TU856/3
UNIVERSAL DESIGN AND ASSISTIVE ICT

ASSIGNMENT 2

PAULINA CZARNOTA C21365726

24/04/2025

PART A

i) Overview of the EU Web Accessibility Directive (WAD) and its implications for EU countries like Ireland

The EU Web Accessibility Directive (Directive (EU) 2016/2102) is a legal instrument adopted by the European Union to improve the accessibility of public sector websites and mobile applications for all users, particularly those with disabilities. It was adopted on 26 October 2016 and became legally binding on 22 December 2016, with full implementation required by September 2018. Its central objective is to ensure that digital public services are inclusive, accessible, and usable by everyone.

The Directive mandates that public sector bodies comply with the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA, which define four foundational accessibility principles: Perceivable, Operable, Understandable, and Robust (POUR). These ensure that digital content is structured to be understandable and functional for users with visual, auditory, cognitive, or motor disabilities.

The scope of the Directive includes websites, mobile applications, documents (e.g., PDFs and Word files), online forms, navigation components, and multimedia content. Exemptions include live video streaming, office documents published before 23 September 2018 (unless updated), and third-party content not funded or controlled by public sector bodies. Despite these exceptions, the Directive has a broad and enforceable reach across public-facing digital services.

Each Member State is responsible for implementation, monitoring, and enforcement. In Ireland, the Directive has been transposed into national law via the European Union (Accessibility of Websites and Mobile Applications of Public Sector Bodies) Regulations 2020. The National Disability Authority (NDA) is the designated monitoring body and is tasked with conducting simplified and in-depth accessibility audits using both manual and automated tools such as WAVE (wave.webaim.org), AChecker, and TAW (tawdis.net). Evaluations are conducted based on the harmonized European standard EN 301 549, which aligns with WCAG 2.1.

Irish public bodies are required to:

- Publish and maintain an Accessibility Statement in line with the EU model template.
- Offer a feedback mechanism to report accessibility issues or request accessible alternatives.
- Participate in regular audits and monitoring as part of Ireland's obligations to report progress to the European Commission every three years.

Implications for Ireland include the legal responsibility for government departments, education institutions (e.g., TU Dublin), health agencies (e.g., HSE), and local authorities to ensure ongoing digital accessibility. These organizations must invest in accessible web design, staff training, assistive technology testing, and periodic evaluations. Institutions such as the Centre for Excellence in Universal Design (CEUD) provide best practice resources and checklists to aid implementation.

The WAD also supports Ireland's obligations under the United Nations Convention on the Rights of Persons with Disabilities (UNCRPD), ratified by Ireland in 2018. It reinforces the commitment to universal design, digital inclusion, and equal access to information.

In summary, the EU Web Accessibility Directive provides a robust framework for ensuring that public sector digital services across Ireland and the EU are accessible to all users. It promotes consistency, accountability, and usability, forming a key pillar of inclusive digital policy and service delivery.

Sources: Directive (EU) 2016/2102, Directive (EU) 2019/882, EN 301 549, W3C (https://www.w3.org/WAI/), WebAIM (https://webaim.org/), CEUD (https://universaldesign.ie/), NDA (https://nda.ie/), EDF (https://edf-feph.org/), UNCRPD (https://www.un.org/disabilities/)

ii) Overview of the 2025 European Accessibility Act (EAA) and its implications for the ICT Industry

The European Accessibility Act (EAA), Directive (EU) 2019/882, is a harmonized legal framework aimed at eliminating accessibility barriers across the European Union for digital products and services used by the general public. Unlike the EU Web Accessibility Directive, which targets public sector bodies, the EAA applies primarily to the private sector, especially the Information and Communications Technology (ICT) industry. Its goal is to establish uniform accessibility standards across Member States, reduce legal fragmentation, and improve the participation of people with disabilities in the digital single market. The enforcement deadline is 28 June 2025.

The scope of the EAA includes a wide range of technologies and services:

- Consumer ICT hardware and operating systems (e.g. computers, smartphones, tablets)
- E-commerce websites, banking services, ATMs, and electronic payment terminals
- Audiovisual media services such as streaming platforms and user interfaces
- Transport information systems and ticket vending machines
- Telecommunications services, including access to emergency services (e.g. 112)
- E-books and associated reading software, which must be compatible with assistive technologies

The EAA mandates adherence to international accessibility standards such as WCAG 2.1 (Level AA), EN 301 549, and ISO/IEC 40500. These standards address critical areas like screen reader compatibility, semantic HTML structure, logical navigation order, sufficient contrast ratios (at least 4.5:1), meaningful link and button labels, keyboard accessibility, and the appropriate use of ARIA roles. The Act promotes "accessibility by design and by default," meaning that inclusivity must be built into every stage of development rather than added as an afterthought.

The implications for the ICT industry are far-reaching. Businesses that fail to meet accessibility criteria risk market exclusion, legal penalties, and reputational harm. National enforcement bodies—such as Ireland's National Disability Authority (NDA)—are responsible for auditing compliance, handling complaints, and issuing corrective measures. The EAA also strengthens consumer rights by giving people with disabilities a legal mechanism to demand accessible services and seek redress when needed.

Importantly, the EAA fosters innovation by encouraging the development of inclusive technologies. Accessibility features like voice controls, dark mode, real-time captions, and smart navigation are increasingly mainstream and benefit all users. The ICT sector is also leveraging AI and machine learning to support auditing tasks such as automated alt text generation, real-time contrast analysis, screen reader testing, and user journey simulations. These tools reduce manual workload and ensure more scalable compliance strategies.

Small and medium-sized enterprises (SMEs) are offered transitional flexibility, yet are encouraged to integrate accessibility to enhance customer reach, meet growing legal expectations, and future-proof their offerings. The EAA also aligns with wider policy goals such as the UN Convention on the Rights of Persons with Disabilities (UNCRPD), the European Pillar of Social Rights, and Sustainable Development Goal 10 (Reduced Inequalities), highlighting accessibility as both a human right and a strategic business priority.

In conclusion, the EAA transforms accessibility into a binding legal and ethical obligation across the European market. It compels the ICT industry to build inclusive products from the outset, ensuring that digital participation is not only possible—but equitable—for all citizens.

Sources: Directive (EU) 2019/882, EN 301 549, W3C (https://www.w3.org/WAI/), WebAIM (https://webaim.org/), CEUD (https://universaldesign.ie/), EDF (https://edf-feph.org/)

PART B

1. Manual Accessibility Evaluation - Website Checkpoint Failures

i) Website 1: TU Dublin Computer Science - https://www.tudublin.ie/explore/schools-and-disciplines/computing-information-technology/computer-science/

This evaluation was conducted using the WCAG 2.1 Level AA guidelines and focused on the four POUR principles: Perceivable, Operable, Understandable, and Robust. Manual testing included:

- Keyboard-only navigation
- NVDA screen reader testing
- Code inspection
- Review of ARIA roles and semantic structure
- Comparison with automated tool results (WAVE, AChecker, TAW, Lighthouse)

1. Perceivable (WCAG Principle 1)

- **Alt Text (1.1.1):** Several images lack descriptive alt attributes. This was confirmed via WAVE and AChecker which flagged "missing alternative text" and "empty links." For example, decorative banners use <img> tags without alt="" or include generic terms like "image."
- **Info and Relationships (1.3.1):** Heading hierarchy is inconsistent. Pages skip heading levels (e.g., from <h1> to <h3>), which disrupts screen reader navigation. Manual inspection and TAW reported incorrect semantic markup, affecting users who rely on structured navigation.
- **Contrast (1.4.3):** Text on image banners and call-to-action buttons fails contrast checks, with ratios below the required 4.5:1. Google Lighthouse flagged multiple instances of insufficient colour contrast, e.g., white text over light backgrounds.
- **Images of Text (1.4.5):** Important information (e.g., dates or titles) embedded in promotional graphics are not available in text format. This fails the requirement for text alternatives for images of text.
- **Text Resize and Reflow (1.4.4, 1.4.10):** Some elements overlap when zoomed to 200%, especially navigation items. This affects users with visual impairments or those using screen magnifiers.

2. Operable (WCAG Principle 2)

- **Keyboard Accessibility (2.1.1):** Dropdown menus and carousels are not operable via keyboard alone. Submenus only expand on hover and cannot be opened using Tab or Enter. Manual NVDA testing confirmed this barrier.
- **Timing Adjustable (2.2.1):** The carousel auto-advances with no visible control to pause or stop, which may disorient users with cognitive or vestibular disabilities. No pause/play buttons were found in code.
- **Navigation (2.4.1, 2.4.3):** No "Skip to main content" link is present. This forces screen reader and keyboard users to navigate repetitive headers every time. There are also no ARIA landmarks such as <nav> or <main>.
- **Link Purpose (2.4.4):** Many links use non-descriptive phrases like "read more," "click here," or buttons with only icons. WAVE flagged multiple vague link texts.
- **Focus Visible (2.4.7):** Keyboard focus indicators are inconsistent. Some buttons and links show no visible focus state, creating barriers for sighted keyboard users.

3. Understandable (WCAG Principle 3)

- **Language of Page (3.1.1):** The page language is set to English and is correctly detected by NVDA and AChecker.
- **Labels and Instructions (3.3.2):** Input fields in the newsletter form do not have <label> elements. This was flagged by AChecker and Lighthouse. Form instructions are missing or rely on placeholder text only.
- **Error Identification (3.3.1):** No error messages are provided for invalid or empty form submission. Users are not alerted when required fields are skipped.

4. Robust (WCAG Principle 4)

- **Parsing (4.1.1):** HTML validation errors include duplicate IDs and unclosed tags, which can interfere with assistive technologies. AChecker flagged these under "known problems."
- **Name, Role, Value (4.1.2):** Some interactive elements (e.g., expandable menus) lack ARIA roles like aria-expanded. This causes screen readers to miss changes in state, violating this checkpoint. TAW confirmed incorrect or missing ARIA markup.
- **Live Regions:** There are no ARIA live regions to notify users of dynamic content changes, such as in carousels or modal popups.

Additional Observations

- **Mobile View and Responsiveness:** While the site is responsive, accessibility issues persist on mobile devices. Button sizes and form fields are too small to interact with comfortably on touchscreens, violating touch target recommendations.
- **PDFs and Downloadables:** Some documents linked on the site (like course brochures) open as PDFs. These have not been evaluated here but must also meet accessibility requirements, especially for tagging and screen reader compatibility.

Summary Table

WCAG Criterion / Issue Type	Manual	WAVE	TAW	AChecker	Lighthouse
Missing/Incorrect Alt Text	✓	✓	✓	✓	✗
Poor Colour Contrast	✓	✓	✓	✗	✓
Missing Labels	✓	✓	✓	✓	✓
Keyboard Navigation Fail	✓	✗	✓	✗	✗
HTML/ARIA Issues	✓	✓	✓	✓	✓
No Skip Link/Focus Issues	✓	✓	✓	✗	✓

Conclusion

The TU Dublin Computer Science website fails to meet several WCAG 2.1 Level AA checkpoints across all four POUR principles. While some issues are visual and semantic (e.g., alt text, contrast), others directly affect navigation and usability for people using assistive technologies. Both manual testing and automated tools such as WAVE, AChecker, TAW, and Lighthouse confirm these failures. The site requires remediation in keyboard navigation, ARIA landmarks, heading structure, form labelling, and contrast compliance to meet both legal requirements under the EU Web Accessibility Directive and best practices in universal design.

ii) Website 2: UCD Computer Science – https://www.ucd.ie/cs/

This evaluation was conducted using the WCAG 2.1 Level AA guidelines. Manual testing methods included:

- Keyboard-only navigation
- NVDA screen reader testing
- Visual inspection and HTML/CSS/ARIA analysis
- Verification with accessibility checklists from WebAIM and CEUD
- Cross-referencing with results from WAVE, TAW, AChecker, and Google Lighthouse

1. Perceivable (WCAG Principle 1)

- **Alt Text (1.1.1):** Many images used across the homepage and research sections lack proper alternative text. For instance, banners labeled as "news image" or "event photo" were flagged by both WAVE and AChecker. Some <img> tags have empty or redundant alt values, failing to convey meaningful content for screen reader users.
- **Info and Relationships (1.3.1):** Heading levels are not consistently structured. Visual elements appear bold and styled like headings (e.g., faculty names), but lack semantic HTML (<h2>, <h3>). This was confirmed by TAW and WAVE, which both flagged skipped heading levels and incorrect hierarchy.
- **Contrast Minimum (1.4.3):** Footer text and navigation links were flagged by Google Lighthouse and WAVE for low colour contrast, especially grey text on a white background. This makes it difficult for low-vision users to read.
- **Images of Text (1.4.5):** Banners with slogans or promotional text embedded in images do not offer text equivalents or alternatives. There is no HTML fallback, violating WCAG 2.1 success criterion 1.4.5.
- **Text Resize and Reflow (1.4.4, 1.4.10):** When zoomed to 200%, layout elements shift awkwardly, overlapping with images or pushing off-screen. Text and buttons in the main navigation do not reflow correctly on smaller devices or magnified screens.

2. Operable (WCAG Principle 2)

- **Keyboard Accessibility (2.1.1):** Several interactive components, including image carousels and the Google Map embed, are not operable via keyboard alone. NVDA testing and tab order analysis showed that these elements either skip focus entirely or trap the user. TAW flagged these as critical failures.
- **Timing Adjustable (2.2.1):** Carousels auto-rotate and lack pause/stop controls. This creates cognitive overload for some users and may lead to disorientation.
- **Navigable (2.4.1, 2.4.3):** There is no "skip to content" or ARIA landmark to allow quick bypass of repetitive headers. This omission forces keyboard users to navigate a large menu each time. Navigation is also missing <nav> or aria-label attributes for screen reader-friendly region identification.
- **Link Purpose (2.4.4):** Anchor texts like "read more," "click here," and "learn more" are used repeatedly without additional context, which fails WCAG requirements for meaningful link purpose. This was also highlighted in AChecker reports.
- **Focus Visible (2.4.7):** Keyboard focus indicators are not consistent across interactive elements. For example, form buttons and nav tabs provide no visible indication of current focus, confirmed by manual testing and flagged in WAVE screenshots.

3. Understandable (WCAG Principle 3)

- **Language of Page (3.1.1):** The language is set correctly to English (lang="en") and was detected properly in NVDA and AChecker scans.

- **Labels and Instructions (3.3.2):** Forms for newsletter subscriptions and contact lack explicit <label> elements or ARIA descriptors. Field grouping using <fieldset> and <legend> is absent. Placeholder-only guidance is not sufficient, as confirmed in both TAW and WAVE.
- **Error Suggestions (3.3.3):** No inline validation or user guidance is provided upon form submission failure. Users receive no feedback unless required fields are submitted correctly.

4. Robust (WCAG Principle 4)

- **Parsing (4.1.1):** HTML validation showed multiple issues, such as non-unique element IDs, improper tag nesting, and missing roles. These affect screen reader compatibility. AChecker flagged these as known errors under parsing compliance.
- **Name, Role, Value (4.1.2):** Buttons and form elements lack aria-label, aria-required, or aria-describedby. For example, collapsible menus do not use aria-expanded or aria-controls, making it hard for assistive technology to interpret component states. WAVE and TAW also reported improper or missing ARIA roles.
- **Dynamic Content and Live Regions:** No ARIA live regions are used to notify screen reader users of dynamic updates such as news carousels or research feature pop-ups.

Additional Manual Observations

- **Mobile Accessibility:** Touch target areas on mobile are small or crowded, especially in dropdown menus. This violates best practices for users with motor impairments.
- **PDF Accessibility:** Linked documents like course brochures are in PDF format. Without tags or OCR, these are unreadable to screen readers, potentially violating WCAG 2.0 AA and PDF/UA guidelines.
- **Use of Icons and Multimedia:** Buttons with icons only (e.g., arrows or "play" symbols) are not labelled for assistive technologies. These need aria-label or -based fallback text.

Summary Table

WCAG Criterion / Issue Type	Manual	WAVE	TAW	AChecker	Lighthouse
Missing/Incorrect Alt Text	✓	✓	✓	✓	✗
Heading Hierarchy Issues	✓	✓	✓	✗	✗
Poor Colour Contrast	✓	✓	✓	✗	✓
Form Labeling Problems	✓	✓	✓	✓	✓
Keyboard Trap/Nav Issues	✓	✗	✓	✗	✗
No Skip Link/ARIA Errors	✓	✓	✓	✓	✓
Parsing/HTML Errors	✓	✗	✓	✓	✓

Conclusion

The UCD Computer Science website demonstrates consistent accessibility issues across all four POUR principles. The combination of manual testing and automated tools revealed a wide range of WCAG 2.1 Level AA violations. Critical areas include missing alternative text, poor contrast ratios, improper heading structures, inaccessible forms, and insufficient ARIA usage.

The site's dynamic elements are not adequately announced by assistive technologies, and it lacks features essential for screen reader users such as skip links and correct role semantics. Given these shortcomings, comprehensive remediation is necessary to align with EU Web Accessibility Directive and upcoming obligations under the European Accessibility Act (2025). Addressing these issues will also enhance usability for all users, especially those with disabilities.

2. Online Evaluation Tools

As per assignment requirements, four automated web accessibility evaluation tools were used to test the TU Dublin and UCD Computer Science websites. The aim was to obtain measurable, standards-based assessments of each site's compliance with WCAG 2.1 Level AA.

The tools used were:

- **WAVE** (Web Accessibility Evaluation Tool) – wave.webaim.org
- **TAW** (Test de Accesibilidad Web) – tawdis.net
- **AChecker** – achecker.achecks.ca
- **Google Lighthouse** – pagespeed.web.dev

Each tool provided a distinct perspective on accessibility, focusing on different technical and usability criteria:

- **WAVE** highlighted visual accessibility issues like low colour contrast, missing alternative text, skipped heading levels, and form labelling problems.
- **TAW** emphasized structural WCAG failures, such as improper use of ARIA roles, lack of semantic landmarks, and navigational barriers.
- **AChecker** classified detected issues into known, likely, and potential problems, helping identify structural, semantic, and form-related accessibility issues.
- **Google Lighthouse** gave a percentage-based accessibility score, pinpointing specific issues like unlabelled form fields, missing button names, and improper ARIA use.

All tool outputs were saved as screenshots, links or PDFs and are included for reference.

3. Comparing Manual Evaluation (Part 1) with Automated Tool Results (Part 2)

i) TU Dublin Website Comparison

The manual evaluation and automated assessments for the TU Dublin Computer Science webpage both revealed widespread accessibility issues, notably with missing alternative text, low colour contrast, poor semantic structure, vague or empty link labels, and inadequate ARIA support. These issues affect conformance with WCAG 2.1 Level A and AA and create barriers for users relying on assistive technologies.

Manual testing using NVDA (screen reader), keyboard-only navigation, and visual inspection highlighted real user difficulties in perceiving and interacting with content. For instance, the absence of a "skip to main content" link forces keyboard users to tab repeatedly through menus on every page. Several dropdown menus were completely inaccessible without a mouse, and focus outlines were either faint or absent, violating SC 2.4.7 (Focus Visible). Forms lacked proper field grouping and label associations, failing SC 3.3.2 (Labels or Instructions), which was clearly apparent during NVDA testing.

The manual process also exposed significant usability concerns not caught by automated tools. Image carousels lacked keyboard operability and failed to offer pause or stop controls, violating SC 2.2.2 (Pause, Stop, Hide). ARIA live regions were missing, so dynamic content updates—such as tab panels or sliders—did not trigger any alerts for screen reader users, leading to confusion and a non-robust experience (SC 4.1.3).

In contrast, the automated tools focused on markup and attribute-level compliance:

- **WAVE** flagged 4 contrast errors, 7 empty links, 2 missing form labels, and 6 heading level skips (e.g., jumping from <h2> to <h4>).
- **TAW** revealed 12 Level A and 6 Level AA failures, particularly around semantic markup, missing navigation landmarks, and poor keyboard accessibility.
- **AChecker** reported over 20 known errors, including improperly nested elements, unlabelled controls, duplicate IDs, missing lang attributes, and vague link text.
- **Google Lighthouse** scored the page at 72% for accessibility, citing contrast issues, missing ARIA roles, and low accessibility for form components.

A key difference is that automated tools do not simulate real-world interactions. While WAVE and Lighthouse noted that certain links lacked context or controls were unlabelled, only manual testing identified that those links were also unreachable or meaningless in context. Furthermore, AChecker's HTML-focused scan did not recognize that tab order was illogical, making keyboard navigation disorienting—especially when modals and dropdowns had no focus traps or dismissal options.

Additionally, tools like TAW and WAVE failed to detect missing ARIA live regions and improper use of ARIA roles that cause assistive technologies to misinterpret page structure. For example, main navigation used generic elements instead of <nav> or role="navigation", making it difficult for screen reader users to locate core areas of the page.

The contrast between automated and manual methods underlines their complementary nature. Automated tools efficiently flag syntactic violations and offer quick snapshots of conformance. However, manual testing is essential for detecting functional and experiential flaws that impact usability and user flow—such as how a screen reader interprets a form or how a visually impaired user accesses a carousel.

In conclusion, the combined use of manual and automated testing provides a holistic view of accessibility. Automated tools are efficient for identifying WCAG failures at code level, while manual

testing uncovers user-facing issues that are otherwise missed. This dual-method approach is essential for compliance with both the EU Web Accessibility Directive and best practices in inclusive design.

ii) UCD Website Comparison

The manual and automated evaluations of the UCD Computer Science website identified consistent and critical accessibility issues, many of which constitute violations of WCAG 2.1 Level AA. Both testing approaches aligned in detecting incomplete ARIA implementation, inconsistent heading structures, low contrast, and barriers to keyboard-only navigation.

Manual testing using keyboard navigation and NVDA screen reader revealed that key interactive elements—such as embedded Google Maps, image carousels, accordions, and collapsible menus—were not accessible without a mouse. Many interactive buttons lacked visible or programmatic focus indicators (SC 2.4.7), and the absence of a "skip to main content" link (SC 2.4.1) forced users to tab repeatedly through lengthy navigation. Accordion controls lacked aria-expanded and aria-controls, meaning screen readers could not announce the expanded/collapsed state, thus violating SC 4.1.2.

The NVDA test also uncovered unlabelled buttons and ambiguous links (e.g., "learn more"), which failed SC 2.4.4 (Link Purpose) and SC 3.3.2 (Labels or Instructions). Additionally, some dynamic content changes—such as news sliders and modal windows—lacked ARIA live regions, so NVDA provided no feedback about state changes, in breach of SC 4.1.3.

Automated testing tools confirmed and elaborated on these issues:

- **WAVE** detected 5 contrast errors, 3 missing form labels, and 4 skipped heading levels, as well as multiple vague anchor texts.
- **TAW** flagged incorrect or missing ARIA attributes and landmark roles. Key structural elements like <nav>, <main>, and <aside> were absent or used improperly.
- **AChecker** reported 18 known issues, including empty buttons, improper HTML nesting, duplicated IDs, and missing alt text for meaningful images.
- **Google Lighthouse** gave a score of 76%, highlighting missing labels, non-descriptive button names, form inputs without aria-describedby, and contrast failures below the minimum ratio of 4.5:1.

Manual inspection revealed nuances not visible in automated reports. For example, while some form labels existed in the code, they were visually disconnected or placed far from the input fields, confusing users with visual impairments. This type of usability issue may not trigger automated alerts but still violates WCAG success criteria. Similarly, keyboard testing showed that users could tab into dropdown menus but could not use arrow keys to select options—breaking navigation continuity and accessibility norms.

A deeper issue involved misuse of ARIA attributes. Some elements contained roles like button or tab, but their states were not updated during interaction. This breaks the robustness (SC 4.1.2) and undermines assistive tech compatibility. Duplicate id attributes on form elements, flagged by AChecker, also violate SC 4.1.1 (Parsing), leading to potential misinterpretation by screen readers.

The combination of tests highlights the importance of using both manual and automated methods. Automated scans efficiently detect markup and attribute-level failures, but they cannot evaluate context, intent, or user experience. Manual testing adds essential context—confirming whether links make sense, whether controls are accessible in real usage, and whether feedback is properly conveyed to users of assistive tech.

In conclusion, the UCD Computer Science site suffers from both technical and experiential accessibility issues. A balanced evaluation approach—combining tools like WAVE, TAW, AChecker, and Lighthouse with structured manual testing—is necessary to uncover and remediate all barriers. This aligns with the accessibility mandates of the EU Web Accessibility Directive and the 2025 European Accessibility Act, both of which emphasize user-centered, inclusive digital design and compliance with WCAG 2.1 Level AA.

4. Remedial Actions to Ensure Website Compliance

i) TU Dublin Website – Remedial Action Plan

To achieve WCAG 2.1 Level AA compliance and meet the requirements of the EU Web Accessibility Directive and upcoming European Accessibility Act (EAA) 2025, the TU Dublin Computer Science website must address key accessibility barriers. The following plan combines findings from manual testing, automated tools (WAVE, TAW, AChecker, Lighthouse), and best practices from CEUD and NDA.

1. Alternative Text and Image Accessibility (WCAG 1.1.1, 1.4.5)

- Provide descriptive alt text for all informative images.
- Use empty alt="" or role="presentation" for decorative images.
- Avoid embedding essential text in images. When necessary, include equivalent text in HTML.
- AChecker identified multiple missing or unclear alt attributes, impacting screen reader navigation.

2. Colour Contrast (WCAG 1.4.3, 1.4.6)

- Ensure all text meets contrast ratios of at least 4.5:1 for normal text and 3:1 for large text using the WebAIM contrast checker.
- Adjust text over promotional banners and buttons, particularly white text over light or dynamic backgrounds.
- WAVE and Lighthouse flagged several colour contrast failures affecting readability.

3. Heading Structure and Semantic HTML (WCAG 1.3.1, 2.4.6)

- Use properly sequenced <h1> to <h6> tags to preserve a logical structure.
- Replace styled or tags with semantic HTML headings.
- Include structural HTML5 elements like <nav>, <main>, and <footer> for better navigation.
- TAW reported skipped heading levels and missing semantic regions.

4. Keyboard Navigation and Focus Indicators (WCAG 2.1.1, 2.4.3, 2.4.7)

- Ensure all interactive elements, including dropdown menus and carousels, are fully keyboard-accessible.
- Implement a visible "Skip to Main Content" link to improve efficiency for screen reader and keyboard users.
- Add visible focus indicators to all focusable elements. WAVE identified several missing focus styles.

5. Link Purpose and Context (WCAG 2.4.4)

- Replace vague links such as “Click here” and “Read more” with descriptive text indicating their purpose or destination.
- Use aria-label or aria-labelledby to clarify the purpose of icon-only or hidden links.

6. Form Accessibility (WCAG 3.3.2, 1.3.1)

- Ensure all form fields have associated <label> elements.
- Use <fieldset> and <legend> to group related form inputs.
- Provide error identification and suggestions using aria-describedby, aria-invalid, and visible inline messages.

- AChecker flagged unlabelled form elements and field association issues.

7. ARIA and Landmark Usage (WCAG 4.1.2)

- Implement correct ARIA roles (e.g., navigation, main, complementary) to enhance semantic clarity.
- Add aria-live regions to inform users of dynamic content updates (e.g., carousel transitions or AJAX-loaded content).
- TAW highlighted missing and misused ARIA roles that must be corrected.

8. Dynamic Content and Carousels (WCAG 2.2.2, 4.1.1)

- Provide pause, stop, and hide controls for auto-rotating carousels.
- Mark slides with role="group" and ensure each has an accessible label.
- Announce slide changes using aria-live="polite" or similar techniques.
- Manual testing revealed auto-rotating content without user control, which violates usability standards.

9. HTML Validation and Code Quality (WCAG 4.1.1)

- Fix all HTML parsing errors (e.g., duplicate IDs, unclosed tags, improperly nested elements) identified by AChecker and Lighthouse.
- Ensure all form elements have unique IDs and valid structure to prevent conflicts with screen readers.

10. Accessibility Statement and Feedback Channel (WAD Requirement)

- Publish an up-to-date accessibility statement detailing WCAG conformance, known issues, and available alternatives.
- Include an easy-to-use feedback form or email for users to report accessibility problems.

11. Ongoing Testing and Monitoring

- Conduct quarterly accessibility audits using WAVE, TAW, Lighthouse, and AChecker.
- Perform screen reader testing with NVDA, JAWS, and VoiceOver, and keyboard-only navigation checks.
- Apply regression testing following content updates or design changes.

12. Staff Training and Inclusive Design Policy

- Deliver accessibility training for web development, content creation, and UX teams.
- Align internal development guidelines with WCAG 2.1 and CEUD recommendations.
- Encourage the use of design systems that integrate inclusive UI components.

Conclusion

These remedial actions, if implemented comprehensively, will significantly enhance the TU Dublin Computer Science website’s accessibility, usability, and legal compliance. Aligning with WCAG 2.1, the EU Web Accessibility Directive, and the upcoming European Accessibility Act will also support TU Dublin's broader commitment to digital inclusion and universal access.

ii) UCD Website – Remedial Action Plan

The UCD Computer Science website needs thorough remediation to meet WCAG 2.1 Level AA, the EU Web Accessibility Directive, and the upcoming EAA, based on findings from manual and automated tools (WAVE, AChecker, TAW, Lighthouse).

1. Alternative Text and Image Accessibility (WCAG 1.1.1, 1.4.5)

- Replace generic or missing alt text (e.g., “image1”, “photo”) with meaningful descriptions that convey purpose and context.
- Decorative images should include alt="" or role="presentation".
- Images containing text (e.g., banners) must provide equivalent textual content elsewhere in the HTML.
- AChecker flagged missing and redundant alt attributes; WAVE identified empty links associated with images.

2. Colour Contrast (WCAG 1.4.3, 1.4.6)

- Revise all low-contrast text, particularly in the footer and navigation bar (e.g., grey on white), to achieve at least 4.5:1 ratio.
- Use WebAIM Contrast Checker to validate each colour combination.
- Lighthouse and WAVE confirmed multiple contrast violations affecting readability.

3. Heading Structure and Semantic Markup (WCAG 1.3.1, 2.4.6)

- Ensure headings follow a logical sequence (e.g., from <h1> to <h2>) without skipping levels.
- Avoid using visual styling to mimic heading appearance.
- Insert missing semantic elements such as <nav>, <main>, and <footer> to structure the page for assistive technologies.
- TAW and WAVE flagged skipped heading levels and improper use of heading tags.

4. Keyboard Navigation and Focus Indicators (WCAG 2.1.1, 2.4.3, 2.4.7)

- Make all components operable via keyboard alone. For instance, embedded Google Maps, carousels, and dropdown menus currently trap or skip focus.
- Add a visible “Skip to main content” link at the top of the page.
- Ensure all focusable elements (e.g., buttons, links) have visible outlines or indicators during keyboard navigation.
- WAVE and manual testing revealed missing focus styles and inaccessible widgets.

5. Link Purpose and Text Clarity (WCAG 2.4.4)

- Avoid vague anchor text such as “Click here” or “Read more”. Use descriptive labels like “View course structure” or “Explore research projects”.
- Use aria-label where links are represented by icons only.
- AChecker and Lighthouse identified multiple vague link instances and missing descriptions.

6. Forms and Input Fields (WCAG 3.3.2, 4.1.2)

- Associate each form control with a visible <label> or ARIA attribute (e.g., aria-labelledby).
- Use <fieldset> and <legend> to group related form fields (e.g., contact or registration forms).
- Include clear error messages and success feedback using aria-describedby and inline text.
- Lighthouse flagged unlabelled inputs and missing descriptions.

7. ARIA Landmarks and Dynamic Updates (WCAG 4.1.2, 4.1.3)

- Correct the misuse of ARIA roles such as aria-expanded, aria-controls, and landmark roles (e.g., role="navigation").
- Implement aria-live regions to announce dynamically updating content, such as rotating banners or tabs.
- Both TAW and AChecker highlighted these deficiencies, which prevent screen reader users from perceiving dynamic changes.

8. Parsing and Validation (WCAG 4.1.1)

- Fix duplicate ID values, unclosed tags, and incorrect nesting, which disrupt accessibility APIs.
- Confirm that all structural and form elements are used appropriately and validated via W3C HTML Validator.
- AChecker listed over 15 known parsing problems, including repeated IDs and broken DOM structure.

9. Auto-Rotating Content and Multimedia Controls (WCAG 2.2.2, 2.2.1)

- Allow users to pause, stop, or hide carousels and modals using clearly visible controls.
- Ensure carousel slides have appropriate ARIA labelling and use role="group" to group them meaningfully.
- Announce content changes using polite or assertive aria-live settings to alert screen reader users.

10. Accessibility Statement and User Feedback (WAD Requirement)

- Publish an accessibility statement disclosing compliance status, known issues, and timelines for remediation.
- Include a feedback mechanism (e.g., email form or contact link) to report problems and request accessible alternatives.
- Link this statement in the website footer and align with NDA and W3C guidance.

11. Testing and Continuous Monitoring

- Perform ongoing accessibility audits after each content or structural update using WAVE, TAW, AChecker, and Lighthouse.
- Include screen reader compatibility checks with NVDA, JAWS, and VoiceOver.
- Engage users with disabilities in usability testing to gather real-world feedback.

12. Staff Training and Institutional Support

- Train web developers, designers, and content editors on accessibility standards (WCAG 2.1, EN 301 549, ISO/IEC 40500).
- Use the CEUD's toolkit and NDA's checklists as references.
- Establish internal accessibility documentation and quality assurance procedures to enforce universal design practices.

Conclusion

Implementing these actions will help UCD's Computer Science website meet WCAG 2.1 Level AA, fulfil legal accessibility requirements, and provide a more inclusive and user-friendly experience for all visitors.

PART C

i) How AI/ML Might Help the Accessibility Auditing Process

Artificial Intelligence (AI) and Machine Learning (ML) are reshaping web accessibility auditing by automating error detection, accelerating compliance checks, and enabling continuous monitoring. These technologies complement traditional manual testing by handling large-scale digital platforms efficiently, offering smart insights, and helping organisations meet standards such as WCAG 2.1, EN 301 549, and legislative frameworks like the EU Web Accessibility Directive and the European Accessibility Act (2025).

AI is integrated into tools like axe DevTools by Deque and Microsoft Azure Cognitive Services, which use machine learning models to identify common violations such as missing alt text (WCAG 1.1.1), incorrect heading structures (1.3.1), low colour contrast (1.4.3), and missing form labels (3.3.2). These tools apply rule-based algorithms alongside deep learning to generate accessible HTML feedback, reducing manual effort and increasing coverage (Deque Systems, 2024; Microsoft, 2024).

Computer vision—a branch of AI—enables platforms like Facebook's Automatic Alternative Text and Microsoft Azure's Vision API to describe images with meaningful text. These systems use deep learning to detect objects, scenes, and context, addressing the non-text content requirement of WCAG. This significantly improves access for screen reader users, particularly where manual descriptions are absent or insufficient (Facebook, 2024; Microsoft, 2024).

AI also enables simulation of assistive technology interactions. Advanced tools mimic keyboard-only navigation, test ARIA role interpretations, and emulate screen reader flows to uncover barriers such as inaccessible menus, focus traps, and unlabelled dynamic components. These simulations validate conformance with the Operable and Robust principles of WCAG and highlight usability issues that basic scanners miss.

Natural Language Processing (NLP) tools improve readability, aligning with WCAG 3.1. AI-driven platforms can assess vocabulary complexity, sentence structure, and tone. They suggest clearer phrasing for users with cognitive impairments or reading difficulties. Benetech and IBM offer accessibility tools that apply NLP to simplify text and enhance comprehension across educational and public-sector content (Benetech, 2024; IBM, 2024).

Continuous auditing is another benefit of AI. Unlike one-time scans, platforms like Siteimprove use AI to monitor websites over time, detecting regressions and triggering alerts when new updates introduce accessibility issues. Real-time dashboards provide actionable reports to help developers maintain long-term compliance, reducing the risk of falling short of accessibility obligations (World Economic Forum, 2024).

However, AI/ML tools are not a replacement for human judgment. They may overlook context-specific issues, such as ambiguous link text, unintuitive navigation, or emotional tone. Therefore, expert manual testing, usability evaluations, and input from users with disabilities remain essential. AI is best viewed as an enhancer—supporting, not replacing, human-centered design.

In conclusion, AI and ML significantly enhance the scope, speed, and accuracy of accessibility auditing. When combined with traditional methods, they form a powerful strategy for ensuring digital inclusivity, compliance with accessibility laws, and a better user experience for all.

References

- IBM. (2024). AI-Powered Accessibility Research. https://www.ibm.com/able/
- Microsoft. (2024). Azure Cognitive Services: Vision AI. https://azure.microsoft.com/en-us/products/cognitive-services/
- Deque Systems. (2024). axe DevTools. https://www.deque.com/products/axe-devtools/
- Benetech. (2024). Inclusive AI for Accessibility. https://benetech.org/what-we-do/accessibility/
- W3C. (2024). Web Accessibility Initiative (WAI). https://www.w3.org/WAI/