2023

IDG2012 – Accessible Website

CANDIDATE NUMBER: **10051**

Introduction

Theme

The website created is called “Saving Wildlife" and serves as an informational site where the organization wants to spread awareness about the wildlife in Norway. It is a made-up website but has added translated text contents from a similar real website. I am linking to the respective website at the very end of the paper. In the code, I have commented which texts that are from the website. Nonetheless, the website has an intention to be easily understood from first glance and simply educational.

Analysis of user needs

The prospect of this website is to be accessible and meet users who have disabilities or impairments. However, this user group is involved with multiple aspects of disabilities and there are different needs that require to be met. A user here may have a visual, auditory, cognitive, or physical disability. These are users that rely on assistive technology and use tools to better access digital content.

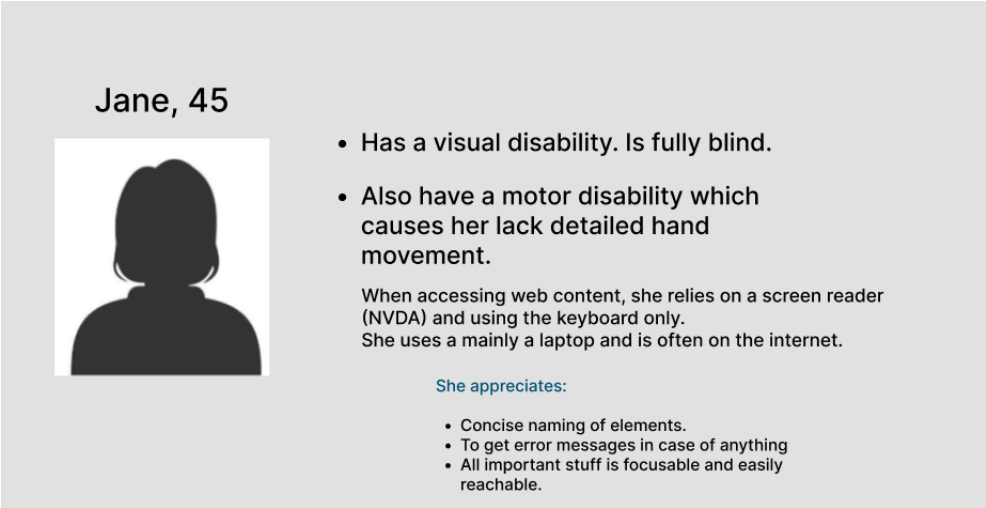
In order to get a greater understanding of these users and their challenges, I did some user research. An informative and trustworthy source I relied on was the W3C WAI website. They provided useful information of their needs and barriers when interacting with content. On their website, I found one page in particular which answered what kind of abilities and barriers a person with a disability can have (Link to page). After some careful reading, I decided to collect and conclude some of the important user needs for each group:

Visual disability	**Auditory disability**	**Cognitive disability**	**Physical disability**
Website is accessible with screen reader. Can	Transcripts and captions are there	Need layout that is not too	Use keyboard only to navigate website

easily navigate website with it. Needs efficient and logical navigation.	in case of audio or video content	complex and intuitive.	
Elements on the page are resizeable, like text, images.	Media player buttons to easily control the video/audio contents	Space between paragraph. Not too much text at once	Larger click/touch areas
Alt texts and and clear descriptions of contexts	Caption customization	Captions to images	Be able to skip tedious content
High contrast between colors and between text and backgrounds.	Description of background sounds in videos/audios.	Clear labels on forms etc	Focusable content should be outlined clearly

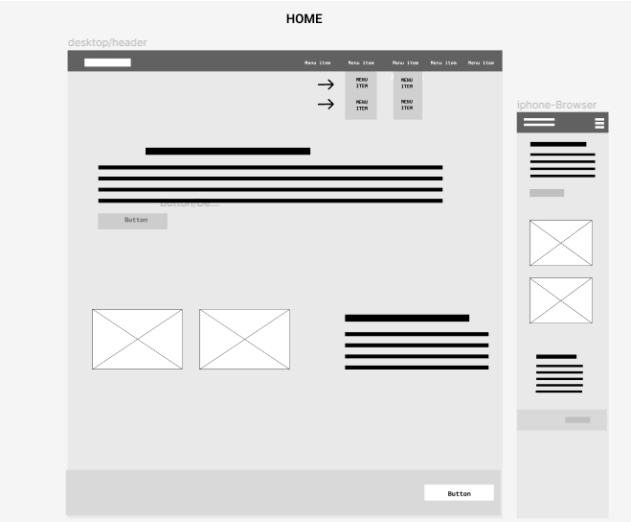
This approach allows me to get an overview of what should be taken considerations towards when designing and building the website.

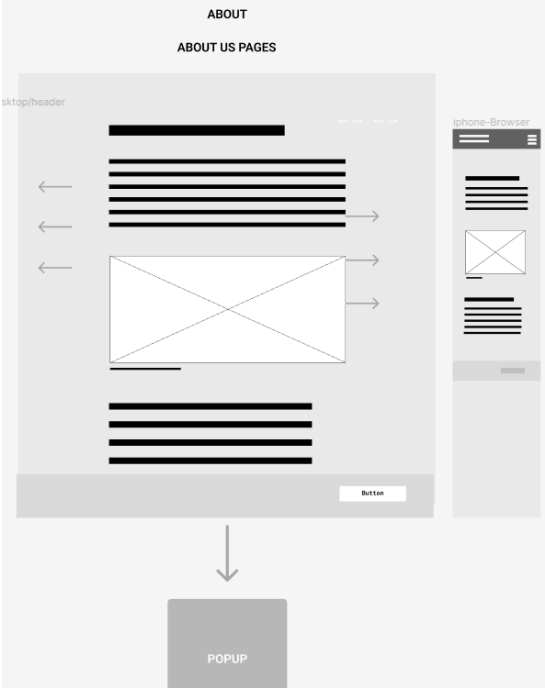
User persona, created in figma

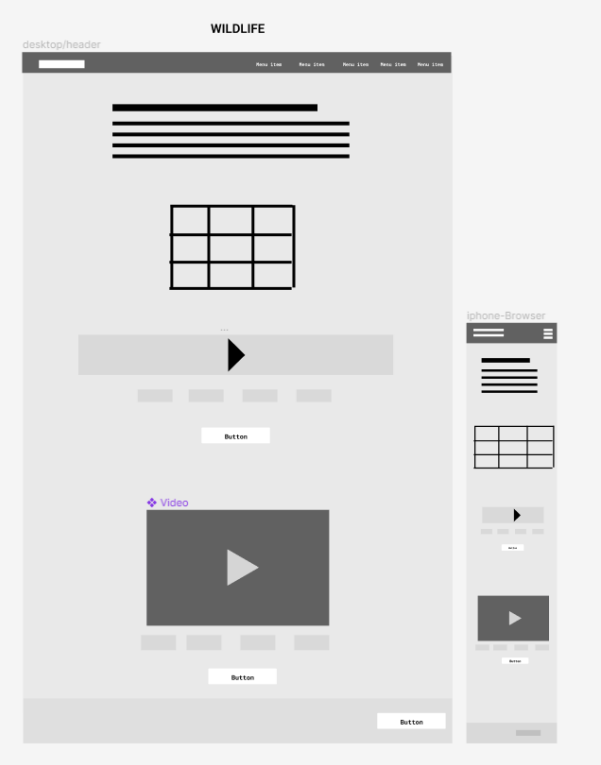


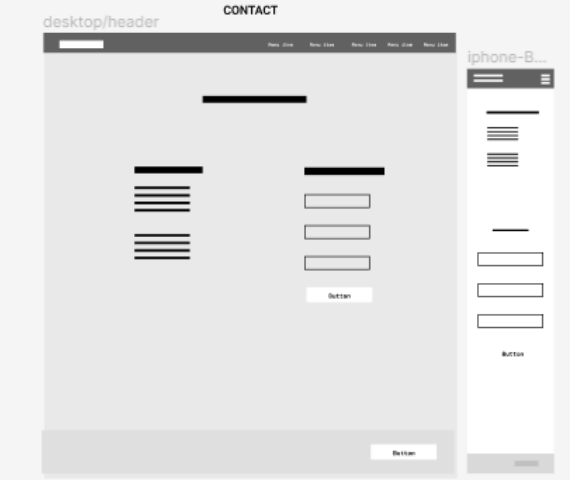
Design and usability

Lofi prototype (wireframe)









Design

Most of the elements seen in the wireframes I did implement. Some things were also added to these during the development. For example, the media buttons were added, since I figured out that I would have to create those media controls in order to make the contents accessible. Other than that, I just made sure to not make anything too complicated.

In the code, I chose to use SASS for styling. This allowed me to create separations between pages and certain elements. It is also more efficient in my opinion, and it certainly contributed to more consistency in the project.

Usability

One of my bigger priorities was to implement responsiveness of the website in a seamless way. I primarily coded with “mobile-first” in mind, since mobile users make up practically “half” of the overall users that are entering the websites. It is an important part to implement correctly, because otherwise the users might be thrown off when going onto the website for the first time. A responsive website is also quite important in terms of accessibility.

The other big priority was navigation. A website without a good functioning navigation will not be favorable to users. For the navigation, I ensured that it was responsive. I implemented a hamburger menu with good visibility and a larger click area. When clicked, a menu comes in play, and the menu-items are now shown in a more spaced column. The hamburger has an animation which indicates whether it is opened or closed too. I also added a “back to top” button on the footer of every page, so that every user can easily make their way back to the main navigation when reaching the bottom.

Lastly, I made sure that all the pages had a structural flow where the different sections make sense. Spacing between elements is an important factor here, as well as logical use of headings.

I made sure to keep testing the website for flaws by resizing the viewport, checking navigation patterns, and perceiving the content as if I was a user myself. By doing so, I would make changes based on the observations made. To mention an example: I made new a media query for “orientation: landscape" for touch devices and made changes in there so that the main navigation is responsive for smaller screens that are in landscape mode.

Accessibility

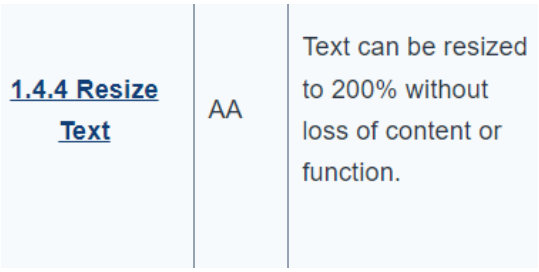
What makes this website accessible?

Colors and font

Let's start with the color palette and font choice. I chose to make the palette myself after some inspirations on websites like “coolors.co". I wanted to make something hamonic, serious and yet allows for high contrast. All the colors do combine well, and the orange is meant primarily as an accent to contrast with the rest. Another intention is to have “a dark" and “a light" to substitute the classic black and white which can be a straining to look at. Using this palette has not caused any issues in regards to accessibility. According to the WAVE tool and WebAim's contrast checker they pass on at least AA level, including focus indicators. I also added more space between focus indicators and the content using “outline-offset", which helps to add more contrast and clarity.

As for the font, I chose to go with “Arial" since it is regarded as one of the best accessible fonts. It is sans-serif, which is has good readability. It is ADA compliant and has good availability. In the code I also made sure to use the REM unit when changing

the font-sizes, since rem makes the text size resizeable and accessible for the user. If they use a software to alter the text size, then the content will be readable.



1.4.4 Resize Text	AA	Text can be resized to 200% without loss of content or function.

https://www.digitala11y.com/wcag-checklist/

Skip to main and show/hide content toggle

To have a skip to main on every page allows the user to not go through the navigation each time. I have implemented this on all the pages and is available as soon as TAB is hit once after entering the page. Another thing I added was a toggle button for each of the transcripts on the “wildlife” page, and these show when the button is clicked and hidden if clicked again. Both the skip to main and toggles are concise for screen readers too.

Video content

To make the video content to be accessible, I had to implement several things. First is captions that come in the VTT format. I first converted the video to a VTT file, and then I made the necessary changes to make the context correct and corresponding to the actual video. I added closed captions too both in video and transcript to include users with an auditory disability. Below the video, I added the transcript. I wanted it to be easy to read, so I spaced the lines according to the dialog in the video. It makes the reading experience more pleasant and less straining for certain users, for example those with a cognitive disability or even visual. Next, I found out that the html5 video player was not accessible to screen readers and I could not add any labels. I therefore had to implement accessible media controls for these users to navigate. I made these buttons: play/pause, forward 10 sec, backward 10 sec, toggle mute. I got some

inspiration from a source, which I list under “references”. From there I learnt how to make these controls functional.

Audio content

When adding the audio element, I saw that its default controls also were not accessible. I therefore found a way to add those same media controls from the video player for the audio too. Again, a source helped me out and is displayed in the code and under “references”. Then, I added a transcript for the audio player for users who rely on that.

Popup

The popup is accessible when it appears on the screen. When it is displayed, it will remember the last item you focused on before it appeared, so when you close it, you will return to that point. When the popup is opened, the focus is placed on the first focusable item inside of it. Then, a focus trap is created so that tab and shift+tab can be used to go between the elements. To close the popup, one clicks on the “close” button, or press esc button. The popup will then be hidden. In the HTML, the popup was given aria attributes to inform screen readers about the content. Visually, the popup also was given a blurred background to add more contrast and visibility.

Table

The table is navigable using tab and shift+tab, and here one can go between each cell. I have used column headers and row headers and used the “scope” attribute. This will better tell screen readers about which column and row corresponds to each cell. However, I noticed that different screen readers had different approaches to how they would say out the information about each cell. I could alternatively add an aria-label here, but I thought maybe that would overdo it and increase the chance of incorrectness.

Form

The form also has focusable input fields. To be more concise for SRs, I added aria-labelledby for each field explaining the content. I also added aria-required to communicate that each input is required. When the submit button is clicked and input is not valid, then the focus will shift to that field. Then an error message will be displayed. I added an aria-describedby in the js scipt too, in order for those errors to be communicated to the screen readers. The errors will go away if the input is valid but relies on the submit button to be pressed again. Overall, the form is quite accessible.

Design with accessibility in mind

In my website, I have made sure to pay attention to spacing between the design elements, paragraphs, headers etc. The buttons and links have decent padding, resulting in more spacious touch areas, which makes it easier for those who need that. For the bigger images displayed, I have also included figcaptions to describe context, which adds clarity for perhaps someone with a cognitive disability.

Keyboard navigation on website

The website with all its pages is navigable using the keyboard only. The interactive elements are focusable by default, but to add focus to also non-interactive elements, I have added “tabindex=0” to elements across the site. This way, tabbing and shift+tabbing between different sections is possible, and then also easily communicated to the screen reader.

Accessibility testing

To test accessibility, I have used these tools:

- Screen readers (google extension and NVDA)
- WebAim contrast checker
- WAVE accessibility checker browser extension
- Google lighthouse extension

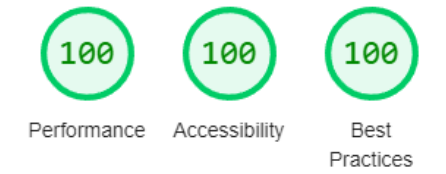
Compliance with lighthouse of all pages:

http://127.0.0.1:5502/index.html

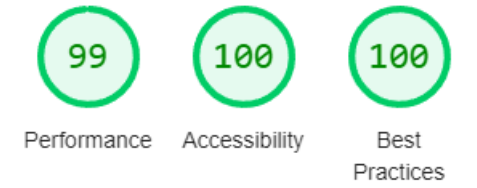
http://127.0.0.1:5502/who-we-are.html

http://127.0.0.1:5502/our-work.html

http://127.0.0.1:5502/make-change.html



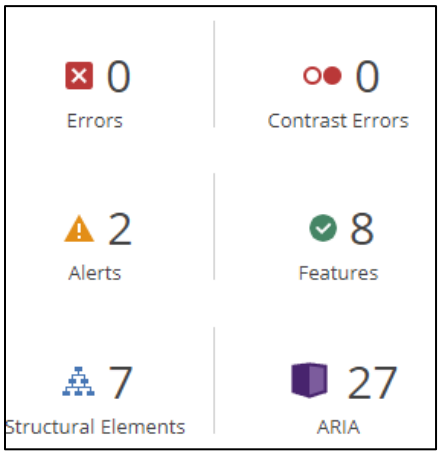
http://127.0.0.1:5502/wildlife.html

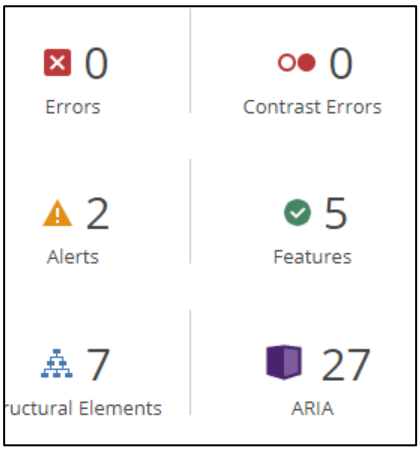


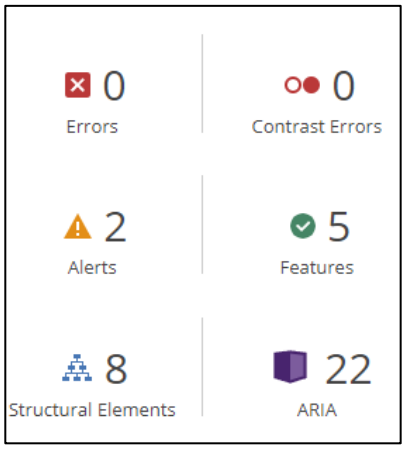
http://127.0.0.1:5502/contact.html

WAVE tool:

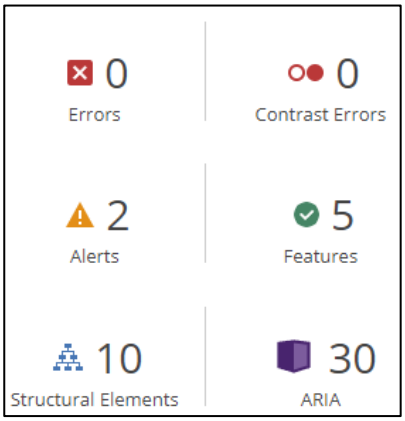
Home, who are we, our work:

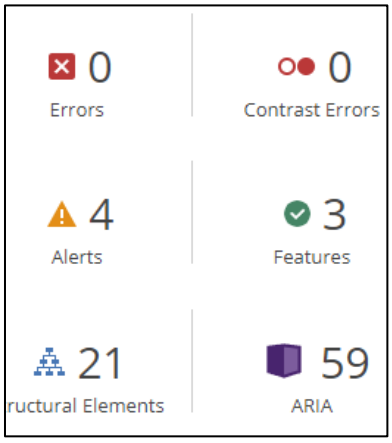


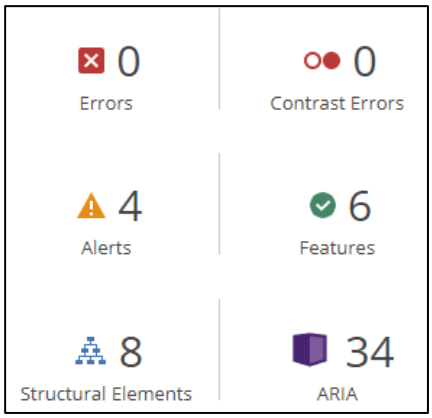




Make a change, wildlife, contact:







Screen readers:

I tested the website manually with two screen readers, primarily with NVDA since it is more used than the google screen reader. I used it consistently during the development.

WCAG 2.1 checkpoints

Criterion	Level	Summary	How the criteria is met
1.1.1 Non-Text Content	A	Provide text alternatives for non-text content that serves the same purpose.	Has been provided. Alt tag on images for example.
1.2.1 Audio-Only and Video-Only	A	Provide an alternative to video-only and audio-only content.	Yes, caption + transcript for video and transcript for audio.
1.2.2 Captions (Prerecorded)	A	Provide captions for videos with audio.	Has been added
1.2.3 Audio Description or Media Alternative	A	Provide audio description or text transcript for videos with sound.	Has been added

1.2.5 Audio Description (Pre-Recorded)	AA	Provide audio descriptions for pre-recorded videos.	Has been provided. Alt tag on images for example.
1.3.1 Info and Relationships	A	Content, structure and relationships can be programmatically determined.	Content and structure has semantic markup. Table, form etc are semantic and form has labels and inputs have names.

1.3.2 Meaningful Sequence	A	Present content in a meaningful order.	There is a meaningful sequence
1.3.3 Sensory Characteristics	A	Instructions don't rely solely on sensory characteristics.	Criteria is met
1.3.4 Orientation	AA	Your website adapts to portrait and landscape views.	Supports landscape view.
1.3.5 Identify Input Purpose	AA	The purpose of input fields must be programmatically determinable.	Has autocomplete on input fields

1.4.1 Use of Color	A	Don't use presentation that relies solely on colour.	Criteria is met
1.4.2 Audio Control	A	Don't play audio automatically.	Media does not autoplay
1.4.3 Contrast Minimum	AA	Contrast ratio between text and background is at least 4.5:1.	Contrasts are all accessible

1.4.4 Resize Text	AA	Text can be resized to 200% without loss of content or function.	Criteria is met
1.4.5 Images of Text	AA	Don't use images of text.	Criteria is met
1.4.10 Reflow	AA	Content retains meaning and function without scrolling in two dimensions.	Criteria is met

1.4.11 Non-Text Contrast	AA	The contrast between user interface components, graphics and adjacent colours is at least 3:1.	Yes, f.ex the focus indicator and background is always above 3:1
1.4.12 Text Spacing	AA	Content and function retain meaning when users change elements of text spacing.	Criteria is met. I have responsive layout and mostly use relative units.
1.4.13 Content on Hover or Focus	AA	When hover or focus triggers content to appear, it is dismissible, hoverable and persistent.	Criteria met

2.1.1 Keyboard	A	All functionality is accessible by keyboard with no specific timings.	It is. Access keys have been removed.
2.1.2 No keyboard Trap	A	Users can navigate to and from all content using a keyboard.	No traps
2.4.1 Bypass Blocks	A	Provide a way for users to skip repeated blocks of content.	Skip to main on all pages

2.4.2 Page Titled	A	Use helpful and clear page titles.	Criteria met
2.4.3 Focus Order	A	Components receive focus in a logical sequence.	Focus order is logical
2.4.4 Link Purpose (In Context)	A	Every link's purpose is clear from its text or context.	They are clear

2.4.6 Headings and Labels	AA	Headings and labels describe topic or purpose.	Criteria met
2.4.7 Focus Visible	AA	Keyboard focus is visible when used.	Works as should
2.5.2 Pointer Cancellation	A	Functions don't complete on the down-click of a pointer.	Criteria met

2.5.3 Label in Name	A	Where a component has a text label, the name of the component also contains the text displayed.	Criteria met

3.1.1 Language of Page	A	Each webpage has a default human language assigned.	Criteria is met
3.2.1 On Focus	A	Elements do not change when they receive focus.	Criteria is met

3.2.3 Consistent Navigation	AA	Position menus and standard controls consistently.	Is consistent	3.3.3 Error Suggestion	AA	Suggest corrections when users make mistakes.	Has a sufficient description. Focus is moved to input fields too
3.3.1 Error Identification	A	Identify and describe input errors for users.	Is provided in the form. Using aria-describedby, and is displayed	4.1.1 Parsing	A	No major code errors	fulfilled
3.3.2 Labels or Instruction	A	Provide labels or instructions for user input.	Criteria met	4.1.2 Name, Role, Value	A	The name and role of user components can be understood by technology.	Criteria met. Mostly using semantic html
				4.1.3 Status Messages	AA	Make sure that all messages indicating success or errors are read out by a screen reader.	Errors are read out to screen readers.

Summary

In this paper, I have described how this project has been made with both design, but mostly accessibility in mind. I have tested the website and found out that it is compliant through technical and manual testing and going through checklist. I think the media controller and the popup dialog are nice highlights from the project. Overall, I am mostly satisfied with the results.

References:

- *WebAIM: Contrast Checker*. Available at: https://webaim.org/resources/contrastchecker/.

- w3c_wai (2017). *Diverse Abilities and Barriers*. [online] Web Accessibility Initiative (WAI). Available at: https://www.w3.org/WAI/people-use-web/abilities-barriers/.

- Making design elements accessible - Accessibility Resources at UNCG (2023). Available at: https://accessibility.uncg.edu/make-content-accessible/design-elements/.

- A guide to designing accessible, WCAG-conformant focus indicators (2021). Available at: https://www.sarasoueidan.com/blog/focus-indicators/.

- Peri, R.S. (2023) WCAG 2.1 and 2.2 AA Checklist: A Guide to Web Accessibility • DigitalA11Y. Available at: https://www.digitala11y.com/wcag-checklist/.
- W3Schools: Accessibility autocomplete. Available at: https://www.w3schools.com/accessibility/accessibility_autocomplete.php.

Video/audio player inspiration/assistance and source

- GeeksforGeeks (2023) Custom Video Player using HTML, CSS, and JavaScript. Available at: https://www.geeksforgeeks.org/custom-video-player-using-html-css-and-javascript/.

- HTML + JavaScript custom player for multiple videos on a website page (2022). Available at: https://stackoverflow.com/questions/71743629/html-javascript-custom-player-for-multiple-videos-on-a-website-page.

- Video and Audio APIs - Learn web development | MDN (2023). Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Video_and_audio_APIs.

- Al Jazeera English (2016) Norway: Cull debate rages over killing wolves. Available at: https://www.youtube.com/watch?v=fERSFh615cc.

- Nat Geo WILD (2019) Arctic Fox in Norway | Wild Nordic. Available at: https://www.youtube.com/watch?v=aSQiLm54gUs.

Hamburger menu inspiration:

- Web Dev Tutorials (2021) JavaScript - How to Create a Responsive Hamburger Menu with HTML, CSS, & JavaScript. Available at: https://www.youtube.com/watch?v=fllItyHiDm7E.

Popup inspiration:

- Creating An Accessible Modal Dialog | bitsofcode (2016). Available at: https://bitsofco.de/accessible-modal-dialog/.

Translated text from these websites (used google translate):

- NOAH - Villedyr (2023). Available at: https://www.villedyr.no/.
- NOAH - for dyrs rettigheter (2023). Available at: https://www.dyrsrettigheter.no/.

Images on website are from https://unsplash.com/