

Accessibility Remediation Plan

E-commerce web app · Homepage, product page, checkout flow (mobile view)

Laura Gilbert

23 April 2026

This report prioritizes 8 accessibility barriers from the sample audit, ranks the top 5 by severity and user impact, and provides a developer-ready remediation plan.

1. Barrier selection overview

The course brief asks for 8 issues spanning different WCAG principles, each with a severity rating and an impact category. The source audit lists barriers across perception, operability, understandability, and technical robustness.

Barrier	WCAG	Severity	Impact
Improper heading structure - multiple H1s on a single page	WCAG 1.3.1	Critical	Understandable
Low contrast between text and background in buttons	WCAG 1.4.3	Low	Perceivable
Form input fields missing labels	WCAG 1.3.1, 3.3.2	Medium	Understandable
Elements with role="button" missing keyboard operability	WCAG 2.1.1	Critical	Operable
Language attribute missing in HTML tag	WCAG 3.1.1	High	Robust
ARIA roles incorrectly applied, confusing screen readers	WCAG 4.1.2	Critical	Robust
Content resizing breaks layout	WCAG 1.4.10	Critical	Perceivable
Page title missing or too generic	WCAG 2.4.2	Low	Perceivable

2. Issue ranking and impact analysis

Prioritized in order of POUR principles and by severity.

Barrier	WCAG	Severity	Impact
1. Content resizing breaks layout	WCAG 1.4.10	Critical ▾	Perceivable ▾
2. Elements with role="button" missing keyboard operability	WCAG 2.1.1	Critical ▾	Operable ▾
3. Improper heading structure - multiple H1s on a single page	WCAG 1.3.1	Critical ▾	Understandable ▾
4. ARIA roles incorrectly applied, confusing screen readers	WCAG 4.1.2	Critical ▾	Robust ▾
5. Language attribute missing in HTML tag	WCAG 3.1.1	High ▾	Robust ▾

3. Remediation planning and examples

Issue	Fix
Content resizing breaks layout - WCAG 1.4.10	Ensure layout adapts up to 400% zoom without horizontal scrolling (except for complex components like tables).
Elements with role="button" missing keyboard operability - WCAG 2.1.1	Either use a native <code><button></code> or fully support keyboard interaction.
Improper heading structure - multiple H1s on a single page - WCAG 1.3.1	Use a logical, hierarchical heading structure with one primary <code><h1></code> .
ARIA roles incorrectly applied, confusing screen readers - WCAG 4.1.2	Avoid unnecessary ARIA. Use native HTML elements first.
Language attribute missing in HTML tag - WCAG 3.1.1	Define the primary language in the <code><html></code> element.

4. Reflections and future applications:

It is critical to not only identify accessibility issues for web and/or email, but also know how to prioritize what to remediate first. There are many tools available to identify the issues and

possible solutions to fix them, but tackling them in an organized way and tracking progress is essential for the business use case and clear communication across teams.