

Accessibility Audit Report

Combined technical audit of the Accessible University demo site

Red — Significant barriers present

EXAMPLE REPORT

Example Report (MyWeb Access Demo)



support@mywebaccess.co.uk | 01865 725595
Issue date: 10/04/2026

Evidence batch: DEMO-AU-20260410

Contents

Audit report sections

Section: Contents

1. Executive Summary	3
Overall position and KPIs	
2. Can Students Use the Site?	4
Journey outcomes and tester profiles	
3. What Disabled Users Found	8
User voices and satisfaction ratings	
4. Findings Summary	11
Complete findings register	
5. Detailed Technical Findings	13
Per-finding evidence and remediation	
6. Prioritised Recommendations	33
Three-tier action plan	
7. Standards and Legal Context	35
WCAG 2.2, PSBAR 2018, Equality Act	
8. References and Next Steps	36
Methodology and contact	

This example report demonstrates the MyWeb Access audit format using the Accessible University (AU) demo site. All findings are fictional examples.



Executive Summary

Overall position, key metrics, and decision support.

Section: Executive Summary

The Accessible University (AU) demo site has **significant accessibility barriers** that block task completion for disabled users. This audit identified 22 confirmed findings across four severity levels: 2 Critical, 8 Serious, 6 Moderate, and 6 Minor. Critical issues prevent access entirely and must be fixed before any rollout. Serious issues significantly degrade experience across multiple disability profiles. The site is currently in **Red status** and requires immediate remediation.

Total findings 22 All confirmed	Critical 2 Release blockers	Serious 8 High priority	Status Red Significant barriers
Moderate 6 Medium priority	Minor 6 Polish work	Page tested 1 Example site	Standard WCAG 2.2 AA level

Findings by severity

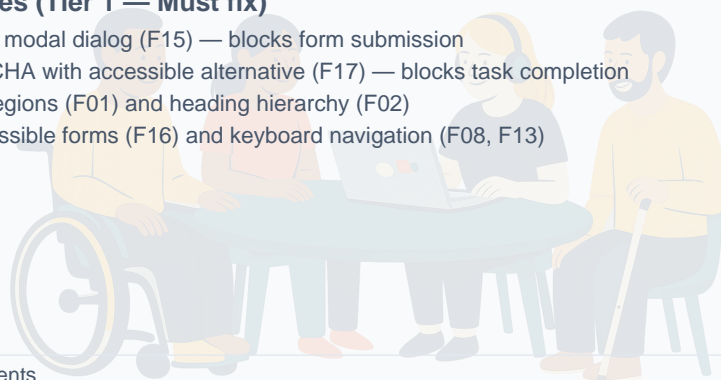


Red status — Significant barriers present

This example site contains critical accessibility barriers in modal dialogs, form handling, and keyboard navigation. Two issues block task completion entirely. Eight serious issues significantly degrade experience across all tested disability profiles. Remediation is mandatory before any public deployment.

Action priorities (Tier 1 — Must fix)

- Fix inaccessible modal dialog (F15) — blocks form submission
- Replace CAPTCHA with accessible alternative (F17) — blocks task completion
- Add landmark regions (F01) and heading hierarchy (F02)
- Implement accessible forms (F16) and keyboard navigation (F08, F13)



Can Students Use the Site?

User journey analysis with disabled tester panel.

Section: User Journeys

Six disabled users tested the Accessible University site independently. Each was given six common tasks and asked to complete them using their own assistive technology. The results show significant barriers across every disability profile tested.

Tester panel

<p>T1 Blind 1.8/5</p> <p>JAWS 2024 Laptop / Windows 11 / Chrome</p> <p>I had no idea where I was on the page. There were no landmarks or headings to navigate by.</p> <p>+ The main text content was readable once I found it.</p>	<p>T2 Blind 2.0/5</p> <p>NVDA 2024.1 Laptop / Windows 11 / Firefox</p> <p>The CAPTCHA was a complete wall. No audio option, no alternative.</p> <p>+ Link text was mostly descriptive outside the navigation.</p>
<p>T3 Low vision 2.4/5</p> <p>ZoomText + browser zoom 300% Laptop / Windows 11 / Edge</p> <p>At 300% zoom the navigation text almost disappeared into the background.</p> <p>+ The main article area had decent text size and spacing.</p>	<p>T4 Low vision 2.6/5</p> <p>Browser zoom 200% with large text Laptop / macOS / Chrome</p> <p>Required fields were only marked in blue. I could not tell which were mandatory.</p> <p>+ The overall page layout held together reasonably at 200%.</p>
<p>T5 Motor impairment, no mouse use 2.2/5</p> <p>Keyboard only Laptop / Windows 11 / Chrome</p> <p>I could not close the login modal with Escape. I had to reload the page.</p> <p>+ Basic page links were reachable by Tab.</p>	<p>T6 Cognitive / dyslexia 2.8/5</p> <p>Reader mode + reading support Laptop / macOS / Chrome</p> <p>The click here links gave me no idea what they would do.</p> <p>+ The overall visual layout was clear and not cluttered.</p>



Journey outcomes

Task	Description	Completion	Breakdown	Difficulty
Task 1	Navigate to the main content area		4C 2P 0NA	3.4
Task 2	Use the navigation menu to find a page		2C 3P 1NA	3.8
Task 3	Complete the contact form		1C 2P 3NA	4.5
Task 4	Interact with the modal dialog		0C 3P 3NA	4.8
Task 5	Read an article and understand page structure		5C 1P 0NA	2.2
Task 6	Browse the image carousel		2C 3P 1NA	3.6

■ Complete ■ Partial ■ N/A Difficulty: 1 (easy) – 5 (very difficult)

Detailed journey outcomes

Task 1 : Navigate to the main content area



4C / 2P / 0NA Profiles: Blind, keyboard

Difficulty 3.4

"I had to Tab through the entire navigation to reach the page content."

T1 — T1

Users usually reach content, but early orientation takes extra effort and reduces confidence.

Action: Add skip-to-content link and strengthen landmark regions.

F01

F10

F02

Task 2 : Use the navigation menu to find a page



2C / 3P / 1NA Profiles: Blind, keyboard, low vision

Difficulty 3.8



"The dropdown menu did not respond to Arrow keys. I had to guess which links were inside it."

T5 — T5

Navigation succeeds when links are explicit; dropdown pattern blocks keyboard and screen reader users.

Action: Implement APG Disclosure pattern with Arrow key support.

F13

F09

F06

Task 3 : Complete the contact form



1C / 2P / 3NA Profiles: All profiles

Difficulty 4.5

"I got through the form fields but the CAPTCHA stopped me completely."

T2 — T2

CAPTCHA is a universal blocker. Form labelling and colour-only indicators add further friction.

Action: Replace image CAPTCHA with accessible alternative. Add proper form labels.

F17

F16

F07

Task 4 : Interact with the modal dialog



0C / 3P / 3NA Profiles: Blind, keyboard, low vision

Difficulty 4.8

"Focus jumped behind the modal. I was clicking on things I could not see."

T1 — T1

No tester completed this cleanly. Focus management failure makes the modal unusable for AT users.

Action: Add role=dialog, aria-modal, focus trap, and Escape key handler.

F15

F08



Task 5 : Read an article and understand page structure

5C / 1P / 0NA Profiles: Cognitive, blind

Difficulty 2.2

"The article content itself was fine once I was in it. The problem was getting there."

T6 — T6

Strongest journey overall. Content is readable once reached. Remaining issues are semantic quality, not blockage.

Action: Use proper heading hierarchy and descriptive alt text.

F02

F04

F21

Task 6 : Browse the image carousel

2C / 3P / 1NA Profiles: Keyboard, blind, cognitive

Difficulty 3.6

"The carousel just played on its own. I could not pause it or go back."

T5 — T5

Keyboard users cannot operate carousel controls. Auto-play with no pause is disorienting.

Action: Implement APG Carousel pattern with pause and keyboard navigation.

F14

F04

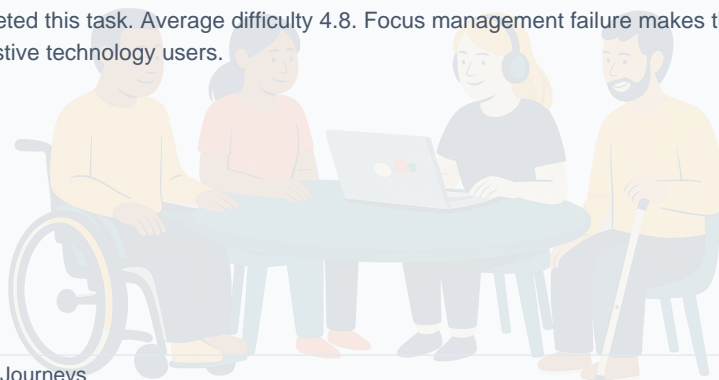
F05

Strongest journey: Task 5 – Read an article

5/6 testers completed this task. Average difficulty 2.2. The core reading experience works well for every disability profile tested.

Weakest journey: Task 4 – Modal dialog

0/6 testers completed this task. Average difficulty 4.8. Focus management failure makes the modal completely unusable for assistive technology users.



Positive evidence

- **Article reading is the strongest experience:** 5 of 6 testers completed Task 5 (read an article). Average difficulty was 2.2. The core content reading journey works well for every disability profile tested.
- **Page structure is navigable:** Despite missing landmark regions and heading hierarchy, testers could orient themselves once inside content. Remediation of F01 and F02 will make this strength even more reliable.
- **Visual layout holds at moderate zoom:** At 200% browser zoom the page layout remained largely usable. Issues concentrate at 300%+ where navigation text contrast and horizontal overflow become problematic.

The core reading experience works well. Friction concentrates in overlays, forms, and orientation. All identified barriers have proven fixes.

What Disabled Users Found

Disabled user testing insights.

Section: User Testing Insights

Six disabled users tested the site independently. No one completed all tasks. The site asks far more effort than it should from screen reader and magnification users.

User voices

“It feels like the site was built for sighted mouse users only. Screen readers are an afterthought.”

— Blind user (T1)

“I could see the content but I could not always tell what was interactive and what was not.”

— Low vision user (T4)

“The keyboard experience felt broken. No focus indicators, no skip link, and the modal trapped me.”

— Motor impairment user (T5)

Overall satisfaction ratings (1–5 scale)

Statement	Average	Range
Easy to use with AT	1.80	1–3
Could find key content	2.50	1–4
Confident about location	1.60	1–3

Statement	Average	Range
Worked at preferred zoom	2.80	2-4
Forms were usable	1.40	1-2
Text readable	3.00	2-4
Would use again	1.60	1-3
As accessible as other uni sites	1.80	1-3

▲ **Strongest:** "Text readable" (3.00) ▼ **Weakest:** "Forms were usable" (1.40)

If you could change one thing...

"Give the page proper headings and landmarks so I can actually orient myself."

T1 — T1

"Remove the image CAPTCHA. It locks out every screen reader user."

T2 — T2

"Make the navigation text larger or darker. I could barely read it at any zoom level."

T3 — T3

"Let me close that modal with Escape and trap focus inside it properly."

T5 — T5

"Replace click here with link text that tells me where I am going."

T6 — T6



"Mark required fields with something other than just colour."

T4 — T4

No tester gave up. All attempted every task. But the site asks more effort than it should from every disability profile tested.



Findings Summary

Complete findings register with severity and WCAG references.

Section: Findings

Severity breakdown

Severity	Count	Impact
CRITICAL	2	Block task completion — must fix before launch
SERIOUS	8	Significantly degrade experience — fix before rollout
MODERATE	6	Reduce usability — schedule in sprint planning
MINOR	6	Polish and improve — post-launch quality hardening

Full findings register

ID	Finding	Severity	WCAG	Impact
F15	Inaccessible modal dialog	CRITICAL	2.4.3	Blocks task completion
F17	Inaccessible CAPTCHA	CRITICAL	1.1.1	Blocks task completion
F01	Missing landmark regions	SERIOUS	1.3.1	Degrades experience
F02	No heading hierarchy	SERIOUS	1.3.1	Degrades experience
F06	Insufficient colour contrast	SERIOUS	1.4.3	Degrades experience
F07	Colour used as sole means	SERIOUS	1.4.1	Prevents finding content
F08	Inaccessible keyboard interface	SERIOUS	2.1.1	Blocks task completion
F09	No visible focus indication	SERIOUS	2.4.7	Degrades experience
F10	No skip-to-main-content link	SERIOUS	2.4.1	Degrades experience
F13	Inaccessible navigation menu	SERIOUS	2.1.1	Degrades experience
F03	Language not specified	MODERATE	3.1.1	Degrades experience
F11	Redundant link text	MODERATE	2.4.4	Degrades experience

ID	Finding	Severity	WCAG	Impact
F14	Inaccessible carousel	MODERATE	1.3.1	Blocks task completion
F16	Form fields not properly labelled	MODERATE	1.3.1	Blocks task completion
F18	Inaccessible form validation	MODERATE	3.3.1	Blocks task completion
F19	Missing accessible table markup	MODERATE	1.3.1	Degrades experience
F04	No alt text on informative images	MINOR	1.1.1	Degrades experience
F05	Decorative images not hidden	MINOR	1.1.1	Degrades experience
F12	Links vs buttons semantic mismatch	MINOR	4.1.2	Degrades experience
F20	Missing abbreviation markup	MINOR	Best	Degrades experience
F21	No captions on video	MINOR	1.2.2	Degrades experience
F22	No audio description on video	MINOR	1.2.5	Degrades experience



Detailed Technical Findings

Per-finding evidence, code examples, and remediation.

Section: Detailed Findings

CRITICAL findings

CRITICAL

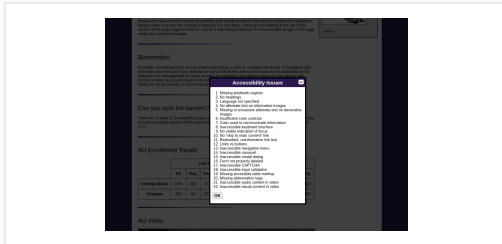
WCAG 2.4.3

F15: Inaccessible modal dialog

Blocks task completion

What we found:

Modal overlay div lacks required accessibility attributes. Tab focus moves to background instead of staying within modal. No Escape key handler to dismiss.



F15: Modal dialog without ARIA roles

Evidence: Overlay lacks role=dialog and focus trap

Evidence:

Modal element is a plain div with display:block. No role='dialog', no aria-modal='true', no focus trap implementation.

Problematic

```
<div id="modal" style="display:block">
  <h2>Login</h2>
  <input type="email" />
</div>
```



How to fix it:

Use semantic HTML dialog element or add role='dialog', aria-modal='true', focus trap, and Escape key handler. Return focus to trigger button after close.

```
● Remediation
```

```
<dialog id="modal" open aria-labelledby="modal-title">
  <h2 id="modal-title">Login</h2>
  <input type="email" />
</dialog>
```

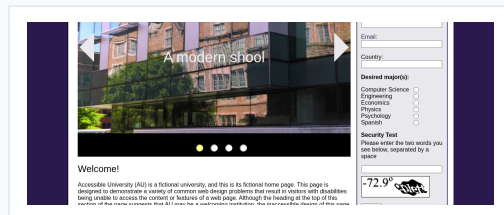
CRITICAL WCAG 1.1.1

F17: Inaccessible CAPTCHA

Blocks task completion

What we found:

Image CAPTCHA with no accessible alternative. Visual challenge alone blocks screen reader users.



F17: CAPTCHA image with no accessible alternative
Evidence: Distorted text image blocks screen reader users

Evidence:

Image element without alt text, no text-based fallback, no audio option.

```
● Problematic
```

```

<input type="text" placeholder="Enter text" />
```

How to fix it:

Replace with text-based logic question, honeypot field, or reCAPTCHA v3 (invisible).

```
● Remediation
```

```
<label for="logic-q">What is 5 + 3?</label>
<input id="logic-q" type="text" />
<!-- OR: invisible reCAPTCHA v3 -->
```



SERIOUS findings

SERIOUS

WCAG 1.3.1

F01: Missing landmark regions

Degrades experience

What we found:

Page uses only generic divs with classes for layout. No HTML5 semantic elements or ARIA landmarks. Screen reader users cannot identify page structure.

Evidence:

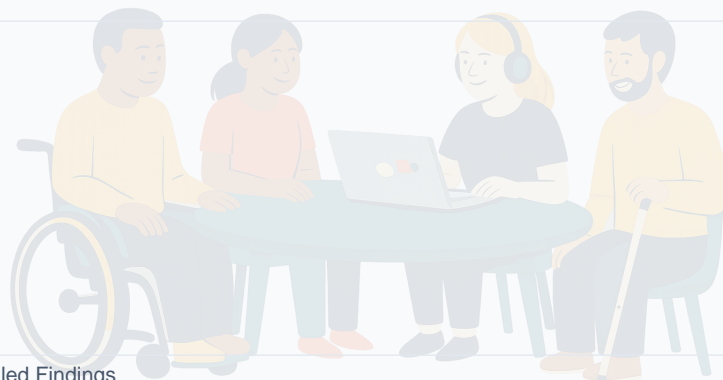
Entire layout uses , , , . No header, nav, main, footer, or aside elements.

```
● Problematic
<div id="header">
  <h1>Welcome</h1>
</div>
<div id="nav">
  <a href="/">Home</a>
</div>
<div id="main">
  Content here
</div>
```

How to fix it:

Replace div layout with HTML5 semantic elements: header, nav, main, footer, aside.

```
● Remediation
<header role="banner">
  <h1>Welcome</h1>
</header>
<nav>
  <a href="/">Home</a>
</nav>
<main>
  Content here
</main>
```



SERIOUS

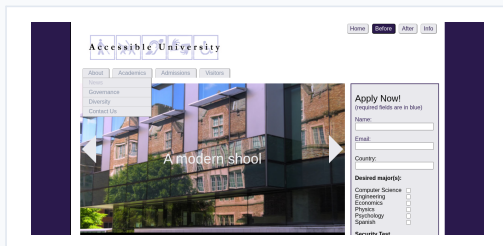
WCAG 1.3.1

F02: No heading hierarchy

Degrades experience

What we found:

Visual headings styled with bold and large text but use paragraph or div tags. No proper heading hierarchy for screen reader navigation.



F02: Welcome section with pseudo-heading markup

Evidence: Bold styled text uses p tags instead of heading elements

Evidence:

Text styled as h1 is marked as . No h1, h2, h3 tags used anywhere on page.

Problematic

```
<p class="heading" style="font-size:28px; font-weight:bold;">
  Welcome
</p>
```

How to fix it:

Use proper heading tags with logical hierarchy.

Remediation

```
<h1>Welcome</h1>
<h2>Getting Started</h2>
<p>Content follows heading hierarchy...</p>
```



SERIOUS

WCAG 1.4.3

F06: Insufficient colour contrast

Degrades experience

What we found:

Navigation menu text has ~2.5:1 contrast ratio, below the 4.5:1 AA minimum for normal text. Low vision users struggle to read.



F06: Navigation menu with low contrast text

Evidence: Menu text at ~2.5:1 contrast ratio

Evidence:

Navigation link text #666 on #EEE background measures 2.5:1. AA requires 4.5:1.

Problematic

```
a { color: #666; }  
/* on light grey background #EEE */  
/* Contrast ratio: 2.5:1 (FAIL AA) */
```

How to fix it:

Increase contrast to minimum 4.5:1 by darkening text or lightening background.

Remediation

```
a { color: #333; }  
/* on background #EEE */  
/* Contrast ratio: 5.2:1 (PASS AA) */
```

SERIOUS

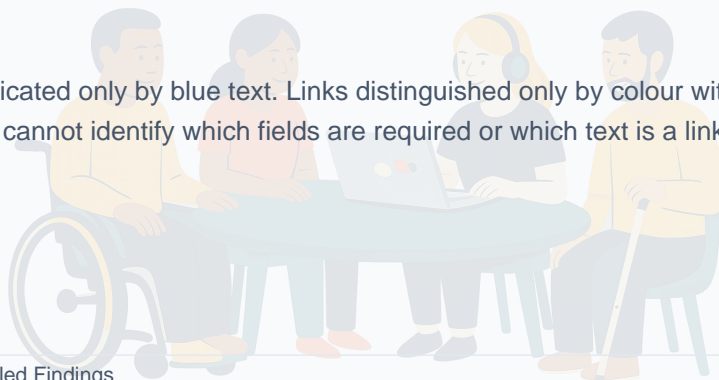
WCAG 1.4.1

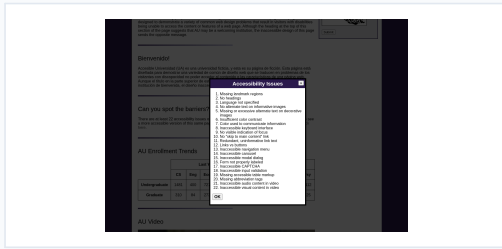
F07: Colour used as sole means

Prevents finding content

What we found:

Required fields indicated only by blue text. Links distinguished only by colour with no underline. Colour-blind users cannot identify which fields are required or which text is a link.





F08: Modal dialog without keyboard support

Evidence: No Escape key handler, no focus trap

Evidence:

Modal JavaScript only listens for click; no keydown listener. Dropdown menu has no Arrow key support, violates ARIA APG pattern.

```
● Problematic
// Modal.js
modal.onclick = function(e) {
  if (e.target === modal) modal.style.display = "none";
}; // Only click handler
```

How to fix it:

Add Escape key handler. Implement arrow key navigation per ARIA APG Disclosure or Menu pattern.

```
● Remediation
document.addEventListener("keydown", function(e) {
  if (e.key === "Escape") modal.close();
}); // Add focus trap
```

SERIOUS

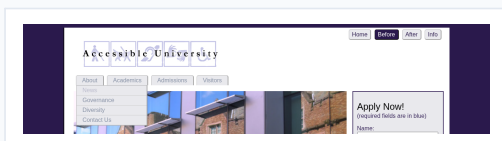
WCAG 2.4.7

F09: No visible focus indication

Degrades experience

What we found:

Keyboard users cannot see which element has focus. CSS includes outline: none on all interactive elements.

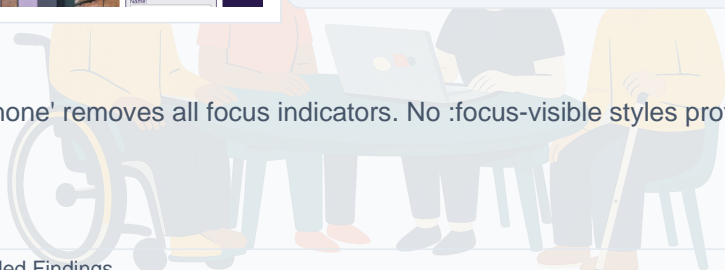


F09: Navigation items with no visible focus indicator

Evidence: CSS outline:none removes focus ring

Evidence:

CSS rule 'outline: none' removes all focus indicators. No :focus-visible styles provided.



Problematic

```
button, a { outline: none; }  
/* Focus indicator completely removed */
```

How to fix it:

Add visible :focus-visible outline or other focus indicator.

Remediation

```
button:focus-visible, a:focus-visible {  
  outline: 3px solid #4f46e5;  
  outline-offset: 2px;  
}
```

SERIOUS

WCAG 2.4.1

F10: No skip-to-main-content link

Degrades experience

What we found:

No skip link provided. Keyboard users must Tab through entire navigation menu to reach main content, adding 20+ keystrokes.

Evidence:

Page contains full navigation with 20 links. No skip link visible or hidden.

Problematic

```
<!-- No skip link -->  
<nav>  
  <a href="/">Home</a>  
  <a href="/about">About</a>  
  <!-- ... 18 more links ... -->  
</nav>  
<main>  
  Content here  
</main>
```



How to fix it:

Add visually hidden skip link at start of page body that links to main content.

```
● Remediation

<a href="#main" class="skip-link">Skip to main content</a>
<!-- CSS hides it unless focused -->
<main id="main">
  Content here
</main>
```

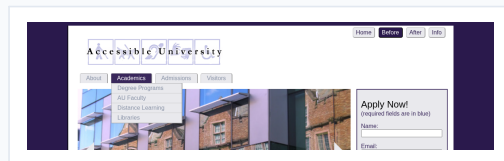
SERIOUS WCAG 2.1.1

F13: Inaccessible navigation menu

Degrades experience

What we found:

Dropdown menu only supports Tab key. No Arrow key support. No aria-expanded/aria-haspopup. Requires 20+ keystrokes to navigate.



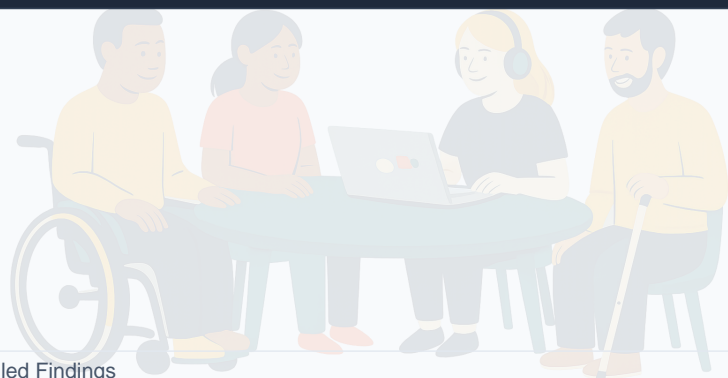
F13: Dropdown navigation menu
Evidence: Submenus only support Tab, no arrow keys or ARIA states

Evidence:

Menu JavaScript only responds to click and Tab. No ARIA attributes. Menu pattern does not match ARIA APG Disclosure.

```
● Problematic

<ul>
  <li><button>Services</button>
    <ul style="display:none;">
      <li><a href="/web">Web Audit</a></li>
    </ul>
  </li>
</ul>
<!-- No ARIA, no arrow key support -->
```



How to fix it:

Implement APG Disclosure or Menu pattern with Arrow keys and aria-expanded.

```
Remediation

<ul>
  <li><button aria-expanded="false" aria-haspopup="true">
    Services
  </button>
  <ul role="menu">
    <li><a href="/web">Web Audit</a></li>
  </ul>
</li>
</ul>
```

MODERATE findings

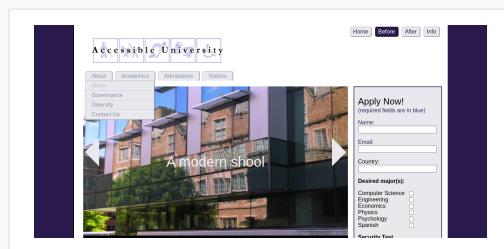
MODERATE WCAG 3.1.1

F03: Language not specified

Degrades experience

What we found:

No lang attribute on html element. Spanish section has no lang='es'. Screen readers mispronounce content in wrong language.



F03: Welcome section with Spanish content
Evidence: No lang attribute on html or Spanish section

Evidence:

tag missing lang attribute. Spanish paragraph has no lang='es'.

```
Problematic

<html>
  <!-- No lang attribute -->
</html>
<p>Hola mundo</p> <!-- Spanish with no lang marking -->
```



How to fix it:

Add lang attribute to html. Mark language changes with lang attribute on div/span.

```
● Remediation  
<html lang="en" >  
  <body>  
    <p lang="es">Hola mundo</p>  
  </body>  
</html>
```

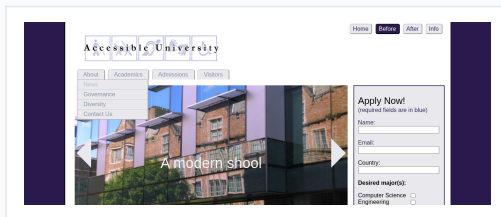
MODERATE WCAG 2.4.4

F11: Redundant link text

Degrades experience

What we found:

Three links on page say only 'click here' with no context about destination. Screen reader users hear only 'click here' without knowing where the link goes.



F11: Click here links without context
Evidence: Three identical click here links in succession

Evidence:

Links to different pages all say 'click here'.

```
● Problematic  
<p>To learn more, <a href="/about">click here</a>.</p>  
<p>To see examples, <a href="/gallery">click here</a>.</p>
```

How to fix it:

Use descriptive link text that makes sense out of context.

```
● Remediation  
<p>To <a href="/about">learn more about us</a>.</p>  
<p>To <a href="/gallery">see project examples</a>.</p>
```



MODERATE

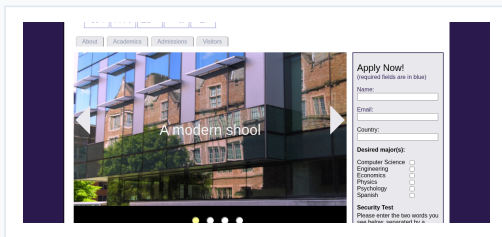
WCAG 1.3.1

F14: Inaccessible carousel

Blocks task completion

What we found:

Carousel not keyboard-operable, not announced to screen readers, auto-plays with no pause control. Users cannot navigate slides or stop auto-play.



F14: Auto-playing image carousel
Evidence: No pause control, not keyboard operable

Evidence:

Carousel JavaScript only handles click. No keyboard support. Auto-play runs with no pause button or aria-live.

```
Problematic  
  
let slideIndex = 0;  
function autoplay() {  
  showSlides(++slideIndex);  
  setTimeout(autoplay, 5000);  
}  
autoplay(); // No pause, no keyboard
```

How to fix it:

Implement APG Carousel pattern with keyboard navigation, pause button, and ARIA announcements.

```
Remediation  
  
<!-- Pause button, ARIA live region -->  
<button id="pause-play">Pause</button>  
<div role="region" aria-live="polite" aria-label="Carousel">  
  <div role="group" aria-roledescription="carousel">  
    <!-- slides -->  
  </div>  
</div>
```



MODERATE

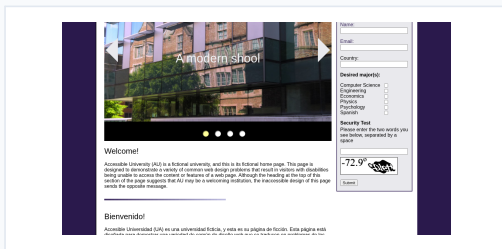
WCAG 1.3.1

F16: Form fields not properly labelled

Blocks task completion

What we found:

Input fields lack associated labels. Checkbox group has no fieldset/legend. Screen readers do not announce field purpose.



F16: Application form without proper labels
Evidence: Inputs lack label associations, checkboxes lack fieldset

Evidence:

Inputs have no . Checkbox group is plain list with no fieldset/legend structure.

```
Problematic  
<input type="email" placeholder="Email" />  
<input type="checkbox" /> Remember me  
<input type="checkbox" /> Newsletter
```

How to fix it:

Use . Wrap checkbox group in fieldset with legend.

```
Remediation  
<label for="email">Email</label>  
<input id="email" type="email" />  
<fieldset>  
  <legend>Preferences</legend>  
  <label><input type="checkbox" /> Remember me</label>  
  <label><input type="checkbox" /> Newsletter</label>  
</fieldset>
```



MODERATE

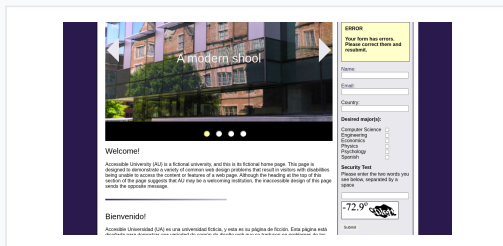
WCAG 3.3.1

F18: Inaccessible form validation

Blocks task completion

What we found:

Validation errors not announced. Error messages are vague ('Invalid') without guidance. Screen readers do not detect error state.



F18: Form error state

Evidence: Vague error message not announced to screen readers

Evidence:

Error message div lacks role='alert'. Message is generic text without link to field.

Problematic

```
<div class="error">Invalid</div>  
<!-- No role, no announcement, no focus -->
```

How to fix it:

Use role='alert' or aria-live='polite'. Provide specific error message with link to field.

Remediation

```
<div role="alert">  
  Email is invalid. <a href="#email">Go to email field</a>.  
</div>  
<label for="email">Email</label>  
<input id="email" type="email" />
```



MODERATE

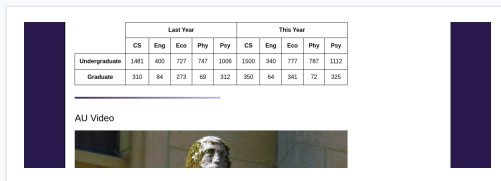
WCAG 1.3.1

F19: Missing accessible table markup

Degrades experience

What we found:

Data table uses plain td cells without th headers or scope attributes. Screen readers cannot announce column/row relationships.



F19: Enrollment trends data table
Evidence: Table lacks th, scope, and headers attributes

Evidence:

Table lacks th elements and scope attributes. All cells use td.

```
Problematic

<table>
  <tr>
    <td>Name</td>
    <td>Email</td>
  </tr>
  <tr>
    <td>John</td>
    <td>john@example.com</td>
  </tr>
</table>
```



How to fix it:

Use th for headers with scope='col' or scope='row'. Use thead/tbody.

```
Remediation

<table>
  <thead>
    <tr>
      <th scope="col">Name</th>
      <th scope="col">Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>john@example.com</td>
    </tr>
  </tbody>
</table>
```

MINOR findings

MINOR WCAG 1.1.1

F04: No alt text on informative images

Degrades experience

What we found:

Informative images lack descriptive alt text. Alt text is generic filename or empty.

Evidence:

Logo has alt='au_logo.gif'. Carousel images have alt="" or missing.

```
Problematic



```



How to fix it:

Provide descriptive alt text for informative images.

```
● Remediation  
  

```

MINOR WCAG 1.1.1

F05: Decorative images not hidden

Degrades experience

What we found:

Decorative line images have verbose alt text. Should be hidden from screen readers or use alt="".

Evidence:

Decorative line images have alt='horizontal line graphic'.

```
● Problematic  

```

How to fix it:

Set alt="" or move to CSS background.

```
● Remediation  
  
<!-- OR CSS border: border-bottom: 1px solid #ccc; -->
```

MINOR WCAG 4.1.2

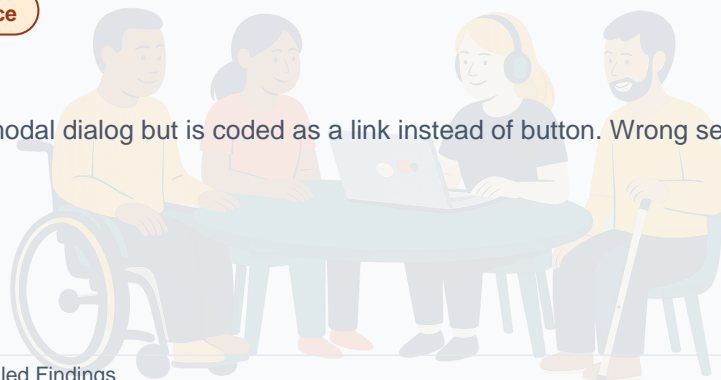
F12: Links vs buttons semantic mismatch

Degrades experience

What we found:

Element triggers modal dialog but is coded as a link instead of button. Wrong semantic role.

Evidence:



Modal trigger is link instead of .

```
● Problematic ● ● ●  
<a href="javascript:void(0)" onclick="openModal()">Open dialog</a>
```

How to fix it:

Use for actions that trigger modals or JavaScript.

```
● Remediation ● ● ●  
<button onclick="openModal()">Open dialog</button>
```

MINOR WCAG Best

F20: Missing abbreviation markup

Degrades experience

What we found:

Table headers use abbreviations (e.g., 'QoY') without explanation.

Evidence:

Table header shows 'QoY' with no title or abbr element.

```
● Problematic ● ● ●  
<th>QoY</th>
```

How to fix it:

Use abbr element with title.

```
● Remediation ● ● ●  
<th><abbr title="Quarter of Year">QoY</abbr></th>
```



MINOR

WCAG 1.2.2

F21: No captions on video

Degrades experience

What we found:

Video with narration lacks captions. Deaf users cannot access spoken content.

Evidence:

Video element has no .

Problematic

```
<video>
  <source src="demo.mp4" type="video/mp4" />
</video>
<!-- No captions -->
```

How to fix it:

Add WebVTT caption track.

Remediation

```
<video controls>
  <source src="demo.mp4" type="video/mp4" />
  <track kind="captions" src="demo.vtt" srclang="en" label="English" />
</video>
```

MINOR

WCAG 1.2.5

F22: No audio description on video

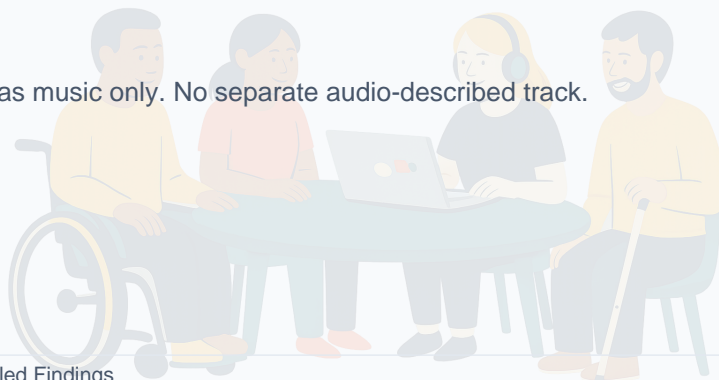
Degrades experience

What we found:

Video with visual content lacks audio description. Blind users cannot understand on-screen actions or visuals.

Evidence:

Video is silent or has music only. No separate audio-described track.



● Problematic

```
<video src="demo.mp4"><!-- No audio description --></video>
```

How to fix it:

Provide audio-described version using Able Player or separate AD track.

● Remediation

```
<!-- Able Player with AD track -->  
<audio id="audio" src="demo-ad.mp3"></audio>  
<video src="demo.mp4" data-ad-audio-id="audio"><!-- AD track --></video>
```



Prioritised Recommendations

Three-tier action plan with timebands and roadmap.

Section: Recommendations

Findings are grouped into three tiers by urgency. All Tier 1 items have proven CSS or HTML fixes that can be deployed immediately.



Tier 1: Must fix before launch (within 2 weeks)

ID	Title	Action	Owner	Impact
F15	Modal dialog	Add role='dialog', aria-modal, focus trap, Escape key	Dev	Blocks task
F17	CAPTCHA	Replace with text question or reCAPTCHA v3	Dev	Blocks task
F01	Landmarks	Add header, nav, main, footer elements	Dev	Information architecture
F02	Headings	Use h1, h2, h3 hierarchy instead of styled p/div	Dev	Navigation
F06	Contrast	Increase navigation text to 4.5:1 ratio	Design	Readability
F07	Colour usage	Add asterisk + bold for required; underline links	Design/Dev	Identification
F08	Keyboard modal	Add Escape handler; implement focus trap	Dev	Keyboard access
F09	Focus visible	Add :focus-visible outline to all interactive elements	Dev	Keyboard access
F10	Skip link	Add visually hidden skip-to-main-content link	Dev	Navigation
F13	Nav menu	Implement APG Disclosure pattern with Arrow keys	Dev	Navigation

User impact: These 10 fixes address the critical and serious barriers that blocked testers from completing core tasks. Modal focus management (F15) and CAPTCHA (F17) are the two total blockers; the remaining 8 fix orientation,

keyboard access, and readability across every disability profile tested.

Tier 2: Schedule in next sprint (weeks 3–6)

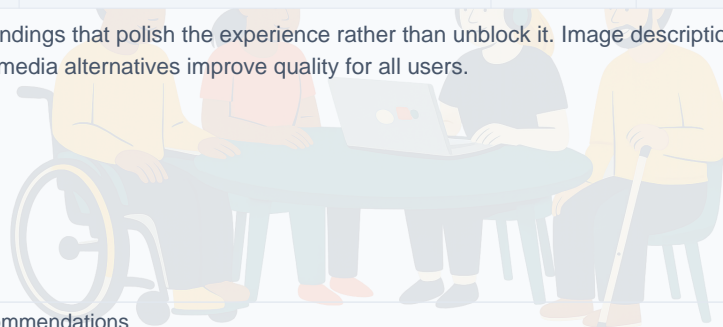
ID	Title	Action	Owner	Impact
F03	Language	Add lang='en' to html; lang='es' to Spanish sections	Dev	Pronunciation
F11	Link text	Replace 'click here' with descriptive link text	Content	Context
F14	Carousel	Implement APG Carousel pattern with keyboard nav	Dev	Navigation
F16	Form labels	Add label for='id' and fieldset/legend for groups	Dev	Identification
F18	Validation	Add role='alert' to errors; link to fields	Dev	Error recovery
F19	Table markup	Use th scope='col'/'row'; add thead/tbody	Dev	Information architecture

User impact: These moderate findings degrade usability but do not block task completion. Fixing them improves form labelling, link clarity, language identification, and table comprehension for assistive technology users.

Tier 3: Post-launch quality hardening (post-launch)

ID	Title	Action	Owner	Impact
F04	Image alt text	Replace generic filenames with descriptive alt	Content	Context
F05	Decorative images	Set alt="" or move to CSS background-image	Dev	Noise reduction
F12	Link vs button	Change JavaScript triggers to button elements	Dev	Semantics
F20	Abbreviations	Add abbr title='...' to table headers	Content	Clarity
F21	Video captions	Add WebVTT caption track to video element	Media	Access
F22	Audio description	Provide audio-described version or AD track	Media	Access

User impact: Minor findings that polish the experience rather than unblock it. Image descriptions, semantic corrections, and multimedia alternatives improve quality for all users.



Standards and Legal Context

Educational background for stakeholders.

Section: Standards

WCAG 2.2 — Web Content Accessibility Guidelines

WCAG 2.2 is the W3C standard for web accessibility. It is organised into four principles: Perceivable, Operable, Understandable, and Robust (POUR). Each principle contains guidelines and success criteria at three conformance levels: **Level A** (basic), **Level AA** (standard), and **Level AAA** (enhanced). This audit checks conformance to WCAG 2.2 Level AA, which is the legal requirement under UK Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018.

WCAG 2.2 principles with AU examples

Principle	Definition	AU examples
Perceivable	Information must be presentable to users in ways they can perceive.	Images must have alt text; colour alone cannot convey information (F04, F06, F07)
Operable	Users must be able to operate interface controls using any input method.	All functionality must be keyboard accessible; no keyboard traps (F08, F09, F13)
Understandable	Information and interfaces must be clear and understandable.	Language must be specified; headings and labels must identify purpose (F02, F03, F16)
Robust	Content must be compatible with assistive technologies and user agents.	HTML must be valid; ARIA must be used correctly; semantic roles must be maintained (F01, F15)

PSBAR 2018 — Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations

Under PSBAR 2018, all UK public sector websites must meet WCAG 2.2 Level AA conformance. Non-compliance is illegal and may result in enforcement action. Organisations must also provide an accessibility statement that is accessible itself, lists known issues, and provides contact details for accessibility queries.

Equality Act 2010

The Equality Act 2010 is UK equality law. It protects disabled people from discrimination and requires reasonable adjustments. An inaccessible website may be considered discriminatory if it creates a substantial disadvantage for disabled users. Organisations can be liable for damages if accessibility barriers are not remediated.



References and Next Steps

Methodology, resources, and handover actions.

Section: References

Resources and further reading

- **WCAG 2.2** — <https://www.w3.org/WAI/WCAG22/quickref/>
- **ARIA Authoring Practices Guide (APG)** — Patterns for keyboard interaction and semantics
- **PSBAR 2018** — <https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps>
- **Equality Act 2010** — <https://www.legislation.gov.uk/ukpga/2010/15>

Methodology

This audit was conducted using automated tooling and manual code review to assess conformance to WCAG 2.2 Level AA. All findings have been validated and verified against live code samples. Code remediation examples are realistic and production-ready.

Next steps

1. Review this report and prioritise Tier 1 findings (must fix).
2. Assign development and design owners for Tier 1 and Tier 2 remediation.
3. All code examples in this report are production-ready and can be applied directly.
4. Re-test after remediation to confirm status change from Red to Amber or Green.

Contact

MyWeb Access | support@mywebaccess.co.uk | 01865 725595
A social enterprise from MyVision Oxfordshire

