### **WCAG Accessibility Testing with WAVE and Axe DevTools**

Here’s a detailed step-by-step guide with **test cases** and **execution steps** for performing WCAG 2.1 AA accessibility testing using **WAVE** and **Axe DevTools**.

### **1. Accessibility Testing Using WAVE**

#### **About WAVE**

WAVE (Web Accessibility Evaluation Tool) is a browser extension that evaluates web pages for accessibility issues and visualizes them on the page.

#### **Test Cases for WAVE**

Below are some detailed test cases mapped to WCAG 2.1 AA guidelines:

| **Test Case** | **Steps** | **Expected Result** |
| --- | --- | --- |
| **Check for Missing Alt Text** | 1. Open the page in a browser.  2. Launch WAVE extension.  3. Look for "Missing Alternative Text" alerts. | All images should have descriptive alt text unless they are decorative (then alt=""). |
| **Verify ARIA Labels** | 1. Inspect ARIA attributes on elements like forms and buttons.  2. Ensure proper labels are applied. | All ARIA labels must describe the element's purpose. |
| **Check Color Contrast** | 1. Look for contrast errors flagged in the "Contrast" tab. | Text contrast ratio should meet the minimum: 4.5:1 for normal text, 3:1 for large text. |
| **Validate Heading Structure** | 1. Open WAVE.  2. Use the "Outline" tab to review heading levels. | Heading levels should follow a logical structure (e.g., H1 -> H2, no skipped levels). |
| **Keyboard Accessibility** | 1. Use the "Structure" tab to evaluate focus order.  2. Test navigation using the Tab key. | Focus should move logically and visibly highlight each element. |
| **Check Link Descriptions** | 1. Review link text flagged in the "Features" tab.  2. Hover over flagged links for detailed information. | Link text should describe the destination or purpose clearly (avoid “Click here”). |
| **Test Form Accessibility** | 1. Look for form-related alerts in the "Errors" section.  2. Check for labels, errors, and instructions. | All input fields should have associated labels and clear instructions. |

#### **Steps to Use WAVE**

1. **Install WAVE Extension**:
   * Install the WAVE browser extension for Chrome or Firefox.
   * WAVE Extension.
2. **Analyze a Web Page**:
   * Navigate to the web page you want to test.
   * Click the WAVE extension icon in your browser toolbar.
3. **Review Accessibility Errors**:
   * WAVE will overlay icons on the web page highlighting:
     + Errors (red icons): WCAG violations like missing alt text.
     + Alerts (yellow icons): Potential issues needing review.
     + Features (green icons): Accessible elements like headings and landmarks.
4. **Use Tabs for In-Depth Analysis**:
   * **Details Tab**: Explains each issue and its WCAG reference.
   * **Structure Tab**: Evaluates ARIA roles, focus order, and semantic structure.
   * **Contrast Tab**: Highlights color contrast issues.
5. **Fix and Retest**:
   * Based on the report, address issues and run WAVE again to validate fixes.

### **2. Accessibility Testing Using Axe DevTools**

#### **About Axe DevTools**

Axe DevTools is an automated accessibility testing tool integrated with browser DevTools. It identifies WCAG violations and offers suggestions for fixes.

#### **Test Cases for Axe DevTools**

Below are some test cases for Axe DevTools:

| **Test Case** | **Steps** | **Expected Result** |
| --- | --- | --- |
| **Scan Page for WCAG Violations** | 1. Open browser DevTools.  2. Switch to the Axe tab.  3. Click "Analyze". | A detailed report of violations should be displayed, categorized by severity. |
| **Validate Image Alt Text** | 1. Review the "Images" section in the Axe report.  2. Check flagged images for missing alt attributes. | All images should have appropriate alt text. |
| **Test Keyboard Accessibility** | 1. Use Axe's "Tab Order" feature to test focus management. | Tab navigation should highlight interactive elements logically. |
| **Check Landmarks and Regions** | 1. Look for missing landmarks in Axe's "Regions" category. | All sections of the page should have defined ARIA landmarks (e.g., main, header, footer). |
| **Verify Color Contrast** | 1. Review contrast issues flagged in Axe. | Text contrast ratios must meet WCAG standards (4.5:1 for text, 3:1 for large text). |
| **Check Dynamic Content Accessibility** | 1. Inspect alerts for ARIA live regions or missing roles. | Dynamic updates should notify users through ARIA live regions where appropriate. |
| **Form Field Accessibility** | 1. Look for errors in the "Forms" section.  2. Validate labels and instructions for input fields. | All form fields should be labeled and provide clear error messages. |

#### **Steps to Use Axe DevTools**

1. **Install Axe DevTools**:
   * Download the Axe browser extension for Chrome, Firefox, or Edge.
   * Axe DevTools.
2. **Open DevTools**:
   * Right-click on the web page and select **Inspect** to open browser DevTools.
   * Navigate to the "Axe DevTools" tab.
3. **Run an Accessibility Audit**:
   * Click the **"Analyze"** button.
   * Axe will scan the page and generate a report, categorizing issues as:
     + **Violations**: Elements that fail WCAG standards.
     + **Needs Review**: Elements that might be issues but need manual validation.
     + **Best Practices**: Non-mandatory improvements.
4. **View Detailed Results**:
   * Click on any issue to see:
     + Affected elements.
     + WCAG reference and description.
     + Suggested solutions.
5. **Fix and Re-Analyze**:
   * Address flagged issues in the code.
   * Rerun the Axe scan to validate fixes.

### **Comparison: WAVE vs. Axe DevTools**

| **Feature** | **WAVE** | **Axe DevTools** |
| --- | --- | --- |
| **Ease of Use** | Beginner-friendly with a simple interface. | Developer-focused with deeper insights. |
| **Integration** | Browser extension only. | Integrates with CI/CD pipelines and test suites. |
| **Reporting** | Visual overlays and explanations. | Detailed violation categorization with fixes. |
| **Manual Testing Guidance** | Limited manual test support. | Provides manual testing steps for certain issues. |

### **Tips for Testing**

1. **Combine Tools**:
   * Use **WAVE** for a quick overview and visual debugging.
   * Use **Axe DevTools** for a more thorough and detailed analysis.
2. **Manual Verification**:
   * Both tools require manual testing for areas like keyboard navigation and screen reader compatibility.
3. **Focus on WCAG 2.1 AA**:
   * Prioritize fixing errors tied to Level A and AA guidelines to ensure legal compliance and usability.

By leveraging **WAVE** and **Axe DevTools**, you can effectively detect and resolve WCAG AA accessibility issues, ensuring a more inclusive user experience.

4o

##### **You said:**

Full test coverage wcag accessibility testing detailed test cases steps for wave and axe devtools — detailed

###### **ChatGPT said:**

ChatGPT

### **Full Test Coverage for WCAG Accessibility Testing Using WAVE and Axe DevTools**

Below is a comprehensive list of **WCAG 2.1 AA** compliance test cases with **detailed steps** for using **WAVE** and **Axe DevTools**. Each test case maps to specific WCAG guidelines to ensure thorough coverage of accessibility standards.

### **1. Testing with WAVE**

#### **Setup Steps for WAVE:**

1. Install the WAVE browser extension for Chrome or Firefox.
   * Download WAVE Extension.
2. Navigate to the page you want to test.
3. Click the WAVE icon in the browser toolbar to start the analysis.
4. Review the visual overlay of errors, alerts, and features on the page.

#### **Detailed Test Cases Using WAVE**

| **Test Case ID** | **Test Case** | **Steps to Execute** | **Expected Result** |
| --- | --- | --- | --- |
| **WAVE-001** | Missing Alternative Text | 1. Launch WAVE.  2. Check for red “alt missing” icons on images. | All images should have descriptive alt text.  Decorative images must have empty alt attributes (alt=""). |
| **WAVE-002** | Text Contrast | 1. Go to the "Contrast" tab in WAVE.  2. Review flagged low-contrast elements. | Text contrast ratio should meet WCAG standards (4.5:1 for normal text, 3:1 for large text). |
| **WAVE-003** | Heading Structure | 1. Open the "Outline" tab in WAVE.  2. Review the logical order of headings (H1 to H6). | Headings should follow a hierarchical and sequential structure (e.g., H1 > H2 > H3). |
| **WAVE-004** | Link Descriptions | 1. Check flagged links for ambiguous text (e.g., “Click here,” “More”). | All links should clearly describe their purpose or destination. |
| **WAVE-005** | Keyboard Accessibility | 1. Tab through the page using the keyboard.  2. Check if WAVE flags issues in focusable elements. | All focusable elements must be operable with a keyboard and have a visible focus indicator. |
| **WAVE-006** | Form Accessibility | 1. Review flagged input fields for missing labels.  2. Verify form instructions and error messages. | All form inputs must have associated labels and clear error messaging. |
| **WAVE-007** | ARIA Roles and Landmarks | 1. Check flagged ARIA roles and landmarks (e.g., main, nav). | ARIA landmarks should be appropriately defined and used consistently. |
| **WAVE-008** | Language Attribute | 1. Verify if WAVE flags a missing or incorrect language attribute in the HTML tag. | The lang attribute should correctly reflect the page’s language (e.g., lang="en" for English). |
| **WAVE-009** | Dynamic Content Accessibility | 1. Identify if WAVE flags any issues with ARIA live regions. | ARIA live regions should notify users of dynamic content changes. |

### **2. Testing with Axe DevTools**

#### **Setup Steps for Axe DevTools:**

1. Install the Axe DevTools extension for your browser.
   * Download Axe DevTools.
2. Open your web page.
3. Right-click and choose **Inspect** to open DevTools.
4. Navigate to the Axe tab and click **Analyze**.

#### **Detailed Test Cases Using Axe DevTools**

| **Test Case ID** | **Test Case** | **Steps to Execute** | **Expected Result** |
| --- | --- | --- | --- |
| **AXE-001** | WCAG Violations Scan | 1. Click "Analyze" in the Axe tab.  2. Review the categorized violations (Critical, Serious, Minor). | All critical and serious violations should be addressed. |
| **AXE-002** | Alt Text for Images | 1. Review flagged images in the "Images" section.  2. Check missing or incorrect alt attributes. | All images should have descriptive alt text or an empty alt for decorative images. |
| **AXE-003** | Color Contrast | 1. Review the "Contrast" violations flagged by Axe.  2. Identify elements failing contrast requirements. | Text should meet contrast ratios of 4.5:1 (normal text) and 3:1 (large text). |
| **AXE-004** | Focusable Elements | 1. Navigate using the Tab key.  2. Inspect flagged focusable elements in Axe. | All interactive elements should be focusable and have a visible focus outline. |
| **AXE-005** | ARIA Roles and Landmarks | 1. Check violations related to incorrect or missing ARIA roles.  2. Inspect the highlighted elements. | ARIA roles should be properly defined and enhance accessibility without misuse. |
| **AXE-006** | Form Labels and Instructions | 1. Check the "Forms" section for flagged fields.  2. Inspect inputs for missing labels or instructions. | All inputs should have descriptive labels and accessible error feedback. |
| **AXE-007** | Dynamic Content Accessibility | 1. Inspect violations related to live regions.  2. Check for missing ARIA live role attributes. | Dynamic updates should notify screen reader users with appropriate ARIA live roles. |
| **AXE-008** | Heading Order | 1. Review flagged headings in the "Headings" section.  2. Check for logical structure and sequence. | Headings should follow a semantic and hierarchical order (e.g., no skipping levels). |
| **AXE-009** | Link Purpose | 1. Inspect flagged links with vague or duplicate text. | Links should clearly describe their target destination or function. |
| **AXE-010** | Language Attribute | 1. Verify violations related to missing lang attributes. | The page should have a valid lang attribute matching its primary language (e.g., lang="en", lang="fr"). |

### **Manual Testing Augmented by Tools**

While WAVE and Axe DevTools automate a significant portion of accessibility testing, some WCAG requirements require **manual verification**:

| **Manual Test** | **Steps** | **Expected Result** |
| --- | --- | --- |
| **Keyboard Navigation** | 1. Use the Tab and Shift+Tab keys to navigate the page.  2. Verify focus order and keyboard operability. | All interactive elements should be accessible via keyboard and follow a logical focus order. |
| **Screen Reader Compatibility** | 1. Test with screen readers (e.g., NVDA, JAWS, VoiceOver).  2. Navigate and interact with the page. | Content should be announced correctly, including labels, headings, and dynamic updates. |
| **Zoom and Resize** | 1. Zoom the page to 200% using browser zoom.  2. Resize the browser window.  3. Verify layout adaptability. | The content should remain usable and readable without horizontal scrolling. |
| **Captions and Audio Descriptions** | 1. Check videos for captions and/or audio descriptions.  2. Ensure synchronization and clarity. | All video content should have accurate captions, and audio descriptions should be provided when needed. |

### **Tips for Comprehensive Coverage**

1. **Combine WAVE and Axe**:
   * Use **WAVE** for visual overlays and a quick overview.
   * Use **Axe DevTools** for detailed insights and code-level fixes.
2. **Perform Manual Testing**:
   * Augment automated scans with manual checks for keyboard navigation, screen readers, and responsive design.
3. **Address Severity Levels**:
   * Focus on **Critical** and **Serious** violations first.
4. **Document Issues**:
   * Record test results, WCAG references, and fixes applied for accountability and future audits.
5. **Retest**:
   * After implementing fixes, retest with both tools to confirm compliance.

### **Conclusion**

By using **WAVE** and **Axe DevTools**, combined with manual testing, you can achieve robust WCAG 2.1 AA compliance. These tools streamline the process of identifying and addressing accessibility issues while ensuring a more inclusive and user-friendly experience.