Accessibility Testing with WCAG PDF

Accessibility testing checks that people with disabilities can use a product. Use the WCAG principles, perceivable, operable, understandable and robust, as a framework. Test keyboard navigation, focus order, text alternatives, colour contrast, form labels and screen reader behaviour, combine automated scans with manual checks, and fix issues early.

Key takeaways

- WCAG is organised around four principles: perceivable, operable, understandable and robust.
- Automated tools find only part of the issues, so manual checks are essential.
- Keyboard-only use is a quick and powerful first test.
- Every image, field and control needs a meaningful accessible name.
- Accessibility benefits many users, not only people with permanent disabilities.

What is accessibility testing?

Accessibility testing is the evaluation of software against criteria such as the Web Content Accessibility Guidelines so that people with visual, hearing, motor or cognitive disabilities can perceive, operate and understand it.

Why does accessibility testing matter?

Many people use assistive technology or have temporary limits, such as a broken arm or bright sunlight on a screen. Accessible products serve more users, support legal and policy requirements in many regions, and usually have cleaner markup. Check which standards apply to your market and product.

What are the WCAG principles?

Principle	Meaning	Example check
Perceivable	Information can be sensed	Images have text alternatives; captions for video
Operable	Interface can be used	Everything works with a keyboard; no time traps
Understandable	Content and behaviour are clear	Clear labels and error messages

Principle	Meaning	Example check
Robust	Works with assistive tools	Valid, semantic markup
WCAG has conformance levels A, AA and AAA. Many policies target AA. Read the current WCAG version and the success criteria for details.		

How do I test with only a keyboard?

Unplug the mouse and use Tab, Shift+Tab, Enter, Space and arrow keys. Check that every control can be reached and used, that the order of focus follows the visual order, that the focus indicator is visible and that you can leave menus and dialogs. Keyboard traps and invisible focus are common defects.

What should I check for visuals?

Text must have enough contrast against its background, usually a ratio of 4.5 to 1 for normal text at level AA. Information must not rely on colour alone, so add icons or text for errors. Pages should work when text is enlarged and zoomed to two hundred percent without losing content. Check that motion can be paused.

How do forms and content get tested?

Every input should have a visible label connected to it, errors should be described in text and linked to the field, and required fields should be identified. Headings should form a logical outline. Links need meaningful text; "click here" tells a screen reader user nothing. Images need alternative text, or empty alternative text when decorative.

How do screen readers fit in?

Try a free screen reader such as NVDA on Windows or the built-in VoiceOver on Apple devices. Listen to how a page is announced: page title, headings, landmarks, button names and form errors. Compare the experience with the visual one. If something is announced as "button" with no name, that is a defect.

Which tools help?

Browser extensions and libraries can scan for missing labels, contrast problems and invalid roles. They are useful for regression checks and can run in an automation pipeline. They find only some issues, so a manual review should always follow. Treat tool output as a list of candidates and confirm each one.

How do I report accessibility defects?

Include the page, the element, the WCAG success criterion, the steps, the assistive technology used and the impact on the user. A clear description such as "Search button has no accessible name, so a screen reader announces only button" helps developers fix it faster.

How do I build accessibility into the process?

Include accessibility acceptance criteria in user stories, review designs before coding, test components in isolation, run automated checks on each build and perform manual audits before major releases. Involve users with disabilities when possible, because their experience reveals issues no checklist covers.

What does a short manual audit routine look like?

Pick the five most important pages. For each, check the title and headings, use the keyboard from top to bottom, zoom the page to two hundred percent, run an automated scan, listen with a screen reader for one flow and note every issue with the success criterion. Fix the critical ones first, such as unreachable buttons and missing form labels. Repeat the routine after each release so that regressions are caught while they are small.

How do I handle common component issues?

Custom dropdowns, modals and tabs are frequent problem areas. Check that a modal moves focus into itself and returns it when closed, that tabs can be changed with arrow keys, that menus announce their expanded state and that live updates such as form errors are announced. When a component is complex, prefer well-tested libraries and verify their accessibility rather than building from scratch.

What is a quick checklist before release?

Confirm the page title is meaningful, headings are in order, every image has suitable alternative text, forms have labels, contrast is sufficient, the keyboard works end to end and the screen reader announces key controls.

Who should own accessibility?

Everyone on the team shares it: designers choose contrast and layout, developers use semantic markup, testers verify and product owners prioritise fixes.

Frequently asked questions

What is WCAG?

WCAG stands for Web Content Accessibility Guidelines, published by the W3C, and it defines criteria to make web content accessible to more people.

What are the four WCAG principles?

They are perceivable, operable, understandable and robust, and each success criterion belongs to one of them.

Can automated tools test accessibility fully?

No. They detect some issues such as missing labels or low contrast, but manual and assistive technology checks are needed for the rest.

What is the quickest accessibility check?

Try using the page with only a keyboard, because it quickly reveals unreachable controls, missing focus indicators and keyboard traps.

Which conformance level should I aim for?

Many policies target level AA, but check the rules that apply to your product, market and contracts.

Sources and further reading

- W3C WCAG overview
- WCAG quick reference

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550