



Website Accessibility Conformance Evaluation Report

WCAG 2.2 Level AA

Prepared for: [Client Organisation]

Report Date: [Date]

Report Version: 1.0

SAMPLE

CONFIDENTIAL

This document contains confidential information prepared exclusively for the named client. It must not be shared, published, or distributed without written consent.

Document Control

Document Title	Website Accessibility Conformance Evaluation Report
Client	[Client Organisation]
Platform Tested	[Platform Name] v[X.X]
Evaluation Standard	WCAG 2.2 Level AA (via WCAG-EM)
Auditor	Dr Alexander Hambley
Date of Testing	[DD Month YYYY] – [DD Month YYYY]
Report Date	[DD Month YYYY]
Report Version	1.0

Executive Summary

Accessible Pixels was engaged by [Client Organisation] to conduct an independent accessibility evaluation of the [Platform Name] web platform against the Web Content Accessibility Guidelines (WCAG) 2.2 at Level AA conformance.

The evaluation was conducted following the WCAG-EM (Website Accessibility Conformance Evaluation Methodology) developed by the W3C, which is the internationally recognised standard for conducting rigorous and reproducible accessibility evaluations.

Overall Conformance Result: Does Not Conform to WCAG 2.2 Level AA. The evaluation identified **3 Critical**, **5 Major**, and **2 Minor** accessibility barriers across the sampled pages and user flows.

A prioritised remediation roadmap is provided in Section 7. With targeted development effort, the identified barriers can be resolved, and the platform can achieve WCAG 2.2 AA conformance.

Findings Summary

Severity	Count	Description
Critical	3	Barriers that completely prevent access to core functionality for users with disabilities.
Major	5	Barriers that significantly impair the user experience but may have workarounds.
Minor	2	Barriers that create friction but do not prevent task completion.

Evaluation Scope & Methodology

Scope Definition

The evaluation covered the [Platform Name] web platform using the WCAG-EM methodology, including the following areas of the platform:

- Authentication and login flow
- Main dashboard and navigation
- Project creation and management
- Document upload, viewing, and collaboration
- User settings and profile management
- Search and filtering functionality
- Reporting and data export

Page Sample

In accordance with WCAG-EM, a structured sample of 12 representative pages/states was selected to cover the breadth of the application's functionality, technologies, and user flows. The sample included pages from each of the functional areas listed above, plus at least one form, one data table, one modal interaction, and one error state.

Pages and States Tested

1. example.com
2. example.com
3. example.com
4. example.com
5. example.com
6. example.com
7. example.com
8. example.com
9. example.com
10. example.com
11. example.com
12. example.com

Testing Methodology

The evaluation employed a multi-layered testing approach combining automated scanning, manual expert review, and assistive technology testing:

Method	Details
Automated Scanning	Full-site scan using axe DevTools, WAVE, Accessibility Insights, IBM Equal Access Checker, and Google Lighthouse to identify programmatically detectable issues.
Manual Expert Review	Each sampled page was manually evaluated against all 50+ WCAG 2.2 Level A and AA success criteria by a qualified accessibility specialist.

Screen Reader Testing	Full workflow testing using NVDA (Windows/Chromium), Narrator (Windows/Chromium) and VoiceOver (macOS/Safari) to evaluate real-world assistive technology compatibility.
Keyboard-Only Testing	All sampled user flows were completed using keyboard-only navigation to verify operability without a pointing device.
Visual Inspection	Colour contrast analysis (using WCAG Contrast Checker), text resize testing (200%, 400% zoom for reflow) and reduced-motion preference testing.

Testing Environment

Component	Version / Details
Browsers	Google Chrome 146, Mozilla Firefox 148, Safari 26
Operating Systems	Windows 11, macOS Tahoe 26.2
Screen Readers	NVDA 2024.1, Narrator (Windows built-in), VoiceOver (macOS built-in)
Automated Tools	axe DevTools, WAVE, Lighthouse, IBM Equal Access, Microsoft Accessibility Insights
Contrast Checker	Google Chrome development tools, WCAG Contrast Checker

Conformance Result

Based on the evaluation, the [Platform Name] web platform does not conform to WCAG 2.2 Level AA. The table below provides a summary of conformance by WCAG principle:

WCAG Principle	Conformance	Issues Found
1. Perceivable	Does Not Conform	4 issues
2. Operable	Does Not Conform	3 issues
3. Understandable	Partially Conforms	1 issue
4. Robust	Does Not Conform	2 issues

Detailed Findings

Each finding below is documented with the WCAG success criterion violated, the severity, location within the application, description, user impact, remediation recommendation, and a code example demonstrating the fix.

A-001: Images Missing Alternative Text

Severity	Critical
WCAG Criterion	1.1.1 Non-text Content (Level A)
Location	Dashboard, Project Overview, Document Library

Description

Multiple informational images, icons, and charts throughout the application lack appropriate alternative text. Screen reader users receive no information about the content or purpose of these visual elements. Decorative images are not correctly marked with empty alt attributes or role="presentation".

User Impact

Users with visual impairments (such as blindness) relying on screen readers cannot perceive or understand the information conveyed by images. This creates a fundamental barrier to using the platform independently.

Recommendation

Add descriptive alt text to all informational images and icons (e.g., alt="Project status: 78% complete"). For decorative images, use alt="" or role="presentation". For complex charts and data visualisations, provide a text summary or data table alternative.

Code Example

```
<!-- Before -->


<!-- After -->

```

A-002: Form Inputs Without Programmatic Labels

Severity	Critical
WCAG Criterion	1.3.1 Info and Relationships (Level A)
Location	Login, Registration, Project Creation, Document Upload

Description

Several form fields across the application rely solely on placeholder text or visual proximity to indicate their purpose. These fields lack associated `<label>` elements or `aria-label` attributes, meaning the relationship between the label and the input is not programmatically determinable.

User Impact

Screen reader users cannot identify the purpose of form fields, making it impossible to complete key workflows such as logging in, creating projects, or uploading documents.

Recommendation

Associate every form input with a visible `<label>` element using the `for/id` attribute pairing. Where a visible label is not appropriate (e.g., search fields), use `aria-label` or `aria-labelledby` to provide an accessible name.

Code Example

```
<!-- Before -->
<input type="text" placeholder="Project name">

<!-- After -->
<label for="project-name">Project name</label>
<input type="text" id="project-name" placeholder="e.g. Bridge Renovation Phase 2">
```

A-003: Keyboard Focus Not Visible on Interactive Elements

Severity	Critical
WCAG Criterion	2.4.7 Focus Visible (Level AA)
Location	Global navigation, toolbar buttons, data table controls

Description

The default browser focus indicator has been suppressed via CSS (`outline: none`) on multiple interactive elements including navigation links, toolbar buttons, and table action buttons. No custom focus indicator has been provided as a replacement.

User Impact

Keyboard-only users and users with motor impairments who rely on keyboard navigation cannot determine which element currently has focus. This makes the application effectively unusable without a mouse.

Recommendation

Remove outline: none declarations or replace them with a clearly visible custom focus indicator. The focus indicator should have a minimum contrast ratio of 3:1 against adjacent colours and should be at least 2px in thickness [SC 2.4.13 AAA guidance].

Code Example

```
/* Before */
button:focus { outline: none; }

/* After */
button:focus-visible {
  outline: 3px solid #2E75B6;
  outline-offset: 2px;
}
```

A-004: Insufficient Colour Contrast on Body Text

Severity	Major
WCAG Criterion	1.4.3 Contrast (Minimum) (Level AA)
Location	Project metadata, table captions, footer content, helper text

Description

Light grey text (#999999) is used on a white background (#FFFFFF) in several locations, producing a contrast ratio of approximately 2.8:1. WCAG 2.2 AA requires a minimum ratio of 4.5:1 for normal text and 3:1 for large text (18pt or 14pt bold).

User Impact

Users with low vision, colour vision deficiencies, or those viewing the platform in bright environments will struggle to read this content. This includes important contextual information such as project metadata and table captions.

Recommendation

Darken the text colour to at least #767676 (4.5:1 ratio on white) for normal-sized text. For large text (18pt+), a minimum of #949494 (3:1) is acceptable. Use a contrast checking tool during design to verify all colour combinations.

Code Example

```
/* Before */
.helper-text { color: #999999; } /* 2.8:1 ratio - FAIL */

/* After */
.helper-text { color: #595959; } /* 7:1 ratio - PASS */
```

A-005: Modal Dialogs Do Not Trap Focus

Severity	Major
WCAG Criterion	2.4.3 Focus Order (Level A)
Location	Confirmation dialogs, file upload modal, user settings modal

Description

When modal dialogs are opened, keyboard focus is not moved to the dialog and is not constrained within it. Users can Tab behind the modal to interact with obscured content, and pressing Escape does not close the dialog. The dialog also lacks appropriate ARIA roles (`role="dialog"`, `aria-modal="true"`).

User Impact

Keyboard and screen reader users may become disoriented when interacting with modals. They may inadvertently trigger actions on background content or be unable to dismiss the dialog, requiring a page refresh.

Recommendation

Implement a robust focus management pattern for all modal dialogs. On open: move focus to the first focusable element within the dialog. While open: constrain Tab cycling within the dialog. On close: return focus to the triggering element. Add `role="dialog"`, `aria-modal="true"`, and `aria-labelledby` pointing to the dialog title.

Code Example

```
<!-- Accessible modal structure -->
<div role="dialog" aria-modal="true" aria-labelledby="dialog-title">
  <h2 id="dialog-title">Confirm Deletion</h2>
  <p>Are you sure you want to delete this?</p>
  <button>Cancel</button>
  <button>Delete</button>
</div>
```

A-006: Data Tables Missing Header Associations

Severity	Major
WCAG Criterion	1.3.1 Info and Relationships (Level A)
Location	Document list, project list, user management, audit log

Description

Data tables throughout the application use `<div>` elements styled to appear as tables rather than semantic HTML `<table>` elements. Where tables are used, column headers lack `<th>` elements and scope attributes, preventing screen readers from associating data cells with their respective headers.

User Impact

Screen reader users cannot navigate tables efficiently or understand the relationship between data and column headers. This severely limits the ability to review and manage project information.

Recommendation

Use semantic HTML table elements (<table>, <thead>, <tbody>, <th>, <td>) for all tabular data. Add scope="col" to column headers and scope="row" to row headers. For complex tables, use headers and id attributes. Provide a <caption> element describing the table purpose.

Code Example

```
<!-- Accessible data table -->
<table>
  <caption>Project document list</caption>
  <thead>
    <tr>
      <th scope="col">Document Name</th>
      <th scope="col">Type</th>
      <th scope="col">Uploaded</th>
      <th scope="col">Actions</th>
    </tr>
  </thead>
  <tbody>...</tbody>
</table>
```

A-007: Status Messages Not Announced to Assistive Technology

Severity	Major
WCAG Criterion	4.1.3 Status Messages (Level AA)
Location	File upload progress, save confirmations, error notifications

Description

Toast notifications, file upload progress indicators, and success/error messages appear visually but are not announced to screen readers. These dynamic content updates lack appropriate ARIA live region attributes.

User Impact

Screen reader users receive no feedback when actions are completed or fail. For example, after uploading a document, there is no audible confirmation that the upload succeeded or failed.

Recommendation

Wrap status messages in an ARIA live region. Use role="status" (polite) for non-urgent confirmations and role="alert" (assertive) for errors and warnings. Ensure the live region container is present in the DOM before content is injected.

Code Example

```
<!-- Status container (present in DOM on page load) -->
<div role="status" aria-live="polite" class="sr-only">
  <!-- Dynamically injected: -->
  Document uploaded successfully.
</div>

<!-- Error alert -->
<div role="alert">
  Upload failed. Please check your file and try again.
</div>
```

A-008: Page Language Not Declared

Severity	Minor
WCAG Criterion	3.1.1 Language of Page (Level A)
Location	All pages

Description

The HTML element does not include a lang attribute specifying the language of the page content (e.g., lang="en").

User Impact

Screen readers may use incorrect pronunciation rules, making content difficult or impossible to understand for users who rely on text-to-speech. This also affects automatic translation tools.

Recommendation

Add lang="en" (or the appropriate language code) to the <html> element on every page. Where content in other languages appears, use the lang attribute on the containing element.

Code Example

```
<!-- Before -->
<html>

<!-- After -->
<html lang="en">
```

A-009: Links and Buttons Lack Descriptive Accessible Names

Severity	Major
WCAG Criterion	2.4.4 Link Purpose (In Context) (Level A)
Location	Document actions, navigation, project cards

Description

Multiple icon-only buttons and links across the application lack accessible names. Buttons using only SVG icons or background images for their visual appearance do not provide text alternatives. Repeated "Click here" and "Read more" links do not describe their destination or purpose.

User Impact

Screen reader users hear only "button" or "link" with no indication of what the element does. When navigating via a links list or buttons list, identical labels ("Click here") provide no way to distinguish between actions.

Recommendation

Add `aria-label` or visually-hidden text to all icon-only buttons (e.g., `aria-label="Download document"`). Replace generic link text with descriptive labels. Where context is needed, use `aria-describedby` to associate additional information.

Code Example

```
<!-- Before -->
<button><svg>...</svg></button>

<!-- After -->
<button aria-label="Download project brief">
  <svg aria-hidden="true">...</svg>
</button>
```

A-010: Resize and Reflow Issues at 200% Zoom

Severity	Minor
WCAG Criterion	1.4.4 Resize Text (Level AA)
Location	Navigation bar, data tables, dashboard widgets

Description

When the browser is zoomed to 200%, some navigation items become truncated or overlap, data table columns collapse causing content to be clipped without scrollability, and dashboard widgets overflow their containers.

User Impact

Users with low vision who enlarge text cannot access the full interface. Content is hidden or obscured, requiring horizontal scrolling that is not intuitive or, in some cases, not possible.

Recommendation

Ensure all content and functionality remains accessible at 200% zoom. Use responsive CSS (relative units like rem and em, flexible grid layouts) to allow content to reflow gracefully. Test at 1280px viewport width at 200% zoom (equivalent to 640px effective viewport).

Code Example

```
/* Use flexible layouts */  
.page-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));  
  gap: 1rem;  
}
```

SAMPLE

Remediation Roadmap

The following roadmap prioritises fixes by severity, user impact and estimated effort to achieve the maximum improvement in accessibility in the shortest time. We recommend addressing **Critical** issues first, followed by **Major**, then **Minor**.

Phase 1: Critical Fixes

These issues completely prevent access to core functionality and should be resolved as the highest priority.

ID	Issue	Effort	Priority
A-001	Images Missing Alternative Text	Medium	Critical
A-002	Form Inputs Without Programmatic Labels	Medium	Critical
A-003	Keyboard Focus Not Visible on Interactive Elements	Medium	Critical

Phase 2: Major Fixes

These issues significantly impact the experience for assistive technology users and should be resolved promptly after critical fixes.

ID	Issue	Effort	Priority
A-004	Insufficient Colour Contrast on Body Text	Low	Major
A-005	Modal Dialogs Do Not Trap Focus	Medium–High	Major
A-006	Data Tables Missing Header Associations	Medium	Major
A-007	Status Messages Not Announced to Assistive Technology	Medium	Major
A-009	Links and Buttons Lack Descriptive Accessible Names	Medium–High	Major

Phase 3: Minor Fixes

These issues create friction but do not prevent task completion. They should be resolved to achieve full conformance.

ID	Issue	Effort	Priority
A-008	Page Language Not Declared	Low	Minor
A-010	Resize and Reflow Issues at 200% Zoom	Medium	Minor