



# Accessibility Testing Templates

## Common WCAG 2.2 AA Failures

Copy-paste-ready bug descriptions for faster, higher-quality accessibility fixes. Designed for testers, QA analysts, and engineers filing accessibility defects.



# Why Use These Templates?



## Speed Up Testing

Pre-written descriptions reduce time spent writing defect reports from scratch



## Improve Quality

Consistent, professional language ensures clarity and actionable guidance for developers



## WCAG Alignment

Each template references specific success criteria for proper documentation

These templates complement the wA11y Tester Toolkit methodology, providing standardized language for common failures discovered during manual testing.

# Keyboard & Focus Issues

## Focus Management

- Focus not trapped in modal dialogs
- Focus order doesn't match visual layout
- No visible focus indicator on interactive elements (2.4.7)
- Focus obscured by sticky headers or footers (2.4.11)

## Navigation Shortcuts

- Skip to content link missing (2.4.1)
- Skip link non-functional or broken (2.4.1)
- Keyboard traps preventing navigation

**WCAG References:** 2.1.2 No Keyboard Trap, 2.4.3 Focus Order, 2.4.7 Focus Visible



# Template Example: Focus Not Trapped in Modal

 **WCAG 2.1.2 No Keyboard Trap – Severity: Critical**

## Problem Description

When a modal dialog opens, keyboard focus is not trapped within the modal boundaries. Users can tab to elements behind the modal, creating confusion and violating expected behavior.

## Expected Result

Focus remains within the modal. Tab cycles through focusable elements inside the dialog, returning to the first element after the last.

## Steps to Reproduce

1. Navigate to [page URL]
2. Activate the [button/link] to open modal
3. Press Tab repeatedly
4. Observe focus moving outside modal

## Actual Result

Focus moves to background page elements while modal remains open. Background content remains interactive despite being visually obscured.

# Forms & Error Handling

1

## Missing Labels

Form fields lack programmatic labels, preventing screen readers from identifying input purpose

**WCAG:** 1.3.1 Info and Relationships, 3.3.2 Labels or Instructions, 4.1.2 Name, Role, Value

2

## Silent Errors

Error messages appear visually but aren't announced to screen reader users, blocking form completion

**WCAG:** 3.3.1 Error Identification, 4.1.3 Status Messages

3

## Color-Only Errors

Invalid fields indicated solely by red borders without text labels or icons

**WCAG:** 1.4.1 Use of Color

4

## Required Fields

Required status not communicated programmatically via aria-required or required attribute

**WCAG:** 1.3.1 Info and Relationships, 3.3.2 Labels or Instructions

# Navigation, Structure & Media Issues



## Page Titles & Headings

Missing or generic page titles fail to describe content. (2.4.2) Heading levels skipped (h1 to h3) or misused for styling create confusing document structure. (1.3.1, 2.4.6)



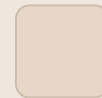
## Landmark Regions

Missing navigation, main, or complementary landmarks make it difficult for screen reader users to navigate efficiently between page sections. (1.3.1, 2.4.1)



## Alternative Text

Informative images missing alt text (1.1.1) or icon-only buttons lacking accessible names (4.1.2) prevent understanding of visual content and functionality.



## Media Accessibility

Videos without captions exclude deaf users. Audio content without transcripts creates barriers for those who cannot hear or process audio. (1.2.1)

# Timing, Motion & ARIA Components

## Timing & Motion Failures

Session timeouts occur without warning, forcing users to restart complex tasks

- Auto-rotating carousels cannot be paused (2.2.2 Pause, Stop, Hide)
- Motion animations lack prefers-reduced-motion support (2.3.3 Animation from Interactions)
- Insufficient timeout warnings (2.2.1 Timing Adjustable)

## Custom ARIA Components (4.1.2)

State changes in custom widgets not announced to assistive technology

- Missing or incorrect ARIA roles on custom controls
- Expandable sections lack aria-expanded state
- Tab panels missing required ARIA attributes

These failures often require manual testing with screen readers and keyboard navigation to detect.

# Using These Templates Effectively

01

---

## Customize Environment Details

Add specific screen reader, browser, OS, and device information before submitting tickets

02

---

## Reference WCAG Criteria

Always include the applicable success criterion number and level in your defect reports

03

---

## Attach Evidence

Include screenshots, screen recordings, or code snippets to support your findings

04

---

## Test Thoroughly

Remember: automated tools detect only 20-30% of accessibility issues. Manual testing is essential

---

**Key Reminder:** Clear, consistent defect reporting accelerates remediation and improves product accessibility. These templates are starting points—adapt them to your project's specific needs and testing context.

 These templates represent common failures, not an exhaustive list of WCAG requirements.