Audit Report - GoMike-Desings

1. **HTML lang attribute was set to default (accessibility)**

The default value of lang is unknown, therefore it is recommended to always specify this attribute with the appropriate value[[1]](#footnote-0).

WCAG Success Criterion 3.1.1 **requires** that a page language is specified in a way which may be 'programmatically determined' (i.e. via the **lang** attribute).

The purpose of these requirements is primarily to allow assistive technologies such as screen readers to invoke the correct pronunciation.

1. **Title was missing**

Page title[[2]](#footnote-1) is important for SEO and Accessibility too.

The contents of a page title can have significant implications for search engine optimization (SEO).

It is important to provide a title value that describes the page's purpose.

A common navigation technique for users of assistive technology is to read the page title and infer the content the page contains.

This is because navigating into a page to determine its content can be a time consuming and potentially confusing process.

1. **Multiple h1 on the page**

Using multiple h1**[[3]](#footnote-2)** on the page won’t result in an error although that is not the best practice. H1 is beneficial for screenreader users and SEO.

A common navigation technique for users of screen reading software is jumping from heading to heading to quickly determine the content of the page. Because of this, it is important to not skip one or more heading levels. Doing so may create confusion, as the person navigating this way may be left wondering where the missing heading is.

1. **Running JavaScript at the right time**

Placing JavaScript files at the end of the \*.html page before the closing body tag will allow that HTML and CSS execute before JavaScript[[4]](#footnote-3).

The reason to place the [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script) element near the bottom of the HTML file is that **the browser reads code in the order it appears in the file**. If the JavaScript loads first and it is supposed to affect the HTML that hasn't loaded yet, there could be problems. Placing JavaScript near the bottom of an HTML page is one way to accommodate this dependency.

1. **Preload key request**

Consider using `<link rel=preload>`[[5]](#footnote-4) to prioritize fetching resources that are currently requested later in page load.

Preloading lets you declare specific resources that a browser must request without blocking the rest of your page’s rendering.

Basically, it gives you more control over how the resources on a web page load, which helps you optimize your site’s performance and improve your site’s perceived page load times.

1. **Image elements do not have explicit width and height**

As images are loaded asynchronously and continue to load after the first paint, if their dimensions aren't defined before load, they can cause reflows to the page content.

This change means that the aspect ratio of the image is calculated by the browser early on and can then be used to correct the size needed to display the image before it has loaded, if CSS has been applied that causes problems with its display size[[6]](#footnote-5).

For this reason, it's critical that you set width and height attributes so that the browser can reserve space for them in the layout.

1. **Large network payloads**

Large network payloads are highly correlated with long load times. They also cost users money; for example, users may have to pay for more cellular data. So, reducing the total size of your page's network requests is good for your users' experience on your site *and* their wallets[[7]](#footnote-6).

* Images size.
* Images instead of text.

1. **Serve images in next-gen formats**

WebP is an excellent choice for both images and animated images. WebP offers much better compression than PNG or JPEG with support for higher color depths, animated frames, transparency etc. (but not progressive display.). Supported by all major browsers except Safari[[8]](#footnote-7).

1. **Names and labels. Links do not have a discernible name (accessibility)**

Website features should be given a descriptive name or label to allow users of assistive technologies to understand what its purpose is and how to operate it correctly[[9]](#footnote-8).

In image maps, give each [<area>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/area) element an alt attribute containing a name that describes what resources the areas links to. Failure to do so makes an image map hard to use for users of assistive technology — they need alternative text to be able to understand the purpose of an image.

1. **Security vulnerabilities (best practices)**

Intruders have automated web crawlers that can scan your site for known security vulnerabilities. When the web crawler detects a vulnerability, it alerts the intruder. From there, the intruder just needs to figure out how to exploit the vulnerability on your site[[10]](#footnote-9). Update jQuery and BootStrap.

1. **Background and foreground colors do not have sufficient contrast ratio**

When choosing a color scheme for your website, choose foreground and background colors that have good contrast. Make the color contrast as good as it can be within your design constraints — ideally go for the AAA rating (see 1.4.6 below), but at least meet the AA rating.[[11]](#footnote-10)

[1.4.3 Minimum contrast (AA)](https://www.w3.org/TR/WCAG21/#contrast-minimum)

The color contrast between background and foreground content should be at a minimum level to ensure legibility:

* Text and its background should have a contrast ratio of at least 4.5:1.
* Heading (or just larger) text should have a ratio of at least 3:1. Larger text is defined as at least 18pt, or 14pt bold.

[1.4.6 Enhanced contrast (AAA)](https://www.w3.org/TR/WCAG21/#contrast-enhanced)

This follows, and builds on, criterion 1.4.3.

* Text and its background should have a contrast ratio of at least 7:1.
* Heading (or just larger) text should have a ratio of at least 4.5:1.

[1.4.11 Non-Text Contrast (AA)](https://www.w3.org/TR/WCAG21/#non-text-contrast) (added in 2.1)

There should be a minimum color contrast ratio of 3 to 1 for user interface components and graphical objects.

1. **Semantic tags**

HTML should be coded to represent the *data* that will be populated and not based on its default presentation styling[[12]](#footnote-11).

Some of the benefits from writing semantic markup are as follows:

* Search engines will consider its contents as important keywords to influence the page's search rankings (see [SEO](https://developer.mozilla.org/en-US/docs/Glossary/SEO))
* Screen readers can use it as a signpost to help visually impaired users navigate a page
* Finding blocks of meaningful code is significantly easier than searching through endless divs with or without semantic or namespaced classes
* Suggests to the developer the type of data that will be populated
* Semantic naming mirrors proper custom element/component naming

1. **Keywords**

Many <meta> features just aren't used any more. For example, the keyword <meta> element (<meta name="keywords" content="fill, in, your, keywords, here">) — which is supposed to provide keywords for search engines to determine relevance of that page for different search terms — is ignored by search engines, because spammers were just filling the keyword list with hundreds of keywords, biasing results[[13]](#footnote-12).

1. **Responsiveness**

If the user had a larger or smaller screen than the designer expected, results ranged from unwanted scrollbars to overly long line lengths, and poor use of space. As more diverse screen sizes became available, the concept of *responsive web design* (RWD) appeared, a set of practices that allows web pages to alter their layout and appearance to suit different screen widths, resolutions, etc[[14]](#footnote-13).

1. **Minification**

Minification is the process of removing unnecessary or redundant data without affecting how a resource is processed by the browser. Minification can include the removal of code comments, white space, and unused code, as well as the shortening of variable and function names. Minification is used to improve web performance by reducing file size[[15]](#footnote-14).

1. "lang - HTML: HyperText Markup Language | MDN - Mozilla." 28 Dec. 2020, <https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/lang>. Accessed 11 Feb. 2021. [↑](#footnote-ref-0)
2. "<title>: The Document Title element - HTML: HyperText ... - MDN." 18 Dec. 2020, <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title>. Accessed 22 Jan. 2021. [↑](#footnote-ref-1)
3. "<h1>–<h6>: The HTML Section Heading elements - MDN." 4 Dec. 2020, <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements>. Accessed 21 Jan. 2021. [↑](#footnote-ref-2)
4. "JavaScript basics - Learn web development - MDN - Mozilla." 8 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics>. Accessed 21 Jan. 2021. [↑](#footnote-ref-3)
5. "Preloading content with rel="preload" - HTML: HyperText ...." 15 Dec. 2020, <https://developer.mozilla.org/en-US/docs/Web/HTML/Preloading_content>. Accessed 24 Jan. 2021. [↑](#footnote-ref-4)
6. "Mapping the width and height attributes of media ... - MDN - Mozilla." 30 Dec. 2020, <https://developer.mozilla.org/en-US/docs/Web/Media/images/aspect_ratio_mapping>. Accessed 11 Feb. 2021. [↑](#footnote-ref-5)
7. "Avoid enormous network payloads - web.dev." 29 May. 2020, <https://web.dev/total-byte-weight/>. Accessed 11 Feb. 2021. [↑](#footnote-ref-6)
8. "Multimedia: Images - Learn web development | MDN - Mozilla." 31 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Learn/Performance/Multimedia>. Accessed 13 Feb. 2021. [↑](#footnote-ref-7)
9. "Text labels and names - Accessibility | MDN - Mozilla." 9 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG/Text_labels_and_names>. Accessed 19 Feb. 2021. [↑](#footnote-ref-8)
10. "Includes front-end JavaScript libraries with known security ... - web.dev." 4 Jun. 2020, <https://web.dev/no-vulnerable-libraries/>. Accessed 19 Feb. 2021. [↑](#footnote-ref-9)
11. "Color contrast - Accessibility | MDN - Mozilla." 24 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG/Perceivable/Color_contrast>. Accessed 19 Feb. 2021. [↑](#footnote-ref-10)
12. "Semantics - MDN Web Docs Glossary: Definitions of Web ... - Mozilla." 26 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Glossary/Semantics>. Accessed 19 Feb. 2021. [↑](#footnote-ref-11)
13. "What's in the head? Metadata in HTML ...." 26 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML>. Accessed 19 Feb. 2021. [↑](#footnote-ref-12)
14. "Responsive design - Learn web development | MDN - Mozilla." 26 Jan. 2021, <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design>. Accessed 19 Feb. 2021. [↑](#footnote-ref-13)
15. "minification - MDN Web Docs Glossary: Definitions of Web ... - Mozilla." 15 Sep. 2020, <https://developer.mozilla.org/en-US/docs/Glossary/minification>. Accessed 20 Feb. 2021. [↑](#footnote-ref-14)