A critical reflection on refactoring the Web Witchcraft and Wizardry project

`https://witches-and-wizards.edmundmulligan.name`

Edmund Mulligan

December 21, 2025

Abstract

This is a personal reflection on the design and implementation of a refactored version of a website. The new architecture is described, along with the navigation design, responsive and accessibility features, testing strategy, known issues and future improvements. Third party credits and resources are also acknowledged.

1 Introduction

Web Witchcraft and Wizardry (Mulligan, 2024) is a website for children to learn the basics of web development through fantasy-themed metaphors. The original version of the website was neither accessible (except by accident) nor responsive. This essay reflects on the design and implementation of a refactored version of the project, focusing on architecture, accessibility and code quality. Detailed decisions are documented as comments in the source code (available on the main branch at `https://github.com/Birkbeck2/puffin-web-development-website-edmundmulligan`), so this essay focuses on higher level design decisions and reflections.

For comparison the old and new landing pages can be seen in figures 1, 2 and 3.

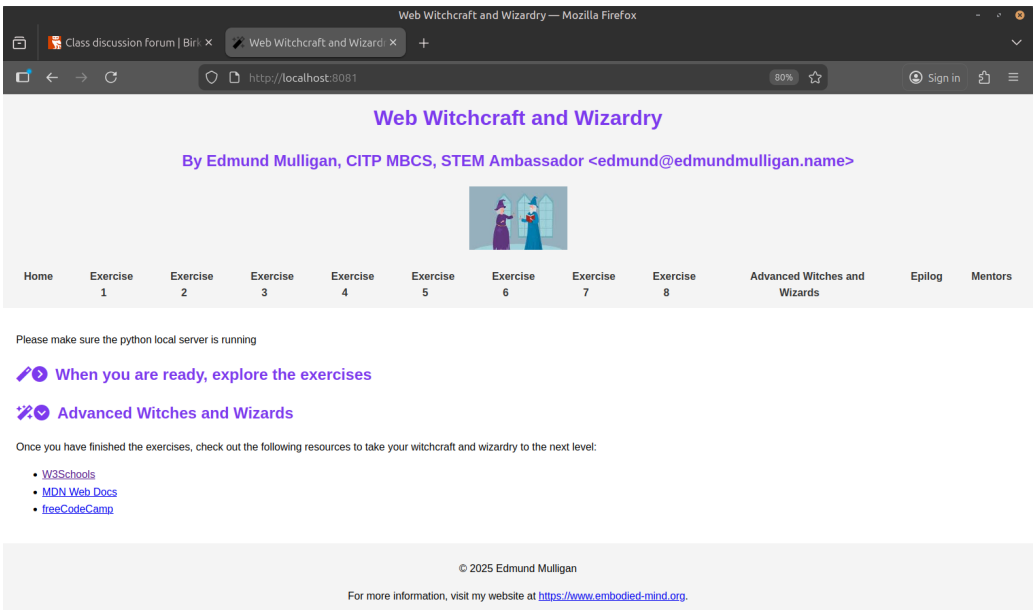

Figure 1: Old landing page

Figure 2: New landing page on a computer

Figure 3: New landing page on a mobile device

The development process used was iterative and incremental (Beck and Andres, 2004), and while there are criticisms of this approach, particularly around its lack of documentation, endless changes and lack of scalability to large teams (see, for example, Boehm and Turner 2004), it was well suited to this small project. The advantage of being able to make quick changes and see immediate results (especially valuable when learning new techniques) was to some extent offset by the extra work required to frequently refactor code to maintain quality and consistency. The process is informally summarised in figure 4.

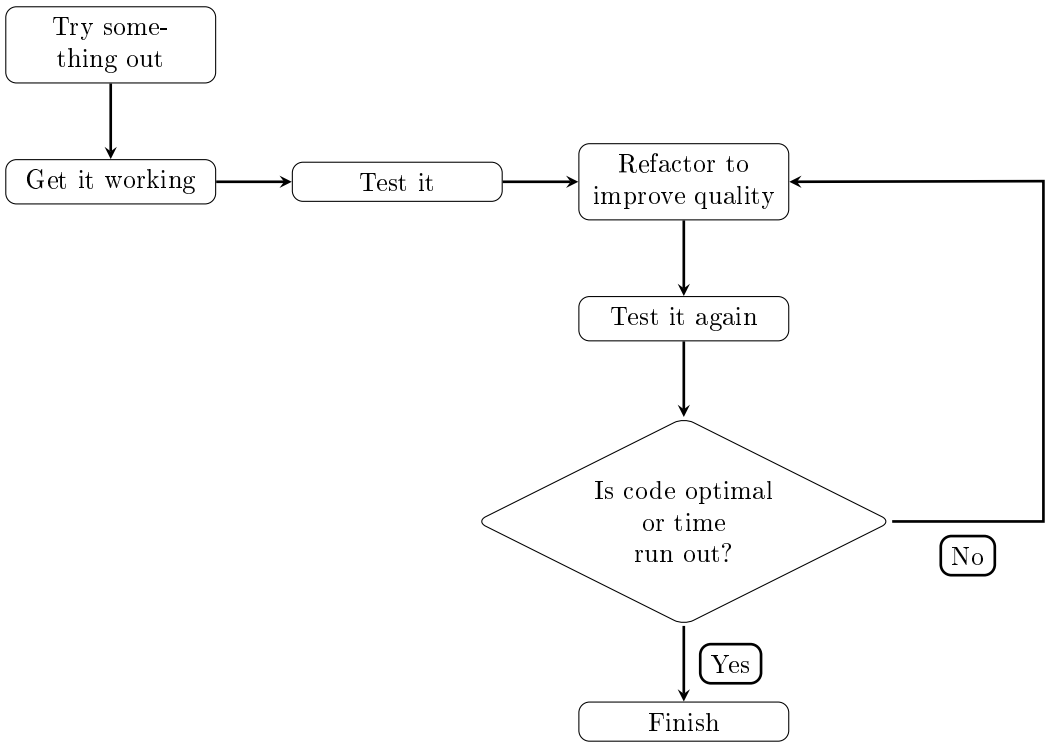

Figure 4: Iterative development process flowchart

This essay was written in parallel with the development of the website, so screenshots may not exactly reflect the final version of the code submitted for assessment, but are intended to illustrate the design decisions discussed.

2 Architecture

Even in a small project, software design principles are important to keep the code maintainable (Gamma et al., 1994). The three main principles applied in this project were separation of concerns (Parnas, 1972), DRY (Don't Repeat Yourself) (Hunt and Thomas, 1999), and self-documenting code (Martin, 2008). The first principle lead to the folder structure shown in figure 5. One advantage of this is that finding and fixing bugs is much easier in well structured code.

The second principle influenced the use of JavaScript to inject common elements into each page at runtime, rather than duplicating the code in each HTML file. This also inspired the use of CSS variables to define common colours and other styles in one place, rather than duplicating these values throughout the CSS files.

The third principle required meaningful names and a consistent naming convention was used throughout.

A fourth (informal) principle is Keep the CSS as simple as possible and make use of cascading styles. This is something experienced web developers know and if I were to redo this project I would put much more effort into writing simple CSS first time rather than trying to refactor complex CSS later. The student.html page is a particular example where the CSS is more complex than necessary and needs more refactoring but I ran out of time to complete this.

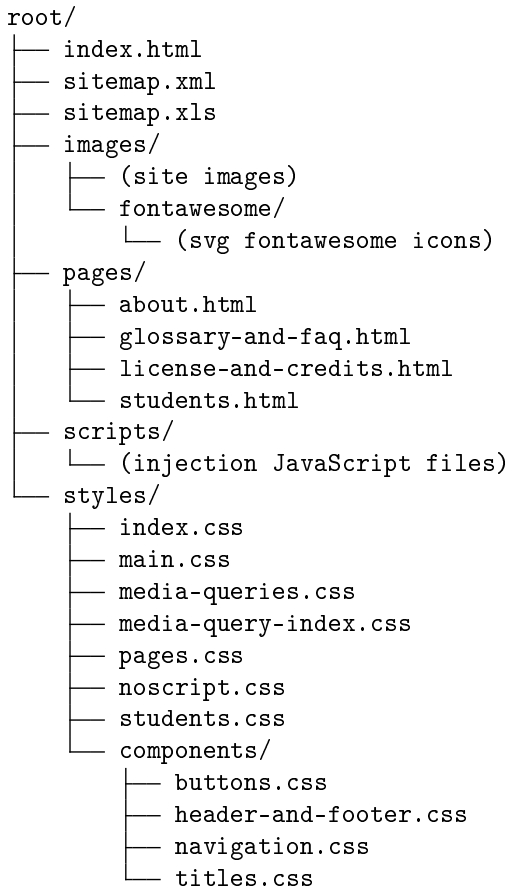

Figure 5: Directory structure of the project

3 Navigation

Navigation design was kept simple with a top navigation bar for site-wide navigation (figure 6) and an in-page navigation bar for navigating within all pages except the landing page (e.g. figure 7, although each page is slightly different). Incidentally, the wizard and witch in the site header are informally known to the development team as 'Joe' and 'Helena' respectively.

Figure 6: Site navigation bar structure

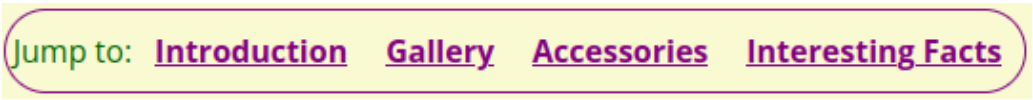

Figure 7: In-page navigation bar structure

4 A Responsive and Accessible Website

The main goal of the refactor was to make the website responsive and accessible. The primary techniques used to achieve responsiveness are media queries and flexible layouts (Marcotte, 2010). Accessibility features were mainly implemented by ensuring the code conformed to web standards, (Caldwell et al., 2024), but also by following best practices in the HTML markup, such as using semantic and landmark HTML elements (WHATWG, 2025),

4.1 Using and avoiding media queries

A point made in several YouTube videos (credited in section 7) is that while media queries are important, they should not be overused. The main layout should be flexible enough to adapt to a wide range of viewport sizes without the need for frequent breakpoints. Exactly which breakpoints to use is a matter of experience (which I lack) and judgement (which I possess in abundance), and I eventually settled on standardised breakpoints across all pages listed in table 1.

Viewport width	**Layout**	**Impact**
0px to 199px	Very small	Warn users but make best effort to display contents reasonably
200px to 799px	Mobile layout	Prefer vertical stacking
800px to 1199px	Desktop layout	Prefer horizontal layout
1200px and above	Large desktop layout	Take advantage of wider screen

Table 1: Viewport width breakpoints

In order to minimise the use of media queries, I used the clamp () CSS function to define fluid typography. A limitation of this is that if the middle value of clamp () is defined using a viewport

width unit (vw), then the text will not scale if the user zooms their browser without changing the viewport window size. This contravenes WCAG 2.2 Success Criterion 1.4.4 (Caldwell et al., 2024) which requires text to be resizable up to 200% without loss of content or functionality. The workaround for this is to calculate the middle value of clamp () from a vw unit and a rem unit, so that zooming the browser window also changes the text size.

One downside of this approach is that it takes a lot of trial and error to find suitable values to pass to clamp () that work well across a wide range of viewport sizes and this took up much of my testing time. The process was made a lot easier when I discovered, at a late stage, an online tool to help calculate fluid typography values (Bece, 2025).

4.2 Flexbox vs Grid

One of the debates I encountered in many tutorials was whether to use CSS Flexbox (Atkins et al., 2018) or CSS Grid (Atkins et al., 2017) for layout. My conclusion was to use flexbox in narrower viewports as this could make vertical stacking of elements easier with flex-direction: column, and to prefer flexbox with flex-direction: row in wider viewports unless there was an obvious case for a grid such as the input form on the student page and the gallery of portraits. Due to my experimental approach to development, early versions of the website had several levels of nested grids and flexboxes which then had to be refactored to simpler layouts later. I also evetually realised that flexbox for vertical stacking is not always necessary as block elements naturally stack vertically. This is another area where more experience would have helped in making better design decisions earlier in the development process.

4.3 Units of measurement

Another way to achieve responsiveness is to use relative units of measurement rather than absolute units. Thus, I used rem units for specifying element sizes. I avoided using px units except in media queries where absolute measurements are required. I preferred rem units over em units as rems are relative to the root font size, making them more predictable, whereas ems are relative to the font size of the parent element, which can (and did) lead to unexpected results.

4.4 Browser compatibility

Ensuring browser compatibility is an important part of web development, particularly when using newer CSS features that may not be supported in all browsers. I used caniuse.com (Can I Use, 2025) and only used HTML/CSS features that were supported by all major browsers (Chrome, Firefox, Safari and Edge). One feature I particularly wanted to try was popover to avoid having to use JavaScript alert (). I was mostly successful but could not get it to work with the form submit button (in student.html) so the Save Information button currently does nothing when clicked. That will be resolved with JavaScript in phase 2 of the project.

On a related note, all file pathes were specified as absolute rather than relative. This is not best practice, but it avoided issues as I moved files around during development and refactoring. The website will break if it is not deployed into the root directory of a web server, but that is an acceptable limitation for now.

4.5 Other accessibility features

The accessibility testing tools sometimes flagged issues with ARIA (Diggs et al., 2023) compliance. I did not have time to fully investigate these issues, but I incorporated sufficient

ARIA features to satisfy the testing tools. This certainly deserves further investigation in the future.

Choice of colours and contrast is very important, not only to make a site accessible but also visually appealing. This is another area where I have little design experience so I defer the aesthetic aspects of colour choice to next term's module, where that is covered more fully. For this phase of the project, I limited myself to ensuring that the colour choices were not too garish and that the choices passed the contrast tests in the accessibility tools. The purple and cyan colours were chosen as they (approximately) match the Embodied Mind logo used in the footer; the yellow and green were more arbitary.

Testing without a mouse and with a screen reader ensured that that the ::focus pseudo-class was styled appropriately to make it clear which element had focus and meaningful alt text was provided for all images. Using the web site in this way was not particularly pleasant or easy and emphasises how important it is to design for accessibility from the start to ensure that those who have no choice but to use assistive technologies have as good a user experience as can be achieved.

5 Testing

Testing software is not only a critical part of development (Myers et al. (2011),Mesbah et al. (2012)), but it is also difficult and time consuming. Many authorities (e.g. Boehm (1981), Jones (2008)) claim that testing can take at least 50% of total development time, and that concurs with my experience.

My approach to testing was to automate as much as possible using open source tools (see Appendix B for results). The tests were designed to check for compliance with web standards (as a proxy for accessibility) and (given the age of the target users) text readability. Both static and dynamic testing tools (Myers et al., 2011) were used — static tools to check code validity, and dynamic tools to check accessibility of the rendered pages.

The tests were run frequently during development to catch issues early and ensure that the code remained compliant as it was refactored (regression testing). A github action was written to ensure the tests were run whenever code was pushed to the remote development and main branches in github. Testing tools were discovered using a google search and the most popular ones were chosen:

- Lighthouse (Google, 2025)
- Axe (Deque Systems, 2025)
- Pa11y (Pa11y Team, 2025)
- Wave (WebAIM, 2025)

A script to check reading age using the Flesch-Kincaid formula (Flesch, 1948) and Gunning Fog index (Gunning, 1952) was developed using the text-readability node.js module (Aashish Khanal, 2025). Code was checked for validity using node.js modules html-validate (html-validate, 2025), stylelint (Stylelint Team, 2025) and eslint (ESLint Team, 2025). The official W3C validator for CSS (W3C, 2025) was not used as this cannot currently validate code with CSS variables. Browser compatibility was tested using Playwright (Microsoft, 2025) to run tests in multiple browsers and I acknowledge, with thanks, the assistance I received to get this working from a senior tester colleague at TfL. This allowed automated testing of the latest versions of all three major browser engines (Chrome, Firefox and Safari). Playwright has potential for much more and could enable a test driven development (TDD) (Beck, 2003) approach in future phases of this project.

Implementing Wave testing was the most difficult as it requires a public URL to test and github pages are based on a private repository. Making the github repository public was

considered and rejected as it could introduce unanticipated security risks so, after consultation with testing colleagues, ngrok (ngrok, 2025) was used to expose the localhost to allow Wave tests to complete.

There was insufficient time to fully evaluate the third party tools used and decide which was the best for this project, so a 'black box' approach was taken without investigating exactly what each tool was testing. This did have the benefit that gaps in one tool's coverage could potentially be caught by one of the other tools, but it did introduce a problem of tools having conflicting standards. Wave, in particular requires the <h1> element to be in the <main> section, which would mean that the site title in the <header> section would have to be either an <h2> or a <p> element, which would be flagged by the other testing tools. Thus a decision to accept the warning from Wave was taken as the least bad alternative. The other alert from Wave was for the use of <noscript>, but this was simply a warning to check this usage as it may have accessibility implications.

Automated testing is not a substitute for manual testing, and the site was also tested manually on a variety of devices, specifically

- A computer with a mouse and keyboard (running Linux Mint)
- A computer without using a mouse (to test keyboard navigation)
- A computer with a screen reader (Linux Orca)
- A touchscreen tablet device without mouse or keyboard (running Android)
- A mobile phone (running Android)

I did not have access to an iOS device so could not test on Apple hardware, nor did I have time to test on non-Linux operating systems. Testing was informal and ad hoc rather than systematic, and with more time, test scripts would be written to ensure consistent coverage across all devices. This would be more necessary if more than one developer were involved in the project to ensure everyone tested the same way.

I did not test the website with JavaScript disabled and I acknowledge this as a limitation of the current testing, driven by the available time. Testing thus far has mainly been verification rather than validation (Myers et al., 2011), (IEEE, 1990), but when actual lessons are implemented user and expert validation from a teacher will be needed to ensure the content is age appropriate and pedagogically sound. Security, and performance testing will also be needed when student data (tracking lesson progress) are collected.

6 Outstanding issues and future improvements

The project submitted for assessment is phase one of a larger project to refactor and improve the Web Witchcraft and Wizardry website. Some content was deliberately omitted from this phase as it will be developed in phase 2 using JavaScript. Table 2 in Appendix A lists some known issues and areas for improvement that could not be addressed within the time constraints of this phase.

7 Third party credits and resources

All code submitted for assessment is my own work. No code has been copied from any external sources, such as online tutorials or code. I have, however, relied heavily on three YouTube channels, those of Kevin Powell (Powell, 2025), Daily CSS (Daily CSS, 2025), and Coding2GO (Coding2GO, 2025), for guidance on best practices and inspiration.

I made extensive use of Google Search, Google Scholar and Stack Overflow to research and troubleshoot specific issues I encountered.

The automation tools used for testing accessibility and performance (Lighthouse, Axe,

pa11y, Wave) are all open source tools developed by third parties and the scripts used to run them were adapted from examples found in their respective documentations and in online resources. I was assisted by a colleague at my employer, Transport for London, in writing the Playwright tests used for testing browser compatibility.

The original artwork used in the website was created by Rachel Mulligan, a professional stained glass artist. I am grateful to her for allowing me to use her work in this project. The images used in this iteration are an early version and will be refined in future phases.

8 Conclusion

This has been a challenging and rewarding project. The challenge has not been coding the website, as I have 40 years of experience doing that, but in learning a lot of new material around modern CSS in a short space of time and applying that in a standards based development paradigm. The refactored Web Witchcraft and Wizardry website, while not yet complete, is a significant improvement on the original version in terms of accessibility, responsiveness and code quality. The use of automated testing has helped ensure that the code meets web standards and is accessible to the target audience. There are still some outstanding issues and areas for improvement that will be addressed in future phases of the project.

Word count

Word count: 2545 words (excluding references)

References

Aashish Khanal. text-readability: An implementation of readability algorithms in javascript. `https://www.npmjs.com/package/text-readability`, 2025. Accessed: December 2025.

Tab Atkins, Elika J. Etemad, and Alex Mogilevsky. Css grid layout module level 1. W3c recommendation, World Wide Web Consortium (W3C), 2017. URL `https://www.w3.org/TR/css-grid-1/`. Accessed: December 2025.

Tab Atkins, Elika J. Etemad, and Alex Mogilevsky. Css flexible box layout module level 1. W3c recommendation, World Wide Web Consortium (W3C), 2018. URL `https://www.w3.org/TR/css-flexbox-1/`. Accessed: December 2025.

Adrian Bece. Modern fluid typography editor. `https://modern-fluid-typography.vercel.app/`, 2025. Accessed: December 2025.

Kent Beck. *Test-Driven Development: By Example*. Addison-Wesley, Boston, MA, 2003.

Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, Boston, MA, 2nd edition, 2004.

Barry Boehm and Richard Turner. Balancing agility and discipline: A guide for the perplexed. *IEEE Software*, 21(6):57–58, 2004.

Barry W. Boehm. *Software Engineering Economics*. Prentice Hall, Englewood Cliffs, NJ, 1981.

Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. Web content accessibility guidelines (wcag) 2.2. W3c recommendation, World Wide Web Consortium (W3C), 2024. URL https://www.w3.org/TR/WCAG22/. Accessed: November 2025.

Can I Use. Can i use... support tables for html5, css3, etc, 2025. URL https://caniuse.com/. Accessed: December 2025.

Coding2GO. Coding2go, 2025. URL https://www.youtube.com/@Coding2GO. Accessed: December 2025.

Daily CSS. Daily css — helping you ... with some nifty css, 2025. URL https://www.youtube.com/@dailycss. Accessed December 2025.

Deque Systems. Axe devtools: Web accessibility testing, 2025. URL https://www.deque.com/axe/devtools/. Accessed: December 2025.

Joanmarie Diggs, Shane McCarron, Michael Cooper, and Richard Schwerdtfeger. Accessible rich internet applications (wai-aria) 1.2. W3c recommendation, World Wide Web Consortium (W3C), 2023. URL https://www.w3.org/TR/wai-aria-1.2/. Accessed: December 2025.

ESLint Team. Eslint: Find and fix problems in your javascript code. https://eslint.org/, 2025. Accessed: December 2025.

Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233, 1948.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1994.

Google. Lighthouse: An open-source, automated tool for improving the quality of web pages, 2025. URL https://developers.google.com/web/tools/lighthouse. Accessed: December 2025.

Robert Gunning. *The Technique of Clear Writing*. McGraw-Hill, New York, 1952.

html-validate. html-validate: Offline html5 validator. https://html-validate.org/, 2025. Accessed: December 2025.

Andrew Hunt and David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley Professional, Boston, MA, 1999.

IEEE. IEEE standard glossary of software engineering terminology. Technical Report IEEE Std 610.12-1990, IEEE Computer Society, 1990.

Capers Jones. *Applied Software Measurement: Global Analysis of Productivity and Quality*. McGraw-Hill, 3rd edition, 2008.

Ethan Marcotte. Responsive web design. *A List Apart*, 306, May 2010. URL https://alistapart.com/article/responsive-web-design/. Accessed: December 2025.

Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, Upper Saddle River, NJ, 2008.

Ali Mesbah, Arie van Deursen, and Danny Roest. Invariant-based automatic testing of modern web applications. In *IEEE Transactions on Software Engineering*, volume 38, pages 35–53. IEEE, 2012.

Microsoft. Playwright: Fast and reliable end-to-end testing for modern web apps. https://playwright.dev/, 2025. Accessed: December 2025.

Edmund Mulligan. Web witchcraft and wizardry. https://github.com/edmundmulligan/Witches-And-Wizards, 2024. Accessed: December 2025.

Glenford J Myers, Corey Sandler, and Tom Badgett. *The Art of Software Testing*. John Wiley & Sons, Hoboken, NJ, 3rd edition, 2011.

ngrok. ngrok – unified application delivery platform for developers, 2025. URL https://ngrok.com/. Accessed: December 2025.

Pa11y Team. Pa11y: Automated accessibility testing. https://pa11y.org/, 2025. Accessed: December 2025.

David L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.

Kevin Powell. Kevin powell — css evangelist, 2025. URL https://www.youtube.com/@KevinPowell. Accessed December 2025.

Stylelint Team. Stylelint: A mighty, modern linter that helps you avoid errors and enforce conventions in your styles. https://stylelint.io/, 2025. Accessed: December 2025.

W3C. W3c css validation service. https://jigsaw.w3.org/css-validator/, 2025. Accessed: December 2025.

WebAIM. WAVE web accessibility evaluation tool, 2025. URL https://wave.webaim.org/. Accessed: November 2025.

WHATWG. Html standard. Technical report, Web Hypertext Application Technology Working Group, 2025. URL https://html.spec.whatwg.org/. Accessed: December 2025.

Appendices

A Known Issues and Workarounds

This appendix contains known issues and workarounds.

Issue Id	Description	Symptoms	Work around
1	Responsive code does not work well at extreme viewport sizes	Text overflows boxes or content does not fill screen properly.	Warning displayed when viewport is too small.
2	Font sizes are not optimal	Text is too small or too large on some devices.	Further refinement of clamp () values is needed.
3	The artwork provided by Rachel Mulligan is in JPEG format with a white background.	The picture gallery looks ugly with white boxes around images.	Images need to be edited to have transparent backgrounds (PNG or SVG format).
4	student.css is a mess.	Doesn't conform to the standards in other css files.	More refactoring needed.
5	Incomplete functionality.	Some features are not implemented yet.	Further development required to complete all features.
6	ARIA compliance is minimal.	Some elements lack full ARIA attributes.	Invesigatin of ARIA, accessibility audit and improvements needed.
7	popovertarget does not work on submit buttons.	The popover does not appear when clicking the Save Information button.	Use JavaScript to show a popover or alert on form submission.
8	Website does not respect system light/dark mode preference.	Website always uses light mode even if system is in dark mode.	Use CSS media query prefers-color-scheme to adapt to system preference.
9	Don't know how acessibility checkers work.	How do they handle checking at different viewports given the responsive design?	Need more invesgation of these tools.

Table 2: Known issues and workarounds

B Automated Test Results

This appendix contains the complete test results from the automated test suite. All tests were executed on December 21, 2025 and demonstrate compliance with web standards, accessibility guidelines, and readability requirements.

Summary of Test Results

```
[Test Results] Test Results Summary
======================

[Code] Code Validation Results:
  Files with issues: 0
  HTML errors: 0
  CSS errors: 0
  Total errors: 0
  Total warnings: 0

[Links] Broken Links Results:
  Pages checked: 5
  Total links: 121
  Broken links: 0

======================
[OK] All tests passed (no critical/serious issues)
[Browser] Browser Compatibility Results:
  Browsers tested: 3
  Passed: 3
  Failed: 0

[Axe] Axe Accessibility Results:
  Total violations: 0
  [Critical] Critical: 0
  [Serious] Serious: 0
  [Moderate] Moderate: 0
  [Minor] Minor: 0

[Lighthouse] Lighthouse Accessibility Results:
  Pages tested: 5
  Average score: 100%
  Pages with issues: 0
  Total failed audits: 0

[Pa11y] Pa11y Accessibility Results:
  Pages tested: 5
  [X] Errors: 0
  [Warning]  Warnings: 0
  Pages with errors: 0

[WAVE] WAVE Accessibility Results:
```

```
  Pages tested: 0
  [X] Errors: 0
  [Warning]  Alerts: 0
  [Contrast] Contrast errors: 0

[Reading] Reading Age Results:
  Pages analyzed: 3
  Total words: 2598
  Average grade level: 9.6
  [OK] All pages at high school level or below
```

Detailed Test Results

The following sections contain the complete JSON output from each test suite.

Code Validation Results


```
{
  "files": [],
  "summary": {
    "htmlErrors": 0,
    "htmlWarnings": 0,
    "cssErrors": 0,
    "cssWarnings": 0,
    "jsErrors": 0,
    "jsWarnings": 0
  }
}
```

File Comments Results

```
File Header Comments Check Results
==================================


====================================
Summary
====================================
Total files checked: 28
Files with issues: 0
  Missing header blocks: 0
  Missing required fields: 0
```

Broken Links Check Results


```
{
  "pages": [
    {
      "url": "index.html",
      "links": [],
      "brokenCount": 0,
      "totalCount": 17
    },
    {
      "url": "pages/students.html",
      "links": [],
      "brokenCount": 0,
      "totalCount": 15
    },
    {
      "url": "pages/glossary-and-faq.html",
      "links": [],
      "brokenCount": 0,
      "totalCount": 15
    },
    {
      "url": "pages/license-and-credits.html",
      "links": [],
      "brokenCount": 0,
      "totalCount": 27
    },
    {
      "url": "pages/about.html",
      "links": [],
      "brokenCount": 0,
      "totalCount": 47
    }
  ],
  "summary": {
    "totalLinks": 121,
    "brokenLinks": 0,
    "excludedLinks": 0
  }
}
```

Axe Accessibility Results

```
{
  "violations": [],
  "passes": [],
  "incomplete": []
}
```

Lighthouse Accessibility Results


```
{
  "pages": [
    {
      "url": "index.html",
      "score": 1,
      "failedAudits": []
    },
    {
      "url": "pages/students.html",
      "score": 1,
      "failedAudits": []
    },
    {
      "url": "pages/glossary-and-faq.html",
      "score": 1,
      "failedAudits": []
    },
    {
      "url": "pages/license-and-credits.html",
      "score": 1,
      "failedAudits": []
    },
    {
      "url": "pages/about.html",
      "score": 1,
      "failedAudits": []
    }
  ]
}
```

Pa11y Accessibility Results


```
{
  "pages": [
    {
      "url": "index.html",
      "documentTitle": "Web Witchcraft and Wizardry",
      "pageUrl": "http://localhost:8080/",
      "issues": []
    },
    {
      "url": "pages/students.html",
      "documentTitle": "Student Dashboard|Web Witchcraft and Wizardry",
      "pageUrl": "http://localhost:8080/pages/students",
      "issues": []
    },
    {
      "url": "pages/glossary-and-faq.html",
      "documentTitle": "Glossary and FAQ|Web Witchcraft and Wizardry",
      "pageUrl": "http://localhost:8080/pages/glossary-and-faq",
      "issues": []
    },
    {
      "url": "pages/license-and-credits.html",
      "documentTitle": "License and Credits|Web Witchcraft and Wizardry
        ",
      "pageUrl": "http://localhost:8080/pages/license-and-credits",
      "issues": []
    },
    {
      "url": "pages/about.html",
      "documentTitle": "About|Web Witchcraft and Wizardry",
      "pageUrl": "http://localhost:8080/pages/about",
      "issues": []
    }
  ]
}
```

WAVE Accessibility Results

```
{"pages":[]}
```

Reading Age Analysis Results


```
{
  "pages": [
    {
      "file": "./pages/students.html",
      "wordCount": 490,
      "fleschReadingEase": 60.9,
      "fleschKincaidGrade": 9.4,
      "smogIndex": 11.5,
      "colemanLiauIndex": 9.2,
      "automatedReadabilityIndex": 10.1,
      "gunningFogIndex": 10.5,
      "daleChallScore": 7.1,
      "averageGradeLevel": 10.1,
      "textStandard": "8th and 9th grade",
      "difficultWords": 79,
      "averageSentenceLength": 18.8,
      "readingLevel": "Fairly Difficult (High School)",
      "readingAge": "17-18 years"
    },
    {
      "file": "./pages/glossary-and-faq.html",
      "wordCount": 81,
      "fleschReadingEase": 68.5,
      "fleschKincaidGrade": 6.5,
      "smogIndex": 10.3,
      "colemanLiauIndex": 8.9,
      "automatedReadabilityIndex": 7.1,
      "gunningFogIndex": 9.1,
      "daleChallScore": 7.8,
      "averageGradeLevel": 8.4,
      "textStandard": "8th and 9th grade",
      "difficultWords": 18,
      "averageSentenceLength": 11.3,
      "readingLevel": "Plain English (High School)",
      "readingAge": "15-16 years"
    },
    {
      "file": "./pages/about.html",
      "wordCount": 2027,
      "fleschReadingEase": 54.9,
      "fleschKincaidGrade": 9.6,
      "smogIndex": 11.7,
      "colemanLiauIndex": 9.8,
      "automatedReadabilityIndex": 9.7,
      "gunningFogIndex": 10.4,
      "daleChallScore": 7.3,
      "averageGradeLevel": 10.2,
      "textStandard": "9th and 10th grade",
      "difficultWords": 364,
      "averageSentenceLength": 16.3,
      "readingLevel": "Fairly Difficult (High School)",
```

```
      "readingAge": "17-18 years"
    }
  ]
}
```

Cross-Browser Compatibility Results

```
{
  "browsers": [
    {
      "name": "chromium",
      "status": "passed",
      "error": null,
      "tests": [
        {
          "name": "Home page title",
          "status": "passed"
        },
        {
          "name": "Home navigation links",
          "status": "passed"
        },
        {
          "name": "Home JavaScript execution",
          "status": "passed"
        },
        {
          "name": "Home CSS Grid",
          "status": "passed"
        },
        {
          "name": "Home CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "Home CSS Variables",
          "status": "passed"
        },
        {
          "name": "Home SVG support",
          "status": "passed"
        },
        {
          "name": "Home responsive viewport",
          "status": "passed"
        },
        {
          "name": "Home CSS calc()",
          "status": "passed"
        },
        {
          "name": "Home font rendering",
          "status": "passed"
        },
        {
          "name": "About page title",
          "status": "passed"
        },
```

```
        {
          "name": "About navigation links",
          "status": "passed"
        },
        {
          "name": "About JavaScript execution",
          "status": "passed"
        },
        {
          "name": "About CSS Grid",
          "status": "passed"
        },
        {
          "name": "About CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "About CSS Variables",
          "status": "passed"
        },
        {
          "name": "About SVG support",
          "status": "passed"
        },
        {
          "name": "About responsive viewport",
          "status": "passed"
        },
        {
          "name": "About CSS calc()",
          "status": "passed"
        },
        {
          "name": "About font rendering",
          "status": "passed"
        },
        {
          "name": "Students page title",
          "status": "passed"
        },
        {
          "name": "Students navigation links",
          "status": "passed"
        },
        {
          "name": "Students JavaScript execution",
          "status": "passed"
        },
        {
          "name": "Students CSS Grid",
          "status": "passed"
        },
        {
```

```
          "name": "Students CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "Students CSS Variables",
          "status": "passed"
        },
        {
          "name": "Students SVG support",
          "status": "passed"
        },
        {
          "name": "Students responsive viewport",
          "status": "passed"
        },
        {
          "name": "Students CSS calc()",
          "status": "passed"
        },
        {
          "name": "Students font rendering",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ page title",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ navigation links",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ JavaScript execution",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ CSS Grid",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ CSS Variables",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ SVG support",
          "status": "passed"
        },
        {
          "name": "Glossary & FAQ responsive viewport",
```

```
        "status": "passed"
      },
      {
        "name": "Glossary & FAQ CSS calc()",
        "status": "passed"
      },
      {
        "name": "Glossary & FAQ font rendering",
        "status": "passed"
      },
      {
        "name": "License & Credits page title",
        "status": "passed"
      },
      {
        "name": "License & Credits navigation links",
        "status": "passed"
      },
      {
        "name": "License & Credits JavaScript execution",
        "status": "passed"
      },
      {
        "name": "License & Credits CSS Grid",
        "status": "passed"
      },
      {
        "name": "License & Credits CSS Flexbox",
        "status": "passed"
      },
      {
        "name": "License & Credits CSS Variables",
        "status": "passed"
      },
      {
        "name": "License & Credits SVG support",
        "status": "passed"
      },
      {
        "name": "License & Credits responsive viewport",
        "status": "passed"
      },
      {
        "name": "License & Credits CSS calc()",
        "status": "passed"
      },
      {
        "name": "License & Credits font rendering",
        "status": "passed"
      }
    ]
  },
  {
```

```
      "name": "firefox",
      "status": "passed",
      "error": null,
      "tests": [
        {
          "name": "Home page title",
          "status": "passed"
        },
        {
          "name": "Home navigation links",
          "status": "passed"
        },
        {
          "name": "Home JavaScript execution",
          "status": "passed"
        },
        {
          "name": "Home CSS Grid",
          "status": "passed"
        },
        {
          "name": "Home CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "Home CSS Variables",
          "status": "passed"
        },
        {
          "name": "Home SVG support",
          "status": "passed"
        },
        {
          "name": "Home responsive viewport",
          "status": "passed"
        },
        {
          "name": "Home CSS calc()",
          "status": "passed"
        },
        {
          "name": "Home font rendering",
          "status": "passed"
        },
        {
          "name": "About page title",
          "status": "passed"
        },
        {
          "name": "About navigation links",
          "status": "passed"
        },
        {
```

```
      "name": "About JavaScript execution",
      "status": "passed"
    },
    {
      "name": "About CSS Grid",
      "status": "passed"
    },
    {
      "name": "About CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "About CSS Variables",
      "status": "passed"
    },
    {
      "name": "About SVG support",
      "status": "passed"
    },
    {
      "name": "About responsive viewport",
      "status": "passed"
    },
    {
      "name": "About CSS calc()",
      "status": "passed"
    },
    {
      "name": "About font rendering",
      "status": "passed"
    },
    {
      "name": "Students page title",
      "status": "passed"
    },
    {
      "name": "Students navigation links",
      "status": "passed"
    },
    {
      "name": "Students JavaScript execution",
      "status": "passed"
    },
    {
      "name": "Students CSS Grid",
      "status": "passed"
    },
    {
      "name": "Students CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "Students CSS Variables",
```

```
      "status": "passed"
    },
    {
      "name": "Students SVG support",
      "status": "passed"
    },
    {
      "name": "Students responsive viewport",
      "status": "passed"
    },
    {
      "name": "Students CSS calc()",
      "status": "passed"
    },
    {
      "name": "Students font rendering",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ page title",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ navigation links",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ JavaScript execution",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Grid",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Variables",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ SVG support",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ responsive viewport",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS calc()",
      "status": "passed"
```

```
        },
        {
          "name": "Glossary & FAQ font rendering",
          "status": "passed"
        },
        {
          "name": "License & Credits page title",
          "status": "passed"
        },
        {
          "name": "License & Credits navigation links",
          "status": "passed"
        },
        {
          "name": "License & Credits JavaScript execution",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Grid",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Variables",
          "status": "passed"
        },
        {
          "name": "License & Credits SVG support",
          "status": "passed"
        },
        {
          "name": "License & Credits responsive viewport",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS calc()",
          "status": "passed"
        },
        {
          "name": "License & Credits font rendering",
          "status": "passed"
        }
      ]
    },
    {
      "name": "webkit",
      "status": "passed",
      "error": null,
      "tests": [
        {
```

```
          "name": "Home page title",
          "status": "passed"
        },
        {
          "name": "Home navigation links",
          "status": "passed"
        },
        {
          "name": "Home JavaScript execution",
          "status": "passed"
        },
        {
          "name": "Home CSS Grid",
          "status": "passed"
        },
        {
          "name": "Home CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "Home CSS Variables",
          "status": "passed"
        },
        {
          "name": "Home SVG support",
          "status": "passed"
        },
        {
          "name": "Home responsive viewport",
          "status": "passed"
        },
        {
          "name": "Home CSS calc()",
          "status": "passed"
        },
        {
          "name": "Home font rendering",
          "status": "passed"
        },
        {
          "name": "About page title",
          "status": "passed"
        },
        {
          "name": "About navigation links",
          "status": "passed"
        },
        {
          "name": "About JavaScript execution",
          "status": "passed"
        },
        {
          "name": "About CSS Grid",
```

```
      "status": "passed"
    },
    {
      "name": "About CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "About CSS Variables",
      "status": "passed"
    },
    {
      "name": "About SVG support",
      "status": "passed"
    },
    {
      "name": "About responsive viewport",
      "status": "passed"
    },
    {
      "name": "About CSS calc()",
      "status": "passed"
    },
    {
      "name": "About font rendering",
      "status": "passed"
    },
    {
      "name": "Students page title",
      "status": "passed"
    },
    {
      "name": "Students navigation links",
      "status": "passed"
    },
    {
      "name": "Students JavaScript execution",
      "status": "passed"
    },
    {
      "name": "Students CSS Grid",
      "status": "passed"
    },
    {
      "name": "Students CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "Students CSS Variables",
      "status": "passed"
    },
    {
      "name": "Students SVG support",
      "status": "passed"
```

```
    },
    {
      "name": "Students responsive viewport",
      "status": "passed"
    },
    {
      "name": "Students CSS calc()",
      "status": "passed"
    },
    {
      "name": "Students font rendering",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ page title",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ navigation links",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ JavaScript execution",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Grid",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Flexbox",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS Variables",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ SVG support",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ responsive viewport",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ CSS calc()",
      "status": "passed"
    },
    {
      "name": "Glossary & FAQ font rendering",
      "status": "passed"
    },
```

```
        {
          "name": "License & Credits page title",
          "status": "passed"
        },
        {
          "name": "License & Credits navigation links",
          "status": "passed"
        },
        {
          "name": "License & Credits JavaScript execution",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Grid",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Flexbox",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS Variables",
          "status": "passed"
        },
        {
          "name": "License & Credits SVG support",
          "status": "passed"
        },
        {
          "name": "License & Credits responsive viewport",
          "status": "passed"
        },
        {
          "name": "License & Credits CSS calc()",
          "status": "passed"
        },
        {
          "name": "License & Credits font rendering",
          "status": "passed"
        }
      ]
    }
  ],
  "summary": {
    "passed": 3,
    "failed": 0,
    "skipped": 0,
    "total": 3
  }
}
```