. This public sample intentionally contains no client-level outcome totals.

Layered review methodology

This is based on Auditzo's documented manual delivery approach: automated detection is an input, not the final decision.

1. **Automated candidate detection**
   Use appropriate accessibility tools to identify elements and states for investigation.
2. **Human treatment decision**
   Confirm, reject or defer candidate findings after evaluating context.
3. **Keyboard navigation**
   Test reachability, focus order, operation, dismissal and focus return.
4. **Assistive technology checks**
   Use screen-reader-oriented review for names, roles, states and meaningful content.
5. **Browser accessibility inspection**
   Inspect accessible names, labels, relationships and the computed accessibility tree.
6. **Responsive/reflow review**
   Test selected narrow viewport states and evaluate visible-content loss.
7. **Evidence curation**
   Preserve the finding, affected state, reproduction instructions and support artifacts.

An automated tool may detect a missing label, but only the review of the actual component and its interaction determines the proper finding language and remediation guidance.

Decisions a client can act on

The items below demonstrate a decision-maker summary format. They are illustrative scenarios, not confirmed issues on a live website.

01 Navigation clarity

Check whether controls expose meaningful names, correct roles and states.

02 Form understanding

Provide persistent labels and instructions instead of placeholder-only context.

03 Keyboard operation

Confirm that core actions can be reached and operated without a mouse.

04 Responsive usability

Verify that meaningful content remains visible at narrow viewport widths.

Representative prioritization

Sample finding	Example impact	Action
DEMO-01 Mobile menu name	Screen-reader orientation	Confirm accessible name and expanded state
DEMO-02 Form labels	Input understanding	Add persistent labels and validate
DEMO-03 Listing action focus	Keyboard access to core actions	Investigate Tab path and widget ownership
DEMO-04 Narrow viewport clipping	Loss of visible content	Fix responsive layout, then retest

Priority is contextual. In live engagements, severity depends on the affected journey, user impact, reproducibility, and available evidence. This sample does not claim that any of the illustrative issues were observed in production.

05 / EXAMPLE FINDING REGISTER

How individual findings are documented

A practical issue register links a concise explanation to the affected state, recommended fix and verification route.

Ref / example	Candidate WCAG criteria	Evidence needed
DEMO-01 Unnamed menu control	4.1.2 Name, Role, Value	Accessibility tree + screen-reader-oriented notes
DEMO-02 Placeholder-only form	3.3.2 Labels or Instructions; 1.3.1	Form states + accessible-name inspection
DEMO-03 Unreachable listing actions	2.1.1 Keyboard; 2.4.3 Focus Order	Natural Tab path and interaction replay
DEMO-04 Narrow-state clipping	1.4.10 Reflow	320 CSS px viewport comparison

Every confirmed client finding should specify

Affected state

Page, viewport, control and interaction path.

Observed behavior

Exactly what the reviewer could reproduce.

User impact

Which users may be disadvantaged and how.

Corrective direction

Practical implementation approach and dependencies.

Verification criteria

Conditions for marking a fix as retested.

Evidence reference

Stable ID, file inventory and relevant notes.

Illustrative WCAG references here identify plausible review criteria; a real report must map each confirmed observation conservatively to the applicable success criterion.

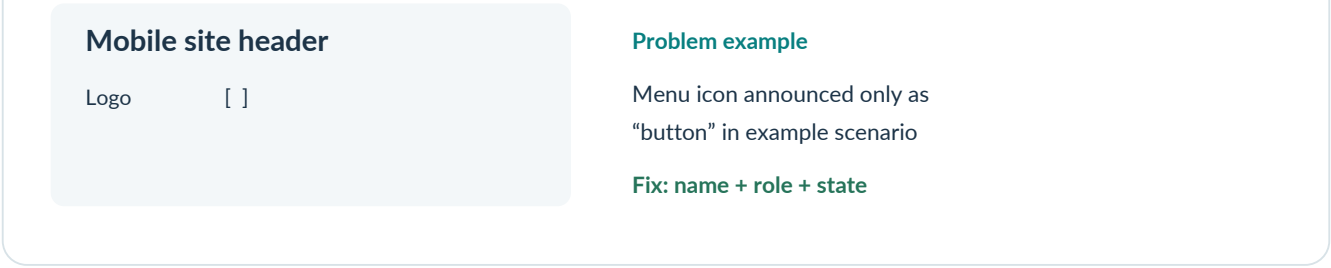
DEMO-01 Menu control lacks a useful name

Illustrative example only • No original customer screenshot or evidence.

Example observation

In this hypothetical mobile state, a menu toggle is visible but the accessibility tree does not expose a useful name. Its operation and expanded/collapsed announcement would need to be checked.

Example interface • diagram only



Why it can matter

A screen-reader user may encounter a control announced simply as "button," without knowing that it opens the site navigation. Keyboard and state behavior must be verified in the live component.

Recommended implementation direction

- Use a native button with a clear accessible name, e.g., "Open navigation."
- Expose state appropriately (such as **aria-expanded** when applicable).
- Verify keyboard activation, Escape/dismissal and the resulting focus location.

Retest criteria

Inspect computed name and role; run a screen-reader-oriented check; confirm the menu operates and its expanded state is exposed in the tested mobile state.

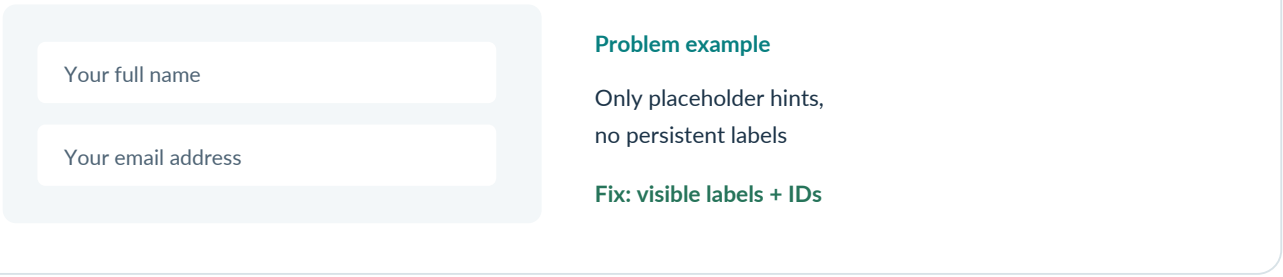
DEMO-02 Form fields need persistent labels

Illustrative example only • No live form was examined for this document.

Example observation

A hypothetical inquiry form uses placeholder text to communicate the purpose of required inputs. Once the visitor starts typing, the contextual hint may disappear.

Example interface • diagram only



Why it can matter

Users with memory, cognitive, low-vision or assistive technology needs may struggle to identify the intended field after a value is entered.

Recommended implementation direction

- Provide a persistent visible label and a reliable programmatic label association.
- Keep required status and instructional hints available after entering text.
- Review real validation messages and the accessible names after interaction.

Retest criteria

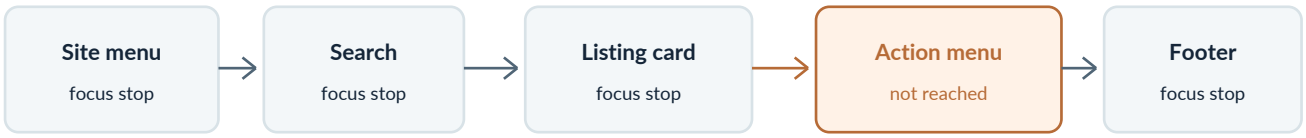
Using keyboard and a screen reader, confirm each selected input communicates the intended purpose and retains understandable context after entry.

DEMO-03 Listing actions need keyboard review

Illustrative example only • The diagram below is not an observed keyboard trace.

Example observation

An action menu associated with a listing card appears visually available but is not encountered in the hypothetical sequential Tab route. A human reviewer must reproduce the behavior before confirming an accessibility issue.



Illustrative focus-path diagram; not a recorded keyboard traversal.

What a reviewer must establish

- Whether the element is expected to be keyboard reachable in the relevant state.
- Whether a different supported keyboard pattern exposes equivalent functionality.
- Whether a third-party widget controls focus behavior and remediation.

Recommended implementation direction

Ensure each actionable control can be reached and operated through a supported keyboard interaction, provides meaningful status, and returns focus appropriately after a dialog or submenu closes.

Retest criteria

Repeat the complete interaction path without a mouse, record the focus sequence and test activation/dismissal in the defined browser and page state.

09 / DEMONSTRATION FINDING

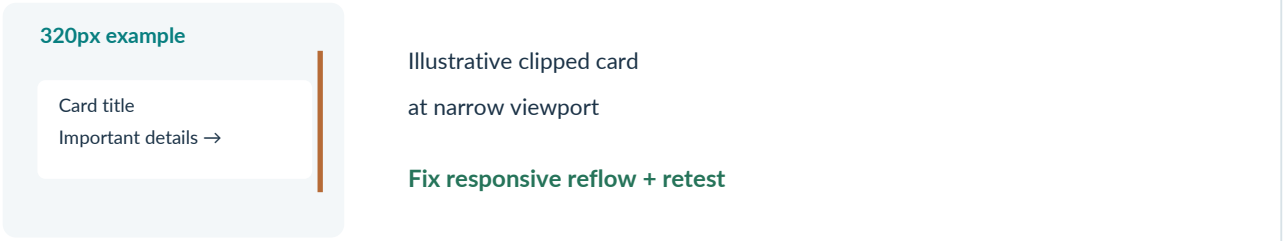
DEMO-04 Content clips at narrow widths

Illustrative example only • This is a schematic, not a captured reflow defect.

Example observation

A hypothetical content card loses visible details at a narrow viewport. The reviewer would distinguish a true information loss or two-dimensional scrolling problem from harmless visual changes.

Example interface • diagram only



Why it can matter

Users who magnify text or rely on small viewports may be unable to find information or controls if the layout obscures meaningful content.

Recommended implementation direction

- Remove fixed-width assumptions and allow card text to wrap naturally.
- Avoid clipping important information behind non-scrollable containers.
- Preserve functional controls and readable content in selected responsive states.

Retest criteria

Verify the defined content and controls at 320 CSS pixels in the chosen configuration, with no unaddressed loss of content or functionality.

What makes a manual finding defensible

For a real engagement, finding references should lead reviewers from a written conclusion to reproducible context and preserved supporting material.

Evidence field	What should be recorded
Finding reference	Unique finding ID used throughout report and remediation register
Review context	Target page, viewport, relevant browser and date/state
Reproduction path	Steps required to observe the relevant behavior
Human-review record	Keyboard/screen-reader/DevTools observations as applicable
Evidence artifact	Selected screenshot, log, transcript or measurement file
Treatment decision	Confirmed, reviewed-not-confirmed, candidate-only or unresolved
Validation instruction	Concrete steps for testing the proposed fix

Illustrative evidence reference

DEMO-03-E01 — hypothetical keyboard-flow capture. This is an example naming convention only. No artifact has been collected, attached or hashed for this public sample.

Integrity and delivery note

A client-specific report may reference a separate curated evidence package and index. The public sample contains no real evidence, cryptographic verification hashes, or private file paths. Actual retention and access controls are defined by the engagement.

Plan fixes around components, not only pages

The practical value of a manual report is also the ability to identify repeated patterns and create a sensible verification plan.

Priority	Workstream	Representative action
1 - urgent	Core interactions	Confirm and address blockers to main navigation, forms and action controls
2 - shared	Templates and components	Resolve repeated label, focus and link-name issues in common UI elements
3 - dependent	Third-party integrations	Coordinate widget, iframe and vendor-controlled remediation
4 - verify	Regression and retesting	Repeat affected keyboard, screen-reader and responsive states

Example developer handoff

Ticket content

Finding ID, page state, affected component, WCAG-oriented reference and reproduction steps.

Suggested owner

Website team for owned components; agency or third-party vendor when required.

Definition of done

Observed behavior corrected and specific accessibility checks completed.

Evidence of closure

Retest notes and updated technical evidence retained under the agreed process.

Fixing a shared component may address several page-level findings. A manual report should identify this relationship rather than imply every finding requires an independent implementation task.

What happens after remediation

A finding should not be closed merely because a CSS or HTML change was deployed. The selected user journey needs to be retested.

Example check	Verification approach	This sample
Menu name and state	Accessibility tree + keyboard + screen reader	Not tested
Form labels	Read names, enter values, review errors	Not tested
Listing action focus	Natural Tab path, activate and dismiss	Not tested
Responsive card content	Reflow at defined viewport; content visibility	Not tested
Updated evidence references	Capture corrected states if in scope	Not applicable

Status meanings used in actual projects

Open

Issue remains to be addressed or verified.

In remediation

Implementation or third-party coordination is in progress.

Ready for retest

Change deployed; behavior not yet confirmed.

Verified in scope

Selected expected behavior reproduced after the fix.

> An issue fixed in one browser or page state does not automatically prove complete website conformance. Closure claims must match the actual retesting scope.

From technical evidence to an actionable audit

The public sample demonstrates reporting format, not a conclusion about any website or a promise of legal compliance.

What a real scoped engagement can deliver

- Agreed representative pages, journeys, viewport and testing boundaries.
- Human-reviewed findings with practical user impact and WCAG-oriented references.
- Curated evidence references, reproduction instructions and remediation guidance.
- Prioritized technical roadmap and a defined retesting approach.

What this sample does not include

- Real client reports, identifying data or proprietary screenshots.
- Executed test results, original audit evidence or verified hashes.
- Legal advice, ADA/WCAG certification or a complete conformance determination.
- Unscoped monitoring, full-site crawling or guaranteed remediation outcomes.

Request a scoped manual accessibility review

Share the website, priority user journeys and your expected outcome. Auditzo can propose a defined human-review scope and deliverables.

auditzo.com/wcag-2-2-aa-website-audit
auditzo.com/contact-us

Publication note

This document is a controlled demonstration derived from Auditzo's documented reporting methodology. It may be distributed publicly. It must not be represented as the underlying report or evidence package from a real customer engagement.