

PUBLIC SAMPLE DELIVERABLE

Sample Accessibility Audit Report

Northstar Checkout - fictional demonstration product

Prepared by	Khushwant Parihar
Target	WCAG 2.2 Level A and AA
Version	1.0
Issued	July 12, 2026

Northstar Checkout is fictional and exists only to demonstrate report structure. This is not a client endorsement, conformance claim, or legal opinion. An accessible HTML equivalent is available at [accessibility.build/sample-audit-report](#).

Executive summary

The fictional checkout sample contains four confirmed accessibility findings across modal focus, error communication, accessible names, and responsive reflow. Findings are prioritized by user impact, not by automated rule count. This limited sample cannot establish full product conformance.

Critical	1	High	1	Medium	2
----------	---	------	---	--------	---

Scope

- Checkout entry page
- Delivery address form
- Payment method step
- Order review dialog
- Confirmation state

Test environments

- Chrome with NVDA on Windows
- Safari with VoiceOver on macOS
- Keyboard-only interaction
- Browser zoom and responsive reflow review
- Automated axe-based checks used as supporting evidence

Limitations

- This sample covers five representative states, not every possible checkout configuration.
- No disabled-user research is represented in this fictional sample.
- Legal compliance and full product conformance cannot be established from this sample.

Detailed findings

Each finding includes a confirmed barrier, criterion mapping, user impact, reproducible evidence, and remediation direction. Verification is a separate retest state.

NS-001	Critical severity	Open in sample
--------	-------------------	----------------

Order review dialog does not keep keyboard focus within the modal

Mapped criteria

2.1.2 No Keyboard Trap; 2.4.3 Focus Order

User impact

Keyboard and screen-reader users can move into content behind the open dialog and lose the relationship between the review task and available actions.

Reproduction evidence

- 1 Open the order review dialog from the payment step.
- 2 Press Tab repeatedly after focus reaches the final dialog action.
- 3 Focus moves to the page navigation behind the modal while the modal remains visible.

Remediation guidance

Use a native dialog implementation or a tested modal pattern. Move focus to a meaningful element when the dialog opens, constrain sequential focus while it is modal, support Escape where appropriate, and restore focus to the invoking control when it closes.

Verification note

Retest the corrected behavior in the agreed environment. A code change alone does not close the finding.

Address validation errors are not announced after submission

Mapped criteria

3.3.1 Error Identification; 4.1.3 Status Messages

User impact

A screen-reader user receives no programmatic notification that submission failed or which fields need correction.

Reproduction evidence

- 1 Leave the postal code and city fields empty.
- 2 Activate Continue.
- 3 Visual error text appears, but focus remains on Continue and no status or error summary is announced.

Remediation guidance

Provide an error summary with links to invalid fields, move focus to the summary after unsuccessful submission, associate each field error using aria-describedby, and maintain aria-invalid until the value is corrected.

Verification note

Retest the corrected behavior in the agreed environment. A code change alone does not close the finding.

Visible payment option text is not included in the accessible name

Mapped criteria

2.5.3 Label in Name; 4.1.2 Name, Role, Value

User impact

Speech-input users cannot reliably activate the option by saying the visible label, and screen-reader output differs from the text presented visually.

Reproduction evidence

- 1 Inspect the option labelled Pay by bank transfer.
- 2 Its accessible name is Transfer payment method.
- 3 The visible words Pay by bank transfer are not contained in the computed name.

Remediation guidance

Use the visible text as the control's accessible name. Prefer an associated native label and remove the conflicting aria-label unless additional context is genuinely required.

Verification note

Retest the corrected behavior in the agreed environment. A code change alone does not close the finding.

Order summary requires horizontal scrolling at 400 percent zoom

Mapped criteria

1.4.10 Reflow

User impact

People using browser zoom must repeatedly scroll in two dimensions to compare item names, quantities, and totals.

Reproduction evidence

- 1 Set a 1280 CSS pixel viewport and zoom the browser to 400 percent.
- 2 Navigate to the order review step.
- 3 The summary table retains a fixed minimum width and creates horizontal page scrolling.

Remediation guidance

Replace the fixed-width layout with a responsive summary pattern. Allow labels and values to wrap, move secondary metadata below the item name, and test at an effective viewport width of 320 CSS pixels.

Verification note

Retest the corrected behavior in the agreed environment. A code change alone does not close the finding.

Delivery and interpretation

Severity is a prioritization aid, not a substitute for reading the user impact and evidence. The final engagement report also identifies scope changes, unresolved dependencies, retest results, and residual risk.

Status model

- Detected - a tool or reviewer identified a potential issue.
- Confirmed - a practitioner validated the barrier and evidence.
- Remediation in progress - an implementation change is under review.
- Verified - the original failure no longer reproduces in the agreed retest environment.
- Residual risk - the issue is partly addressed, out of scope, or dependent on another party.

Important qualification

An audit reflects the agreed sample, product version, environment, standard, and date. Product changes may create new barriers. Accessibility.build does not describe a tool score, one-time scan, or limited sample as certification of legal compliance.

Accessibility.build

Founder-led accessibility audits, remediation, training, and practical WCAG tools.

contact@accessibility.build | accessibility.build