



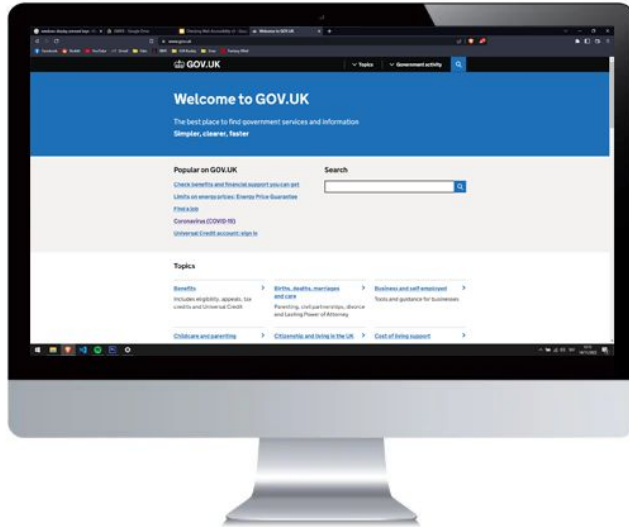
Checking Web Accessibility

Ožbej Golob, Alexander Thien, Olli-Pekka Riikola, Robin Karlsson

Information Architecture and Web Usability 2022/23

Using Keyboard to Navigate a Site

Good website: <https://www.gov.uk/>



Bad website: <https://www.mirror.co.uk/>



Keylogging captured with <http://code52.org/carnac/>.

Checking Web Accessibility

Showcase Video

Ožbej Golob

18.11.2022

Information Architecture and Web Usability WS 2022/2023

Group 4

Using keyboard to navigate gov.uk site

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

https://youtu.be/RMw_rKEzXdQ

Checking Web Accessibility

Showcase Video

Ožbej Golob

18.11.2022

Information Architecture and Web Usability WS 2022/2023

Group 4

Using keyboard to navigate mirror.co.uk site

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

https://youtu.be/48Qixhs4_Rg

Text-Only Browsers



- Originally, text browsers were the only way to browse the web.
- Nowadays, text browsers are useful on poor internet connection.
- Also, text browsers help to reduce distraction when browsing the web.

Note:

- Since most text browsers do not support JavaScript, some functionality will not be available.

Source: <https://linuxhint.com/best-linux-text-based-browsers/>

Known Text-Only Browsers



Text-browsers:

- *Lynx* - Probably most well known
<https://lynx.browser.org/>
- *WebbIE* - Dedicated for a screen reader
<https://www.webbie.org.uk/>
- *W3M* - Another stable browser
<https://w3m.sourceforge.net/>
- *Links* or *Links2* - Stable indieproject
<http://links.twibright.com/>

Other:

- *Browsh* - cli-browser with JS
<https://www.brow.sh/>
- Text mode - Chrome extension
[https://chrome.google.com/webstore/detail/text-mode/adelhekhakakacomdfejiipdnaadiiib](https://chrome.google.com/webstore/detail/text-mode/adelhekhakakocomdfejiipdnaadiiib)
- Firefox Built-In Reader View - F9
<https://support.mozilla.org/en-US/kb/firefox-reader-view-clutter-free-web-pages>

Showcase - Text Browsers

Checking Web Accessibility

Showcase Video

Olli-Pekka Riikola

20.11.2022

Information Architecture and Web Usability WS 2022/2023

Group 4

Browsing gov.uk and mirror.co.uk with Lynx

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

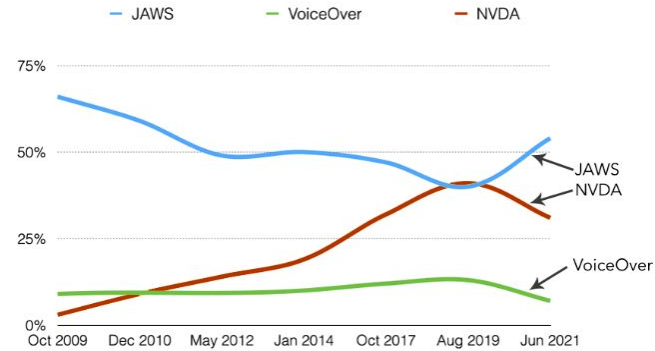
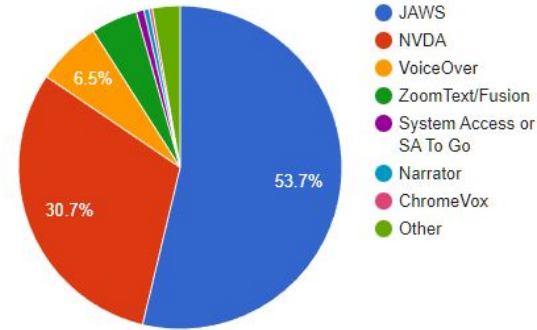
Short - <https://youtu.be/MyQjRt1pwwE>

Long - <https://youtu.be/fSYBxFXR4Q0>

Screen Readers

- JAWS (Windows, paid, 10 languages)
<https://webaim.org/articles/jaws/>
- NVDA (Windows, free and open-source, 63 languages)
<https://webaim.org/articles/nvda/>
- VoiceOver (iOS, free, 64 languages)
<https://webaim.org/articles/voiceover/>
- Narrator (Windows, free, 49 languages)
<https://webaim.org/articles/narrator/>
- TalkBack (Android, free, 63 languages)
<https://webaim.org/articles/talkback/>

Primary screen reader:



Screen Readers Detailed

Screen reader	Pros	Cons
<i>JAWS</i>	Most popular, most configurable	Learning curve, not free
<i>NVDA</i>	Free and open-source	Learning curve
<i>VoiceOver</i>	Bundled with Apple devices, free, user friendly and configurable	Learning curve
<i>Narrator</i>	Bundled with Windows, free	Limited functionality, navigation issues
<i>TalkBack</i>	Free and open-source	Learning curve

Showcase - Screen Readers

Checking Web Accessibility

Showcase Video

Ožbej Golob

18.11.2022

Information Architecture and Web Usability WS 2022/2023

Group 4

Using NVDA screen reader on gov.uk site

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

<https://youtu.be/oTBNtaBVXM0>

Screen Readers Showcase Videos



- JAWS:
 - gov.uk: https://youtu.be/aX_vK0JpYZE
 - mirror.co.uk: https://youtu.be/4zxA2TmmN_A
- NVDA
 - gov.uk: <https://youtu.be/oTBNtaBVXM0>
 - mirror.co.uk: <https://youtu.be/SY2njevCTul>
- VoiceOver
 - gov.uk: <https://youtu.be/UBPnCozggK4>
 - mirror.co.uk: <https://youtu.be/4sDqR2oT0c>
- Narrator
 - gov.uk: https://youtu.be/ZVTOI2_vw1s
 - mirror.co.uk: <https://youtu.be/LktZKcB7xNo>
- TalkBack
 - gov.uk: https://youtu.be/Hx_pl2bZ930
 - mirror.co.uk: <https://youtu.be/EyYMB9Kn8Xw>

Browser Extensions for Accessibility Audit



1. axe DevTools

<https://deque.com/axe/devtools/>

2. Accessibility Insights for Web

<https://accessibilityinsights.io/>

3. Google Lighthouse

<https://developer.chrome.com/docs/lighthouse/accessibility/>

4. Siteimprove

<https://chrome.google.com/webstore/detail/siteimprove-accessibility/efcfolpjihicnikpmhnmphjhhpiclljc>

5. WAVE

<https://wave.webaim.org/extension/>

Browser Extensions for Accessibility Audit

Browser extension	Browser	Licence
<i>axe DevTools</i>	Chrome, Firefox, Edge	Free, Commercial
<i>Accessibility Insights for Web</i>	Chrome, Edge	Free, Open-source
<i>Google Lighthouse</i>	Chrome	Free, Open-source
<i>Siteimprove</i>	Chrome, Firefox, Edge, Opera	Free
<i>WAVE</i>	Chrome, Firefox, Edge	Free

Sources: <https://developer.chrome.com/docs/lighthouse/accessibility/>, <https://deque.com/axe/devtools/>, <https://wave.webaim.org/>, <https://www.siteimprove.com/>, <https://accessibilityinsights.io/docs/web/overview/>

1. axe DevTools

- Uses axe-core library
- Very good issue descriptions
- Paid version has many extra features
 - Audit to other than WCAG 2.1 AAA
 - Guided tests
 - Scan part of page
 - Page structure
 - Exporting report
- Great element highlighting

The image displays two screenshots of the axe DevTools interface. The left screenshot shows the 'Overview' tab for a test on 'https://www.gov.uk/'. The interface includes a 'Test Name' field, a 'Test URL' field, and a 'TOTAL ISSUES' summary card showing 2 issues. Below the summary, there is a table of issue counts: AUTOMATIC ISSUES (2), GUIDED ISSUES (0), Needs Review (1), Best Practice (1), Critical (0), Serious (0), Moderate (1), and Minor (0). The 'Best Practices' section is set to 'ON' and 'WCAG 2.1 AAA'. The 'Issues: Total Issues' section shows 2 issues, with two specific issues listed: 'All page content should be contained by landmarks' (1) and 'Elements must have sufficient color contrast' (1). A green cross is overlaid on the left side of this screenshot.

The right screenshot shows the 'Overview' tab for a test on 'https://www.mirror.co.uk/'. The interface includes a 'Test Name' field, a 'Test URL' field, and a 'TOTAL ISSUES' summary card showing 268 issues. Below the summary, there is a table of issue counts: AUTOMATIC ISSUES (268), GUIDED ISSUES (0), Needs Review (28), Best Practice (26), Critical (15), Serious (204), Moderate (12), and Minor (9). The 'Best Practices' section is set to 'ON' and 'WCAG 2.1 AAA'. The 'Issues: Total Issues' section shows 268 issues, with several specific issues listed: 'Elements must only use allowed ARIA attributes' (1), 'Buttons must have discernible text' (1), 'Elements must have sufficient color contrast' (25), 'IDs of active elements must be unique' (1), 'id attribute value must be unique' (4), 'Frames must have an accessible name' (2), and 'Images must have alternate text' (14). A red box is overlaid on the right side of this screenshot.

2. Accessibility Insights for Web

- Uses axe-core library
- Very good issue descriptions
- Problematic elements clearly visible on page
- Full assessment (guided test)
- Exporting as HTML or to GitHub, Azure Boards

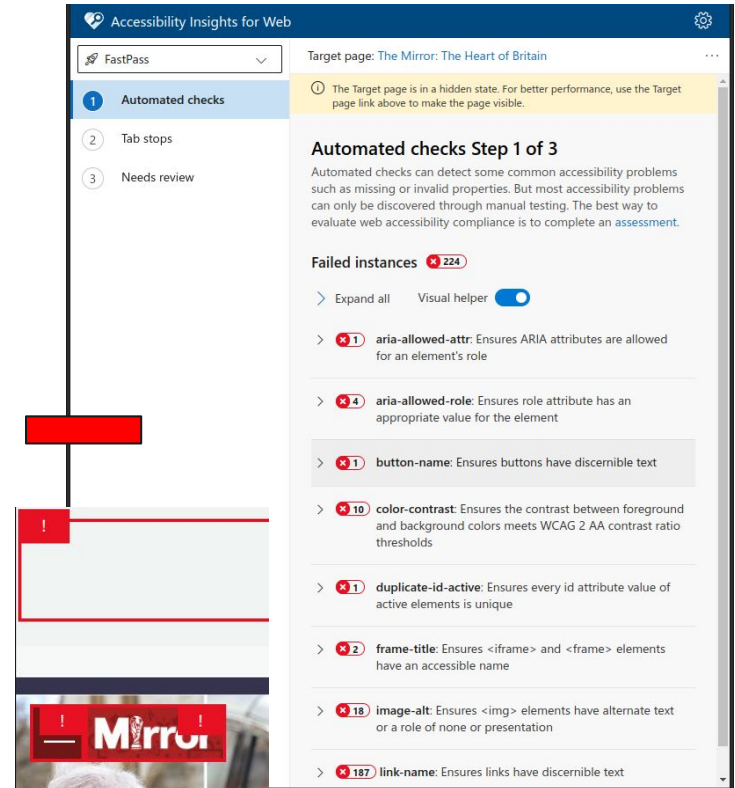

Needs review Step 3 of 3

Sometimes automated checks identify *possible* accessibility problems that need to be reviewed and verified by a human. Because most accessibility problems can only be discovered through manual testing, we recommend an [assessment](#).

Instances to review 1

> Expand all Visual helper

> 1 **color-contrast**: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds



Accessibility Insights for Web

FastPass

Target page: The Mirror: The Heart of Britain

1 Automated checks

2 Tab stops

3 Needs review

The Target page is in a hidden state. For better performance, use the Target page link above to make the page visible.

Automated checks Step 1 of 3

Automated checks can detect some common accessibility problems such as missing or invalid properties. But most accessibility problems can only be discovered through manual testing. The best way to evaluate web accessibility compliance is to complete an [assessment](#).

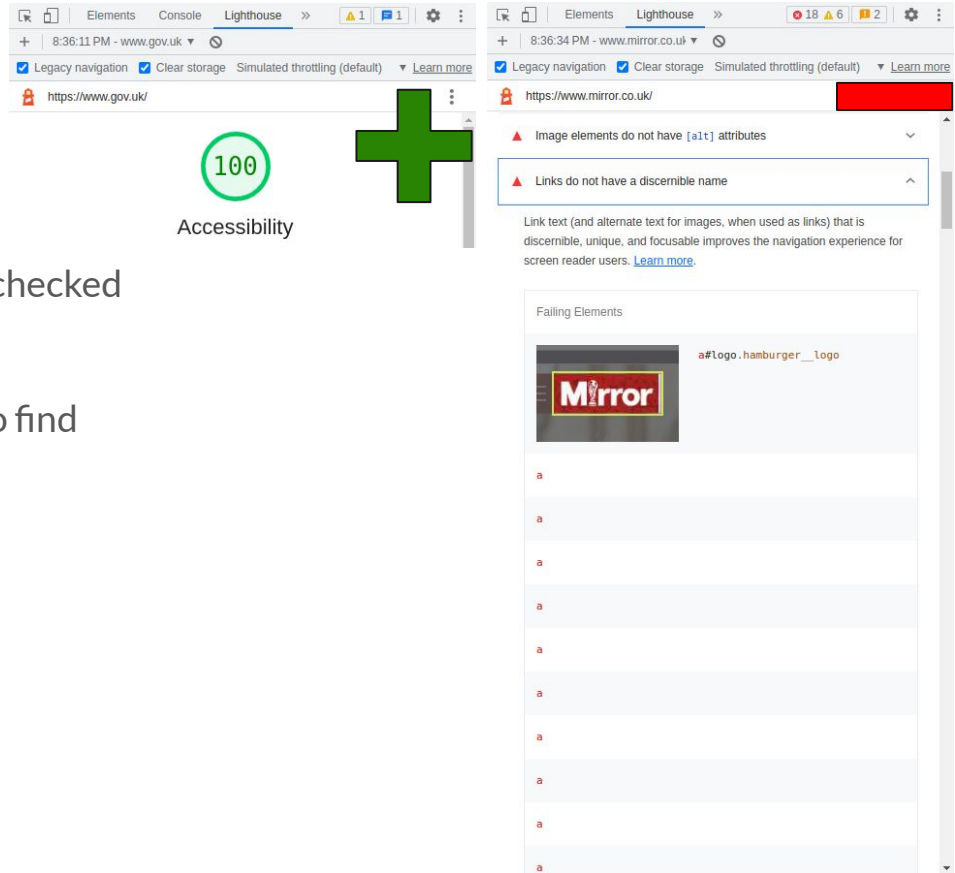
Failed instances 224

> Expand all Visual helper

- > 1 **aria-allowed-attr**: Ensures ARIA attributes are allowed for an element's role
- > 4 **aria-allowed-role**: Ensures role attribute has an appropriate value for the element
- > 1 **button-name**: Ensures buttons have discernible text
- > 10 **color-contrast**: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
- > 1 **duplicate-id-active**: Ensures every id attribute value of active elements is unique
- > 2 **frame-title**: Ensures <iframe> and <frame> elements have an accessible name
- > 18 **image-alt**: Ensures elements have alternate text or a role of none or presentation
- > 187 **link-name**: Ensures links have discernible text

3. Google Lighthouse

- Uses axe-core library
- Good issue descriptions
- Links to learn more about issues
- List of items that were checked and not checked
- Exporting report to JSON
- Problematic elements not always easy to find



4. Siteimprove

- Good descriptions
- Very good filtering of issues
 - E.g. by role, difficulty, WCAG level
- Simulate color blindness
- Code examples (somewhat useless)
- No export of report

Siteimprove Accessibility Checker interface for 'Welcome to GOV.UK'. The page shows 2 issues. The first issue is 'Text not included in an ARIA landmark' (ARIA WAI-ARIA authoring practices) with 6 instances. The second issue is 'Line height is below minimum value' (AAA 1.4.8 Visual Presentation) with 26 instances.

Siteimprove Accessibility Checker interface for 'The Mirror: The Heart of Britain'. The page shows 7 issues. The first issue is 'Link without a text alternative' (2.4.4 Link Purpose (In Context), 2.4.9 Link Purpose (Link Only), 4.1.2 Name, Role, Value) with 199 instances. The second issue is 'Button without a text alternative' (4.1.2 Name, Role, Value) with 1 instance. The third issue is 'Visible label and accessible name do not match' (2.5.3 Label in Name) with 199 instances. The fourth issue is 'Image without a text alternative' (1.1.1 Non-text Content) with 10 instances. The fifth issue is 'Element IDs are not unique' (4.1.1 Parsing) with 10 instances. The sixth issue is 'Text not included in an ARIA landmark' (ARIA WAI-ARIA authoring practices) with 4 instances. The seventh issue is 'Color contrast is not sufficient' (1.4.3 Contrast (Minimum), 1.4.6 Contrast (Enhanced)) with 5 instances.

Browser Extensions for Accessibility Audit

Overall:

- Many good options to choose from
- Accessibility Insights for Web, axe DevTools and Lighthouse use axe-core library
 - axe DevTools reports some additional issues
- The best tool depends on your priorities

Recommendation:

1. axe DevTools
 - Use if budget allows
 - Good set of features
2. Accessibility Insights for Web:
 - Good free alternative to axe DevTools
 - Same library, overall nicer than Lighthouse
 - Free guided test
3. Siteimprove:
 - Good tool to get more accessibility issues than reported by axe-core
 - Excellent filtering

Web Tools For Accessibility Audits



1. Accessi- <https://www.accessi.org/>
2. WAVE - <https://wave.webaim.org/>
3. Lighthouse (Page Speed Insights)- <https://pagespeed.web.dev/>
 - Tenon - <https://tenon.io/>
 - SiteImprove - <https://www.siteimprove.com/toolkit/accessibility-checker/>
 - WebAccessibility - <https://www.webaccessibility.com/>

Showcase - Web Tools For Accessibility Audits

Checking Web Accessibility

Showcase Video

Alexander Thien

20.11.2022

Information Architecture and Web Usability WS 2022/2023

Group 4

Using Lighthouse web tool for accessibility audit

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

Video: https://youtu.be/ui_RQ2O-zhM

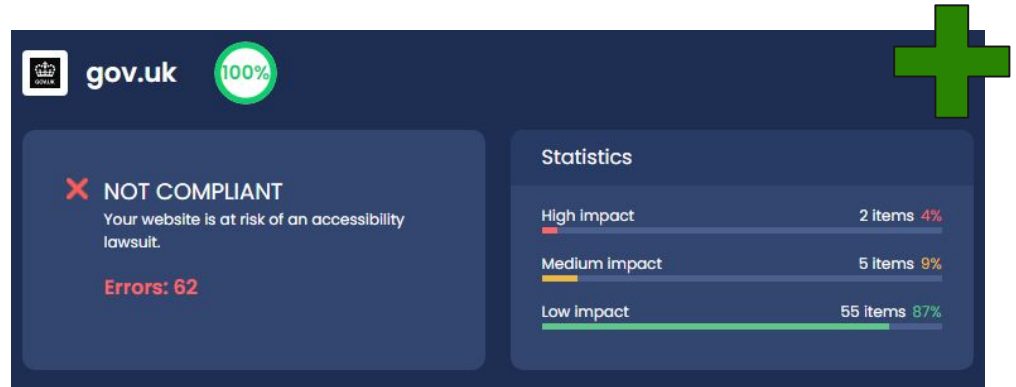
1. Accessi

Positive example:

- Reached and evaluated
- 2 high impact errors
- 5 medium impact errors
- 55 medium impact errors

Negative example:

- Reached and evaluated
- 221 high impact errors
- 220 medium impact errors



2. WAVE



Positive example:

- Reached and evaluated
- No ERRORS
- little problems detected

Negative example:

- Scripting so bad the tool cant even work properly

WAVE powered by WebAIM
web accessibility evaluation tool

Address: <https://gov.uk>

Styles: OFF ON

Summary

Summary Details Reference Structure Contrast

0 Errors	1 Contrast Errors
5 Alerts	16 Features
68 Structural Elements	38 ARIA

View details >

WAVE powered by WebAIM
web accessibility evaluation tool

Address: <https://www.mirror.co.uk/>

Error
There is a scripting conflict with content on this page. We recommend using the WAVE [Chrome or Firefox Extension](#) to evaluate this page.

Styles: OFF ON

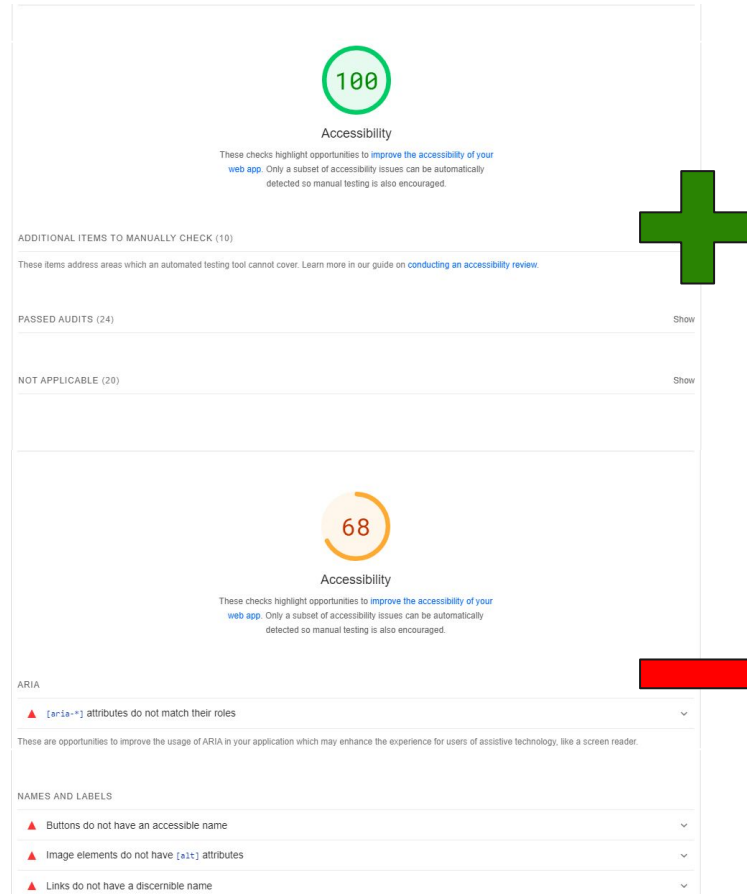
3. Lighthouse

Positive example:

- Accessibility Score 100
- No errors found

Negative example:

- Accessibility Score 70
- Several errors with explanations



The image shows two screenshots of the Lighthouse Accessibility report. The top screenshot shows a score of 100, indicating a perfect accessibility score. The bottom screenshot shows a score of 68, indicating several accessibility issues. A large green plus sign is overlaid on the right side of the top screenshot, and a red rectangle is overlaid on the right side of the bottom screenshot.

100
Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)
These items address areas which an automated testing tool cannot cover. [Learn more in our guide on conducting an accessibility review.](#)

PASSED AUDITS (24) Show

NOT APPLICABLE (20) Show

68
Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ARIA

▲ `[aria-*]` attributes do not match their roles ▼

These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

NAMES AND LABELS

▲ Buttons do not have an accessible name ▼

▲ Image elements do not have `[alt]` attributes ▼

▲ Links do not have a discernible name ▼

Conclusion Web Tools



Overall:

- Many free solutions
- Mostly registering alone already lets you view more details
- Some tools are stricter than others
- Overall the evaluation of each tool came to the same conclusions

Recommendation:

1. Accessi:

- Detailed
- Gives Examples
- PDF report


2. WAVE:

- Interactive tool
- Can click on every element and inspect
- Graphically pleasing and well structured

3. Lighthouse:

- Detailed and very explanatory
- Seems to be regarded a standard

Accessibility Audit Tools in Build Process

- 
- Lighthouse
 - axe Linter
 - WAVE Site-wide tools

These tools were not a part of our survey.