

Category	Tool/Technology	Purpose	Features	Platform	Website
ARIA - Accessible Rich Internet Applications		A set of roles and attributes that enhance web content and applications accessibility , especially for JavaScript-based apps.			
Accessibility Testing Tools	Accessibility Insights	Suite of tools for finding and fixing accessibility issues, with both automated and manual testing functionality	Automated WCAG 2.1 compliance checks, FastPass (automated checker), manual inspection, provides guidance for developers on fixing issues.	Browser extension (Chrome, Edge), Windows app	Accessibility Insights
	Axe Accessibility Checker	Browser extension for automated accessibility testing, helping developers identify common issues based on WCAG standards.	Detailed reports with recommendations, supports web, mobile, and responsive design testing.	Browser extension (Chrome, Firefox)	Axe Accessibility Checker

	WAVE	Web-based tool that visually highlights accessibility issues on web pages.	Visual feedback on accessibility issues, detailed error reports with explanations, supports both manual and automated testing.	Web-based, Browser extension (Chrome, Firefox)	WAVE
	Lighthouse	Open-source, automated tool to improve web page quality, including accessibility, performance, SEO, and more.	Detailed reports on accessibility scores and actionable feedback, covers WCAG 2.1 guidelines, good for PWAs.	Integrated in Chrome DevTools, Node.js standalone	Lighthouse
	Tenon.io	Web-based tool focusing on WCAG 2.1 compliance.	API access for automated testing, detailed reports with advice for fixing issues, supports continuous integration.	Web-based API, Browser extension	Tenon.io
	Color Contrast Analyzer	Tool for evaluating color contrast ratio between text and background to ensure it	Tests color contrast to meet WCAG standards, provides color recommend	Standalone tool (Windows, macOS)	Color Contrast Analyzer

		meets WCAG standards.	ations for compliance.		
Screen Readers	Narrator	Built-in screen reader for Windows that reads aloud text and describes elements.	Speech output, Braille display, keyboard shortcuts, integrates with Windows OS.	Windows	Narrator
	JAWS (Job Access With Speech)	Popular screen reader for blind or low-vision users, with extensive app compatibility .	Customizabl e voice settings, Braille support, web browser integration, advanced navigation.	Windows	JAWS
	NVDA (NonVisual Desktop Access)	Free, open-source screen reader for Windows.	Speech output for browsers and desktop apps, supports multiple languages and Braille devices, compatible with screen reader commands.	Windows	NVDA
	VoiceOver	Built-in screen reader for macOS and iOS devices, providing spoken feedback.	Spoken feedback, gesture navigation (iOS), Braille support, customizabl e options.	macOS, iOS	VoiceOver

	TalkBack	Screen reader for Android devices providing feedback for blind and low-vision users.	Converts text to speech, auditory feedback, customizable settings, gesture navigation.	Android	TalkBack
Other Assistive Technologies	Dragon NaturallySpeaking	Speech recognition software for voice-based control and dictation.	Dictation and command-based navigation, supports various applications, specialized for physical disabilities.	Windows	Dragon NaturallySpeaking
	ZoomText	Screen magnification software for low-vision users with screen reading capabilities.	Screen magnification, color enhancements, speech features, works with Windows apps and web browsers.	Windows	ZoomText
	Aira	Real-time visual assistance service for blind and low-vision users through trained agents.	Provides real-time assistance via smartphone app, helps with navigation, digital content interpretation, and more.	iOS, Android	Aira
	BeeLine Reader	Tool designed for	Color gradient for	Browser extension	BeeLine Reader

		users with dyslexia or reading difficulties, improving text readability.	improved focus, enhances reading comprehension, works with web content and eBooks.	(Chrome, Firefox), mobile app	
	GazeTalk	Communication tool for people with motor impairments using eye-tracking technology.	Eye-tracking for computer control, writing text, and interacting with apps.	Windows	GazeTalk

Common Techniques & Heuristics in Exploratory Testing

- **Use Case Testing:** Test the application based on real-world scenarios or how end users will interact with the system.
- **Error Guessing:** Leverage your experience to anticipate areas where errors are likely to occur, such as complex forms, edge cases, or poorly defined features.
- **Boundary Testing:** Test the limits of the application, such as input fields with the maximum or minimum values, empty values, or large data sets.
- **State Transition Testing:** Verify the application's behavior when moving from one state to another (e.g., user authentication, shopping cart states, etc.).
- **Pair Testing:** Two testers collaborate, one focusing on the test execution and the other on thinking of new test ideas based on current results.
- **Mind Maps or Charters:** Create diagrams to visualize the areas of the application that need testing and how to explore them systematically.

Tools for Exploratory Testing

While exploratory testing is often more human-centric, several tools can help document the process and track issues:

- **Session Recording Tools:** Tools like TestRail, Bugzilla, Jira, and Trello are useful for logging discovered issues.
- **Session Management:** Tools like Exploratory Testing Tools or Session-based Test Management (SBTM) systems help to manage exploratory testing sessions by tracking charters, time-boxing, and session outcomes.
- **Browser Developer Tools:** For web-based applications, browser developer tools can help quickly identify performance issues, inspect DOM elements, or debug JavaScript errors during the exploration.
- **Mind Mapping Tools:** XMind or MindMeister can be used to visually organize exploratory test ideas and areas of focus.

When to Use Exploratory Testing

1. Early in the Development Cycle
2. When There Are Incomplete Requirements
3. During Maintenance or Regression Testing
4. When Testing Is Time-Pressed
5. For User Experience (UX) Testing