HOW TO CREATE AN ACCESSIBILITY AUDIT

By John Walker

Accessibility Audit Considerations

A complete evaluation requires both AI-based and manual surveys

- Some issues can't be detected or properly assessed via AI and require human evaluation

I suggest using two different assessment methods

- You could combine the results from two browser-based AI tools (axe, Lighthouse, WAVE etc)
- Or use a combination of a third-party vendor evaluation and a browser tool

Doing this audit, I found disparities in the issues detected

- I wouldn't have captured all the issues without using two different AI tools
- Even if the tools had detected the same issues, the differing presentations of results would have been helpful in fully assessing the issues

Case Study: Accessibility Assessment Process and Tools

1. **I scanned the AllianceBernstein homepage using axe**
   - This code evaluation tool is cited in most ADA-related lawsuits
2. **I rescanned the homepage using Lighthouse**
3. **I also consulted survey results from an outside vendor, SiteMorse**
   - Though I eventually concluded results from two AI-based surveys is sufficient, I used three sources for this case study
   - I also scanned the homepage again using SiteMorse's user-operated AI tool
4. **I manually surveyed the homepage**
   - I found many issues the AI didn't detect, sometimes for contextual reasons (semantic H1-6 structure could be improved) or because of tool limitations (color-contrast standards etc)
5. **I combined the results of these surveys for a final assessment**

NOTE: This presentation was created in 2020 and the tools' functionalities may have evolved

Axe Evaluation Overview

Axe is an AI browser-based code evaluation tool by Deque Systems, Inc

Axe runs automated tests and auto-flags results by priority

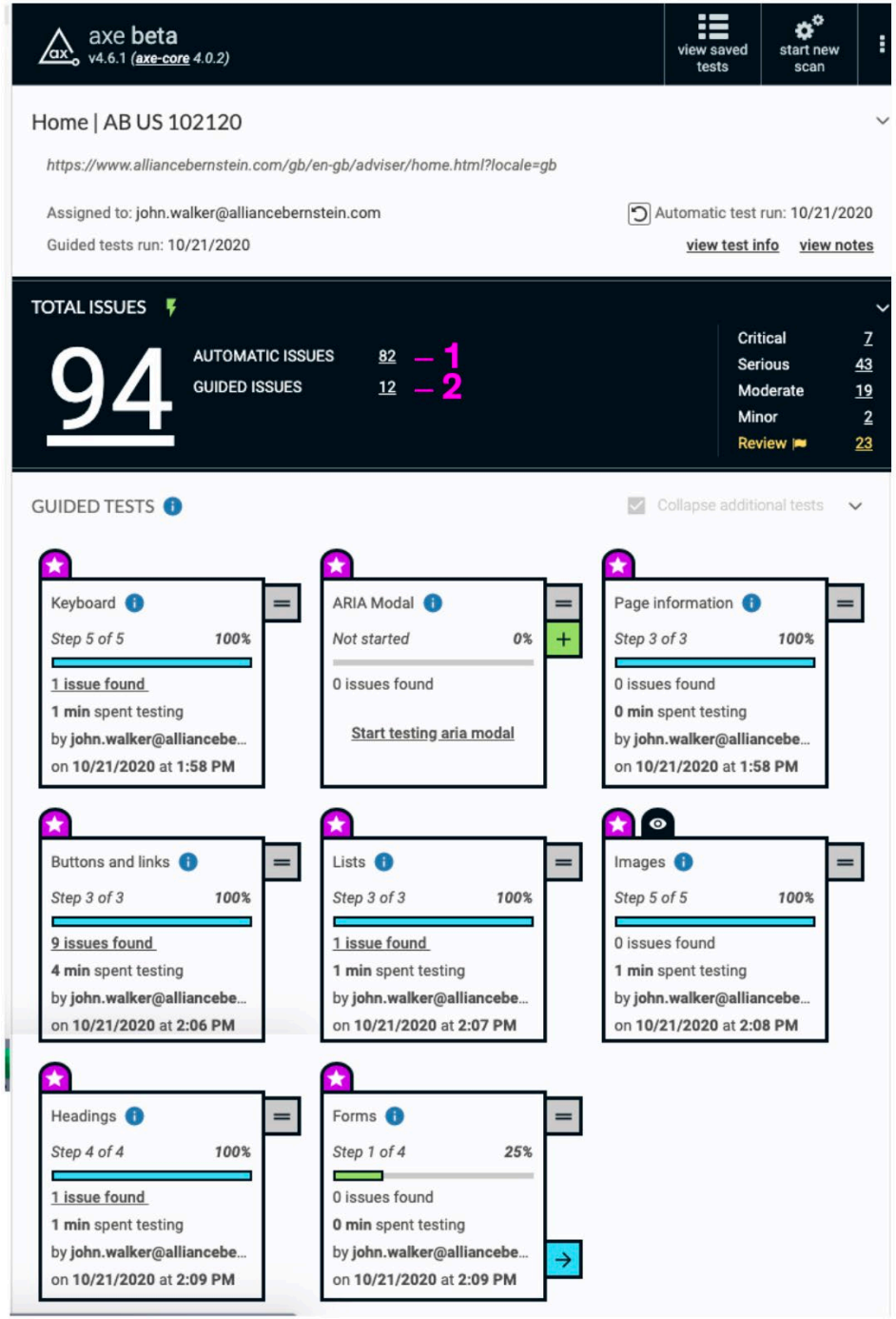
- Prioritization categories are Critical, Serious, Moderate, Minor, and Review

Axe also suggests eight guided tests for a manual survey

- The test categories are: Keyboard, ARIA Model, Page Information, Buttons and Links, Lists, Images, Headings, and Forms
- A manual survey using the guided tests is required since AI can’t judge context well enough to determine WCAG conformance

Axe Evaluation Sample

1. Issues automatically flagged
2. Issues found after manual survey
3. Guided test categories for manual survey



Lighthouse Evaluation Overview

Lighthouse is an AI browser-based code evaluation tool by Google

Lighthouse runs automated tests and segments results

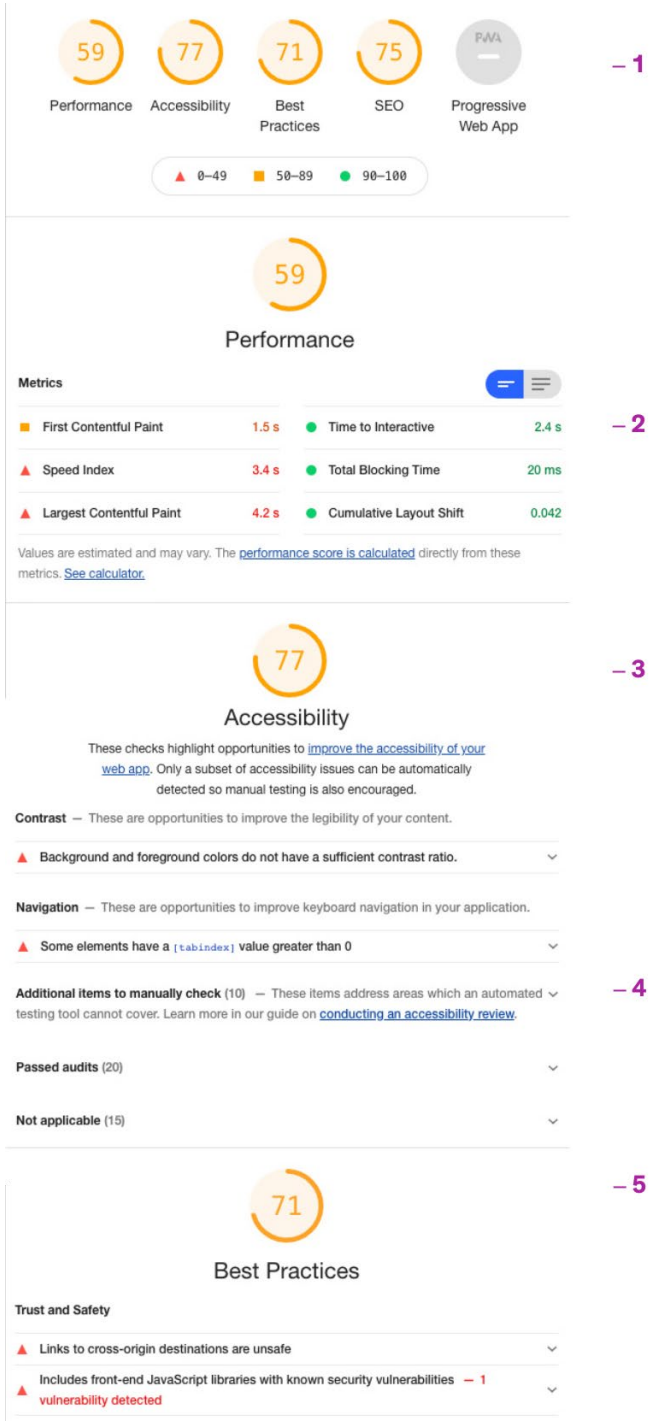
- Accessibility is its own self-contained segment
- The other segments (Performance, Best Practices, SEO, and Progressive Web App) only tangentially address accessibility concerns

Lighthouse also offers suggestions for manual evaluations

- It provides a 10-item checklist with a link to more information about conducting manual evaluations

Lighthouse Evaluation Sample

1. Segments results into five categories
2. Prioritizes results into three categories of importance
3. Accordions can be opened to show more information
4. Suggests manual surveys
5. Best Practices are general and not necessarily related to accessibility



SiteMorse Vendor Survey Overview

SiteMorse is UK-based vendor whose website evaluations can include accessibility

SiteMorse runs automated tests across specified pages and delivers a report

- Results are delivered in three views: (1) Priorities, (2) Report, and (3) Inventory

The Report view has most detailed and actionable reporting

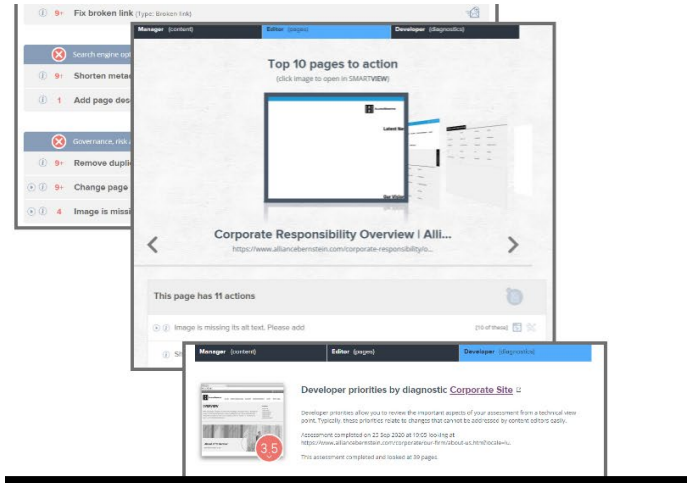
- The other views are too high-level to capture all issues

I also scanned the homepage again using SiteMorse's user-operated AI tool, SmartView

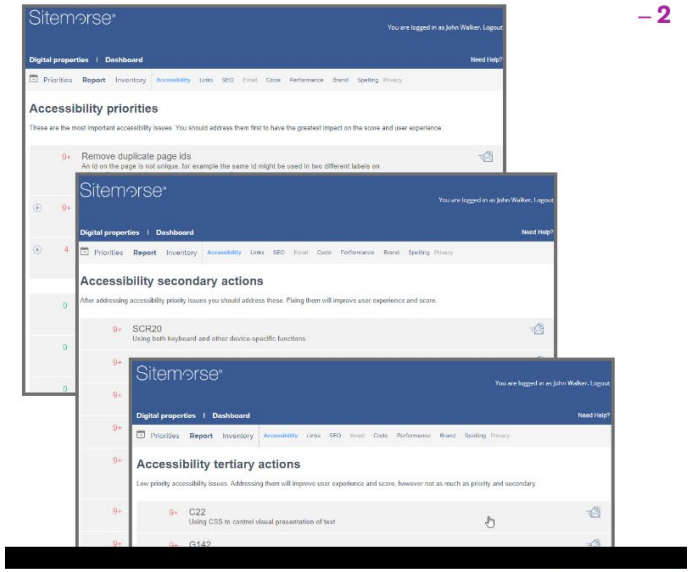
- The tool did uncover additional issues not documented in their report

SiteMorse Sample

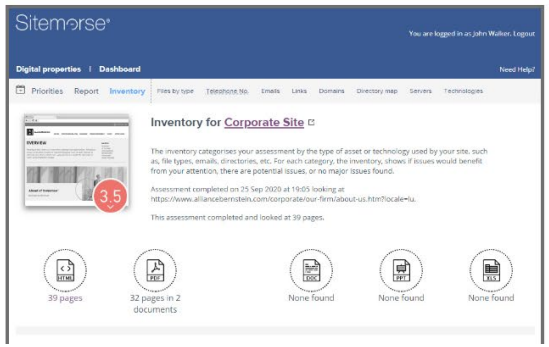
1. **Priorities can be filtered by Manager, Editor, Developer**
   - Provides high-level overviews with some clickable links
2. **Report view**
   - Segmented into primary, secondary and tertiary priorities
   - Clickable links, some of which lead to specific lines of code
3. **Inventory view**
   - Segmented into eight topic areas
   - Each topic is clickable and leads to a clickable list of flagged items



– 2



– 3



Manual Survey Checklist for Accessibility

I've created a list of items that benefit from a manual survey

JW

MANUAL SURVEY CHECKLIST FOR ACCESSIBILITY

AI tools don't catch all accessibility issues; human evaluation is also required.

Here's a list of some items to survey manually.

ACCESSIBILITY ISSUES

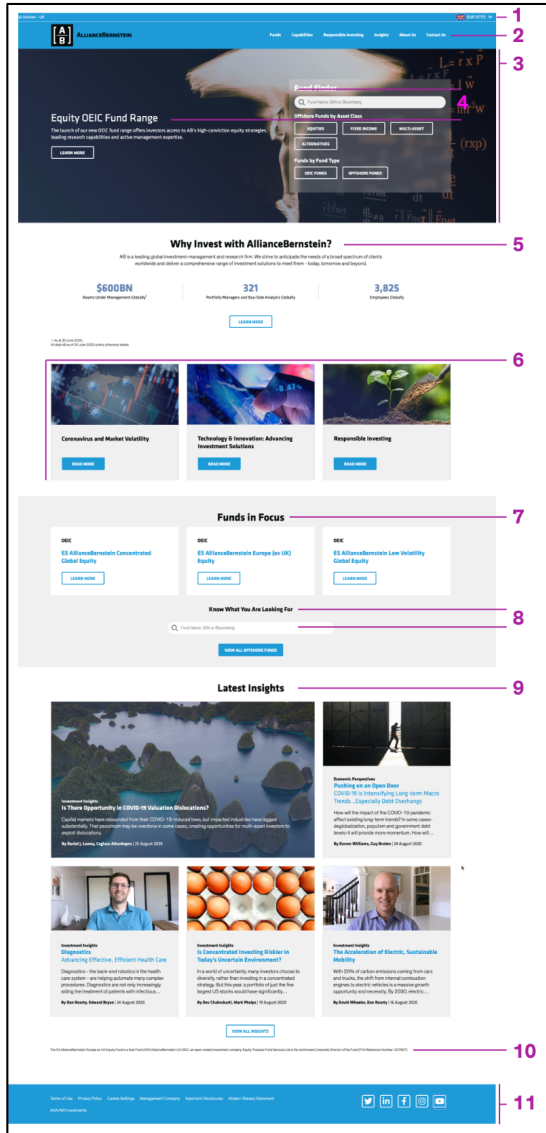
- ☐ Buttons, links, toggles all have accessible names
  - Including non-text controls such as X-close buttons
- ☐ Image elements have alt attributes
  - Decorative images can have empty alt attributes so screen readers ignore them
  - Images loaded as background images can't have alt text
- ☐ Videos include at least transcripts, and possibly closed-captioning
- ☐ Audio-only components include transcripts
- ☐ The page has a logical tab order
  - Tab through the page—the order in which elements are focused should follow the DOM order
  - Ensure you can access all the interactive elements easily
  - https://web.dev/logical-tab-order/?utm_source=lighthouse&utm_medium=devtools
- ☐ Interactive controls are keyboard focusable
  - Ensure controls are focusable and display a focus indicator
  - https://web.dev/focusable-controls/?utm_source=lighthouse&utm_medium=devtools
- ☐ Interactive elements indicate their purpose and state
  - Visual focus is indicated by a visual cue (color change, underline etc)
  - Screen readers announce the (specific—not generic) name of the control and its current state
  - https://web.dev/interactive-element-affordance/?utm_source=lighthouse&utm_medium=devtools
- ☐ The user's focus is directed to new content when added to the page
  - If new content is added to the page (via Javascript etc), the user's focus must be directed to it
  - The controller that updates the content should be positioned before the content, not after it, so users don't have to go backwards to see refreshed content
  - https://web.dev/managed-focus/?utm_source=lighthouse&utm_medium=devtools
  - https://www.w3.org/WAI/tutorials/carousels/

Manual Survey Overview

I did this manual survey using my checklist

There were many screen-reader issues the AI wasn't able to detect

- The header functionalities weren't fully operable
- The semantic CSS headers weren't optimally applied
- The page wasn't correctly divided into navigable, understandable sections
- The footer wasn't correctly implemented



1. Country navigation is inoperable

2. Some site navigation options are operable, some aren't

3. Hero is a "complementary" component (usually used for a sidebar to main content)

4. "Equity OEIC Fund Range" is an H1
   - "Fund Finder" is not an H and can't be found easily

5. "Why Invest with AB" is an H2 and creates a section

6. Cards have no context (no headline/no separate container)—confusing
   - Images lack alt text

7. "Funds in Focus" is an H3 and creates a section

8. Headline doesn't make sense in context so search bar is unexpected here

9. "Latest Insights" is an H3 and creates a section

10. Disclaimer is set up as both an article and a list (probably shouldn't be set up as either)

11. Not set up as a footer
    - Text inks are set up as navigation; social icons are set up as separate links

Matrix of Issues Detected by Method

I created a matrix to compare all issues detected

No single AI method captured all the issues

- In a perfect world, a combination of multiple tool scans and a third-party audit would be used
- In the real world, I imagine a combination of any two will suffice
- **But always in combination with a manual survey!**

	TOPIC	SITEMORSE*	SMARTVIEW*	LIGHTHOUSE*	AXE*
2	Broken links	X Priority: UX	X (Editor)		
3	Blue color not compliant	X (UX)	X (Dev)	X (Identified)	
4	Shorten metadata/description	X (SEO)			
5	Remove dupe page IDs	X (Governance)	X (Editor)		
6	Change page head order	X (Gov)			
7	Add alt text to images	X (Gov)	X (Editor)	X (Identified)	X (Critical)
8	Spelling	X (Gov)			
9	Fix broken links	X (Editor)			
10	Using CSS incorrectly	X (Dev)			
11	Link text lacks specificity		X (Dev)	X (Identified)	X (Serious)
12	Aseets size too large		X (Dev)		
13	Performance issues		X (Dev)	X (Identified)	
14	Spelling		X (Dev)		
15	Privacy questions		X (Dev)		
16	HTML questions	X (Dev)	X (Dev)		X (Moderate)
17	Form elements must have labels			X (Identified)	X (Critical)
18	Keyboard nav could be improved			X (Identified)	
19	Tab order			X (Manual)	
20	Keyboard focusable interactive elements			X (Manual)	
21	Interactive elements properly labeled			X (Manual)	
22	Focus on updated items			X (Manual)	
23	Focus trapped in region			X (Manual)	
24	Custom controls have proper labels/ARIA			X (Manual)	
25	Visual order matched DOM order	X (Dev)		X (Manual)	
26	Offscreen content hidden from reader			X (Manual)	
27	HTML landmarks used			X (Manual)	X (Moderate)
28	Zooming/scaling	X (Dev)			X (Critical)
29	Text used as header				X (Serious)
30	Lists improperly used				X (Moderate)
31	ID attributes must be unique				X (Minor)
32					
33		UX, SEO, Governance, Editor, Dev=Results filters	Editor, Dev=Results filters	Identified=Flagged by AI	Critical, Serious, Moderate, Minor=Results filters
34				Manual=Requires human supervision	
35					
36		** NOTE: SiteMorse results derive from 125+ pages and omits some issues (see other sheet)*	**NOTE: SMARTVIEW is a browser tool provided by SiteMorse but run by users*	**NOTE: Lighthouse is a tool built into the Chrome browser being used by our devs*	**NOTE: axe is a free Chrome browser extension used by most lawyers filing ADA lawsuits*