

Assignment 4: Report

For assignment 4, I have delivered:

- This report: the use of AI, log interactions and assumptions/comments
- Link to the homepage: <https://viktorialangaas.github.io/assignment4/index.html>
- Link to sub-page: <https://viktorialangaas.github.io/assignment4/tickets.html>
- Code in GitHub repository: <https://github.com/ViktoriaLangaas/assignment4>

Assumptions/Additional comments

- I assume that the website does not have to be responsive, since its not specified in the assignment description. The website designed for laptop use, where a keyboard navigation and screen reader might be used.
- In the sub-page I wanted to make input field for event and ticket as a select, where the user can select one out of three options. However, since the image of subpage in the assignment description shows the form with six input fields, I decided to use simple input field in the form. It would be more natural if select with options was used.

The use of AI

As a part of this assignment, I used AI to help me with certain parts. I used AI to create and reflect the website, as well as understanding the user requirements. I also used AI to briefly test my website, as a “blind person” and as a “deaf person”.

I gave AI all the assignment requirements, what each webpage should contain and a theme to make the webpages about.

After getting the code to create the website from AI, this is some of the changes I did:

- ✓ Changed the CSS styling to match the assignment images of the pages
- ✓ Added images, with relevant alternative text
- ✓ Changed the navigation elements to be similar in each page
- ✓ Added the error message, disappears when the input is valid

I used the WAVE accessibility evaluation tool to test both websites, where both webpages passed the test and had no errors.

When I was mostly done with the webpage, I used AI to simulate how a blind person and a deaf person would navigate the website. The task was to find the price of a VIP ticket to a

basketball event. Both simulations were successful and managed to find the right price. However, the blind user used more time, which is natural because she/him has more steps to go through and needs to “wait” for the screen reader. For the deaf user, it was straightforward, as they were no auditory elements involved.

With the help of AI, I understood more what a website must have to be accessible, especially for those with a visual impairment. Its important to have a proper structure to your website and use ARIA roles. The different elements of the website must be accessible via keyboard navigation, otherwise users could get stuck or leave the website. Elements like links and buttons also needs clear labels, for a blind person to understand what action the element perform. Additionally, blind users must get error messages that the screen reader can read, otherwise the person will not receive the error message.

Log interactions with AI

Me: I have an assignment, where I need your help developing a website that is fully accessible via keyboard (navigation) and screen reader. I will provide you with requirements, and what content each website must have. Based on those requirements and theme given, you will develop one homepage and one sub-page.

ChatGPT (4.0): I'd be happy to help with that! Building a website that's fully accessible to keyboard navigation and screen readers is a great project, and it'll involve applying various web accessibility principles. Please go ahead and share the requirements, content, and any specific themes or style guides you have in mind. From there, I can help you design the homepage and sub-page layout and guide you on how to implement accessible elements such as semantic HTML, ARIA roles, focus management, and screen reader-friendly navigation.

Me: For the assignment, the website must be fully accessible via keyboard navigation and screen reader. The website must be in English, and the page title will be "Welcome to my virtual home" for both pages. The theme for the pages is sports in general. The homepage shows different images for sports and has sports news as one of the sections. The subpage is a ticket page, where you can book tickets for sports events through the form. The structure of the pages: Homepage --> link (skip to content link), banner (image), navigation with 6 elements, image (big), three sections (each containing: image, h1, and h2). Sub-page --> link (skip to content link), banner (image), navigation, h1, h2, table (4x4) containing ticket prices, h1, h2, form (with 6 labels and 6 inputs, and a button). For the form, take this into consideration: Follow logical navigation order in the form. Provide an error message in case an input field is missing. Make sure that the screen reader reads the message. For both pages the link, banner and navigation is similar - only the main content is different for each page. Additionally, take into considerations that you are developing webpages for a visual impairment. Therefore, use proper structure and proper semantic elements. Place the link (skip to main content), at the top left of the page. Provide clear alternative text for each image, do not use "image of..". Use ARIA and make sure to have color contrast. Provide me with each html file for each webpage, a css file for styling and a short explanation. Only use HTML and CSS to complete the assignment. Do not use JavaScript either in HTML or as an own file.

ChatGPT (4.0): Here's the solution based on the requirements provided:

Project Structure

1. HTML Files:

- `index.html` (Homepage)
- `tickets.html` (Sub-page)

2. CSS File:

- `styles.css`

Each page has a `skip to content` link, accessible navigation, screen-reader-friendly elements, and error message guidance (although error messages will only be visually displayed without JavaScript). Semantic HTML, ARIA attributes, and clear alternative text are used for accessibility, and color contrasts are considered in CSS styling.

1. `index.html` (Homepage)

```

` `` `html

```

```

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1.0"> <title>Welcome to my virtual home</title> <link
rel="stylesheet" href="styles.css"> </head> <body> <!-- Skip to Content Link --> <a href="#main-content"
class="skip-link">Skip to main content</a> <!-- Banner --> <header class="banner" role="banner" aria-
label="Sports theme banner"> </header> <!-- Navigation --> <nav aria-label="Main navigation" class="nav">
<ul> <li><a href="#news">News</a></li> <li><a href="tickets.html">Tickets</a></li> <li><a
href="#section1">Football</a></li> <li><a href="#section2">Basketball</a></li> <li><a
href="#section3">Tennis</a></li> <li><a href="#contact">Contact</a></li></ul></nav> <!-- Main Content -->
<main id="main-content"> <!-- Hero Image --> <section>  </section> <!-- Sections for Various Sports --> <section id="section1"
class="sport-section"> 
<h1>Football</h1> <h2>Latest Updates</h2> </section> <section id="section2" class="sport-section">  <h1>Basketball</h1>
<h2>News and Highlights</h2> </section> <section id="section3" class="sport-section">  <h1>Tennis</h1> <h2>Upcoming
Tournaments</h2> </section></main></body></html>

```

```

` `` `

```

2. `tickets.html` (Ticket Booking Sub-page)

```

` `` `html

```

```

<!DOCTYPE html><html lang="en"><head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1.0"> <title>Welcome to my virtual home</title> <link
rel="stylesheet" href="styles.css"></head><body> <!-- Skip to Content Link --> <a href="#main-content"
class="skip-link">Skip to main content</a> <!-- Banner --> <header class="banner" role="banner" aria-
label="Sports theme banner">  </header> <!-- Navigation --> <nav aria-label="Main navigation" class="nav">
<ul> <li><a href="index.html">Home</a></li> <li><a href="tickets.html">Tickets</a></li> <li><a
href="#prices">Prices</a></li> <li><a href="#booking">Book Tickets</a></li> <li><a

```

```

href="#contact">Contact</a></li> </ul> </nav> <!-- Main Content --> <main id="main-content"> <!-- Ticket
Prices Table --> <section id="prices"> <h1>Ticket Pricing</h1> <h2>Available Ticket Options</h2> <table>
<thead> <tr> <th>Event</th> <th>Standard</th> <th>VIP</th> <th>Student</th> </tr> </thead> <tbody>
<tr> <td>Football</td> <td>$50</td> <td>$150</td> <td>$40</td> </tr> <tr> <td>Basketball</td>
<td>$40</td> <td>$120</td> <td>$30</td> </tr> <tr> <td>Tennis</td> <td>$60</td> <td>$180</td>
<td>$50</td> </tr> <tr> <td>Swimming</td> <td>$30</td> <td>$90</td> <td>$20</td> </tr> </tbody>
</table> </section> <!-- Ticket Booking Form --> <section id="booking"> <h1>Book Your Tickets</h1>
<h2>Fill out the form to secure your seat</h2> <form aria-label="Ticket booking form"> <label
for="name">Name:</label> <input type="text" id="name" aria-required="true"> <span class="error" aria-
live="assertive" id="nameError">Please enter your name</span> <label for="email">Email:</label>
<input type="email" id="email" aria-required="true"> <span class="error" aria-live="assertive"
id="emailError">Please enter your email</span> <label for="phone">Phone:</label> <input type="tel"
id="phone" aria-required="true"> <span class="error" aria-live="assertive" id="phoneError">Please enter your
phone number</span> <label for="event">Event:</label> <input type="text" id="event" aria-required="true">
<span class="error" aria-live="assertive" id="eventError">Please enter the event name</span> <label
for="ticketType">Ticket Type:</label> <input type="text" id="ticketType" aria-required="true"> <span
class="error" aria-live="assertive" id="ticketTypeError">Please select a ticket type</span> <label
for="quantity">Quantity:</label> <input type="number" id="quantity" aria-required="true"> <span
class="error" aria-live="assertive" id="quantityError">Please specify the quantity</span> <button
type="submit">Submit</button> </form> </section> </main> </body> </html>

```

...

3. `styles.css` (Styling for both pages)

```

` `` `css

```

```

/* General Styling */

```

```

body {font-family: Arial, sans-serif;color: #333;background-color: #f4f4f4;} /* Skip Link */.skip-link
{position:absolute; top: 0; left: 0; background: #005f5f; color: #fff; padding: 10px;text-decoration: none;}.skip-
link:focus {top: 10px;} /* Banner */.banner {text-align: center; background-color: #005f5f; color: #fff; padding:
20px 0;} /* Navigation */.nav ul {list-style: none; padding: 0; text-align: center;}.nav ul li {display: inline;
margin: 0 15px;}.nav ul li a { color: #333; text-decoration: none;} /* Sport Sections */.sport-section {padding:
20px;margin: 20px;background-color: #fff;border: 1px solid #ddd;}.hero-img, .banner-img, .sport-img { max-
width: 100%; height: auto;} /* Table */table { width: 100%; border-collapse: collapse;}table th, table td {
padding: 10px; text-align: left; border: 1px solid #ddd;} /* Form */form { display: flex; flex-direction: column;}
label { margin-top: 10px;}input {padding: 8px; margin-top: 5px;border: 1px solid #ddd;}.error {color: red; font-
size: 0.9em; display: none;}

```

...

Explanation of Key Accessibility Features

1. **Semantic HTML:** HTML5 tags (e.g., `header`, `nav`, `main`, `section`, `table`, `form`) provide a meaningful document structure.
2. **ARIA Attributes:** Used for better screen reader support (e.g., `aria-label`, `aria-live`).
3. **Skip Link:** Enables users to bypass navigation and go directly to the main content.
4. **Keyboard Navigation:** All elements can be navigated via keyboard using semantic tags and skip links.

- 5. Form Accessibility:** Each input has a corresponding label, `aria-required` attributes, and error spans with `aria-live` for screen reader feedback.

Let me know if you need further customization or clarification on any specific part!

Me: When it says the website has to be fully accessible, what does that involve? If you were blind and trying to access and use the website, what is the main frustrations you would meet? Explain shortly.

ChatGPT (4.0): When a website needs to be *fully accessible*, especially for visually impaired users who rely on screen readers, this involves several key considerations to ensure a smooth, frustration-free experience:

1. Logical Structure and Navigation:

- **Clear Structure:** The website should have a logical, consistent structure with semantic HTML tags (like `header`, `nav`, `main`, `footer`) that help screen readers understand the layout. Without these, screen readers may read content out of order, making it hard to follow.
- **Keyboard Navigation:** Every part of the site should be accessible via keyboard alone, including menus, buttons, forms, and links. Many blind users navigate primarily by keyboard, so inaccessible elements can prevent them from using parts of the site.

2. Descriptive Content and Alternative Text:

- **Image Descriptions:** Every image must have clear and relevant alternative text. Missing or vague descriptions (like “image” or “click here”) can make content meaningless, since images are skipped or poorly described.
- **Accessible Links and Buttons:** Screen readers rely on descriptive text to navigate links and buttons. Generic labels (“Click here”) create confusion and lead to difficulty finding specific information.

3. Screen Reader Feedback and Form Validation:

- **ARIA Attributes:** These enhance the experience by describing elements (like form controls) and providing role-specific context. Without ARIA, screen readers may misinterpret the elements function.
- **Error Messages:** Input errors (like incomplete forms) need immediate feedback read by screen readers. If error messages aren’t screen reader-accessible, users may be unaware of what went wrong and why they can’t proceed.

Main Frustrations for a Blind User:

1. **Inability to Navigate:** If elements aren’t keyboard-accessible, a blind user may get “stuck” and unable to interact with or leave certain areas of the site.
2. **Lack of Context:** Missing structure and ARIA roles can make it hard to understand the content’s purpose (e.g., distinguishing a menu from a main section).
3. **Ambiguous Labels:** Links and buttons that lack clear labels are frustrating, as it’s unclear what action they’ll perform.
4. **Hidden Error Feedback:** If form errors aren’t screen reader-friendly, the user may repeatedly submit a form without knowing why it’s not going through.

Ensuring a website is fully accessible means removing these barriers, making the site logical, and ensuring all elements are clear and usable for those navigating in non-visual ways.

Me: you are a blind person which will navigate a website to purchase a ticket. You will navigate with the help of screen reader and keyboard navigation. Simulate the step-by-step process, start on the home page. Your task is to find the price for VIP ticket to the basketball event. Link to homepage:

<https://viktorialangaas.github.io/assignment4/index.html> Link to sub-page:

<https://viktorialangaas.github.io/assignment4/tickets.html>

ChatGPT (4.0):

Searched in 1 website:

Starting on the homepage of the provided website, I simulate screen reader navigation to locate and proceed to the "Tickets" page:

1. **Start:** On the homepage, a screen reader announces the main navigation links, which include sections like "Home," "Tickets," and specific sports categories.
2. **Locate 'Tickets':** I navigate to the "Tickets" link using the Tab key and activate it with Enter to proceed to the tickets page.

Now, let's open the tickets page and continue.

Searched in 1 website:

On the tickets page, the screen reader identifies a table listing ticket prices for different events. For a **basketball VIP ticket**, the listed price is **\$120**.

To proceed with booking, a form is available where I would enter details like name, email, and ticket type, navigating fields via the Tab key.

Me: Simulate the same test one more time, with the same links and task. But you are now a deaf person, how would you navigate the website?

ChatGPT (4.0): As a deaf user, the navigation process remains largely the same but without reliance on audio cues. Using visual and text-based elements alone:

1. **Homepage Navigation:** I would use visual navigation aids to locate the "Tickets" link and click it to reach the appropriate page.
2. **Tickets Page:** On this page, I'd visually scan for the section with ticket types and prices. Here, the price for a basketball VIP ticket is clearly listed as **\$120**.
3. **Booking Process:** I would then proceed to fill out any booking form fields as needed.