ARIA- Accessible Rich Internet Applications:

It is a set of roles and attributes that define ways to make web content and web applications (especially those developed with JavaScript) more accessible to people with disabilities.

1. Accessibility Testing Tools

These tools are used by developers and testers to evaluate the accessibility of digital products and ensure they meet accessibility standards like WCAG (Web Content Accessibility Guidelines), ADA, and Section 508.

a) **Accessibility Insights**

Purpose: A suite of tools developed by Microsoft to help find and fix accessibility issues. It provides both automated and manual testing functionality.

Features:

Automated checks for WCAG 2.1 compliance.

FastPass (automated accessibility checker) and manual inspection tools.

Can be used for web and Windows application accessibility testing.

Includes guidance for developers on how to fix issues.

Platform: Browser extension for Chrome and Edge, as well as a Windows app.

Website: Accessibility Insights

b) **Axe Accessibility Checker**

Purpose**:** A powerful browser extension for automated accessibility testing that helps developers find common issues based on WCAG standards.

Features:

Provides detailed reports with recommendations for fixing accessibility issues.

Can be used directly within the browser (Chrome and Firefox).

Supports web, mobile, and responsive design testing.

Platform: Browser extension (Chrome and Firefox).

Website: axe Accessibility Checker

c) **WAVE (Web Accessibility Evaluation Tool)**

Purpose: A web-based tool that visually highlights accessibility issues on web pages.

Features:

Provides visual feedback about the accessibility of web content.

Offers detailed error reports with explanations and suggestions.

Supports both manual and automated testing.

Platform: Web-based and browser extension (Chrome, Firefox).

Website: WAVE

d) **Lighthouse**

Purpose: An open-source, automated tool for improving the quality of web pages, including accessibility, performance, SEO, and more.

Features:

Generates detailed reports, including accessibility scores and actionable feedback.

Covers a broad range of accessibility tests based on WCAG 2.1 guidelines.

Works well for testing progressive web apps (PWAs).

Platform: Integrated into Chrome DevTools, or available as a standalone tool in Node.js.

Website: Lighthouse

e) Tenon.io

Purpose: A web-based accessibility testing tool that focuses on WCAG 2.1 compliance.

Features:

Provides API access for automated testing.

Provides detailed reports with specific advice on fixing issues.

Includes support for continuous integration (CI) tools.

Platform: Web-based API and browser extension.

Website: Tenon.io

f) Color Contrast Analyzer

Purpose: A tool designed to help evaluate the contrast ratio between text and background, ensuring it meets WCAG standards for color accessibility.

Features:

Helps test color contrast to ensure it meets accessibility requirements (4.5:1 for normal text, 3:1 for large text).

Provides color recommendations to meet contrast standards.

Platform: Available as a standalone tool for Windows and macOS.

Website: Color Contrast Analyzer

2. Screen Readers

Screen readers are essential tools for users who are blind or visually impaired. These tools convert text on the screen into spoken words or Braille, providing accessibility to digital content.

a) Narrator

Purpose: A built-in screen reader for Windows that reads aloud the text on the screen and describes elements.

Features:

Provides basic screen reading features for Windows users.

Supports speech output, Braille display, and keyboard shortcuts.

Integrated directly into the Windows operating system, making it easy to use.

Platform: Windows.

Website: Narrator

b) JAWS (Job Access With Speech)

Purpose: One of the most popular and powerful screen readers for users who are blind or have low vision.

Features:

Supports a wide range of applications, including web browsers, email clients, and office software.

Offers customizable voice settings, Braille support, and advanced navigation features.

Integrated with web browsers for improved web accessibility.

Platform: Windows.

Website: JAWS

c) NVDA (NonVisual Desktop Access)

Purpose: A free, open-source screen reader for Windows.

Features:

Provides speech output for web browsers, email clients, and desktop applications.

Includes support for multiple languages and Braille devices.

Offers compatibility with popular screen reader commands.

Platform: Windows.

Website: NVDA

d) VoiceOver

Purpose: A built-in screen reader for macOS and iOS devices, enabling users to interact with the device through spoken feedback.

Features:

Provides spoken feedback for all content on macOS and iOS devices.

Supports gestures on touch devices and keyboard shortcuts on macOS.

Offers Braille support and customization options.

Platform: macOS and iOS.

Website: VoiceOver

e) TalkBack

Purpose: A screen reader for Android devices that provides feedback for blind and low-vision users.

Features:

Converts text into speech, provides auditory feedback, and supports gesture navigation.

Customizable settings for users' preferences and needs.

Works seamlessly with Android apps and features.

Platform: Android.

Website: TalkBack

3. Other Assistive Technologies and Tools

a) Dragon NaturallySpeaking

Purpose: A speech recognition software that enables users to control their computer and write text through voice commands.

Features:

Supports dictation and command-based navigation.

Provides specialized features for individuals with physical disabilities.

Compatible with various applications including word processors and email clients.

Platform: Windows.

Website: Dragon NaturallySpeaking

b) ZoomText

Purpose: A screen magnification software for low-vision users, providing both magnification and screen reading features.

Features:

Allows for screen magnification and color enhancements.

Includes speech capabilities for reading text aloud.

Compatible with Windows-based applications and web browsers.

Platform: Windows.

Website: ZoomText

c) Aira

Purpose: A service that provides real-time visual assistance to blind and low-vision users through trained agents.

Features:

Users can connect to trained agents via a smartphone app for assistance with reading, navigating, or interpreting the environment.

Aira agents can assist with digital content, text recognition, navigation, and more.

Platform: iOS, Android.

Website: Aira

d) BeeLine Reader

Purpose: A tool that helps individuals with dyslexia or reading difficulties by changing the way text appears on the screen.

Features:

Provides a color gradient that helps users focus on text more effectively.

Improves reading comprehension for individuals with learning disabilities.

Works with web content and eBooks.

Platform: Browser extension (Chrome, Firefox), and mobile app.

Website: BeeLine Reader

e) GazeTalk

Purpose: A communication tool for people with motor impairments that uses eye-tracking technology to help them control the computer.

Features:

Allows users to control their computer, write text, and interact with applications using eye movements.

Supports communication for people with severe motor disabilities.

Platform: Windows.

Website: 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**: