

Public Summary of “Accessibility Audit report Sofie”

Based on Document Version Number: V1.1

Test performed on: 10/11/23

Table of Contents

Contents

Public Summary of “Accessibility Audit report Sofie”	1
Table of Contents.....	2
Report Summary.....	3
Tools & Browsers.....	4
Tools.....	4
Browsers.....	4
Policy, Standards and Methods.....	4
Detailed Findings.....	5
Empty Buttons.....	5
Colour contrast issues.....	6
Zoom issues on the page.....	7
Form label.....	8
Skipped headings.....	9
Sensory characteristics.....	10
Keyboard accessibility issues.....	12
Table structure.....	13
The Toggle button was not announced.....	14
Interactive elements.....	14
Unlabelled form fields.....	15
WCAG Test Table.....	17
Appendix I.....	29
JAWS.....	29
Dragon Naturally Speaking.....	29
ZoomText.....	30

Report Summary

There were a significant number of accessibility issues identified during the testing process which would impact multiple user groups including screen reader users, keyboard-only users, Dragon users, and users with limited vision. Keyboard-only users, VI users Motor impaired users will be largely impacted along with other disabilities.

There are form fields that do not have a programmatically associated or a form control does not have a corresponding label which would prevent screen reader users from being able to interact with the product.

Many times meaningful sequence is not seen, and keyboard focus is not visible most of the time. Inaccessible expanders, and resize issues Also there are several components that are inaccessible for users who navigate with a keyboard only would be unable to complete several key processes with this product.

KO users can use some elements but when the keyboard is used with a screen reader expander element and some elements like toggle buttons etc., are inaccessible.

A sampling of the in-scope pages and some components has been done however issues highlighted in this document are impacted on all the pages as similar components are used across the application. Due to multiple issues, it was impossible to note down all issues on all the pages in the document, but we have tried to highlight the major ones.

A summary of the issues is contained in the report below.

Tools & Browsers

Tools

- JAWS 2023
- Dragon naturally speaking 15.
- Zoom text 2023

Browsers

- Google Chrome 118.0.5993.90 (Official Build) (32-bit)

Policy, Standards and Methods

The main supported technology consists of:

- [Job Access with Speech](#) (JAWS) v2023
- [Dragon Naturally Speaking](#) v15.3
- [ZoomText](#) v 2023

These software tools run on Windows 10 (release 20h2) and Firefox (latest release) and Chrome (Latest release) browsers.

Note: Additional AT software might be added to this list e.g. [NVDA](#) and [Apple Mac Voiceover](#).

Detailed Findings

Empty Buttons

Issue 1: The drag to reorder or move out of the playlist button is empty or has no value text.

Page title: Sofie

Screen Shot:

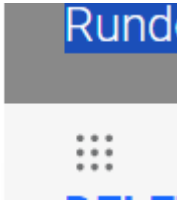


Figure 1: drag to reorder or move out of playlist button on the screen is empty



Figure 2: All of these buttons are unnamed and empty.

Drag to reorder or move out of playlist button is empty or has no value text. When navigating to a button, descriptive text must be presented to screen reader users to indicate the function of the button.

Recommendation

Place text content within the <button> element or give the <input> element a value attribute. When navigating to a button, descriptive text must be presented to screen reader users to indicate the function of the button. This needs to be fixed across all the buttons on the table.

Priority: High

Impact: High

1.1.1 Non-text Content (Level A)

2.4.4 Link Purpose (In Context) (Level A)

Colour contrast issues

Issue 2: Colour contrast failing for text.

Page title: Sofie

Screen Shot:

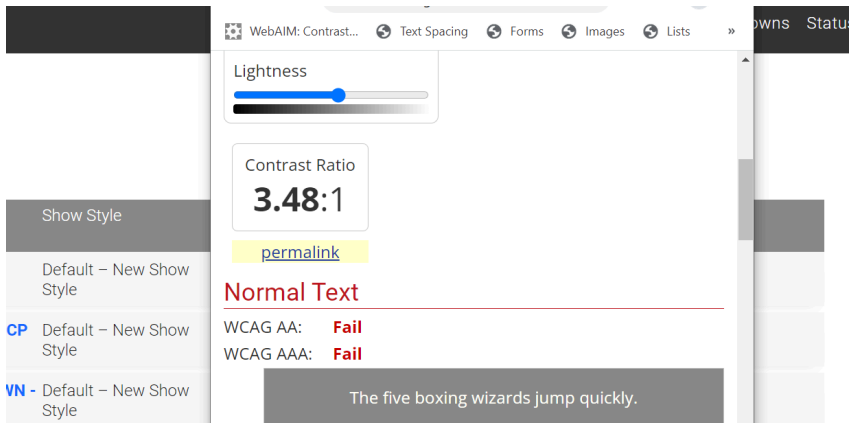


Figure 3: Colour contrast ratio failing for text on the table header.

Show Style	On Air Start Time	Duration	Expected End Time	Last updated
Default - New Show Style	21 days ago	Not set	Not set	Yesterday 15:06:38
Default - New Show Style	2 months ago	00:10:00	2 months ago	Last Saturday 00:00:08
Default - New Show Style	3 months ago	00:04:29	3 months ago	Last Monday 16:04:53
Default - New Show Style	Not set	Not set	Not set	Last Thursday 20:51:08
Default - New Show Style	2 months ago	00:26:52	2 months ago	Last Monday 16:05:01
Default - New Show Style	a month ago	00:27:16	a month ago	Last Monday 16:04:54

Figure 4: Colour contrast issues for text on the table.

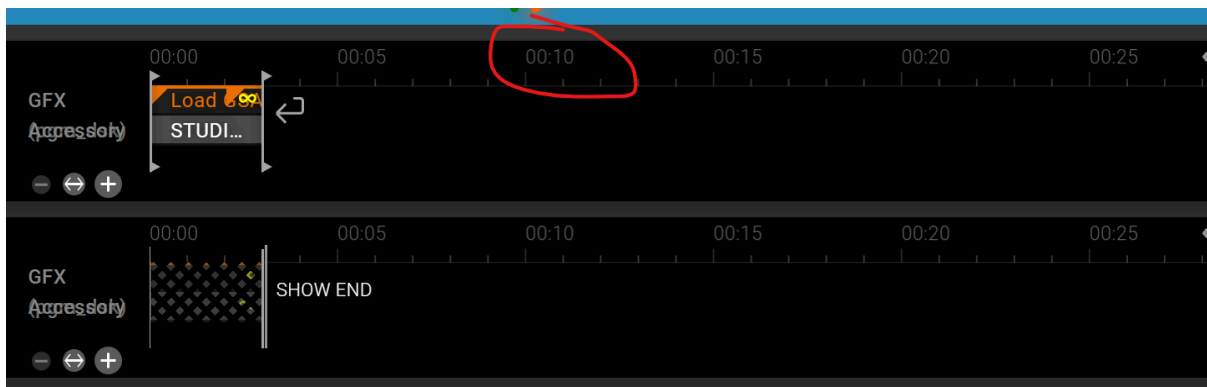


Figure 5: Colour contrast issues on the page.

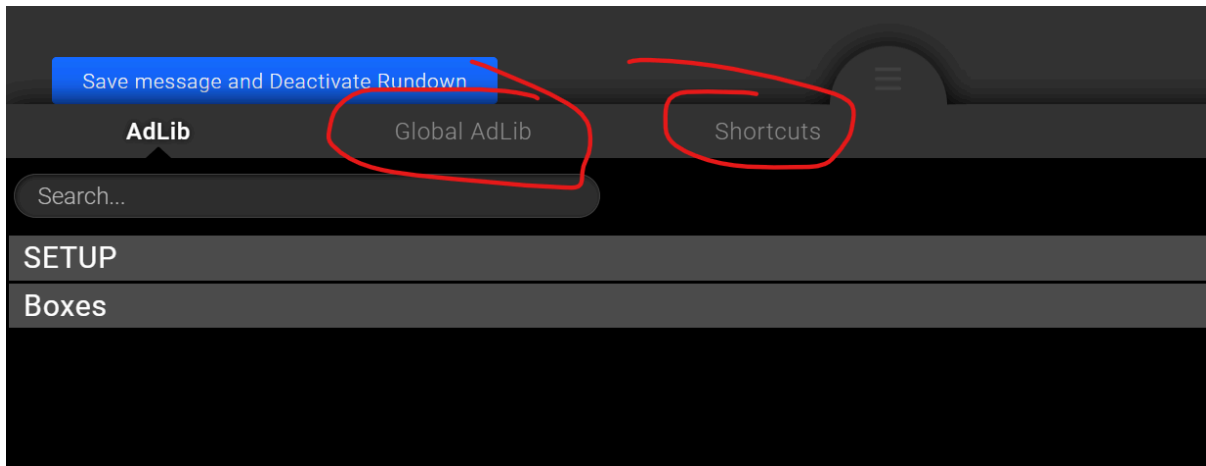


Figure 6: Colour contrast issues for text.

Very low contrast between the text and background colours.

Recommendation

Adequate contrast of text is necessary for all users, especially users with low vision. Increase the contrast between the foreground (text) colour and the background colour. Large text (larger than 18 point or 14 point bold) does not require as much contrast as smaller text. Colour contrast ratio should minimum should be 4.5:1.

Priority: Medium

Impact: Medium

1.4.3 Contrast (Minimum) (Level AA)

Zoom issues on the page

Issue 3: Zoom up to 200% on the browser overlaps the text

Page title: Sofie

Screen Shot:

Rundown	Show Style	On Air Start Time	Duration	Expected End Time	Last updated
⋮ Peter's Test Rundown - DO NOT DELETE	Default - New Show Style	21 days ago	Not set	Not set	Yesterday 15:06:38
⋮ 0000 Screen Control Mosart VDCP Test	Default - New Show Style	2 months ago	00:10:00	00:00:08	Last Saturday

Figure 7: Zoom up to 200% overlaps the text and issues need to be fixed across all the pages.

When zooming up to 200% in a web browser causes text to overlap.

Recommendation

When zooming up to 200% in a web browser causes text to overlap and become unreadable, it's often a sign of poor text scaling or layout issues.

Use Relative Units: Ensure that your web page uses relative units (such as "em" or "%" for font sizes and element dimensions) rather than fixed units (like "px"). Relative units adapt better to zooming, as they scale proportionally.

Create a responsive web design that adjusts the layout and spacing of elements based on screen size and zoom level. This can help prevent overlapping when users zoom in.

Priority: Medium

Impact: Medium

Form label

Issue 4: A form control does not have a corresponding label.

Page title: Sofie

Screen Shot:

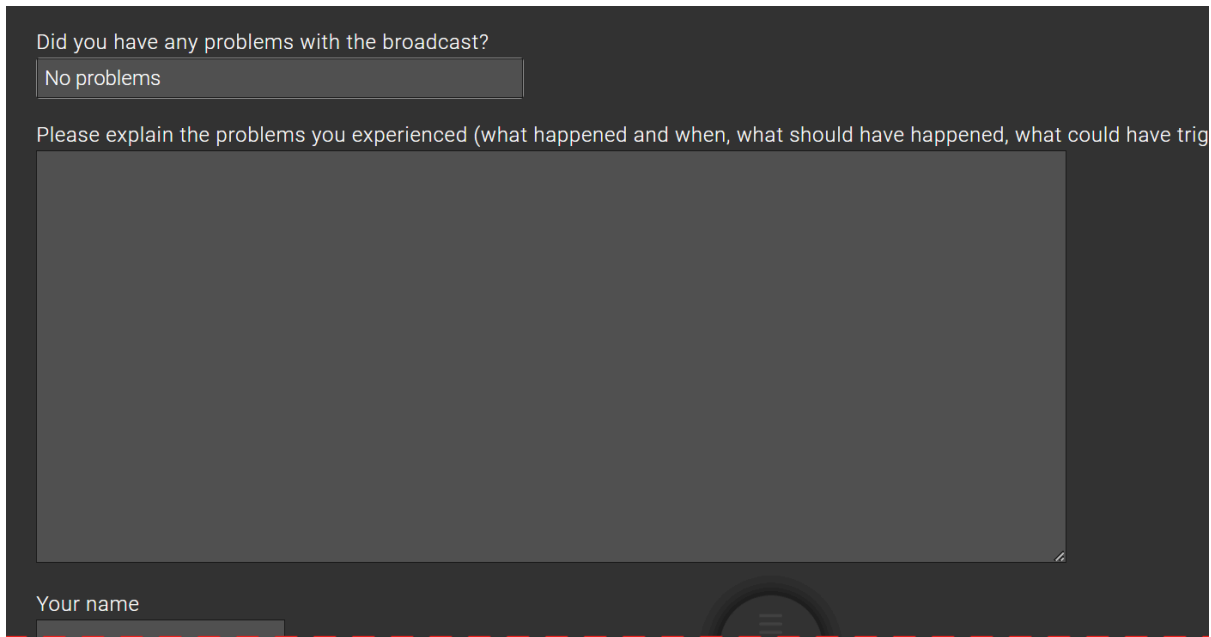


Figure 8: Screenshot displaying form fields do not have corresponding labels.

The dropdown text box and your name field do not have corresponding labels associated programmatically.

Recommendation

If a form control does not have a properly associated text label, the function or purpose of that form control may not be presented to screen reader users. Form labels also provide visible descriptions and larger clickable targets for form controls. If a text label for a form control is visible, use the <label> element to associate it with its respective form control. If there is no visible label, either provide an associated label, add a descriptive title attribute to the form control, or reference the label(s) using aria-labelledby. Labels are not required for image, submit, reset, button, or hidden form controls.

Priority: Medium

Impact: High

1.3.1 Info and Relationships (Level A)

3.3.2 Labels or Instructions (Level A)

Skipped headings

Issue 5: A page does not have a first-level heading.

Page title: Sofie

Screen Shot:

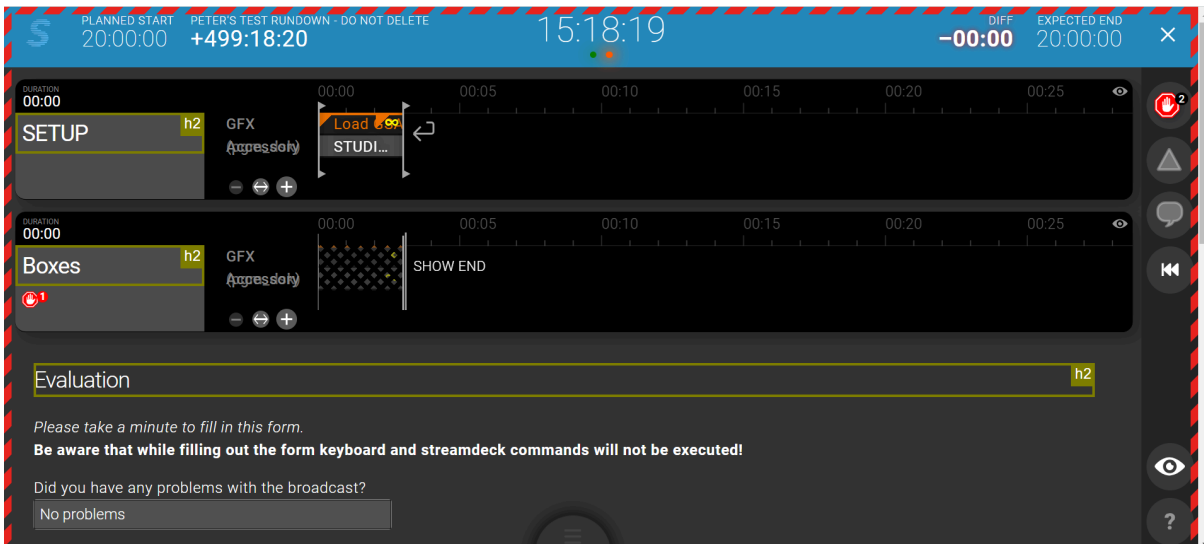


Figure 9: Heading levels are skipped, and no H1 heading is present.

Headings facilitate page navigation for users of many assistive technologies. They also provide semantic and visual meaning and structure to the document. A first-level heading (<h1>) should be present on all pages. It should contain the most important heading on the page (the document title).

Recommendation

If the page presents a main heading, place it within an <h1> element. Add other sub-headings as necessary.

Priority: Medium

Impact: Medium

1.3.1 Info and Relationships (Level A)

2.4.6 Headings and Labels (Level AA)

???skipped headings does not fail wcag, it's still a usability issue though

Sensory characteristics

Issue 6: Buttons and shapes do not have visible form labels.

Page title: Sofie

Screen Shot:

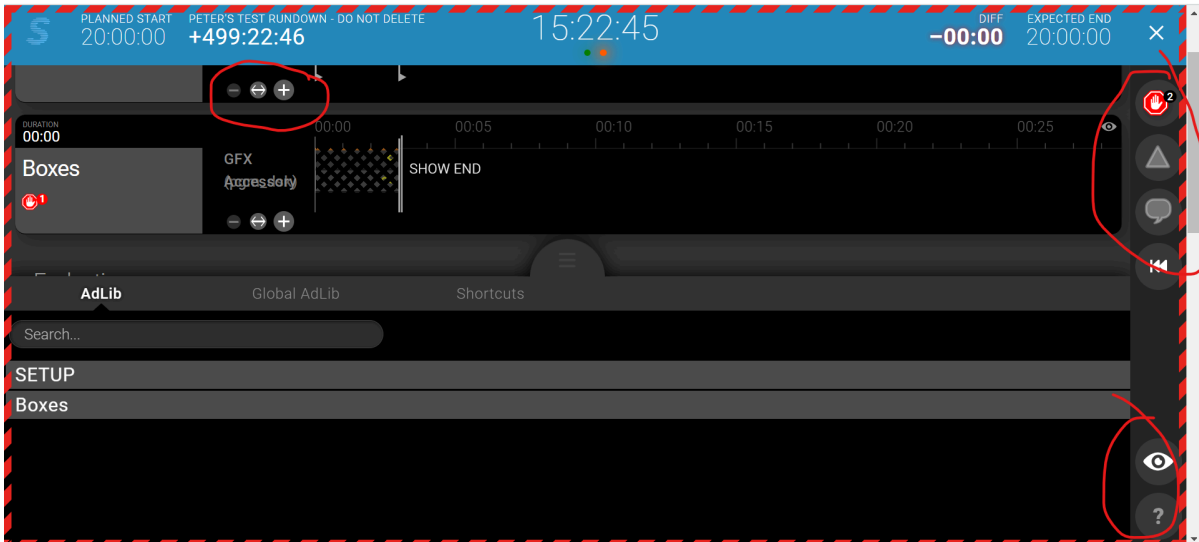


Figure 10: Buttons and shapes do not have visible form labels.

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound.

Recommendation

When sensory characteristics (e.g., buttons and shapes) rely on visual cues such as shapes alone, it can create accessibility issues for individuals who may not be able to perceive these visual cues.

Use Semantic HTML Elements: Instead of relying solely on visual cues like shapes, use semantic HTML elements to describe the purpose or function of buttons and shapes. For example, use `<button>` elements with appropriate labels or descriptive text to convey their meaning.

Provide Text Labels: Ensure that all buttons and interactive elements have clear and concise text labels. These labels should indicate their function or purpose. If the shape itself is significant, you can include text labels alongside it.

Implement Alternative Text (Alt Text): For non-text content like icons or images used as buttons, provide meaningful alt text that describes their purpose. Screen readers can convey this information to users who cannot see the visual cues.

Use ARIA Roles and Attributes: If you're using custom or non-standard elements, use ARIA (Accessible Rich Internet Applications) roles and attributes to provide additional accessibility information. For example, you can use `aria-label` or `aria-labelledby` to associate text with elements.

Priority: High

Impact: High

Keyboard accessibility issues

Issue 7: Overlays or modal dialogs that are inaccessible by keyboard

Page title: Sofie

Screen Shot:

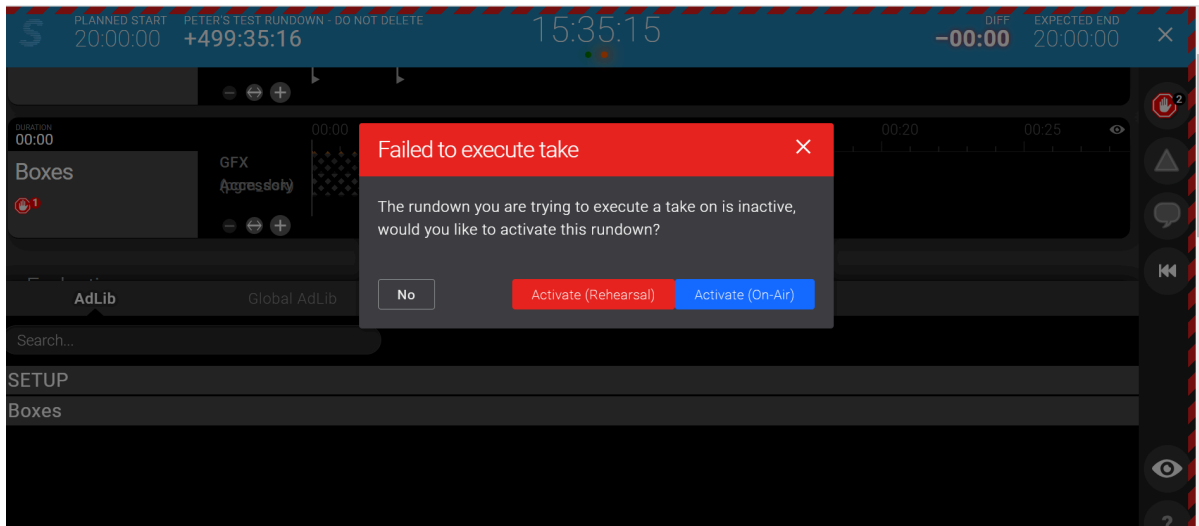


Figure 11: Modal dialogue is inaccessible by keyboard.

Overlays or modal dialogs that are inaccessible by keyboard navigation create usability problems for individuals who rely on keyboard input, including those with motor impairments or vision disabilities. When overlays cannot be operated with a keyboard, users may be unable to interact with or dismiss them, effectively locking them out of content or functionality.

Recommendation

Focus Management: Ensure that keyboard focus is managed within the overlay. When an overlay appears, move the keyboard focus to an element within the overlay, making it the active element. When users press the "Tab" key, the focus should cycle through all interactive elements within the overlay in a logical order.

Close or Dismiss Overlay: Provide a keyboard-accessible way to close or dismiss the overlay. Typically, the "Esc" key serves as a common keyboard shortcut for closing overlays. Make sure this functionality is indicated to users.

Trap Focus: Implement keyboard focus trapping to prevent users from navigating outside of the overlay while it is open. This ensures that users can only interact with the overlay's content until it is closed.

ARIA Roles: Use ARIA roles, such as "dialog" or "alertdialog," to semantically identify the overlay to assistive technologies. This helps screen readers understand the purpose of the overlay and its relationship to the rest of the page.

Priority: High
Impact: High

2.1.1 Keyboard (Level A)

2.4.3 Focus Order(Level A)

Table structure

Issue 8: information is presented in rows and columns, but a table is not properly defined.

Page title: Sofie

Screen Shot:



	Show Style	On Air Start Time	Duration	Expected End Time	Last updated
DO NOT	Default – New Show Style	23 days ago	Not set	Not set	Last Tuesday 15:06:38
osart VDCP	Default – New Show Style	2 months ago	00:10:00	2 months ago	Last Wednesday 14:49:26
RUNDOWN -	Default – New Show Style	3 months ago	00:04:29	3 months ago	Yesterday 16:04:28
p	Default – New Show Style	Not set	Not set	Not set	Last Wednesday 14:49:26
TREAMS OT DELETE	Default – New Show Style	2 months ago	00:26:52	2 months ago	Yesterday 16:04:19

Figure 12: Information not presented in the table, no caption and summary provided.

When information is presented in rows and columns, but a table is not properly defined in the markup, it can lead to accessibility issues. Assistive technologies, such as screen readers, rely on the semantic structure of HTML to convey information effectively to users. Failing to define a table may result in a loss of context and make it challenging for users, especially those with disabilities, to understand the relationships between data elements.

Recommendation

Use Semantic HTML: Enclose tabular data within the appropriate HTML `<table>` element. Define the structure using `<thead>`, `<tbody>`, and `<tfoot>` as needed. Use `<th>` for header cells and `<td>` for data cells.

Use Scope Attribute: For header cells (`<th>`), use the scope attribute to indicate whether they apply to a column (`scope="col"`) or a row (`scope="row"`).

Provide Captions: Include a `<caption>` element within the `<table>` to provide a brief, descriptive title for the table.

Priority: Medium

Impact: Medium

1.3.1 Info and Relationships (Level A)

4.1.2 Name, Role, Value (Level A)

The Toggle button was not announced

Issue 9: The Toggle button is not announced to screen reader users.

Page title: Sofie

Screen Shot:

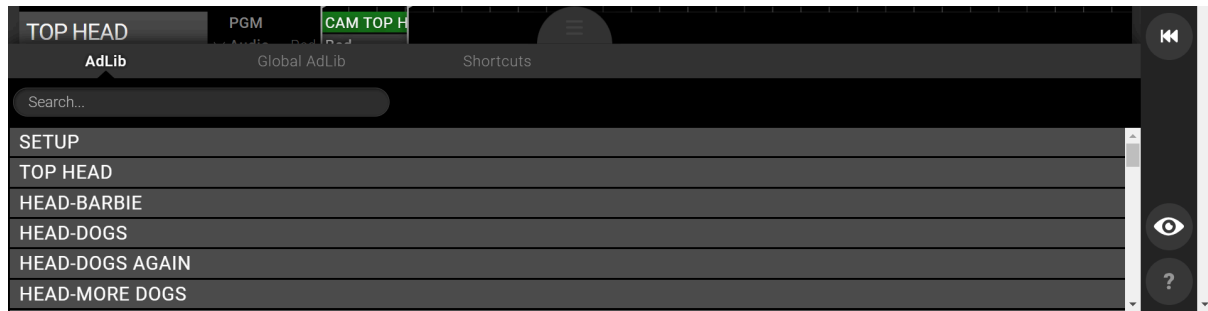


Figure 13: Toggle button not announced to screen reader user

When a toggle button is not announced to screen reader users, it creates a significant accessibility barrier. Screen reader users rely on audible feedback to understand the state and functionality of interactive elements on a web page. Failing to announce a toggle button's status can lead to confusion and a lack of awareness about its purpose and whether it is currently activated or deactivated.

Recommendation

Ensure that you are using semantic HTML elements such as `<button>` for your toggle button. This helps assistive technologies properly identify the interactive nature of the element.

Use ARIA (Accessible Rich Internet Applications) attributes to convey additional information about the toggle button. In this case, you can use the `aria-pressed` attribute to indicate the current state (pressed or not pressed).

Ensure that your JavaScript logic for toggling the button updates the `aria-pressed` attribute accordingly. When the button is pressed, set `aria-pressed="true"`, and when it is not pressed, set `aria-pressed="false"`

Priority: High/Medium/Low

Impact: High/Medium/Low

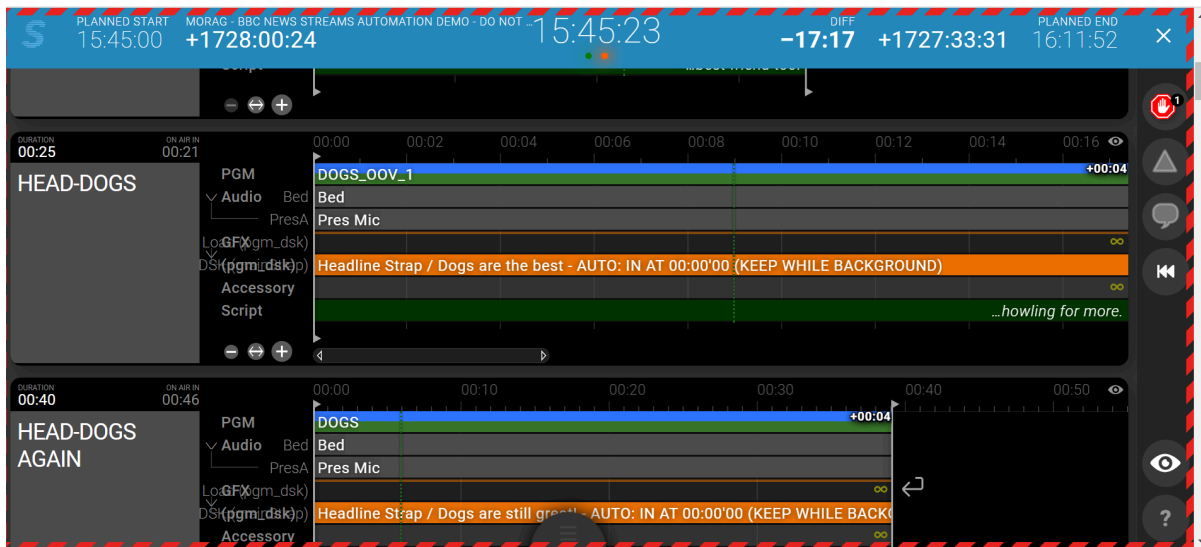
4.1.2 Name, Role, Value (Level A)

Interactive elements

Issue 10: while navigating through the page using the keyboard's Tab and Shift+Tab keys, users are unable to access all page functionalities and interactive components.

Page title: Sofie

Screen Shot:



while navigating through the page using the keyboard's Tab and Shift+Tab keys, users are unable to access all page functionalities and interactive components. However, these functionalities become accessible when using arrow keys.

Recommendation

Ensure that all interactive elements (buttons, links, form fields, etc.) are coded using semantic HTML. Use appropriate HTML tags (<button>, <a>, <input>, etc.) to represent the nature of each interactive component.

Ensure that the tab index is correctly set for interactive elements. The tab index determines the order in which elements receive focus when navigating with the Tab key.

Utilize ARIA roles and attributes where necessary to enhance the accessibility of complex components. For example, ARIA roles like role="button" can be used to convey the purpose of an element.

Priority: Medium

Impact: Medium

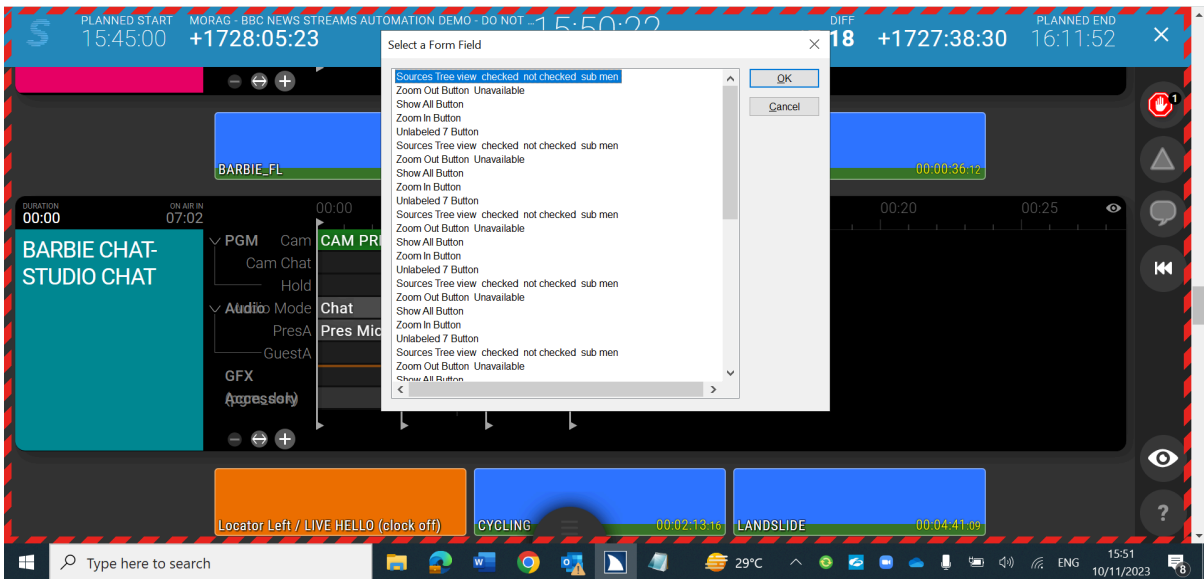
2.1.1 Keyboard (Level A)

Unlabelled form fields

Issue 11: Empty button on the page

Page title: Sofie

Screen Shot:



Unlabelled form fields can pose a significant accessibility challenge, as users may not be able to understand the purpose of each form element, especially when using assistive technologies.

Recommendation

Every form field should have a clear and descriptive label associated with it. Use the <label> element to provide explicit labels for each input.

If a visual label is not feasible, you can use the aria-label attribute to provide a label for screen readers.

Ensure that labels are visible to all users. Avoid hiding labels visually while relying solely on placeholders, as this can cause accessibility issues.

For related form fields, use the <fieldset> and <legend> elements to create a semantic grouping and provide a title for the group.

Priority: High

Impact: High

3.3.2 Labels or Instructions (Level A)

WCAG Test Table

This table indicates where WCAG success criteria have been met, further details of instances where the WCAG guidelines have not been achieved are listed towards the end of this report in the detailed findings section.

Key

Pass – Product passes the WCAG criteria

Fail – Product fails to meet the WCAG criteria

N/A – The WCAG criteria does not apply

WCAG A	Summary	Status	Description
1.1.1 Non-text Content (Level A)	Where appropriate, non-text content that is presented to the user has a text alternative	Failed	Drag to reorder or move out of playlist button is empty or has no value text.
1.2.1 Audio-only and Video-only (Pre-recorded) (Level A)	An alternative is provided for pre-recorded audio/video-only content.	N/A	No Audio and Video pre-recorded are present on the screens.
1.2.2 Captions (Pre-recorded) (Level A)	Captions are provided for all pre-recorded audio content in synchronized media.	N/A	No pre-recorded content hence no captions are needed.
1.2.3 Audio Description or Media Alternative (Pre-recorded) (Level A)	Audio description, or another alternative, of any pre-recorded video content is provided.	N/A	No Audio description, or another alternative, as no pre-recorded video content is provided.
1.3.1 Info and Relationships (Level A)	Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.	Failed	Heading issues, structure issues are identified and mentioned in detail.

1.3.2 Meaningful Sequence (Level A)	When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.	Failed	Focus cannot be tracked hence reading sequence is incorrect at many places.
---	---	---------------	---

1.3.3 Sensory Characteristics (Level A)	<p>Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound.</p>	<p>Failed</p>	<p>Instructions provided reply only upon shape.</p>
1.4.1 Use of Colour (Level A)	<p>Colour is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.</p>	<p>Failed</p>	<p>In many places, only colours are used without context.</p>
1.4.2 Audio Control (Level A)	<p>If any audio on a Web page plays automatically for more than 3 seconds, a mechanism is available to either: pause or stop it; or control its volume independently from the overall system volume.</p>	<p>N/A</p>	<p>No audio is available on the page.</p>

2.1.1 Keyboard (Level A)	All functionality of the content is operable through a keyboard interface.	Failed	Functionality is not available via keyboard.
2.1.2 No Keyboard Trap (Level A)	Focus can be moved both to and from every component using only a keyboard.	Failed	Focus is not visible across all the application and pages.
2.1.4 Character Key Shortcuts (Level A 2.1 only)	If a keyboard shortcut is implemented using only individual characters, then the shortcuts can be turned off, remapped or are active only when a certain component is focused on.	Pass	
2.2.1 Timing Adjustable (Level A)	Users should have enough time to read and interact with content.	Pass	

2.2.2 Pause, Stop, Hide (Level A)	<p>For any moving, blinking scrolling or auto-updating information, there is a mechanism for the user to pause, stop, hide or (in the latter case) control its frequency. Unless the movement or auto-updating is part of an activity where it is essential.</p>	<p>N/A</p>	
2.3.1 Three Flashes or Below Threshold (Level A)	<p>Web pages do not contain anything that flashes more than three times in any one second.</p>	<p>N/A</p>	
2.4.1 Bypass Blocks (Level A)	<p>A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.</p>	<p>N/A</p>	
2.4.2 Page Titled (Level A)	<p>Web pages have titles that describe topic or purpose.</p>	<p>Failed</p>	<p>No unique page titles are given to pages, everywhere it says the title as Sofie.</p>

2.4.3 Focus Order (Level A)	<p>If a webpage can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.</p>	<p>Failed</p>	<p>Focus is not visible hence focus order cannot be identified.</p>
2.4.4 Link Purpose (In Context) (Level A)	<p>The purpose of each link can be determined from the link text alone.</p>	<p>Failed</p>	
2.5.1 Pointer Gestures (Level A 2.1 only)	<p>All functions can be accessed via a single pointer, and with non-path-based commands.</p>	<p>Pass</p>	
2.5.2 Pointer Cancellation (Level A 2.1 only)	<p>If the user performs an action by mistake, then there must be a simple way to abort and/or undo it using a single pointer.</p>	<p>N/A</p>	<p>No such feature to validate on the application.</p>

2.5.3 Label in Name (Level A 2.1 only)	User interface components with labels that include text or images of text, the name contains the text that is presented visually.	N/A	Nothing specific on the application to test against this guideline.
2.5.4 Motion Actuation (Level A 2.1 only)	Device or user motion should not be the only means to complete an action; and the user should be able to turn off any such function.	N/A	Desktop application hence this guideline is not applicable.
3.1.1 Language of Page (Level A)	The default human language of each Web page can be programmatically determined.	Failed	The language of the page is not defined.
3.2.1 On Focus (Level A)	When any component receives focus, it does not initiate a change of context.	Pass	No such component was identified.
3.2.2 On Input (Level A)	Changing the setting of any user interface component does not automatically cause a change of context.	N/A	No interface changes after input.
3.2.6 Consistent Help (Level A)	Make Web pages appear and operate in predictable ways.	Pass	
3.3.1 Error Identification (Level A)	If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.	Failed	Form fields do not have appropriate validations for input.

3.3.2 Labels or Instructions (Level A)	Labels or instructions are provided when content requires user input.	Failed	Label issues are mentioned in the report.
3.3.7 Redundant Entry (Level A)	Help users avoid and correct mistakes - don't ask for the same information twice in the same session.	Pass	
4.1.1 Parsing (Level A)	Where markup languages have been used, elements: have complete start and end tags, are nested according to their specifications, do not contain duplicate attributes; and any IDs are unique, except where the specifications allow these features.	N/A	Was not able to verify this guideline as html parsing tools cannot validate this issue.
4.1.2 Name, Role, Value (Level A)	For all user interface components, the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.	Failed	Failure to these guideline issues is mentioned in the report.

WCAG AA

Status

Description

1.2.4 Captions (Live) (Level AA)	Captions are provided for all live audio content in synchronized media.	N/A	No audio on the application to validate this test.
1.2.5 Audio Description (Pre-recorded) (Level AA)	Audio description is provided for all pre-recorded video content in synchronized media.	N/A	No audio on the application to validate this test.
1.3.4 Orientation (Level AA 2.1 only)	Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.	N/A	As this is a desktop only site this test is not applicable.
1.3.5 Identify Input Purpose (Level AA 2.1 only)	Entry fields should be programmed to accept only the relevant data type. E.g., phone number fields should accept numerical data only.	Passed	
1.4.3 Contrast (Minimum) (Level AA)	Text and images of text have a contrast ratio of at least 4.5:1, unless specified as exempt.	Failed	Issues are highlighted in the issues section, please refer to the report for issue description and details.
1.4.4 Resize text (Level AA)	Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.	Failed	When resized text overlaps for most of the page and components..
1.4.5 Images of Text (Level AA)	If the technologies permit, text is used to convey information rather than images of text; unless the image of text can be visually	N/A	No images of text are present on the page.

	customized to the user's requirements, or the presentation of text is essential to the information being conveyed.		
1.4.10 Reflow (Level AA 2.1 only)	Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions when scrolling: vertically at a width equivalent to 320 CSS pixels; and horizontally at a height equivalent to 256 CSS pixels. Except for if/where the content requires two-dimensional layout for usage or meaning.	Failed	Text overlaps when verified in suggested width and height.
1.4.11 Non-text Contrast (Level AA 2.1 only)	The visual presentation of user Interface Components and graphical Objects have a contrast ratio of at least 3:1 against adjacent colour(s).	N/A	No such non-text content is available for contrast.
1.4.12 Text Spacing (Level AA 2.1 only)	No meaning is lost when character, word, line and/or paragraph spacing is increased.	Passed	
1.4.13 Content on Hover or Focus (Level AA 2.1 only)	If receiving focus triggers additional content to become visible, then the guidelines of 'Dismissible', 'Hoverable' and 'Persistent' should be met.	N/A	No such functionality on this application.

2.4.5 Multiple Ways (Level AA)	More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process.	Failed	More than one way is not available to locate a Web page within a set of Web pages.
2.4.6 Headings and Labels (Level AA)	Headings and labels describe topic or purpose.	Failed	Heading and labels are not descriptive.
2.4.7 Focus Visible (Level AA)	Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.	Failed	Focus is not visible across all the pages and application.
2.4.11 Focus Not Obscured (Level AA)	Provide ways to help users navigate, find content, and determine where they are.	Failed	Invisible focus hence this is failing on the application.
2.5.7 Dragging Movements (Level AA)	Make it easier for users to operate functionality through various inputs beyond keyboard.	Failed	Dragging functionality is not accessible by keyboard and not easy to operate.
2.5.8 Target Size (Level AA)	Ensure targets meet a minimum size or have sufficient spacing around them.	N/A	Not application for the current application
3.1.2 Language of Parts (Level AA)	The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the	N/A	No components or content identified in different language

	immediately surrounding text.		
3.2.3 Consistent Navigation (Level AA)	Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.	Pass	
3.2.4 Consistent Identification (Level AA)	Components that have the same functionality within a set of Web pages are identified consistently.	Pass	
3.3.3 Error Suggestion (Level AA)	If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.	Failed	No suggestions were given to the user.
3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)	For web pages that: cause legal commitments or financial transactions for the user to occur; modify or delete user-controllable data in data storage systems; or submit user test responses, either: submissions are reversible; user-entered data is checked for errors, and they are given an opportunity to correct them; and/or the user is given the	N/A	Not a legal application.

	chance to review, confirm, and correct information before finalizing the submission.		
3.3.8 Accessible Authentication (Level AA)	Don't make people solve, recall, or transcribe something to log in.	N/A	No such feature is present on the application.
4.1.3 Status Messages (Level AA 2.1 only)	Status messages do not automatically divert focus.	N/A	No such messages are present on the application.

Appendix I

JAWS

There are specific screen reader issues present within the pages tested:

- It was not possible to navigate the content of the screen in a meaningful manner.
- All form controls were not properly labelled, both explicitly and implicitly except a few.
- Show all/Zoom in / Zoom out controls had no proper groupings and inaccessible by keyboard when screen reader was on.
- Not all Links were meaningful when read out of context.
- The underlying structure of the page was not programmatically set (i.e. headings).
- Table captions are not programmatically coded hence it was difficult for the screen reader user to understand the table on the pages.

Dragon Naturally Speaking

Dragon Naturally Speaking is a voice recognition application that allows the user to control navigation, and operation and directly input into pages and the operating system via voice alone. It is intended to replace most instances where a user needs to use a mouse and all instances where they are reliant upon a keyboard.

Using Dragon Naturally Speaking, it was possible to:

- All controls were not correctly “flagged” by the Dragon number functionality.

ZoomText

ZoomText is a screen magnifier with built-in screen reader functionality, it allows the user to magnify their screen and move the magnified area so that users with visual impairments can have text enlarged to a size they feel comfortable reading.

Using the ZoomText application:

- It was not possible to navigate the content of the screen in a meaningful manner.
- All magnification functionality worked as expected but text was not that clear at greater than 8x magnification.
- All colour change functionality issues were identified because of the colours and combinations used on the pages.