

Accessibility regression triage

Controlled demonstration report for the fictional Northstar Studio campaign review page.

<div>DEMONSTRATION PROJECT</div> <div>Northstar Studio synthetic test site</div>	<div>REPORT DATE</div> <div>October 31, 2026</div>
<div>COMPARED STATES</div> <div>Baseline → Controlled regression → Remediated</div>	<div>REVIEW TYPE</div> <div>Local automated HTML heuristic checks</div>

Executive summary

The baseline page produced zero findings from the project’s four selected HTML heuristic checks. A controlled website change then removed the primary button’s accessible name, removed the email field’s associated label, removed the informative image’s alt attribute, and changed a section heading from h2 to h3. The checker identified those four potential accessibility regressions: one critical, two high, and one medium under the demonstration’s triage model. Each finding is presented with a stable identifier, affected element, source location, observed problem, prioritization rationale, reproduction steps, controlled evidence, suggested remediation, and developer acceptance guidance. The simulated remediation restored all four patterns. A repeated run of the same checker against the remediated page reported zero remaining findings, and that retest state is recorded below.

<div>Critical</div> <div>1</div>	<div>High</div> <div>2</div>	<div>Medium</div> <div>1</div>	<div>Retest open</div> <div>0</div>
----------------------------------	------------------------------	--------------------------------	-------------------------------------

Agency takeaway

This proposed workflow could give development teams a consistent way to investigate selected accessibility regressions introduced by routine website releases: preserve the before-and-after evidence, route prioritized observations to developers, and document what the same bounded checks report after a fix. It supplements—not replaces—human accessibility testing and release-specific judgment.

Interpretation boundary: These are genuine results from a limited checker against a synthetic page. They are not a complete accessibility audit, do not establish WCAG compliance, and do not represent a manual NVDA/Chrome or qualified specialist review.

Test scope

- Three local HTML files under direct project control: baseline, controlled regression, and remediated states.
- Four selected deterministic patterns: accessible button name, form-control label association, image alt attribute, and sequential heading levels.
- Comparison of finding identities between versions, using the checker’s rule name as the stable identity.
- Local machine-readable evidence and a human-readable demonstration transcript.
- No network requests, customer data, third-party scans, production deployment, manual keyboard review, or assistive-technology review.

Severity and prioritization rationale

Critical is used when the demonstrated change can prevent understanding or operation of the primary action for users relying on accessible names. **High** indicates material loss of programmatic information for a form control or informative image. **Medium** indicates structural degradation that may impair navigation or comprehension but does not block the demonstrated primary action by itself. These labels support demonstration triage and require context-specific human review before use on real work.

Technical methodology

The demonstration runs in a controlled synthetic test environment so the baseline, introduced changes, and remediated state are known in advance. Automated heuristic checks are intentionally separated from human accessibility testing.

<p>1 · Establish baseline</p> <p>Run the checker against demo/baseline/index.html. The selected patterns produce zero findings.</p>	<p>2 · Introduce controlled changes</p> <p>Use demo/regression/index.html, where four known HTML changes create four detectable patterns.</p>
<p>3 · Compare finding identities</p> <p>Compare rule identities across baseline and regression outputs. A rule present only after the change is recorded as introduced.</p>	<p>4 · Assign demonstration severity</p> <p>The checker maps each selected rule to a fixed triage level based on the demonstrated interaction or information loss. This is prioritization, not a legal conclusion.</p>
<p>5 · Simulate remediation</p> <p>The remediated state restores button text, the label relationship, image alternative text, and sequential heading order.</p>	<p>6 · Generate final retest</p> <p>Run the identical checker against demo/remediated/index.html. The committed result reports zero findings.</p>

Current versus proposed capability

<p>Current demonstration capabilities</p> <ul style="list-style-type: none">• Controlled synthetic HTML testing• Four selected automated heuristic checks• Regression comparison and demonstration severity• Developer-oriented findings• Simulated remediation and retesting• Sample report generation	<p>Proposed future service capabilities</p> <ul style="list-style-type: none">• Broader accessibility regression testing• Manual keyboard testing• NVDA/Chrome review• Qualified specialist triage• Recurring release-batch evaluation• White-label reporting and managed retesting <p>Status: proposed only; not represented as operational.</p>
---	--

Findings

CRITICAL • MV-001

Primary action button has no accessible name

RETEST: PASS

AFFECTED ELEMENT OR COMPONENT

“Request a review” primary action button.

SOURCE LOCATION

demo/regression/index.html, line 16.
Exact column data is not generated by the demonstration checker.

OBSERVED ACCESSIBILITY PROBLEM

The button contains only an arrow inside `aria-hidden="true"`, so the selected heuristic finds no accessible name.

PRIORITY RATIONALE

Critical in this demonstration because the primary action’s purpose is not exposed by text the checker can treat as an accessible name.

REPRODUCTION INSTRUCTIONS

1. Open the regression file.
2. Locate the line 16 button.
3. Run `node tools/a11y-check.mjs demo/regression/index.html --json`.
4. Observe rule `button-name`.

SUGGESTED REMEDIATION

Restore descriptive visible text such as “Send review request,” or provide an accurate accessible name when an icon-only design is genuinely required.

CONTROLLED REGRESSION EVIDENCE

Regression line 16: `<button type="button"><span aria-hidden="true">></span></button>`
Checker message: “Button has no accessible name.”

RETEST OUTCOME

PASS. Remediated line 16 restores visible text: `<button type="button">Send review request</button>`. The repeated controlled checker reports zero total findings.

Suggested developer ticket

Restore an accessible name for the request action. Acceptance: the control exposes an accurate name and the controlled `button-name` finding is absent on retest.

AFFECTED ELEMENT OR COMPONENT

Work email form field.

SOURCE LOCATION

demo/regression/index.html, line 15.
Exact column data is not generated by the demonstration checker.

OBSERVED ACCESSIBILITY PROBLEM

The input keeps a placeholder but has no associated `label` or other accessible-name relationship recognized by the selected heuristic.

PRIORITY RATIONALE

High because the demonstrated form control loses programmatic identification even though visual placeholder text remains.

REPRODUCTION INSTRUCTIONS

1. Open the regression file.
2. Locate the line 15 email input.
3. Run the checker against the regression file.
4. Observe rule `form-label`.

SUGGESTED REMEDIATION

Restore a visible `label` whose `for` value matches the input `id`. Do not rely on placeholder text as the label.

CONTROLLED REGRESSION EVIDENCE

```
Regression line 15: <input id="email" type="email" autocomplete="email" placeholder="Work email">  
Checker message: "Form control has no programmatically associated label."
```

RETEST OUTCOME

PASS. Remediated line 15 restores `<label for="email">Work email</label>` and the matching input. The repeated checker reports zero total findings.

Suggested developer ticket

Restore the visible work-email label and its `for/id` relationship. Acceptance: the selected `form-label` finding is absent on retest.

AFFECTED ELEMENT OR COMPONENT

Campaign planning-board illustration.

SOURCE LOCATION

demo/regression/index.html, line 13.
Exact column data is not generated by the demonstration checker.

OBSERVED ACCESSIBILITY PROBLEM

The `img` element has no `alt` attribute, so its text alternative is undefined in the controlled regression.

PRIORITY RATIONALE

High because the demonstration treats the image as informative and its equivalent text is removed.

REPRODUCTION INSTRUCTIONS

1. Open the regression file.
2. Locate the line 13 image.
3. Run the checker against the regression file.
4. Observe rule `image-alt`.

SUGGESTED REMEDIATION

Add concise equivalent text. If product owners determine the image is decorative, use `alt=""` instead; that content decision is outside this heuristic.

CONTROLLED REGRESSION EVIDENCE

```
Regression line 13 begins:   
Checker message: "Image is missing an alt attribute."
```

RETEST OUTCOME

PASS. Remediated line 13 adds `alt="Abstract blue shapes representing a campaign planning board"`. The repeated checker reports zero total findings.

Suggested developer ticket

Add an appropriate text alternative based on the image's intended purpose.
Acceptance: an explicit `alt` decision is present and the selected `image-alt` finding is absent.

Heading structure skips from level one to level three

AFFECTED ELEMENT OR COMPONENT

"Request a review" section heading.

SOURCE LOCATION

demo/regression/index.html, line 14.
Exact column data is not generated by the demonstration checker.

OBSERVED ACCESSIBILITY PROBLEM

The heading after the page h1 is an h3, creating a skipped level in the controlled document outline.

PRIORITY RATIONALE

Medium because the structural sequence is degraded, but the demonstrated primary action remains present.

REPRODUCTION INSTRUCTIONS

1. Open the regression file.
2. Review the h1 on line 11 and h3 on line 14.
3. Run the checker.
4. Observe rule heading-order.

SUGGESTED REMEDIATION

Use an h2 for the section so the heading hierarchy reflects the page structure.

CONTROLLED REGRESSION EVIDENCE

h1 "Campaign review workspace" → h3 "Request a review"
Checker message: "Heading hierarchy skips a level."

RETEST OUTCOME

PASS. Remediated line 14 uses `<h2 id="contact">Request a review</h2>`. The repeated checker reports zero total findings.

Suggested developer ticket

Restore the section heading to h2. Acceptance: the document uses sequential levels and the selected heading-order finding is absent.

Retest status

The remediated synthetic page restores visible button text, a label association, an explicit image text alternative, and sequential heading order. The repeated controlled checker run reports zero findings. Machine-readable outputs are preserved in `artifacts/evidence/`.

Limitations and exclusions

- The checker is intentionally narrow and recognizes only four selected HTML patterns used by this controlled demonstration.
- It does not test color contrast, keyboard order, focus visibility, dynamic state, broader ARIA semantics, multimedia, documents, mobile assistive technology, or full WCAG success criteria.
- Line locations in this report were confirmed from the controlled source files; the checker itself does not generate line or column coordinates.
- Severity is a demonstration triage judgment and requires context-specific human review before use on real work.
- No manual keyboard test, NVDA/Chrome review, qualified specialist review, legal review, production monitoring, or customer-site testing occurred.
- The report is representative presentation material, not evidence of a completed commercial engagement.

Macelyr Ventures is an independently developed, early-stage B2B technology initiative presented for concept validation. Proposed service scope and pricing remain under validation. The service is not currently available for purchase.