1. HW3 Scope and What Was Tested

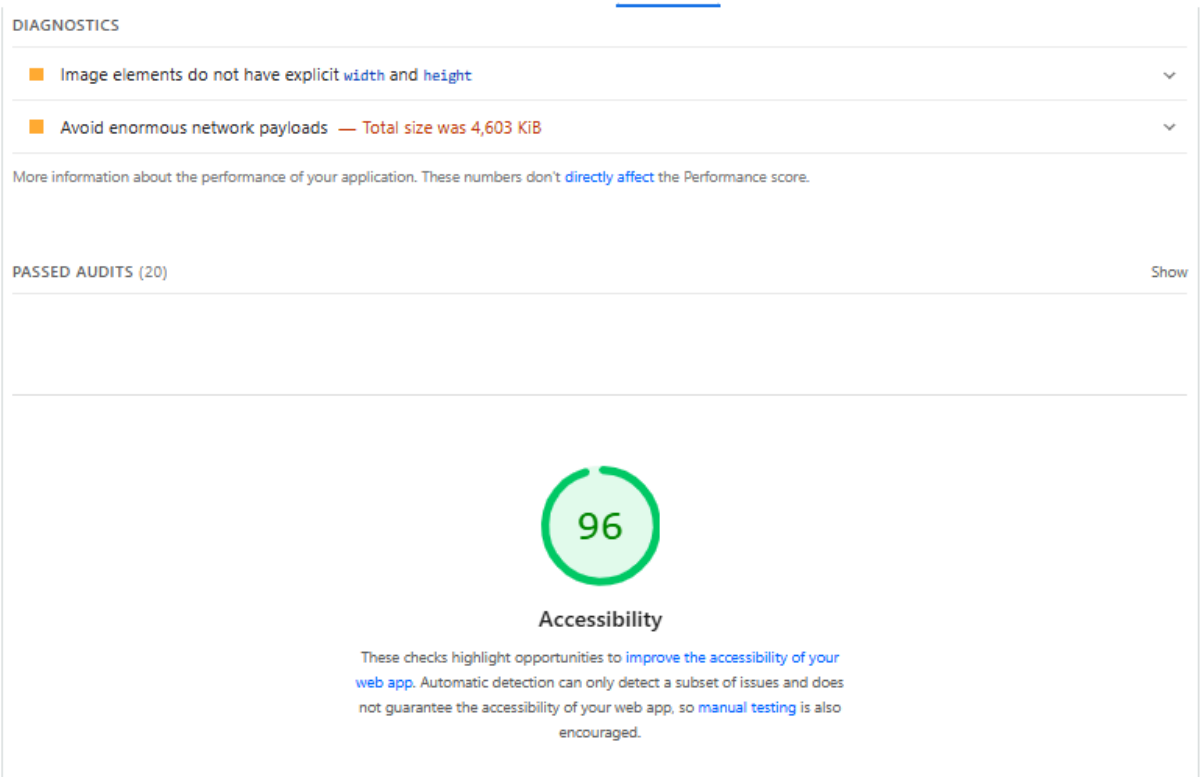
HW3 expanded the HW2 email campaign prototype by improving button design, responsive hero behavior, single-column layout, background images behind text, updated image assets, and accessibility validation using professional tools.

Pages Tested: index.html and signup.html

2. Accessibility Tools Used

Chrome DevTools Lighthouse/PageSpeed Checker (Accessibility)

Pros: Built-in, fast scan, numeric score, highlights common WCAG issues. I used PageSpeed to check by using the pages url (the full report here: https://pagespeed.web.dev/analysis/https-m-ph484-github-io-comp584-hw3MK/7aw3aa214i?form_factor=desktop)

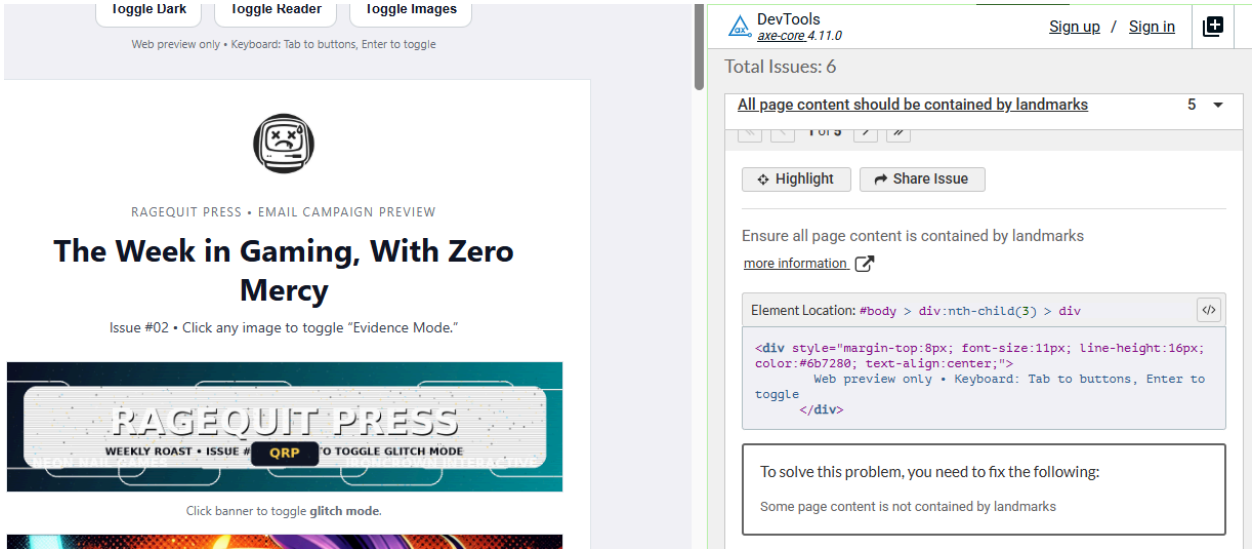


Cons: Limited contextual understanding; automated checks only.

axe DevTools

Pros: Auto-rule checks, actionable remediation guidance.

Cons: Automated; does not replace manual testing. Fixing one thing led to flags saying there were a lot more errors than before.



(Some of the recommendations that axe gave were helpful. This was one of the issues that had to be fixed)

Manual Testing

Keyboard navigation validation, focus testing, skip link verification, reader mode testing.

(sometimes glitches happened and tests had to be done to troubleshoot, shown above with the dark mode css conflict)

3. Accessibility Changes Implemented

Skip Link + Main Landmark

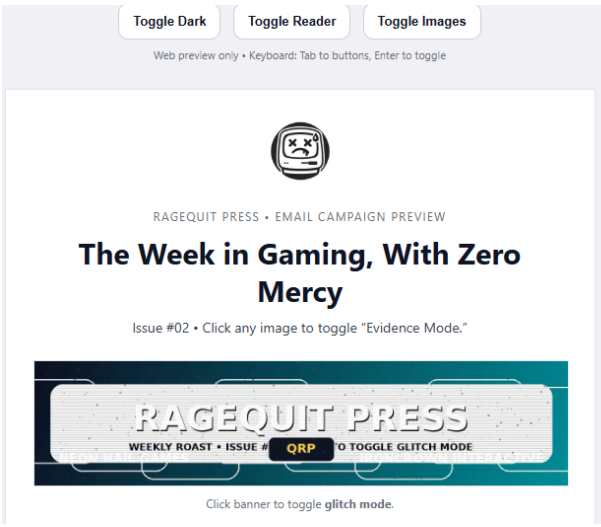
Added skip link and role='main' landmark to improve navigation for assistive technologies. Press tab to focus on main content

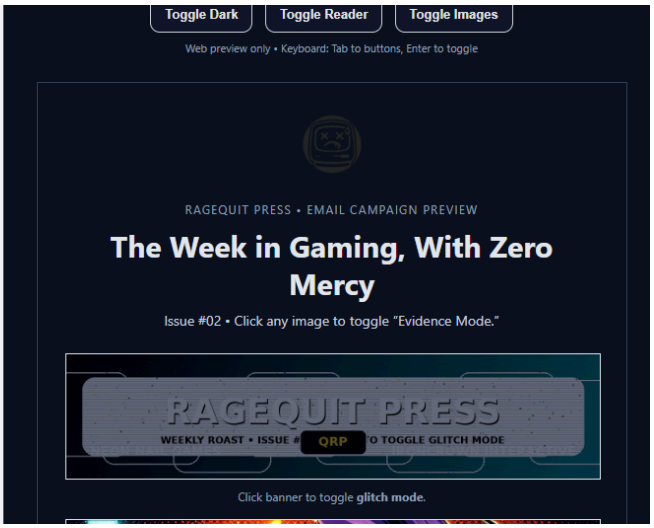
```
</head>

<body id="body" style="margin: 0 !important; padding: 0 !important; background-color: #f3f4f6;">
  <!-- Skip link improves keyboard navigation (especially for screen reader and power users) -->
  <a href="#main" style="position:absolute; left:-9999px; top:auto; width:1px; height:1px; overflow:hidden;">
    Skip to main content
  </a>
```

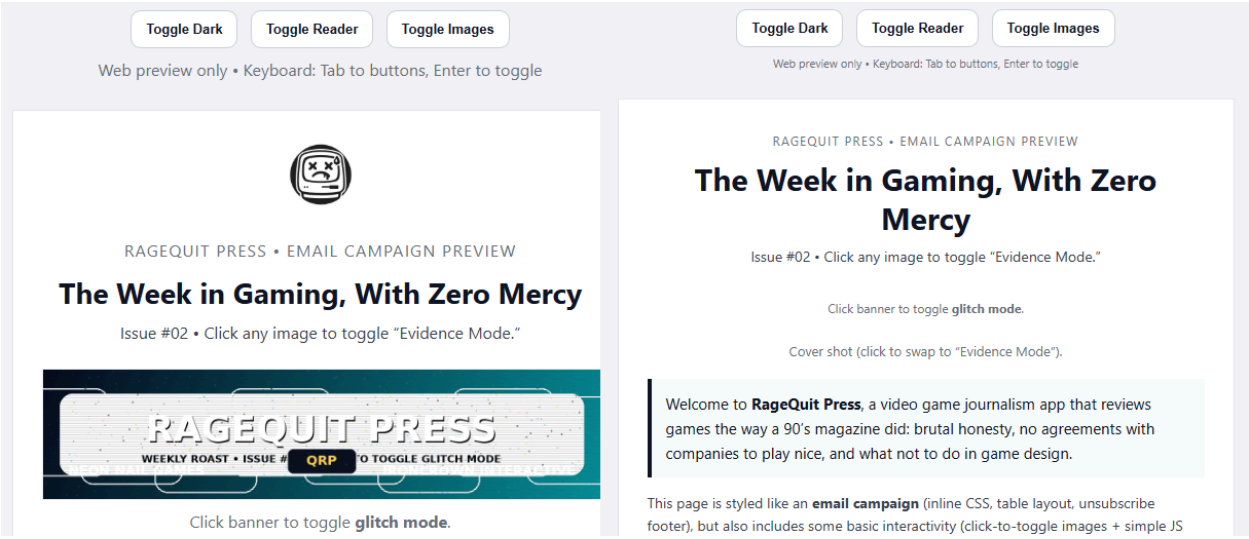
Dark Mode, Reader Mode, Skip Images Mode

User-controlled accessibility toolbar implemented. Dark mode required CSS overrides due to email-style inline backgrounds.





(Light vs Dark mode - Banner changes too)



(reader-mode on the left, no-image mode on the right)

Dark Banner Fix

Implemented JavaScript-based banner swapping because banner is an <img>, not a CSS background.

(banner changes in dark mode, too, and is mindful of keeping the 'darker' theme)

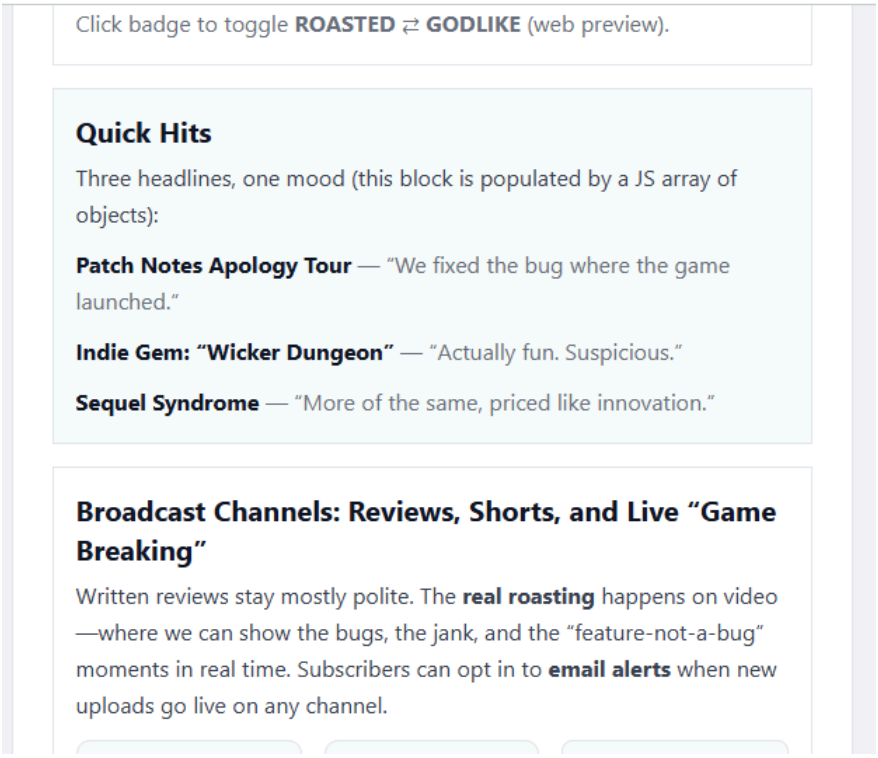
Button Accessibility Improvements

Implemented both bulletproof (table-based) and standard anchor buttons, centered with larger tap targets.

```
<!-- ========================
     PRIMARY CTA
     Purpose:
     - Clear conversion action (subscribe) in layout
     - When this is made to work for real, this will go to an external landing page
     Edit points:
     - Change href to a real route later (signup.html / React route / etc.)
     ======================== -->
<div style="margin-top:18px; text-align:center;">
  <a
    href="signup.html"
    id="ctaSubscribe"
    style="display:inline-block; padding:12px 18px; background-color:#111827; color:#ffffff; text-decoration:none; border-radius:6px; font-size:14px; line-height:14px;
    Subscribe for Roasts & Alerts
  </a>
  <div style="margin-top:10px; font-size:12px; line-height:18px; color:#6b7280;">
    (CTA now links to signup.html — a frontend-only landing page preview.)
  </div>
</div>
```

Single-Column Layout

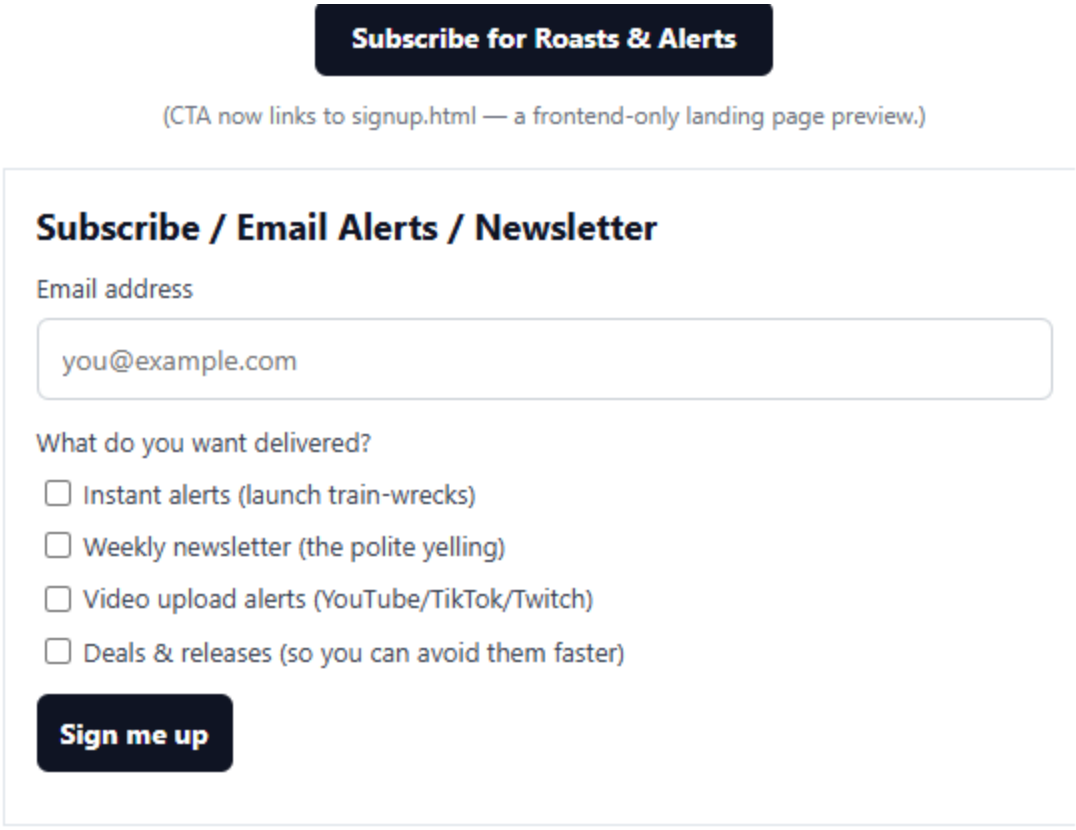
Converted feature section to stacked single-column format for reliability and accessibility.



(Before it would only turn to single-column mode if window was shrunk)

Form Accessibility (Signup Page)

Proper label-input pairing, clear CTA text, improved spacing, and validation structure.



4. Findings Summary from Tools

Contrast issues in early dark mode resolved via override rules. Heading hierarchy validated. ARIA roles and labels confirmed compliant.

5. Accessibility Pros and Cons

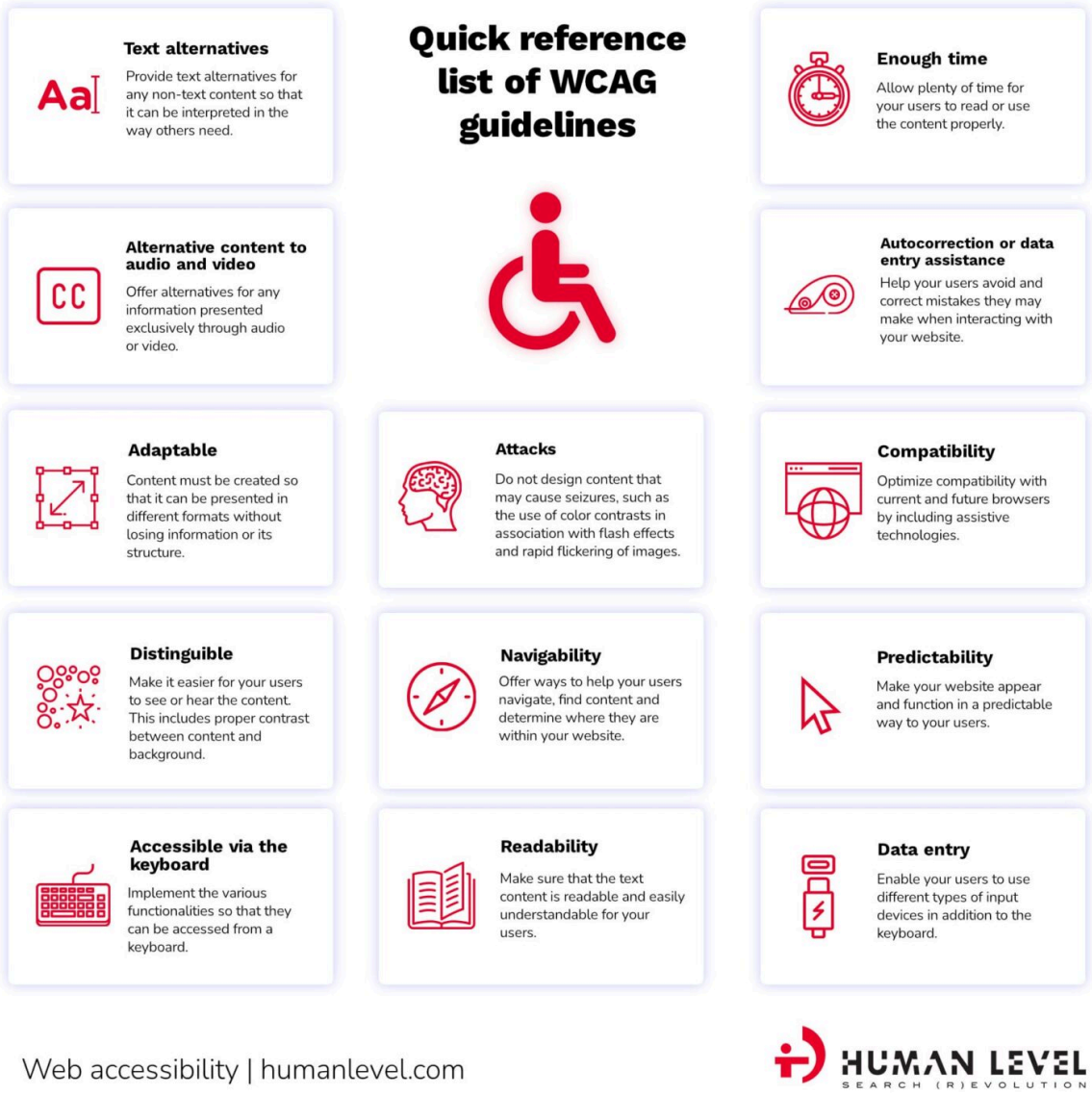
Pros: Strong keyboard support, clear landmarks, adaptive viewing modes, improved button sizing.

Cons: JavaScript-dependent features not supported in real email clients; inline CSS complicates theming.

6. Connection to Senior Design — WCAG Framework

Inclusion and Accessibility
Web **C**ontent **A**ccessibility **G**uidelines

My senior Design course covered WCAG principles: Perceivable, Operable, Understandable, Robust. These were taken into account when doing research and changing the design for HW 3 requirements. (What is Web Accessibility (WCAG) and what are its guidelines? | Human Level)



(the full document, separated from comp 491 slides screenshot)

Perceivable

Dark mode, reader mode, alt text, contrast fixes.

Operable

Keyboard support for toggles and navigation.

Understandable

CTA labels, logical heading hierarchy

Robust

Semantic HTML, ARIA roles, form label structure.

7. Influence of External Sources (ByteByteGo + Medium)

Research from ByteByteGo and Medium influenced architecture planning and future stack decisions. These resources were recommended by my senior project professor.

Why MUI Would Be Useful Later

Centralized theming, built-in accessible components, improved dark mode implementation, standardized focus behavior.

Why Django Is Relevant for Accessibility

Server-side validation, accessible error binding, secure form processing, scalable structure.

8. Conclusion

HW3 requirements were satisfied and used some extra interactivity in the campaign. Accessibility tools were applied, WCAG principles followed, and a roadmap for scalable accessibility (MUI + Django) established.