

# Can That Be Accessible?



## Documentation – How did you make this accessible?

William – 23 June 2026

### Scope

My portfolio is the [\*Can That Be Accessible?\*](#) website.

This is solo exploration of whether certain things can be made accessible. I fundamentally believe that when you plan properly, you plan for accessibility.

In *Can a Medieval Fayre be accessible?*, I explore whether a rustic, loud, and crowded event can be immersive while being accessible. In *Should accessibility ever be de-linked from disability?* I explore an underlying tension in justifications for accessibility. This blog post is based on a real conversation I had at work about how we frame accessibility to a wider audience.

My four documents are:

- [\*Can a Medieval Fayre be accessible?\*](#) (Multimedia)
- [\*Can a Medieval Fayre be accessible?\* Word transcript](#) (DOCX, 25 KB)
- [\*Should accessibility ever be de-linked from disability?\*](#) (HTML)
- [\*Should accessibility ever be de-linked from disability?\* PDF version](#) (PDF, 214 KB)

### Introduction

A good accessibility advocate should follow what they preach.

While I'm no preacher, I've gone beyond simply choosing a template—I've actively built accessibility into the structure of this website and its supporting materials.

## 1. Website

### Hugo and Zen

I used [Hugo](#), an open-source static website generator, to create this website. It's command line-based software. I chose it because earlier in 2026, I learnt how to use Hugo so that I could create websites unfettered by subscription fees. Static websites have identical content for every user, which was not an issue for being the website for *Can That Be Accessible?*

Hugo offers a wide range of community themes that define layout and include CSS, reducing the need to build everything from scratch. With basic HTML and CSS, these themes can be customised further.

## Documentation – How did you make this accessible?

I went with [Zen by frjo](#) (GPL-2.0), which prioritises accessibility. As frjo notes:

[Zen] uses HTML5 with a modern CSS grid and flex layout. Care has been taken to produce semantic and accessible code.

Unlike other Hugo themes, Zen was built with accessibility in mind. I found Zen through [a forum post made by another accessibility-interested Hugo user](#). As suggested in that thread, I ran the Zen demo site through [WAVE web accessibility evaluation tool](#). This returned a high score and no critical issues for manual review.

### Making an accessible theme more accessible

Despite its strengths, Zen does not have a “Skip to main content” link—one that benefits keyboard and screen reader users. I added a skip link in the base HTML file (baseof.html).

```
<a class="skip-link" href="#main-content">
  Skip to main content
</a>
```

I ensured all pages had the matching target this by updating taxonomy.html:

```
<main id="main-content" class="main layout__main" tabindex="-1">
<header>
<h1 class="title">{{ .Title }}</h1>
</header>

{{ .Content }}

{{ range .Pages.ByTitle -}}
{{ .Render "taxonomy-summary" }}
{{ end -}}
</main>
```

This created a consistent landmark for the skip link to land on. While the indentation may not be perfect, the “Skip to main content” link is functional.

### Assessment

I tested the [Should accessibility ever be de-linked from disability?](#) page using WebAIM WAVE, which returned an AIM Score of 9.9/10 with zero errors and four alerts.

The 0.1 reduction was due to:

- Two redundant links (the logo and website title linking to the home page)
- Links to DOCX and PDF alternatives which WAVE could not fully evaluate.

These were intentional trade-offs: I provided alternative formats to improve user choice, even if WAVE could not verify them.

## 2. YouTube

The YouTube video, *[Can a Medieval Fayre be accessible?](#)*, was produced from scratch, combining video editing and accessibility design.

YouTube supports audio description tracks for some creators, however I was not able to use the feature. According to the [YouTube Help page about multi-language features](#), this was only available to a “subset of creators with access to Advanced features as we are expanding access gradually.” I therefore uploaded a separate audio-described version.

Recording the audio description required matching the length of the original video precisely. I could only describe during pauses in my original narration. I chose to read Dr Stephen Stone’s quote verbatim rather than using my identity-first re-phrasing so that I was accurate to what was on screen. Similarly, I continued reading the credits even after it disappeared from the video.

Additional accessibility features include:

- Closed captioning, including description of music
- On-screen text being held for at least one second
- Clean, clear audio using a lapel microphone with light post-processing
- Controlled background noise with minimal ambient elements
- High-contrast text over background elements
- Alternative formats (HTML, PDF, and DOCX transcripts and audio-described video)
- Keyboard-accessible embedded YouTube player.

Stock footage was primarily sourced from Pexels, with attention to diversity and inclusion, including people without visible disabilities.

## 3. Website Documents (PDF and DOCX)

### Approach

The PDF and DOCX files were designed as alternative formats rather than primary content. The HTML versions were prioritised for accessibility, but the downloadable formats were still carefully structured.

The Word documents made use of:

- Semantic structuring with only one H1-level heading per document
- Proper use of heading styles
- Document title in the metadata
- Alt text for images that convey meaning.

These documents therefore passed the basic Microsoft accessibility checker.

Well-structured Word documents allowed for straightforward PDF export with minimal remediation.

## Documentation – How did you make this accessible?

Caption text was standardised at 12pt for readability, overriding Word's default 9pt caption style.

### Assessment

The *Should accessibility ever be de-linked from disability?* PDF version passed Adobe Acrobat's accessibility checker with no critical errors detected. I manually examined whether the document had a logical reading order and sufficient colour contrast.

Both navigation using keyboard and the tab order were logical. I ensured there was sufficient contrast using WebAIM contrast checker.

Overall, the PDF met accessibility standards.

### Reflections

Planning for accessibility upfront significantly reduces effort. Choosing an accessible Hugo theme covered most structural requirements, and properly formatted Word documents naturally produced accessible PDFs.

However, accessibility was demanding when making the video. This was the case for writing and then timing the audio description narration. I think this came more from my inexperience in audio description.

At my workplace, I often use GenAI to draft alt text, which I then refine. A similar workflow for audio description (e.g. having a GenAI model come up with descriptions based on excerpts or single frames) would improve efficiency.

This project reinforced a key lesson: designing for accessibility is far easier than retrofitting it.

When I began working at an unnamed government agency, I thought cramming everything into one complex table was best approach—it displayed all information at once. In hindsight, those tables were not user friendly and would be difficult to remediate. In many cases, it's easier to just have three simple tables that can get the information across more effectively and accessibly.