

Issues for 5055 MY241 - Desktop Chrome & Mobile iOS 2.2

Web Audit AA of 14 pages



Report generated on 21 June 2024

Table of contents

[Executive summary](#)

[Issues by priority](#)

[Breakdown of issues](#)

[Additional issues by priority](#)

[Breakdown of Additional issues](#)

[Links and references](#)

[ADR_20590-25 Footer - Significant elements overlapped at default text size and zoom levels](#) HIGH

[ADR_20590-38 Significant elements overlapped at default text size and zoom levels](#) HIGH

[ADR_20590-46 Interactive elements do not receive keyboard focus](#) HIGH

[ADR_20590-32 Informative images have no alt attribute](#) MEDIUM

[ADR_20590-77 Functional image has no text alternative](#) MEDIUM

[ADR_20590-102 Prerecorded video has no audio description or media alternative](#) MEDIUM

[ADR_20590-101 Prerecorded video has no audio description](#) MEDIUM

[ADR_20590-13 Unordered list is malformed](#) MEDIUM

[ADR_20590-26 Heading not marked up semantically](#) MEDIUM

[ADR_20590-29 Heading structure is inconsistent with the visual structure](#) MEDIUM

[ADR_20590-33 Unordered list is malformed](#) MEDIUM

[ADR_20590-37 Presentational table exposed to assistive technology](#) MEDIUM

[ADR_20590-41 Text uses heading semantics inappropriately](#) MEDIUM

[ADR_20590-42 Form fields and visual labels are not associated](#) MEDIUM

[ADR_20590-73 Text uses heading semantics inappropriately](#) MEDIUM

[ADR_20590-91 Button group not associated with legend](#) MEDIUM

[ADR_20590-92 Additional information is not associated with input field](#) MEDIUM

[ADR_20590-19 Reading order includes off-screen elements](#) MEDIUM

[ADR_20590-23 Listbox does not follow combobox in reading order](#) MEDIUM

[ADR_20590-61 Screen reader cursor not set onto modal dialog](#) MEDIUM

[ADR_20590-62 Screen reader cursor not restricted to modal dialog](#) MEDIUM

[ADR_20590-67 Meaningful visual structure is not represented by reading order](#) MEDIUM

[ADR_20590-109 Reading order of card disclosure elements includes off-screen elements](#) MEDIUM

[ADR_20590-96 Colour relied upon to communicate information](#) MEDIUM

[ADR_20590-11 Regular link text does not have sufficient contrast](#) MEDIUM

[ADR_20590-54 Regular text does not have sufficient contrast](#) MEDIUM

[ADR_20590-60 Regular button text does not have sufficient contrast](#) MEDIUM

[ADR_20590-69 Large text does not have sufficient contrast](#) MEDIUM

[ADR_20590-72 Regular text does not have sufficient contrast on two-coloured background \(blue text\)](#) MEDIUM

[ADR_20590-76 Regular combobox text does not have sufficient contrast](#) MEDIUM

[ADR_20590-82 Regular text does not have sufficient contrast on gradient background](#) MEDIUM

[ADR_20590-83 Regular link text does not have sufficient contrast on gradient background](#) MEDIUM

[ADR_20590-84 Large text does not have sufficient contrast on gradient background](#) MEDIUM

[ADR_20590-86 Regular button text does not have sufficient contrast on keyboard focus or mouse hover](#) MEDIUM

[ADR_20590-89 Regular text does not have sufficient contrast on two-coloured background \(white text\)](#) MEDIUM

[ADR_20590-103 Large text does not have sufficient contrast in video](#) MEDIUM

[ADR_20590-106 Large text does not have sufficient contrast on two-coloured background](#) MEDIUM

[ADR_20590-8 Zoom causes loss of meaningful content](#) MEDIUM

[ADR_20590-30 Footer - Zoom causes loss of meaningful content](#) MEDIUM

[ADR_20590-71 Zoom causes loss of functionality](#) MEDIUM

ADR_20590-63 Image contains embedded text MEDIUM
ADR_20590-6 Reflow causes loss of functionality MEDIUM
ADR_20590-7 Reflow causes loss of content MEDIUM
ADR_20590-20 Reflow requires users to scroll in multiple directions (no exceptions) MEDIUM
ADR_20590-34 Footer - Reflow causes loss of content MEDIUM
ADR_20590-12 Functional non-text content fails contrast when receiving keyboard focus and mouse hover MEDIUM
ADR_20590-40 Custom interactive element boundary has insufficient contrast MEDIUM
ADR_20590-85 Functional non-text content fails contrast MEDIUM
ADR_20590-87 Functional non-text content fails contrast on background image MEDIUM
ADR_20590-95 Functional non-text content fails contrast (ratings buttons) MEDIUM
ADR_20590-110 Custom focus indication fails contrast MEDIUM
ADR_20590-5 Text spacing causes content or functionality to overlap MEDIUM
ADR_20590-51 Text spacing causes content or functionality to be lost MEDIUM
ADR_20590-105 Content triggered by mouse hover or keyboard focus is not dismissable MEDIUM
ADR_20590-9 Same page link does not move focus when activated MEDIUM
ADR_20590-94 Interactive element cannot be activated with keyboard MEDIUM
ADR_20590-45 Mechanism for pausing automatically moving content does not work with keyboard MEDIUM
ADR_20590-49 Automatically moving content cannot be paused MEDIUM
ADR_20590-18 Navigation skip link does not move keyboard focus MEDIUM
ADR_20590-22 Background elements receive focus MEDIUM
ADR_20590-56 Non-interactive elements receive keyboard focus MEDIUM
ADR_20590-57 Focus is not managed appropriately when page is dynamically updated MEDIUM
ADR_20590-58 Focus order includes off-screen interactive elements MEDIUM
ADR_20590-59 Focus does not return to triggering element MEDIUM
ADR_20590-65 Focus not set onto expanded content MEDIUM
ADR_20590-68 Focus is not managed appropriately when content is removed from the page MEDIUM
ADR_20590-78 Links not self-explanatory or disambiguated by context. MEDIUM
ADR_20590-52 Interactive elements have no visible focus indicator MEDIUM
ADR_20590-66 Focused element is completely hidden by chat popup which cannot be dismissed MEDIUM
ADR_20590-15 Accessible name does not include visual name MEDIUM
ADR_20590-39 Target size for pointer input does not meet minimum size requirements MEDIUM
ADR_20590-16 Text content has no lang attribute MEDIUM
ADR_20590-64 Keyboard focus is moved to another element when element is adjusted MEDIUM
ADR_20590-74 Components with the same functionality are inconsistently identified MEDIUM
ADR_20590-75 Programmatic order of help is inconsistent within a set of pages MEDIUM
ADR_20590-55 Interactive elements do not have a permanent label MEDIUM
ADR_20590-14 Interactive element has no role MEDIUM
ADR_20590-17 Missing or malformed ARIA on editable combobox (with autocomplete) MEDIUM
ADR_20590-24 Expandable element is not programmatically identified MEDIUM
ADR_20590-43 ARIA has been defined on the incorrect element MEDIUM
ADR_20590-44 Video element with no accessible name MEDIUM
ADR_20590-47 Interactive elements with no accessible name MEDIUM
ADR_20590-48 Button element has no role MEDIUM
ADR_20590-93 Pressed element is not programmatically identified MEDIUM
ADR_20590-107 Card disclosure element has no role MEDIUM
ADR_20590-111 Interactive elements relies on accessible description for its accessible name MEDIUM
ADR_20590-53 Search results not programmatically identified MEDIUM
ADR_20590-79 Decorative icon not hidden from assistive technology LOW
ADR_20590-35 Unordered list not marked up semantically LOW
ADR_20590-70 Block quote not marked up semantically LOW
ADR_20590-90 Ordered list not marked up semantically LOW
ADR_20590-98 Colour relied upon to communicate information (job results) LOW
ADR_20590-100 Informative non-text content fails contrast (job results) LOW
ADR_20590-50 Inappropriate page titles LOW
ADR_20590-112 iFrame has no title LOW

Executive Summary

This report details the conformance of the project 5055 MY241 - Desktop Chrome & Mobile iOS 2.2 Web Audit AA of 14 pages. A summary of pages evaluated, and the dates the review was carried out on, can be found in the separate audit summary file, [adr_20590_summary.pdf](#).

Based on this evaluation, the project currently **does not meet** the target conformance standard.

This report has found that there are:

- **1 high priority** issue
- **84 medium priority** issues
- **8 low priority** issues
- **2 additional** issues

The [review summary section](#) gives an overview of the issues found, whilst detailed recommendations for these issues are covered in the [review breakdown](#).

The [additional issue review summary section](#) gives an overview of the additional issues found, whilst detailed recommendations for these issues are covered in the [additional issue breakdown](#). They are not strictly required by the target conformance standard, but project accessibility and usability would benefit if they were addressed.

Issues by Priority

These are issues which fail the target conformance standard. They require resolving for the project to be conformant with the standard.

High priority (1 issue)

Can't use: some users are excluded from using part of the product.

Specific user group(s) are excluded from using part of the product. For example, widgets cannot be opened with keyboard or critical visual information is not communicated by screen readers.

Issue	Success criteria Users affected	Status
ADR_20590-46 Interactive elements do not receive keyboard focus	2.1.1: Keyboard [2.0 A] 	OPEN

Medium priority (84 issues)

Causing problems: some users will experience problems using the product.

Specific user group(s) will experience problems but they are not prevented from using the product.

Nonetheless, they may find accessing and operating content frustrating, confusing, or otherwise challenging. For example, expanding and collapsing elements do not communicate this behaviour to screen reader users.

Issue	Success criteria Users affected	Status
ADR_20590-32 Informative images have no alt attribute	1.1.1: Non-text content [2.0 A] 	OPEN
ADR_20590-77 Functional image has no text alternative	1.1.1: Non-text content [2.0 A] 	OPEN
ADR_20590-102 Prerecorded video has no audio description or media alternative	1.2.3: Audio Description or Media Alternative (Prerecorded) [2.0 A] 	OPEN
ADR_20590-101 Prerecorded video has no audio description	1.2.5: Audio Description [2.0 AA] 	OPEN
ADR_20590-13 Unordered list is malformed	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-26 Heading not marked up semantically	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-29 Heading structure is inconsistent with the visual structure	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-33 Unordered list is malformed	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-37 Presentational table exposed to assistive technology	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-41 Text uses heading semantics inappropriately	1.3.1: Info and Relationships [2.0 A] 	OPEN

Issue	Success criteria Users affected	Status
ADR_20590-42 Form fields and visual labels are not associated	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-73 Text uses heading semantics inappropriately	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-91 Button group not associated with legend	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-92 Additional information is not associated with input field	1.3.1: Info and Relationships [2.0 A] 	OPEN
ADR_20590-19 Reading order includes off-screen elements	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-23 Listbox does not follow combobox in reading order	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-61 Screen reader cursor not set onto modal dialog	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-62 Screen reader cursor not restricted to modal dialog	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-67 Meaningful visual structure is not represented by reading order	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-109 Reading order of card disclosure elements includes off-screen elements	1.3.2: Meaningful Sequence [2.0 A] 	OPEN
ADR_20590-96 Colour relied upon to communicate information	1.4.1: Use of Color [2.0 A] 	OPEN
ADR_20590-11 Regular link text does not have sufficient contrast	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-54 Regular text does not have sufficient contrast	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-60 Regular button text does not have sufficient contrast	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-69 Large text does not have sufficient contrast	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-72 Regular text does not have sufficient contrast on two-coloured background (blue text)	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-76 Regular combobox text does not have sufficient contrast	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-82 Regular text does not have sufficient contrast on gradient background	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-83 Regular link text does not have sufficient contrast on gradient background	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-84 Large text does not have sufficient contrast on gradient background	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN

Issue	Success criteria Users affected	Status
ADR_20590-86 Regular button text does not have sufficient contrast on keyboard focus or mouse hover	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-89 Regular text does not have sufficient contrast on two-coloured background (white text)	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-103 Large text does not have sufficient contrast in video	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-106 Large text does not have sufficient contrast on two-coloured background	1.4.3: Contrast (Minimum) [2.0 AA] 	OPEN
ADR_20590-8 Zoom causes loss of meaningful content	1.4.4: Resize text [2.0 AA] 	OPEN
ADR_20590-30 Footer - Zoom causes loss of meaningful content	1.4.4: Resize text [2.0 AA] 	OPEN
ADR_20590-71 Zoom causes loss of functionality	1.4.4: Resize text [2.0 AA] 	OPEN
ADR_20590-63 Image contains embedded text	1.4.5: Images of Text [2.0 AA] 	OPEN
ADR_20590-6 Reflow causes loss of functionality	1.4.10: Reflow [2.1 AA] 	OPEN
ADR_20590-7 Reflow causes loss of content	1.4.10: Reflow [2.1 AA] 	OPEN
ADR_20590-20 Reflow requires users to scroll in multiple directions (no exceptions)	1.4.10: Reflow [2.1 AA] 	OPEN
ADR_20590-34 Footer - Reflow causes loss of content	1.4.10: Reflow [2.1 AA] 	OPEN
ADR_20590-12 Functional non-text content fails contrast when receiving keyboard focus and mouse hover	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-40 Custom interactive element boundary has insufficient contrast	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-85 Functional non-text content fails contrast	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-87 Functional non-text content fails contrast on background image	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-95 Functional non-text content fails contrast (ratings buttons)	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-110 Custom focus indication fails contrast	1.4.11: Non-text Contrast [2.1 AA] 	OPEN
ADR_20590-5 Text spacing causes content or functionality to overlap	1.4.12: Text Spacing [2.1 AA] 	OPEN
ADR_20590-51 Text spacing causes content or functionality to be lost	1.4.12: Text Spacing [2.1 AA] 	OPEN

Issue	Success criteria Users affected	Status
ADR_20590-105 Content triggered by mouse hover or keyboard focus is not dismissable	1.4.13: Content on Hover or Focus [AA] 	OPEN
ADR_20590-9 Same page link does not move focus when activated	2.1.1: Keyboard [2.0 A] 	OPEN
ADR_20590-94 Interactive element cannot be activated with keyboard	2.1.1: Keyboard [2.0 A] 	OPEN
ADR_20590-45 Mechanism for pausing automatically moving content does not work with keyboard	2.2.2: Pause, Stop, Hide [2.0 A] 	OPEN
ADR_20590-49 Automatically moving content cannot be paused	2.2.2: Pause, Stop, Hide [2.0 A] 	OPEN
ADR_20590-18 Navigation skip link does not move keyboard focus	2.4.1: Bypass Blocks [2.0 A] 	OPEN
ADR_20590-22 Background elements receive focus	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-56 Non-interactive elements receive keyboard focus	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-57 Focus is not managed appropriately when page is dynamically updated	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-58 Focus order includes off-screen interactive elements	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-59 Focus does not return to triggering element	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-65 Focus not set onto expanded content	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-68 Focus is not managed appropriately when content is removed from the page	2.4.3: Focus Order [2.0 A] 	OPEN
ADR_20590-78 Links not self-explanatory or disambiguated by context.	2.4.4: Link Purpose (In Context) [2.0 A] 	OPEN
ADR_20590-52 Interactive elements have no visible focus indicator	2.4.7: Focus Visible [2.0 AA] 	OPEN
ADR_20590-66 Focused element is completely hidden by chat popup which cannot be dismissed	2.4.11: Focus Not Obscured (Minimum) [2.2 AA] 	OPEN
ADR_20590-15 Accessible name does not include visual name	2.5.3: Label in Name [2.1 A] 	OPEN
ADR_20590-39 Target size for pointer input does not meet minimum size requirements	2.5.8: Target Size (Minimum) [2.2 AA] 	OPEN
ADR_20590-16 Text content has no lang attribute	3.1.2: Language of Parts [2.0 AA] 	OPEN
ADR_20590-64 Keyboard focus is moved to another element when element is adjusted	3.2.2: On Input [2.0 A] 	OPEN

Issue	Success criteria Users affected	Status
ADR_20590-74 Components with the same functionality are inconsistently identified	3.2.4: Consistent Identification [2.0 AA] 	OPEN
ADR_20590-75 Programmatic order of help is inconsistent within a set of pages	3.2.6: Consistent Help [2.2 A] 	OPEN
ADR_20590-55 Interactive elements do not have a permanent label	3.3.2: Labels or Instructions [2.0 A] 	OPEN
ADR_20590-14 Interactive element has no role	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-17 Missing or malformed ARIA on editable combobox (with autocomplete)	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-24 Expandable element is not programmatically identified	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-43 ARIA has been defined on the incorrect element	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-44 Video element with no accessible name	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-47 Interactive elements with no accessible name	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-48 Button element has no role	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-93 Pressed element is not programmatically identified	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-107 Card disclosure element has no role	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-111 Interactive elements relies on accessible description for its accessible name	4.1.2: Name, Role, Value [2.0 A] 	OPEN
ADR_20590-53 Search results not programmatically identified	4.1.3: Status Messages [2.1 AA] 	OPEN

Low priority (8 issues)**Minor and cosmetic issues: something is wrong, but it won't affect users too much.**

Specific user group(s) will experience minor problems. They are not excluded from the product and won't face major obstacles. Nonetheless, these issues will make interaction bothersome and the product seem unpolished. For example, a small number of cosmetic images with redundant alt text.

Note that our priorities are context-sensitive. For example, if a page is swamped with occurrences of the same low priority issue, it may be bumped to medium priority.

Issue	Success criteria Users affected	Status
ADR_20590-79 Decorative icon not hidden from assistive technology	1.1.1: Non-text content [2.0 A] 	OPEN

Issue	Success criteria Users affected	Status
ADR_20590-35 Unordered list not marked up semantically	1.3.1: Info and Relationships [2.0 A] 	 OPEN
ADR_20590-70 Block quote not marked up semantically	1.3.1: Info and Relationships [2.0 A] 	 OPEN
ADR_20590-90 Ordered list not marked up semantically	1.3.1: Info and Relationships [2.0 A] 	 OPEN
ADR_20590-98 Colour relied upon to communicate information (job results)	1.4.1: Use of Color [2.0 A] 	 OPEN
ADR_20590-100 Informative non-text content fails contrast (job results)	1.4.11: Non-text Contrast [2.1 AA] 	 OPEN
ADR_20590-50 Inappropriate page titles	2.4.2: Page Titled [2.0 A] 	 OPEN
ADR_20590-112 iFrame has no title	4.1.2: Name, Role, Value [2.0 A] 	 OPEN

Breakdown of issues

ADR_20590-46 HIGH PRIORITY OPEN

Interactive elements do not receive keyboard focus

Success Criteria 2.1.1: Keyboard [2.0 A]

Devices Desktop

Pages Affected 03

Users Affected Motor

Component Button

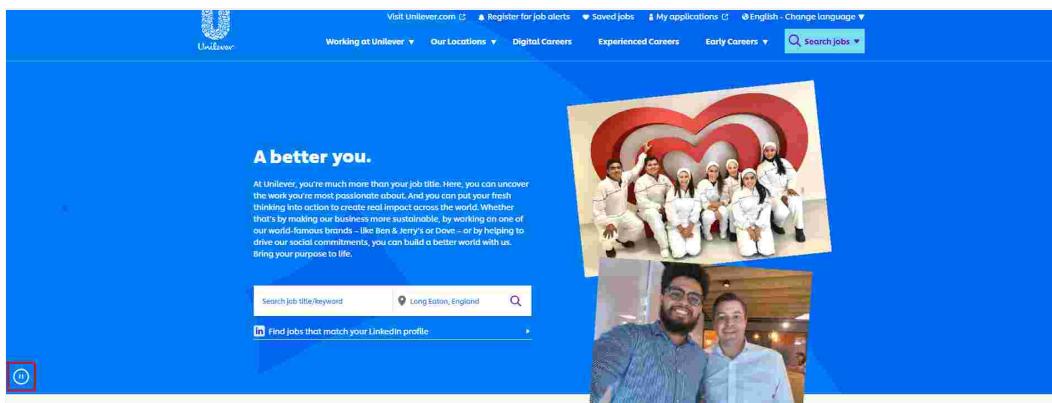
What happened

Interactive elements must be entirely compatible with a keyboard.

Users should never need to use a mouse. Otherwise, screen reader, keyboard, and switch users, may be unable to control the interactive elements. There are interactive elements that do not receive focus when tabbing through the page. As a result, they cannot be operated with a keyboard.

Where it happened

This can be seen on page 3, on desktop, with the pause/play button provided for the hero banner video. The pause/play button does not receive keyboard focus. As a result, it cannot be operated using a keyboard.



Page 3: desktop, hero banner video pause/play button

```
<div style="position: absolute; z-index: -1; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-position: 50% 0%; background-image: none;">
  <video loop muted style="margin: auto; position: absolute; z-index: -1; top: 0%; left: 50%; transform: translate(-50%, 0%); visibility: visible; opacity: 1; width: 1899px; height: auto;"></video>
</div>
<div class="play-pause-button" paused="false"></div>
```

Page 3: DOM, hero banner video pause/play button

Steps to reproduce

1. Observe the interactive elements.
2. Attempt to navigate onto them with a keyboard using Tab.
3. Observe that the elements do not receive keyboard focus.
4. Observe that there is not an alternative way to access the same information or functionality.

Recommendation

Make sure that interactive elements can be navigated onto with a keyboard. They should receive focus when tabbing through the page. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` which can be tabbed onto by default.

To add custom components into the focus order, use `tabindex="0"`. Make sure that this is used on the container with the interactive role. For example:

```
<button>Submit</button>
<span tabindex="0" role="button">Submit</span>
```

Make sure that when the component has been added to the focus order that it responds to keyboard commands. For example, make sure that it can be tabbed onto and activated with Return or Space. Alternatively, make sure that it can be adjusted with arrow keys, typed into, or whatever is appropriate for the component.

If you cannot make the component compatible with a keyboard, provide an alternative that gives users access to the same functionality. For example, a date-entry input field can be provided as an alternative to a date picker component.

For more information, see [Google Developer - Using Tabindex](#), [MDN - Keyboard Accessibility](#), [Keyboard-navigable JavaScript widgets](#), and [WAI ARIA - Developing a Keyboard Interface](#). For practical examples, see [WAI ARIA Practices](#) and [Inclusive Components](#).

[Return to priority table](#)

ADR_20590-32 MEDIUM PRIORITY OPEN

Informative images have no alt attribute

Success Criteria 1.1.1: Non-text content [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12

Users Affected Vision

Component Image

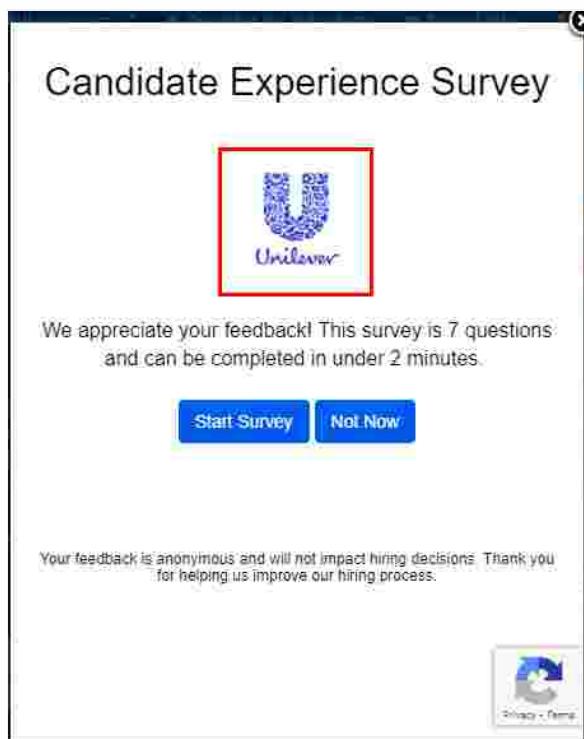
What happened

Non-text content must have appropriate alternatives. This is required for the content to be understood by all users. Images, icons and so on may otherwise be miscommunicated by assistive technology. There are informative images with no **alt** attribute whatsoever.

We have classed the images as informative as they communicate unique or meaningful information.

Where it happened

This can be found on page 12 on desktop and iOS, with the “Unilever” logo present below the “Candidate Experience Survey” heading. This image has no **alt** attribute.



Page 12: desktop, Unilever logo image

```
<div ng-if="noIntro" class="intro-question ng-scope compact" dir="ltr" ng-class="(compact:compactView, 'margin-top-40':multipleLanguages)" ng-hide="clicked">
  <h1 class="survey-title ng-binding compact" ng-class="(compact:compactView)" ng-style="['color':survey.titleColor || 'black']" style="color: black;">
    <div class="intro-html-wrap ck-content ng-hide" ng-hide="compactView"></div>
  </div>
  <div class="intro-html-wrap" ng-hide="!compactView">
    <div class="ck-content">
      <div ng-bind-html="surveyIntroScaledImages[0].trusted" class="survey-intro_compact ng-binding" ng-style="['color':survey.titleColor || 'black']">
        <figure class="Image">
          
        </figure>
        <p style="text-align:center;">...</p>
      </div>
    </div>
  </div>
</div>
```

Page 12: DOM, Unilever logo image with no alt attribute

Steps to reproduce

1. Inspect the informative images.
2. Confirm that they do not have an **alt** attribute.

Recommendation

Images marked up with an `` tag must have an **alt** attribute. Use the attribute to provide an appropriate text alternative. Informative images that convey meaningful or unique information need a descriptive alt attribute. For example:

```

```

For more information, see [Web Accessibility Tutorials - Informative Images](#), and [Alt Attribute Decision Tree](#).

[Return to priority table](#)

ADR_20590-77 MEDIUM PRIORITY OPEN

Functional image has no text alternative

Success Criteria 1.1.1: Non-text content [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 08, 11

Users Affected Vision

Component Image

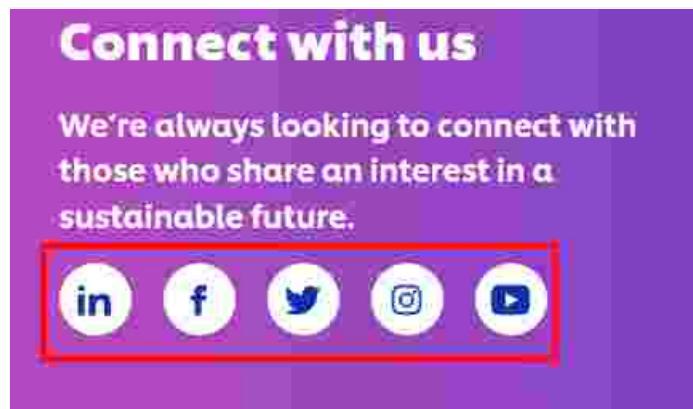
What happened

Non-text content must have appropriate alternatives. This is required for the content to be understood by all users. Images, icons and so on may otherwise be miscommunicated by assistive technology. There are functional images without appropriate text alternatives.

We have classed the images as functional because they have functionality associated with them.

Where it happened

This can be found on page 8 on desktop and iOS with the social media icon links under “Connect with us”. The images do not have alternative text.



Page 8: desktop, social media icon links

```
<ul class="social_media"> #new
  <li> #item
    <a href="https://www.linkedin.com/company/unilever/"> 
    </a>
  </li>
  <li> #item
    <a href="https://www.facebook.com/UnileverCareers"> 
    </a>
  </li>
  <li> #item
    <a href="https://twitter.com/unilever?ref_src=twsrc%5Egoogle%5Ctwcamp%5Eescript%5Ctwterm%5Eauthor"> 
    </a>
  </li>
</ul>
```

Page 8: DOM, social media icon links do not have alternative text

This can also be found on page 11 on desktop and iOS with the step numbers under “Application process”. The numbers are embedded in images, which do not have alternative text.



Page 11: desktop, informative number images under “Application process”

A screenshot of a browser's developer tools showing the DOM structure for the 'Application process' section. The DOM includes an `<h2>`, a `<div>` for the grid, and a `<div>` for the first step. Inside the first step, there is a `<button>` element with the class `application-process_number-button_one`. The `Name` property of this button is highlighted with a red box in the developer tools' computed properties panel. The value of the `Name` property is `"Online application +"`.

Page 11: DOM, number “1” under Application process does not have alternative text

Steps to reproduce

1. Inspect the functional images.
2. Confirm that they do not have appropriate text alternatives.

Recommendation

Images marked up with an `` tag must have an `alt` attribute. Use the attribute to provide an appropriate text alternative. Functional images need to communicate the action or outcome of the functionality, such as the destination with image links. For example:

```
<a href="#"></a>
```

Inline SVG images can use `<title>`, `<desc>`, and `aria-labelledby` for the same purpose. For example:

```
<svg aria-labelledby="uniqueTitleID uniqueDescID">
<title id="uniqueTitleID">The title of the SVG</title>
<desc id="uniqueDescID">A longer, more complete description for complex graphics.</desc>
...
</svg>
```

For more information, see [Web Accessibility Tutorials - Functional Images](#), [Alt Attribute Decision Tree](#), and [CSS Tricks - Accessible SVGs](#).

[Return to priority table](#)

ADR_20590-102 MEDIUM PRIORITY OPEN

Prerecorded video has no audio description or media alternative

Success Criteria 1.2.3: Audio Description or Media Alternative (Prerecorded) [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Video

What happened

Videos must have accurate and informative audio descriptions or media alternative.

There are prerecorded videos without audio descriptions or media alternative, such as a transcript. The video content relies on perceiving the visuals. This may particularly disadvantage low vision and blind users who may be excluded by the reliance on the visuals. Users may be unable to access the information in the video.

Note that this WCAG A success criterion permits transcripts, unlike its AA counterpart (Audio Descriptions) which requires audio descriptions.

Where it happened

This can be seen on page 13 on desktop and iOS with the “Digital Careers at Unilever” video. The video conveys information through images and on-screen text, such as “Tackle inequality in the Metaverse”. This information is only available visually. There is no transcript or audio description.



Page 13: desktop, “Digital Careers at Unilever” video conveys information through on-screen text but does not have audio description

Steps to reproduce

1. Navigate onto the prerecorded videos.

2. Observe that there are no audio descriptions.
3. Observe that there is no media alternative, such as a transcript.

Recommendation

Make sure that videos have accurate audio descriptions or media alternative, such as a transcript. Meaningful visual action and content needs to be included, such as informative on-screen text, such that the meaningful information in the video can be entirely understood from the audio descriptions or media alternative.

The audio descriptions can be provided as a secondary audio track or burned-in to a separate video. For more information, see [WAI - Audio Description of Visual Information](#), [ACB - Guidelines for Audio Descriptions](#), and [WAI - Accessibility Perspective Videos](#) which includes videos with audio descriptions.

The alternative can be provided on the same page. For example, a transcript can be provided underneath the video. Another option is providing a link to the alternative, even if this loads a new page. For more information, see [WAI Tutorials - Transcripts](#) and [WAI - Accessibility Perspectives](#) which includes videos with transcripts.

[Return to priority table](#)

ADR_20590-101 MEDIUM PRIORITY OPEN

Prerecorded video has no audio description

Success Criteria 1.2.5: Audio Description [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Video

What happened

Videos must have accurate and informative audio descriptions.

There are prerecorded videos without audio descriptions. The video content relies on perceiving the visuals. This may particularly disadvantage low vision and blind users who may be excluded by the reliance on the visuals. Users may be unable to access the information in the video.

Note that this WCAG AA success criterion requires audio descriptions, unlike its A counterpart (Audio Description or Media Alternative) which permits transcripts.

Where it happened

This can be seen on page 13 on desktop and iOS with the “Digital Careers at Unilever” video. The video conveys information through images and on-screen text, such as “Tackle inequality in the Metaverse”. This information is only available visually. There is no audio description.



Page 13: desktop, “Digital Careers at Unilever” video conveys information through on-screen text but does not have audio description

Steps to reproduce

1. Navigate onto the prerecorded videos.
2. Observe that there are no audio descriptions.

Recommendation

Make sure that videos have accurate audio description. Meaningful visual action and content needs to be included in the audio descriptions, such as informative on-screen text, such that the meaningful information in the video can be entirely perceived from the audio track.

Note that there is a relaxed provision for “talking head” videos, such as those with a single speaker talking straight to the camera. With such videos, this success criterion can be passed with a text alternative. For more information, see [WCAG - Alternative to Describe a Talking Head Video](#).

The audio descriptions can be provided as a secondary audio track or burned-in to a separate video. For more information, see [WAI - Audio Description of Visual Information](#), [ACB - Guidelines for Audio Descriptions](#), and [WAI - Accessibility Perspectives](#) which includes videos with audio descriptions.

[Return to priority table](#)

ADR_20590-13 MEDIUM PRIORITY OPEN

Unordered list is malformed

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop

Pages Affected 01

Users Affected Vision

Component List

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There are elements that are visually communicated as being interrelated, unordered list items. However, malformed list semantics have been used.

Where it happened

This can be seen on page 1 with the list containing the top row of navigation links and buttons such as “Visit Unilever.com”, “Register for job alerts”, and so on. This list has been marked up with **role="list"** on the container and **role="listitem"** on its children. However, the list is malformed because:

- The “Change language” button is not marked up as a list item.
- The list of language options is inappropriately nested as a direct child of the list despite not being a list item.

Note that, for the elements with **role="listitem"**, the list role overrides the native link or button role. This has been raised in ADR_20590 - 14.



Page 1: desktop, navigation links beginning “Visit Unilever.com”

A screenshot of the Unilever website's DOM structure. The code shows a list item (``) containing several navigation links and a "Change language" button. The "Change language" button is marked with the `aria-label` "Submenu for international versions of this website". The list item is highlighted with a red border, and the entire code block is shown below:

```
<div class="core-header__right"> ...
<div class="upper-nav" role="list">
  <a class="upper-nav__element external-link" href="https://unilever.com" target="_blank" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Unilever.com">Visit Unilever.com</a>
  <span class="visually-hidden" opens in a new window/>
  ...
</li>
<button class="desktop-only desktop-only--inline upper-nav__element upper-nav__element--alerts" id="hide-in-russia" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Unilever.com">Contact Us</button>
<a class="upper-nav__element upper-nav__element--saved-jobs" href="/saved-jobs" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Saved jobs">Saved jobs</a>
<a class="upper-nav__element upper-nav__element--login external-link" href="https://unilever-wd3.myworkdayjobs.com/unilever_Experience" target="_blank" data-custom-event="true" data-custom-category="AT&T_login" data-custom-label="AT&T Sign-in button">AT&T Sign-in button</a>
<button class="upper-nav__element upper-nav__element--language-switcher" aria-expanded="false" aria-label="Submenu for international versions of this website">
  ...
  <span id="lang-switcher-current-lang">
    <span class="lang-reduced">EN</span>
    <span class="lang-extended">English</span>
    ...
    <span class="lang-extended">language</span>
  </span>
  ...
</button>
<div class="language-switcher-list" role="list">
  <a href="https://careers.unilever.com.br" role="listitem">Português</a>
  <a href="https://karriere.unilever.de/" role="listitem">Deutsch</a>
  <a href="https://careers.unilever.es/" role="listitem">Español</a>
  <a href="https://careers.unilever.fr/" role="listitem">Français</a>
  <a href="https://careers.unilever.com.tr" role="listitem">Türkçe</a>
</div>
</div>
```

Page 1: DOM, list beginning with “Visit Unilever.com” is malformed

This can also be seen on page 1 with the list for the second level of navigation links and buttons such as “Working at Unilever”, “Our Locations”, and so on. Within this list, the container for each submenu, such as the “Working at Unilever” submenu, is inappropriately marked up as a list item with **role="listitem"**.



Page 1: desktop, navigation links beginning "Working at Unilever", with "Working at Unilever" submenu expanded

```
<nav class="lower-nav" aria-label="Lower navigation menu"></nav>
<button class="mobile-menu-toggle mobile-only" aria-expanded="false" aria-label="Navigation menu"></button>
<nav class="main-navigation" role="list"></nav>
<button class="main-navigation__link sub-links-trigger open" aria-expanded="true" data-submenu-trigger="1" role="listitem" aria-label="Working at Unilever, Toggles submenu">
  "Working at Unilever"
  <ul>
    <a href="/working-at-unilever" role="listitem" data-custom-event="true"></a>
    <a href="/diversity-and-inclusion" role="listitem" data-custom-event="true" data-custom-category="Custom" data-custom-label="Home - D&I"></a>
    <a href="/benefits-learning-wellbeing" role="listitem">Benefits, Learning & Wellbeing</a>
  </ul>
</button>
<button class="main-navigation__link sub-links-trigger" aria-expanded="false" data-submenu-trigger="2" role="listitem" aria-label="Our locations, Toggles submenu"></button>
```

Page 1: DOM, container for "Working at Unilever" submenu is marked up as a list item

Steps to reproduce

1. Observe the unordered list items.
2. Inspect the elements in the DOM.
3. Observe that the unordered list semantics are malformed.

Recommendation

Mark up the unordered list using `` and ``. Make sure that each set of list items is contained in a single `` and that the list items themselves are contained within individual `` tags. Only use separate `` tags when the list items are contained logically and visually in separate, distinct lists.

Make sure not to override the semantics of interactive elements with list item semantics. Nest interactive elements inside list items instead of giving them list item roles.

For example:

```
<ul>
  <li>
    <a href="...">Visit Unilever.com<span class="visually-hidden">opens in a new window</span></a>
  </li>
  ...
  <li>
    <button aria-expanded="true">EN English - Change language</button>
    <ul>
      <li>
        <a href="..." lang="pt">Português</a>
      </li>
      ...
    </ul>
  </li>
</ul>
```

For more information, see [WAI Tutorials - Unordered Lists](#).

[Return to priority table](#)

ADR_20590-26 MEDIUM PRIORITY OPEN

Heading not marked up semantically

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 04, 05, 08

Users Affected Vision

Component Accordion or Disclosure, Heading

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganised.

There are elements that title or categorise other content. These elements have not been marked up as headings.

Where it happened

This can be seen on page 4 on desktop and iOS with the name of each job in the results list. These elements title the text underneath them and the “Save Job” button for each job. However, they are not semantically marked up as level 3 headings.

The screenshot shows a search results page with a header for filtering by category, location, state, city, and remote working status, with a 'Remove filters' link. Below the header, a large blue header reads "27 Search results". Six job listings are displayed in a grid. Each listing includes the job title, location, and a heart icon. The job titles are highlighted with red boxes: "Senior Product Development Scientist - Health and Wellbeing", "Territory Development Manager - Indiana and Ohio", "Territory Development Manager - New York", "Associate Finance Manager, Yasso", "Full-Time - Guest Experience Host, Scoop Shop", and "Associate Finance Manager - Multiple Openings !".

Page 4: desktop, headings bookmarklet, job names are not marked up as headings

```
<h1>27 Search results</h1>
<section id="search-results-list">
  <ul class="global-job-list-white" role="list">
    <li class="color-default">
      <a href="/job/englewood-cliffs/senior-product-development-scientist-health-and-wellbeing/34155/65273848352" data-job-id="65273848352" data-org-id="34155" data-aria-label="Job">
        Senior Product Development Scientist - Health and Wellbeing</a>
        Englewood Cliffs, New Jersey
      <button class="js-save-job-btn" aria-pressed="false" data-job-id="65273848352" data-org-id="34155" role="button" aria-label="Save Job">
        Save job
      </button>
    </li>
  </ul>
</section>
```

Page 4: DOM, “Senior Product Development Scientist - Health and Wellbeing” job name is not marked up as a heading

This can be seen on page 5 on desktop and iOS, with “Background & Purpose of the Job”, “Who You Are & What You’ll Do”, “What You’ll Need To Succeed” and “What We Can Offer You” headings. These headings are not semantically marked up as headings of level 2.

The screenshot shows a job listing page. At the top, there are three buttons: 'Apply now' (with a checkmark icon), a blue arrow pointing right, 'Save job', and a heart icon. Below these buttons, there are two sections with red borders: 'Background & Purpose of the Job' and 'Who You Are & What You'll Do'. The 'Background & Purpose of the Job' section contains text about Yasso's growth trajectory and creative marketing. The 'Who You Are & What You'll Do' section contains text about the role's responsibilities.

Page 5: desktop, "Background & Purpose of the Job" and "Who You Are & What You'll Do" are visually styled headings

```
<section class="job-description" data-selector="name"="jobdetails" data-org-id="34155" data-job-id="64916262304" data-save-job="true">
  > <div class="save-and-apply-buttons"> ... </div> +[flex]
  > <div class="ats-description">
    > <div>
      > <div>
        > <p>
          > <b>
            <span>Background & Purpose of the Job</span> == #B
          </b>
          <span></span>
        </p>
      </div>
    > <div> ... </div>
    > <div> ... </div>
    > <div> ... </div>
    > <div> ... </div>
  > </div>
```

Page 5: DOM, "Background & Purpose of the Job" is not marked up semantically as a heading

This can be seen on page 5 on desktop and iOS, with "Plan", "Bonus", "Benefits" and "Long-Term Incentive (LTI)" headings. These headings are not semantically marked up as headings of level 3.

The screenshot shows a job listing page with several sections highlighted with red boxes. The first section, 'Pay', contains text about compensation decisions. The second section, 'Bonus', contains text about bonus eligibility. The third section, 'Long-Term Incentive (LTI)', contains text about LTI eligibility. The fourth section, 'Benefits', contains text about the company's benefits plan.

Page 5: desktop, "Plan", "Bonus" and so on visually styled headings

```

<p></p>
  <p>
    <b>Pay:</b>
    "The pay range for this position is $110,700 to $166,000 . Unilever takes into consideration a wide range of factors that are utilized in making compensation decisions including, but not limited to, skill sets, experience and training, licensure and certifications, qualifications and education, and other business and organizational needs."
  </p>
  <p>
    <b>Bonus:</b>
    "This position is bonus eligible."
  </p>
  <p>
    <b>Long-Term Incentive (LTI):</b>
    "This position is LTI eligible."
  </p>
  <p>
    <b>Benefits:</b>
    "Unilever employees are eligible to participate in our benefits plan. Should the employee choose to participate, they can choose from a range of benefits to include, but is not limited to, health insurance (including prescription drug, dental, and vision coverage), retirement savings benefits, life insurance and disability benefits, parental leave, sick leave, paid vacation and holidays, as well as access to numerous voluntary benefits. Any coverages for health insurance and retirement benefits will be in accordance with the terms and conditions of the applicable plans and programs." 
  </p>

```

Page 5: DOM, “Plan”, “Bonus” and so on are not marked up semantically as a heading

This can be seen on page 8 on desktop and iOS with the accordion buttons under “Explore Where You Fit Across Our Digital Teams”, such as “Digital R&D”, “Digital Supply Chain”, and so on. These elements title the contents of the accordion panels and the “View jobs” buttons. However, they are not marked up as headings.

Page 8: desktop, accordion buttons under “Explore Where You Fit Across Our Digital Teams”

```

<div class="accordionLeft">
  <div>
    <a href="#accordion1" aria-expanded="true" aria-controls="accordion1" class="accordionTitle js-accordionTrigger is-collapsed is-expanded">
      :before
      "Digital R&D"
    </a>
  </div>
  <div class="accordionContent left accordionItem is-expanded animateIn" aria-hidden="false">
    <div class="accordionRight">
      <div class="accordionContent right accordionItem is-expanded animateIn" id="accordion1" aria-hidden="false">
        <p>
          "Our R&D teams believe that innovation occurs in truly connected environments where creative ideas are encouraged. We have more than 5,000 research and development experts working across 60 countries, including in six global labs and ten regional hubs, working to create tomorrow's innovation through the power of consumer insights, best in class research, and technology and data."
        </p>
        <p>
          "People capabilities: innovation mindset, agile, E2E consumer experience creation."
        </p>
      </div>
    </div>
  </div>

```

Page 8: DOM, accordion button “Digital R&D” is not marked up as a heading

Steps to reproduce

1. Visually note the headings.
2. Inspect the elements in the DOM.
3. Note that the headings have not been marked up, such as `<h1>`, `<h2>`, and so on.

Recommendation

When elements title or categorise other content, use headings with appropriate heading levels. Make sure that the heading levels used correctly communicate the structure of the page, such as the parent and child relationship between the headings. For example:

```
<h1>Cooking techniques</h1>
<!-- some content here -->
<h2>Cooking with oil</h2>
<!-- some content here -->
<h3>Sautéing</h3>
<!-- some content here -->
<h3>Deep frying</h3>
<!-- some content here -->
<h2>Cooking with butter</h2>
<!-- some content here -->
```

For more information, see [WAI Tutorials - Headings](#).

[Return to priority table](#)

ADR_20590-29 MEDIUM PRIORITY OPEN

Heading structure is inconsistent with the visual structure

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 04, 08, 12

Users Affected Vision

Component Heading

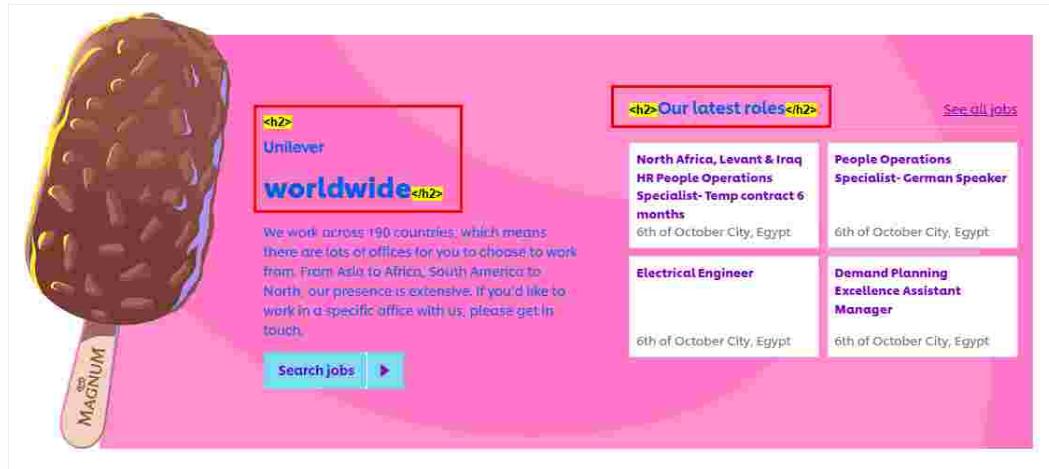
What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There are headings that have a visual parent-child relationship. However, this structure is not correctly expressed by the heading levels.

Where it happened

This can be seen on page 3, on desktop and iOS, with the headings in the “Unilever worldwide” section. Visually, the “Our latest roles” heading is presented as a child of “Unilever worldwide”. However, they are both declared as heading level 2.



Page 3: desktop, headings bookmarklet, “Out latest roles” and “Unilever worldwide” are both declared as headings level 2

```
<section class="localisation-jobs-section max-width-wrapper max-width-wrapper--narrow"> flex
  <div class="localisation-jobs-section_left">
    <div class="personalization" data-post-module="true" data-ajax-url="//careers.unilever.com/module/postmodule" data-module-name="localised_copy_in_homepage" data-module-type="Personalization" data-personalized="true" data-cookie="PersonalizationCookie" data-svs="2">
      <div class="localisation-jobs-section_image localisation-jobs-section_image--magnum"></div>
    </div>
    <div class="localisation-jobs-section_copy">
      <h2>
        <span>Unilever</span>
        <span>worldwide</span>
      </h2>
      <p>We work across 190 countries, which means there are lots of offices for you to choose to work from. From Asia to Africa, South America to North, our presence is extensive. If you'd like to work in a specific office with us, please get in touch.</p>
      <a href="/search-jobs" class="button-with-arrow button-with-arrow--teal">Search jobs</a>
    </div>
  </div>
  <div class="localisation-jobs-section_right">
    <div class="job-list" data-selector-name="joblist" data-post-module="true" data-ajax-url="//careers.unilever.com/module/postmodule" data-module-name="location dependent job list" data-module-type="JobList">
      <div class="job-list_header" flex>
        <h2>Our latest roles</h2>
        <a href="/search-jobs">See all jobs</a>
      </div>
      <ul class="global-job-list--blue" flex>...</ul>
    </div>
  </div>
</section>
```

Page 3: DOM, “Out latest roles” and “Unilever worldwide” are both declared as headings level 2

This can also be seen on page 4 on desktop and iOS with the search results heading (such as “6 Search results”) and the “Filtered by” heading. The “6 Search results” heading is inappropriately marked up as a level 1 heading, which does not match the visual structure where it is a sibling of the level 2 heading “Filter your search”. The heading “Filtered by:”, which is a child of “6 Search results”, is level 2 instead of level 3.

The screenshot shows a web page with a yellow header bar containing the text "Recruitment scams & fraud warning". Below this is a message about avoiding recruitment scams. The main content area has a yellow header "Filter your search" followed by dropdown menus for Category, Location, State, and City, and a "Remove filters" link. A yellow box highlights the heading "6 Search results". Below this, another yellow box highlights the heading "Filtered by:" followed by the text "Supply Chain" and a close button. Three job listing cards are shown: "Market Collaboration Planner, Yasso" (Englewood Cliffs, New Jersey), "Associate Innovation Planning Manager, Yasso" (Englewood Cliffs, New Jersey), and "Manufacturing Lead" (Silkston, Missouri). Each card has a heart icon.

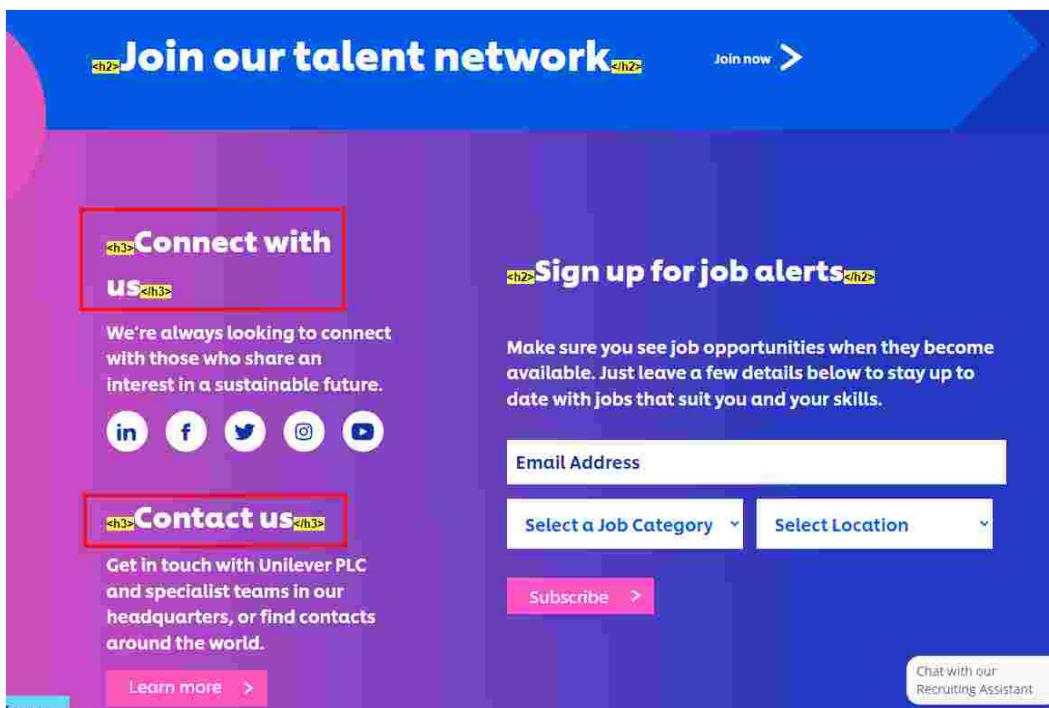
Page 4: desktop, headings bookmarklet, “6 Search results” is marked up as level 1 and “Filtered by:” is marked up as level 2, which does not match the visual structure

A screenshot of the DOM showing the HTML structure. It includes a red box around the heading "6 Search results" and a red box around the heading "Filtered by:". The code snippet shows the following structure:

```
<h1>6 Search results</h1>
<section id="search-results-list">
  <div id="applied-filters" class="search-results-options">
    <h2 id="applied-filters-label">Filtered by:</h2>
    <ul aria-labelledby="applied-filters-label" role="list"><li></li></ul>
  </div>
  <ul class="global-job-list--white" role="list"><li></li></ul>
</section>
```

Page 4: DOM, “6 Search results” is marked up as level 1 and “Filtered by:” is marked up as level 2

This can also be seen on page 8 on desktop and iOS with the “Connect with us” and “Contact us” headings. These headings are marked up as level 3, but they are not children of the level 2 heading “Join our talent networks” above them.



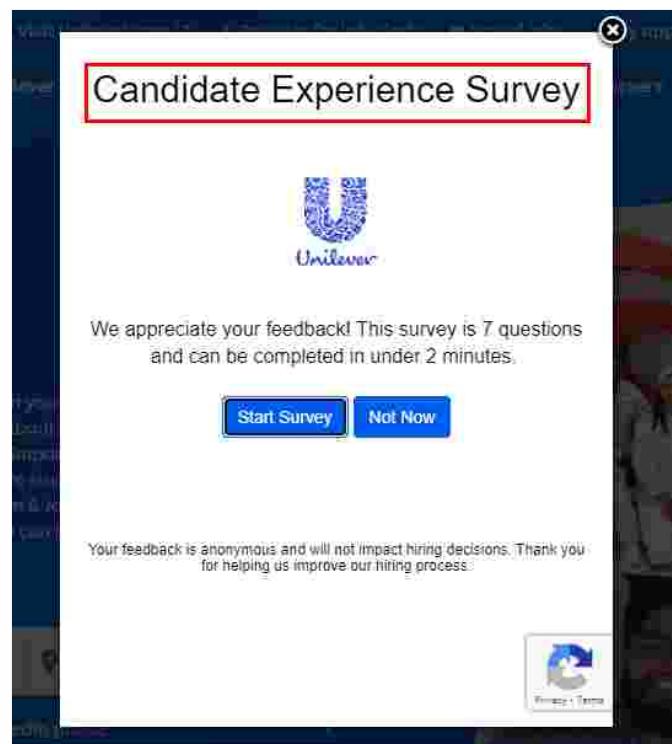
Page 8: desktop, headings bookmarklet, “Connect with us” and “Contact us” are marked up as level 3

```
<div class="two-two-grid max-width-wrapper max-width-wrapper--narrow">
  <div class="two-two-grid_inner">
    <div class="two-two-grid_child first">
      <div class="two-two-grid_child-copy">
        <div class="connect">
          <h3>Connect with us</h3>
          <p>We're always looking to connect with those who share an interest in a sustainable future.</p>
          <ul class="social_media"><li></li><li></li><li></li><li></li><li></li></ul>
        </div>
        <div class="contact">
          <h3>Contact us</h3>
          <p>Get in touch with Unilever PLC and specialist teams in our headquarters, or find contacts around the world.</p>
          <a href="/search-jobs" class="button-with-arrow button-with-arrow--pink" data-custom-label="About Us - Search Jobs"></a>
        </div>
      </div>
    </div>
  </div>

```

Page 8: DOM, “Connect with us” and “Contact us” are level 3 headings

This can also be seen on page 12 on desktop and iOS, with the “Candidate Experience Survey” heading level 2. However, it is defined as heading level 1, which does not match the visual structure if the page.



Page 12: desktop, "Candidate Experience Survey" heading

```
<!-- ngIf: noSurveyAvailable -->
<!-- ngIf: !surveyExpired && (survey.status == 'CLOSED') && !surveyDateClosed && !noSurveyAvailable -->
<!-- ngIf: !surveyExpired && (survey.status != 'CLOSED') && !surveyDateClosed && !noSurveyAvailable -->
  ><div ng-show="multipleLanguages" class="pull-left font-size-13 ng-hide">...</div>
  <!-- ngIf: noIntro -->
<!-- ngIf: noIntro --> class="intro-question ng-scope compact" dir="rtl">(compact:compactView,'margin-top-40';multipleLanguages
  <h1 class="survey-title ng-binding compact" ng-class="(compact:compactView)" ng-style="{color:survey.titleColor
    || 'black'}" style="color: black;">Candidate Experience Survey</h1>
  <div class="intro-html-wrap ck-content ng-hide" ng-hide="compactView">...</div>
  <div class="intro-html-wrap" ng-hide="!compactView">...</div>
  <div class="clear"></div>
  <div class="button-wrap">...</div>
  <div class="text-align-right">...</div>
```

Page 12: DOM, heading structure is inconsistent with the visual structure

Steps to reproduce

1. Navigate to the above page.
2. Inspect the headings and note their levels.
3. Observe that the heading levels do not correctly express the visual structure.

Recommendation

When elements title or categorise other content, use headings with appropriate heading levels. Make sure that the heading levels used correctly communicate the structure of the page, such as the parent and child relationship between the headings. For example:

```
<h1>Cooking techniques</h1>
<!-- some content here -->
<h2>Cooking with oil</h2>
<!-- some content here -->
<h3>Sautéing</h3>
<!-- some content here -->
<h3>Deep frying</h3>
<!-- some content here -->
<h2>Cooking with butter</h2>
<!-- some content here -->
```

For more information, see [WAI Tutorials - Headings](#).

[Return to priority table](#)

ADR_20590-33 MEDIUM PRIORITY OPEN

Unordered list is malformed

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 05

Users Affected Vision

Component List

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There are elements that are visually communicated as being interrelated, unordered list items. However, malformed list semantics have been used.

Where it happened

This can be seen on page 5 on desktop and iOS, with the list items present under “Who You Are & What You’ll Do” section. These list items are marked up using three different unordered lists, due to which the list is malformed.

The screenshot shows a document page with the title "Who You Are & What You'll Do". Below the title is a paragraph of text: "You are a passionate marketer who always puts the consumer at the center of business decisions. The Yasso Innovation Brand Manager will be held accountable for leading innovation, developing unmissable campaigns and assets, superior tasting products and developing long term strategy to sustain core portfolio." Below this text is a bulleted list of tasks, each preceded by a small square icon. The entire list is enclosed in a red rectangular box. The tasks listed are:

- Lead category trend work and consumer research by partnering with consumer insights team to conduct research on innovation and product concepts.
- Navigate the Unilever systems for speedy integration
- Develop innovation pipeline by working with cross functional teams to ensure product mix is launched on time and in full
- Partner closely with Procurement and Supply Chain to manage capacity. Identify contract manufacturers where relevant
- Partner with R&D to manage product and packaging transitions, improvements and updates
- Work hand in hand with the operations marketing team to ensure flawless execution in market
- Develop owned channel assets including on time, in full delivery of all materials for eCommerce pages
- Have an appreciation for better for you propositions and trends
- Day-to-day management includes:

Page 5: desktop, list items present under “Who You Are & What You’ll Do” section

```
<ul>
  <li>
    <span>
      <p>
        "Lead category trend work and consumer research by partnering with consumer insights team to conduct research on innovation and product concepts"
      </p>
    </span>
    <span></span>
  </li>
  <li>
    <span>
      <p>
        "Develop owned channel assets including on time, in full delivery of all materials for eCommerce pages"
      </p>
    </span>
    <span></span>
  </li>
</ul>
```

Page 5: DOM, different unordered lists are used for list items present under “Who You Are & What You’ll Do” section

Steps to reproduce

1. Observe the unordered list items.
2. Inspect the elements in the DOM.
3. Observe that the unordered list semantics are malformed.

Recommendation

Mark up the unordered list using `` and ``. Make sure that each set of list items is contained in a single `` and that the list items themselves are contained within individual `` tags. Only use separate `` tags when the list items are contained logically and visually in separate, distinct lists. For example:

```
<ul>
  <li>Pears</li>
  <li>Apples</li>
  ...
</ul>
```

For more information, see [WAI Tutorials - Unordered Lists](#).

[Return to priority table](#)

ADR_20590-37 MEDIUM PRIORITY OPEN

Presentational table exposed to assistive technology

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop

Pages Affected 12

Users Affected Vision

Component Modal Dialog

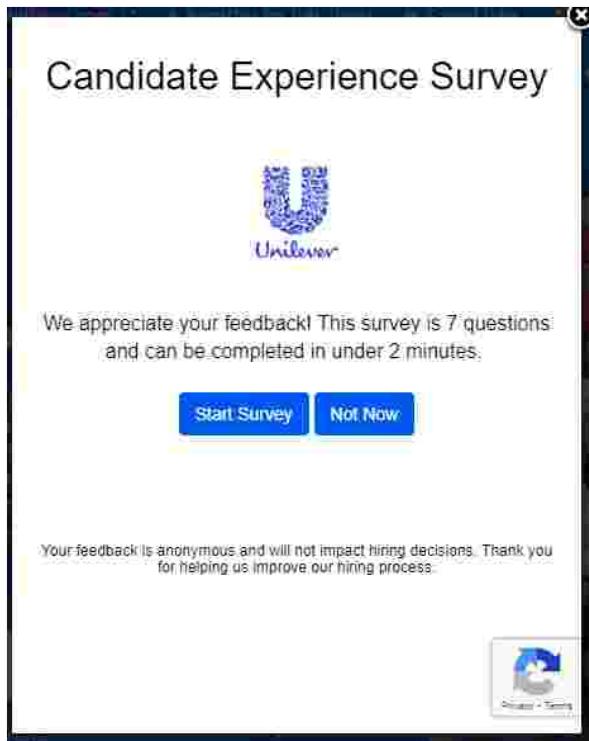
What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There is content contained in `<table>` tags, despite not being tabular data. The table semantics are inappropriately exposed to assistive technology.

Where it happened

This can be seen on page 12 on desktop. There are tables with no content positioned before and after the “Candidate Experience Survey” modal in the reading order. These tables are inappropriately exposed to assistive technology.



Page 12: desktop, “Candidate Experience Survey” modal

Page 12: DOM, table present before “Candidate Experience Survey” modal

```
</div>
</div>
<div class="padding: 0px; border: none; margin: 0px; position: absolute; width: 458px; height: 557px; visibility: visible; overflow: visible; z-index: auto; top: 0px; left: 0px;" style="position: absolute; visibility: visible; width: auto; direction: ltr; opacity: 1; right: 0px; height: auto; top: 0px; z-index: 4;*>
  <div id="ns-htl" style="padding: 0px; border: none; margin: 0px; position: absolute; visibility: visible; width: auto; direction: ltr; opacity: 1; right: 0px; height: auto; top: 0px; z-index: 4;*>
    <div style="display: block;">
      <button id="ns-close-button" class="closebutton" title="Close" style="border: none; height: 30px; width: 30px; background-color: transparent; border: none; border-radius: 50%; font-size: 1em; color: inherit; text-decoration: none; outline: none; position: absolute; right: -15px; top: -15px; z-index: 1000; font-weight: bold; font-family: sans-serif; transition: all 0.3s ease; cursor: pointer; opacity: 0.8;*>X</button>
    </div>
  </div>
</div>
<div class="highslide-dimming highslide-viewport-size" style="padding: 0px; border: none; margin: 0px; visibility: visible; opacity: 0.75;*></div>
<table style="display: none; padding: 0px; border: none; margin: 0px; visibility: hidden; position: absolute; border-collapse: collapse; width: 0px;">
  <tbody style="padding: 0px; border: none; margin: 0px; width: 0px;">
    <tr style="padding: 0px; border: none; margin: 0px; height: auto;">
      <td style="padding: 0px; border: none; margin: 0px; line-height: 0; font-size: 0px; background: url('https://static.surveylive.com/ext/assets/graphics/outline/drop-shadow.png') 0px 0px; height: 28px; width: 28px;*>
        <td style="padding: 0px; border: none; margin: 0px; line-height: 0; font-size: 0px; background: url('https://static.surveylive.com/ext/assets/graphics/outline/drop-shadow.png') -40px 0px; height: 28px; width: 28px;*>
          <td style="padding: 0px; border: none; margin: 0px; line-height: 0; font-size: 0px; background: url('https://static.surveylive.com/ext/assets/graphics/outline/drop-shadow.png') -28px 0px; height: 28px; width: 28px;*>
            <tr style="padding: 0px; border: none; margin: 0px; height: auto;">
              <td style="padding: 0px; border: none; margin: 0px; height: auto;"></td>
            </tr>
            <tr style="padding: 0px; border: none; margin: 0px; height: auto;">
              <td style="padding: 0px; border: none; margin: 0px; height: auto;"></td>
            </tr>
          </td>
        </td>
      </td>
    </tr>
  </tbody>
</table>
```

Page 12: DOM, table present after “Candidate Experience Survey” modal

Steps to reproduce

1. Observe that the content is not tabular data.
 2. Inspect the content and review the DOM.
 3. Note the use of `<table>` tags.
 4. Note the lack of `role="presentation"` to suppress the semantics.

Recommendation

If possible, use pure CSS to position the content. Do not use `<table>` tags, unless creating a table with tabular data. If this is not possible and the tags must be used, add `role="presentation"` to the `<table>` element. This instructs browsers to suppress the table semantics. For example:

```
<table role="presentation">
  ...
</table>
```

For more information, see [WAI ARIA Practices - Intentionally Hiding Semantics](#) and [WAI Tutorials - Tables](#).

[Return to priority table](#)

ADR_20590-41 MEDIUM PRIORITY OPEN

Text uses heading semantics inappropriately

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12

Users Affected Vision

Component Heading

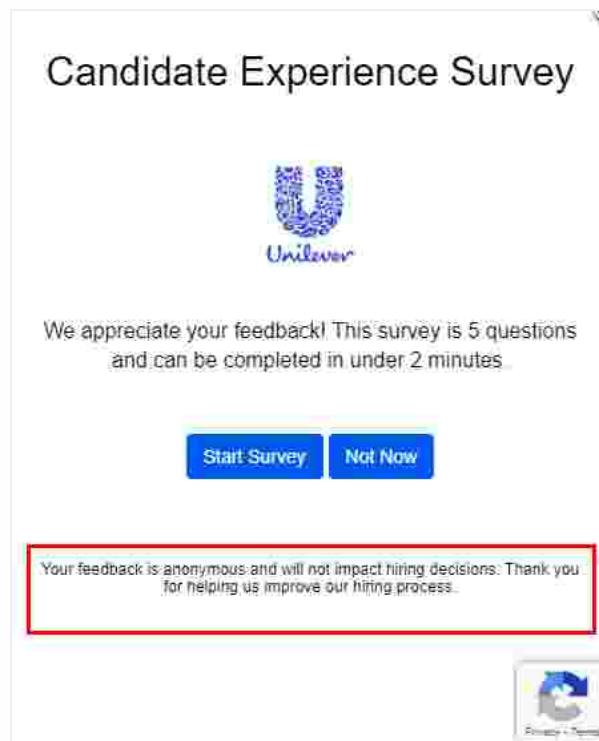
What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Similarly, content must not use structural semantics inappropriately. Content may otherwise seem unstructured or disorganised.

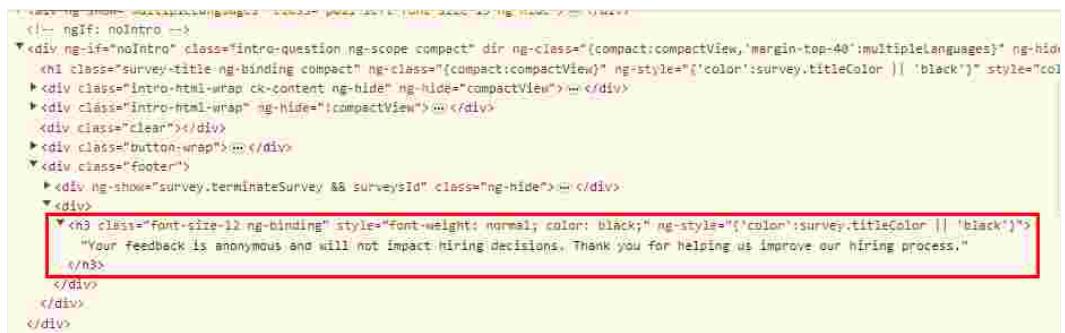
There are elements contained in heading tags that do not title or categorise other content.

Where it happened

This can be seen on page 12 on desktop and iOS, with the “Your feedback is anonymous...” text. This text does not title anything, but it has been inappropriately marked up as `<h3>`.



Page 12: desktop, “Your feedback is...” text



The screenshot shows a portion of the DOM structure. A red box highlights an `<h3>` element located within a `<div>` block that has the class `"ng-hide"`. The `ng-style` attribute of this `<div>` includes the value `{"color":survey.titleColor || 'black'}`, which is intended to style the heading. However, the heading itself contains the text: "Your feedback is anonymous and will not impact hiring decisions. Thank you for helping us improve our hiring process." This usage of a heading tag for descriptive text is considered inappropriate.

```
<!-- ngIf: noIntro -->
<div ng-if="noIntro" class="intro-question ng-scope compact" dir="compact:compactView" margin-top-40":multipleLanguages}* ng-hide="true">
  <h1 class="survey-title ng-binding compact" ng-class="{compact:compactView}" ng-style="{'color':survey.titleColor || 'black'}" style="color: black;">
    <div class="intro-html-wrap ck-content ng-hide" ng-style="compactView">...</div>
  <div class="intro-html-wrap" ng-hide="!compactView">...</div>
  <div class="clear"></div>
  <div class="button-wrap">...</div>
<div class="footer">
  <div ng-show="survey.terminateSurvey && surveyId" class="ng-hide">...</div>
</div>
<h3 class="font-size-12 ng-binding" style="font-weight: normal; color: black;" ng-style="{'color':survey.titleColor || 'black'}">
  "Your feedback is anonymous and will not impact hiring decisions. Thank you for helping us improve our hiring process."
</h3>
</div>
</div>
</div>
```

Page 12: DOM, “Your feedback is...” text uses heading semantics inappropriately

Steps to reproduce

1. Observe that the content is not titling or categorizing other content.
2. Inspect the content and review the DOM.
3. Note the inappropriate use of heading tags.

Recommendation

Use CSS to style the content. Do not use heading tags, such as `<h1>`, `<h2>`, and so on, unless the text elements title or categorise other content. Use neutral tags that do not have any structural meaning, such as `<div>`, ``, and `<p>` tags. For example:

```
<h1>Contact us</h1>
<p>Use the form below to get in contact</p>
```

For more information, see [MDN - Styling Text](#), [WAI Tutorials - Headings](#) and [WAI Tutorials - Content Structure](#).

[Return to priority table](#)

ADR_20590-42 MEDIUM PRIORITY OPEN

Form fields and visual labels are not associated

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 14

Users Affected Vision

Component Input Field

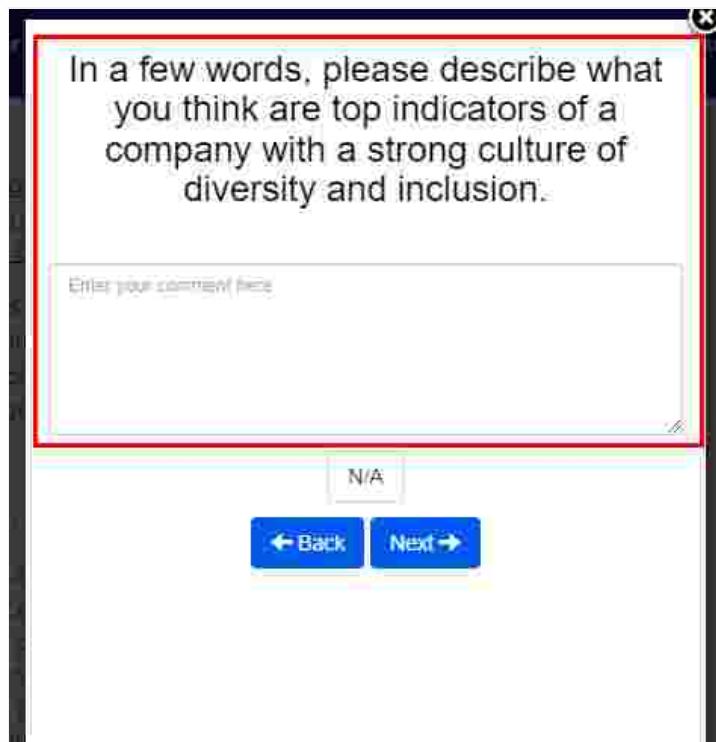
What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationship to users. Content may otherwise seem unstructured or disorganized.

There are form fields that are not associated with their visual labels. The purpose of the form fields may be unclear to users.

Where it happened

This can be seen on page 14 on desktop and iOS, with the “Large Comment” form field present under “In a few words, please describe...” text. This text is not associated with its input field.



Page 14: desktop, the “In a few words, please describe...” text

Page 14: DOM, the “In a few words, please describe...” text is not associated with the input field

Steps to reproduce

1. Observe the visual labels for the form fields.
 2. Inspect the form fields and review the DOM.
 3. Observe that the visual labels are not associated with the form fields.
 4. Observe that there is not an alternative labelling method that serves the same purpose.

Recommendation

Any relationship presented through visual cues must also use the appropriate semantics. Make sure that form fields are associated with their visual labels. This can be done with `<label>` or `aria-labelledby`. Make sure that the `id` values used match appropriately. For example:

```
<label for="book">Favourite book</label>
<input id="book" type="text">
```

Note that `aria-labelledby` is useful when you need to construct the label using multiple elements. For example:

```
<span id="firstPart">Favourite</span>
<span id="secondPart">colour</span>
<input aria-labelledby="firstPart secondPart" type="text">
```

Avoid using multiple labelling techniques on the same form field. For example, using `<label>` and `aria-label` together will cause the `aria-label` to overwrite the `<label>`. As well, using `aria-labelledby` will overwrite any other labelling techniques.

For more information, see [WAI Tutorials - Labelling Controls](#).

[Return to priority table](#)

ADR_20590-73 MEDIUM PRIORITY OPEN

Text uses heading semantics inappropriately

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop

Pages Affected 08

Users Affected Vision

Component Heading

What happened

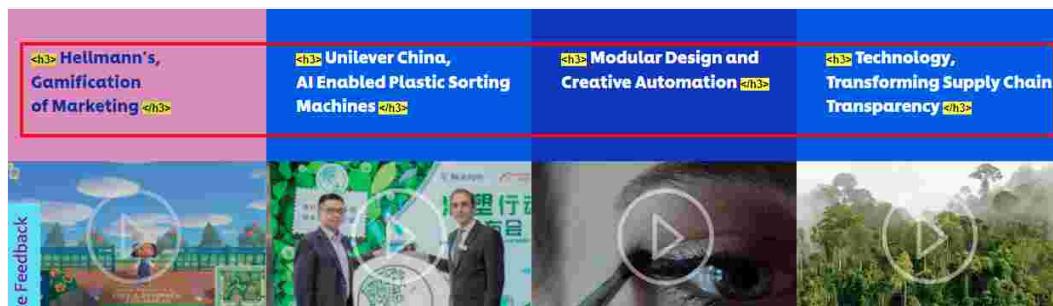
If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Similarly, content must not use structural semantics inappropriately. Content may otherwise seem unstructured or disorganized.

There are elements contained in heading tags that do not title or categorise other content.

Where it happened

This can be seen on page 8 on desktop with the button text under “Purpose Driven Work with Global Impact”, such as “Hellmann’s, Gamification of Marketing”. These elements do not have content under them, except for decorative images, but they are marked up as headings.

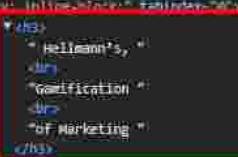
Note that the heading tags are also nested within buttons, which is not valid HTML.



Page 8: desktop, headings bookmarklet, button text under “Purpose Driven Work with Global Impact” is marked up as headings

```
<div class="slick-slide slick-current slick-active" data-slick-index="0" aria-hidden="false" style="width: 316px;">

### * Hellmann's, * * Gamification * * of Marketing * *




```

A screenshot of the DOM code for the Hellmann's Gamification of Marketing card. The heading text 'Hellmann's, Gamification of Marketing' is highlighted with a red box. The code shows the heading is nested within a button element.

Page 8: DOM, “Hellmann’s, Gamification of Marketing” is marked up as a heading

Steps to reproduce

1. Observe that the content is not titling or categorising other content.
2. Inspect the content and review the DOM.
3. Note the inappropriate use of heading tags.

Recommendation

Use CSS to style the content. Do not use heading tags, such as `<h1>`, `<h2>`, and so on, unless the text elements title or categorise other content. Use neutral tags that do not have any structural meaning, such as `<div>`, ``, and `<p>` tags. For example:

```
<h1>Contact us</h1>
<p>Use the form below to get in contact</p>
```

For more information, see [MDN - Styling Text](#), [WAI Tutorials - Headings](#) and [WAI Tutorials - Content Structure](#).

[Return to priority table](#)

ADR_20590-91 MEDIUM PRIORITY OPEN

Button group not associated with legend

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Button

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganised.

There are buttons that are not associated with their visual group label. The purpose of the form fields may be unclear to users.

Where it happened

This can be seen on page 13 on desktop and iOS with the ratings buttons. The buttons are visually labelled by the modal heading “How important is it that a company disclose its workforce demographics and progress towards its workforce diversity and equity goals on its careers site?”. However, the group is not associated with this label.

How important is it that a company disclose its workforce demographics and progress towards its workforce diversity and equity goals on its career site?

Please rate on scale below.

(Scale=Low to High)

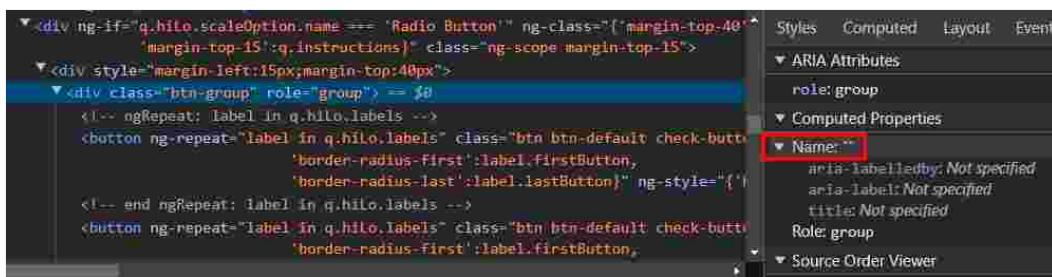
Not rated

N/A

Next ➔

Privacy & Terms

Page 13: desktop, ratings buttons are labelled by modal heading



Page 13: DOM, ratings button group is not associated with modal heading

Steps to reproduce

1. Observe the visual group label for the buttons.
2. Inspect the buttons and review the DOM.
3. Observe that the visual group label is not associated with the buttons.
4. Observe that there is not an alternative labelling method that serves the same purpose.

Recommendation

Any relationship presented through visual cues must also use the appropriate semantics. Make sure that form fields are associated with their visual labels. This can be done with `<fieldset>` and `<legend>`. For example:

```
<fieldset>
  <legend>I am the group label</legend>
  <input type="radio" name="example" id="exampleOne" checked>
  <label for="exampleOne">I am the first radio button</label>
  <input type="radio" name="example" id="exampleTwo">
  <label for="exampleTwo">I am the second radio button</label>
</fieldset>
```

If this is not possible, you can use `aria-labelledby`. This can be used with custom radio buttons created with ARIA. For example:

```
<h2 id="radioGroupLabel">I am the group label</h2>
<div role="radiogroup" aria-labelledby="radioGroupLabel">
  <!-- radio buttons -->
</div>
```

Avoid using multiple labelling techniques on the same form field. For example, using `<label>` and `aria-label` together will cause the `aria-label` to overwrite the `<label>`. As well, using `aria-labelledby` will overwrite any other labelling techniques.

For more information, see [WAI Tutorials - Grouping Controls](#) and [Labelling Controls](#).

[Return to priority table](#)

ADR_20590-92 MEDIUM PRIORITY OPEN

Additional information is not associated with input field

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Input Field

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationship to users. Content may otherwise seem unstructured or disorganized.

There are form fields with additional information that is necessary to understand or complete the form fields. This additional information is not associated with the relevant form fields. The complete purpose of the form fields may be unclear to users.

Where it happened

This can be seen on page 13 on desktop and iOS with the ratings buttons and the text “(Scale-Low to High)”. The text element provides additional instructions for the buttons, but is not associated with the button group.

Note that the buttons also do not have accessible names. This is raised in ADR_20590 - 47.

How important is it that a company disclose its workforce demographics and progress towards its workforce diversity and equity goals on its career site?

Please rate on scale below.



Not rated

N/A

Next ➔



Page 13: desktop, ratings button group and hint text

```
<div style="margin-left:15px; margin-top:40px">
  <div class="btn-group" role="group" style="border:1px solid #ccc; border-radius:4px; padding:5px; width:100px; height:30px; display:flex; align-items:center; justify-content:center; gap:10px; font-size:14px; font-weight:bold; background-color:#fff; position:relative; z-index:1;>
    <!-- ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.bgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.hgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.bgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.hgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.bgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-wid
      'border-radius-first':label.firstButton,
      'border-radius-last':label.lastButton" ng-style="{background-color:label.hgColor}"
    <!-- end ngRepeat: label in q.hilo.labels -->
  </div>
</div>
<p class="font-size-10 ng-binding" style="font-style: italic;">(Scale Low to High)
<div style="margin-top:10px;font-size:16px;" class="ng-binding">Not at all important
```

</div>
<!-- end ngIf: q.hilo.scaleOption.name === 'Radio Button' -->

Page 13: DOM, ratings button group is not associated with hint text

Steps to reproduce

1. Observe the additional information for the form fields.
 2. Inspect the form fields and review the DOM.
 3. Observe that the additional information is not associated with the form fields.
 4. Observe that there is not an alternative labelling method that serves the same purpose.

Recommendation

Any relationship presented through visual cues must also use the appropriate semantics. Make sure that the necessary-to-know information is correctly associated with form fields. This can be done with **aria-describedby**. Make sure that the **aria-describedby** value and **id** match correctly. For example:

```
<div role="group" aria-labelledby="..." aria-describedby="hintText">
    ...
</div>
<p id="hintText">(Scale-Low to High)</p>
```

Note that `aria-describedby` can reference multiple `id` values, such as hints and error messages.

Avoid using multiple labelling techniques on the same form field. For example, using `<label>` and `aria-label` together will cause the `aria-label` to overwrite the `<label>`. As well, using `aria-labelledby` will overwrite any other labelling techniques.

For more information, see [WAI Tutorials - Labelling Controls](#).

[Return to priority table](#)

ADR_20590-19 MEDIUM PRIORITY OPEN

Reading order includes off-screen elements

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 02

Users Affected Vision

Component Link, Image

What happened

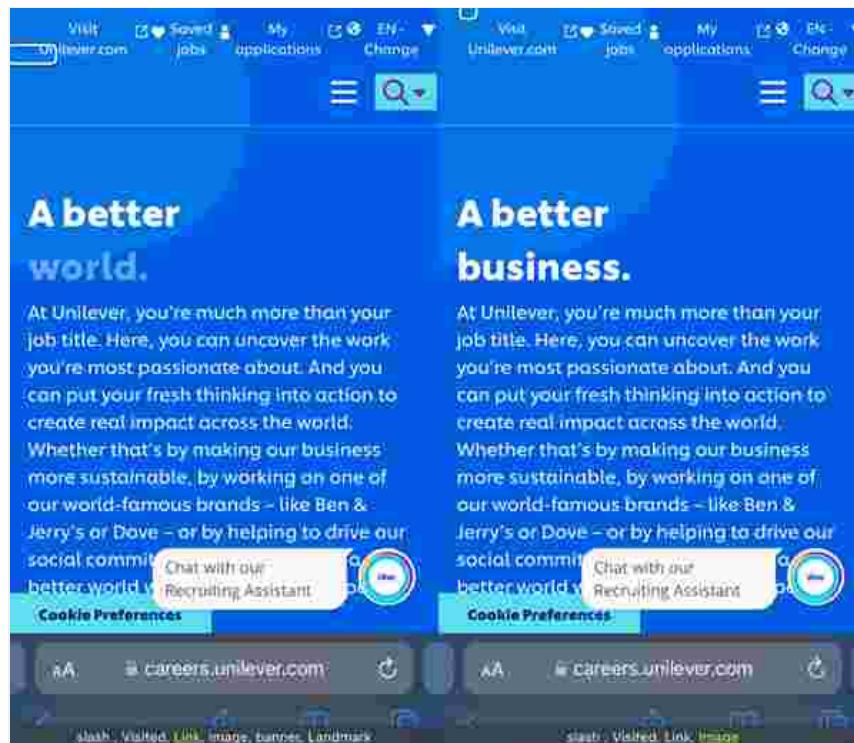
When content has a meaningful order, an appropriate reading order needs to be programmatically determined. This enables those using assistive technology to effectively navigate the content and understand it. Content may otherwise seem unstructured or disorganized.

There are off-screen elements that receive the screen reader cursor. This may particularly disadvantage screen reader users who often navigate by swiping through the interactive elements. The logical reading order is that the screen reader cursor cannot be set to invisible off-screen elements, which is not reflected by the programmatic reading order.

Where it happened

This can be seen on page 1 on iOS. There are 2 off-screen image links that receive the screen reader cursor after the “Skip to main content” link and before the “Visit Unilever.com” link.

Note that it is unclear what these links correspond to. However, they may be connected to the “Unilever careers home” link, which is off-screen but exposed to assistive technology when the page is scrolled to the top.

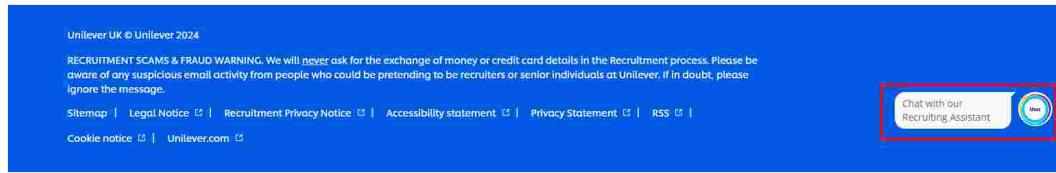


Page 1: iOS, off-screen links receive the screen reader cursor before the “Visit Unilever.com” link

```
<div id="page">
  <a id="skip-nav" class="access-link" href="#content">Skip to main content</a>
  <div class="header-background header-background--video header-background--video--core" data-vide-bg="mp4:https://tcdn.talentbrew.com/company/34155/v1/home-video-76.mp4, poster:https://tcdn.talentbrew.com/company/34155/v1/img/video-poster-99.jpg" data-video-options="posterType: jpg, loop: true, muted: true, position: 50% 0%, autoplay: false" style="background-image: url('https://tcdn.talentbrew.com/company/34155/v1/img/video-poster-99.jpg'); position: relative;">
    <div style="position: absolute; z-index: 3; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-d-position: 50% 0%; background-image: url('https://tcdn.talentbrew.com/company/34155/v1/img/video-poster-99.jpg');"></div>
    <div class="play-pause-button" paused="false" style="display: none;"></div>
  </div>
  <header class="core-header">
    <div class="max-width-wrapper"> </div>
    <div class="core-header_left">
      <a href="#" class="header-logo">
        
      </a>
    </div>
    <div class="core-header_right"> </div>
    <div class="uppernav" role="list"> </div>
      <a class="uppernav_element external-link" href="https://unilever.com" target="_blank" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Unilever.com" role="listitem" rel="noopener">
        Visit Unilever.com
        <span class="visually-hidden">opens in a new window</span>
      </a>
    </div>
  </header>
```

Page 1: DOM, off-screen “Unilever careers home” image link

This can also be seen on page 2 on desktop and iOS with the content “Una said, Hi! I am Una, your personal job assistant at Unilever! You can ask me anything about our business, culture, team or I can help you start your job search!” in the chat widget. When the chat widget is collapsed, this content is still included in the reading order.



Page 2: desktop, chat widget is collapsed

```
<div tabindex="0" role="button" aria-haspopup="true" aria-expanded="false" data-testid="chatbox_lbl_prompt_text" class="apply-chat-prompt _0286a8 c5883">
  <span>
    <div> Chat with our Recruiting Assistant </div>
  </span>
</div>
<div id="new-message-alert" aria-live="polite" aria-atomic="false" class="f845a8">
  "Una said, Hi! I am Una, your personal job assistant at Unilever! You can ask me anything about our business, culture, team or I can help you start your job search!"
</div>
</div>
```

Page 2: DOM, chat widget message is exposed to assistive technology when widget is collapsed

Steps to reproduce

1. Using just your screen reader, swipe through the above.
2. Observe that the screen reader cursor goes onto elements that are not visible on the page.

Recommendation

Make sure that the content is programmatically ordered in such a way that it provides an appropriate reading order. It does not always need to exactly match the visuals, but it must provide users with a meaningful reading order. In this case, make sure that elements that are not visible on the page are not exposed to assistive technologies. Screen reader users should be able to swipe through the page in a meaningful order.

If they are not meant to be accessed at any point, remove the elements from the DOM, comment them out, or remove them from the focus and reading orders. If they are meant to be interactive at some point, make sure that they are excluded or included in the focus and reading orders as appropriate.

Removing elements from the focus and reading orders can be achieved with the HTML `hidden` or CSS `display:none`. Both will hide the element from all users. This same can be programmatically achieved by using both `aria-hidden="true"` and `tabindex="-1"` on the same element. For example:

```
<button hidden> ... </button>
<button class="visuallyOffScreen" aria-hidden="true" tabindex="-1"> ... </button>
```

However, note that `aria-hidden="true"` and `tabindex="-1"` will not visually remove elements from the page.

Removing these attributes will re-insert the elements into the focus order based on their location in the DOM, allowing users to navigate onto them with a keyboard. Make sure that this is done as appropriate, such as when the user toggles a parent control that shows and hides these interactive elements.

For more information, see [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - Using Tabindex](#), [MDN - aria-hidden](#), and [MDN - hidden](#).

[Return to priority table](#)

ADR_20590-23 MEDIUM PRIORITY OPEN

Listbox does not follow combobox in reading order

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Mobile (iOS)

Pages Affected 01

Users Affected Vision

Component Combobox

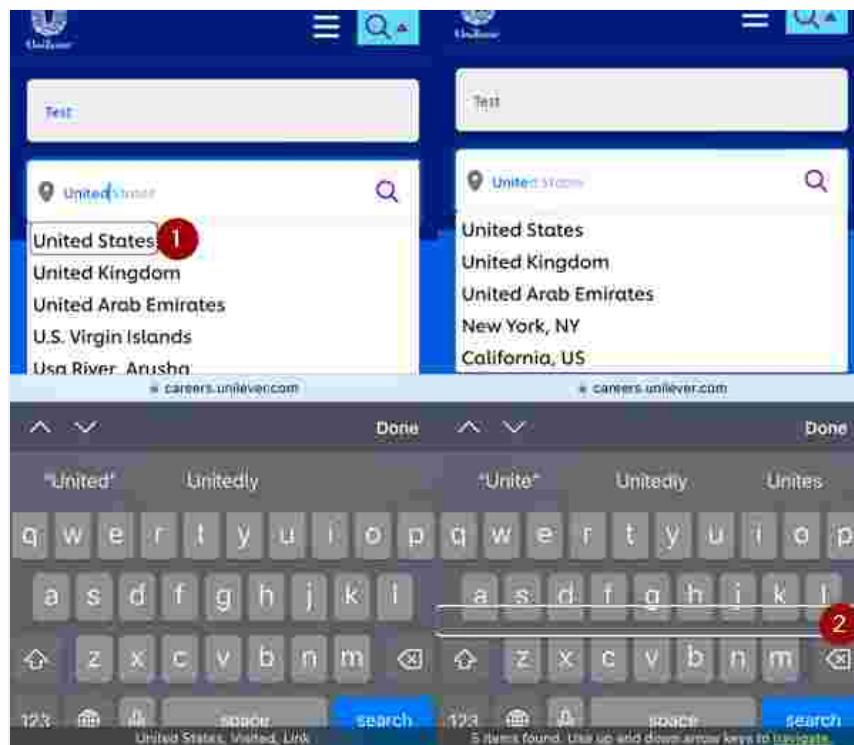
What happened

When content has a meaningful order, an appropriate reading order needs to be programmatically determined. This enables those using assistive technology to effectively navigate the content and understand it. Content may otherwise seem unstructured or disorganized.

There is content with a meaningful visual reading order. However, an appropriate reading order has not been programmatically determined.

Where it happened

This can be seen on page 1 on iOS with the list of results in the location combobox. The list of options does not follow the input field in the reading order. For example, when swiping backwards from the first option, the screen reader cursor moves to a live region at the bottom of the page instead of to the input field.



Page 1: iOS, when swiping backwards from location list with screen reader, the cursor moves to the bottom of the page instead of to the combobox input field

Steps to reproduce

1. Observe that the content has a meaningful visual reading order.
2. Navigate through the content with a screen reader.
3. Observe that an appropriate reading order has not been programmatically determined.

Recommendation

Assistive technologies navigate content using the order in which elements appear in the DOM. Make sure that the content is programmatically ordered in such a way that it provides an appropriate reading order. It does not always need to exactly match the visuals, but it must provide users with a meaningful reading order. For example:

```
<button aria-label="Close modal">X</button>
<h2>Modal heading</h2>
<p>Modal text content</p>
<button>OK</button>
<button>Cancel</button>
```

In this case, make sure the listbox follows the combobox input field in the DOM.

For more information, [WCAG - G57 Ordering the content](#), [C57 DOM order match the visual order](#), [The Dark Side of the Grid](#), and [Web Dev - Content Reordering](#).

[Return to priority table](#)

ADR_20590-61 MEDIUM PRIORITY OPEN

Screen reader cursor not set onto modal dialog

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Mobile (iOS)

Pages Affected 04

Users Affected Vision

Component Modal Dialog

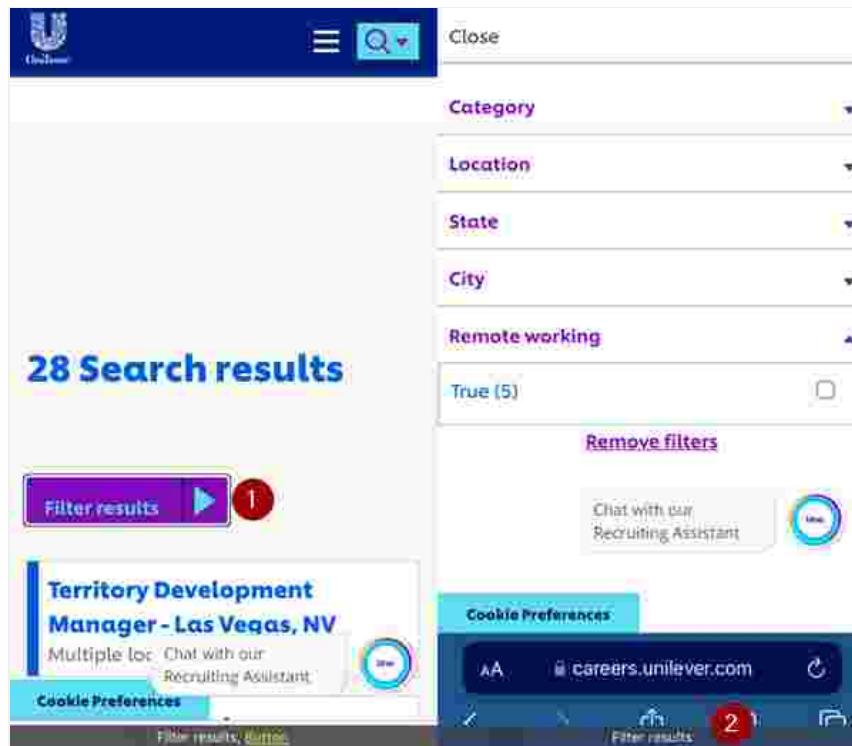
What happened

When content has a meaningful order, an appropriate reading order needs to be programmatically determined. This enables those using assistive technology to effectively navigate the content and understand it. Content may otherwise seem unstructured or disorganized.

There are modal dialogs where the programmatic reading order is not logical. When the modal dialog is opened, users are not routed onto the modal dialog. Instead, users remain on the background element that trigger the modal dialog, which is no longer part of the visual reading order.

Where it happened

This can be seen on page 4 on iOS with the “Filter results” modal. When the modal is triggered with a screen reader, the cursor is not placed on the modal. Instead, it remains on the “Filter results” button in the background content.



Page 4: iOS, screen reader cursor is not placed on “Filter results” modal

Steps to reproduce

1. Using a screen reader, activate the modal dialog.
2. Observe the screen reader cursor location.
3. Observe that the screen reader cursor is not automatically set onto the modal dialog.

Recommendation

Make sure that the content is programmatically ordered in such a way that provides an appropriate reading order. It does not always need to exactly match the visuals, but it must provide users with meaningful reading order. In this case, when the modal dialog is open, make sure that users are routed onto the modal dialog.

In this case, when the modal dialog is open, make sure that the programmatic keyboard focus is set onto the modal dialog. This can be achieved with the **focus** method. As well, if you need the non-interactive elements to be capable of receiving focus through the **focus** method, use **tabindex="-1"**. For example:

```
<!-- background content -->
<div aria-hidden="true">
  <p>Example plain text in background</p>
  <button tabindex="-1">Example button in background</button>
  ...
</div>
<!-- modal dialog container -->
<div role="dialog" aria-labelledby="dialogName" aria-modal="true">
  <h2 id="dialogName" tabindex="-1">Modal Dialog Heading</h2>
  ...
</div>
```

For more information, see [WAI ARIA - Dialog Example](#), [MDN - focus method](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), and [Google Developer - Using Tabindex](#).

[Return to priority table](#)

ADR_20590-62 MEDIUM PRIORITY OPEN

Screen reader cursor not restricted to modal dialog

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Mobile (iOS)

Pages Affected 04, 09

Users Affected Vision

Component Modal Dialog

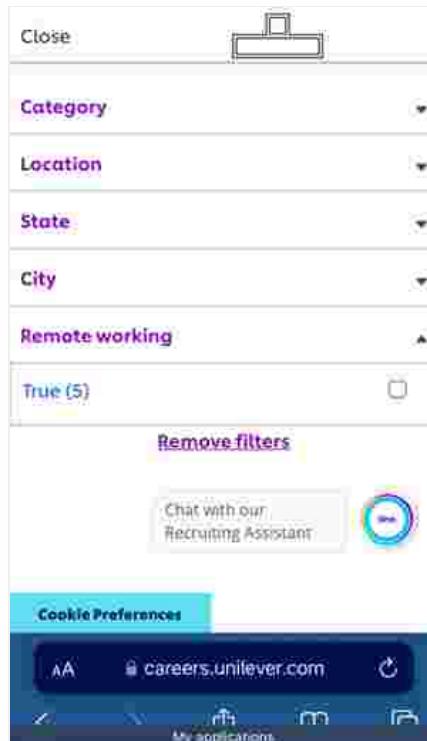
What happened

The assistive technology focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There is a modal dialog where the background elements can still be accessed with a screen reader. This may particularly disadvantage screen reader users. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be found on page 4 on iOS with the filters modal. The elements in the background content can receive the screen reader cursor while the modal is open.



Page 4: iOS, screen reader cursor is on “My applications” link while filters modal is open

This can also be seen on page 9 on iOS, with the “Be a part of something bigger” modal. When swiping through the modal, background elements can still receive the screen reader cursor.



Page 9: iOS, background link exposed to assistive technology as "Plus" receives screen reader cursor

Steps to reproduce

1. Using a screen reader, activate the modal dialog.
2. Navigate into the modal dialog.
3. Attempt to swipe away from the modal dialog.
4. Observe that the screen reader cursor can access the background elements.

Recommendation

Make sure that the content is programmatically ordered in such a way that provides an appropriate focus order. It does not always need to exactly match the visuals, but it must provide users with meaningful focus order. When the modal dialog is open, make sure that users are routed onto the modal dialog.

In this case, when the modal dialog is open, make sure that programmatic keyboard focus is set onto the modal dialog. As well, make sure that keyboard focus and the screen reader cursor are restricted to the dialog. This can be achieved with the following.

- Send the focus to the first element on the modal container. This can be done using the JavaScript `.focus()` method.
- Hide all of the other content of the page with `aria-hidden="true"`.
- Remove the interactive elements in the background from the focus order with `tabindex="-1"`.
- Revert the above when the modal is closed, removing the `aria-hidden` and `tabindex` properties.
- When the dialog is dismissed, use the `.focus()` method to set focus back onto the element that opened it.

The `aria-hidden` and `tabindex` can be supplemented with an inert polyfill. For examples, see [GitHub - WICG - Inert](#), and [inert-polyfill](#).

For more information, see [WAI ARIA - Dialog Example](#), [MDN - aria-hidden](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - DOM Order Matters](#), and [Using Tabindex](#).

[Return to priority table](#)

ADR_20590-67 MEDIUM PRIORITY OPEN

Meaningful visual structure is not represented by reading order

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 08, 11

Users Affected Vision

Component Button, Plain Text, Heading

What happened

When content has a meaningful order, an appropriate reading order needs to be programmatically determined. This enables those using assistive technology to effectively navigate the content and understand it. Content may otherwise seem unstructured or disorganized.

Note that this is distinct from the focus order, which is about interactive elements receiving focus when users keyboard tab through the page. There is content with a meaningful visual reading order. However, an appropriate reading order has not been programmatically determined.

Where it happened

This can be seen on page 8 on desktop and iOS with the “Previous” button in the “Meet Our People” carousel. The button comes before the heading “Meet Our People” in the reading order, which is not logical because it controls the carousel content under “Meet Our People”.



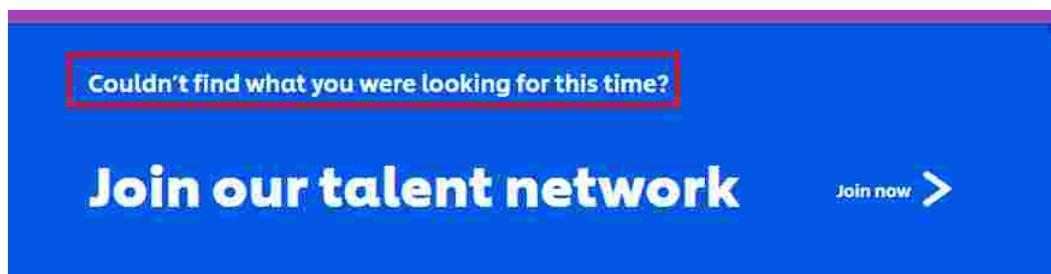
Page 8: desktop, previous button for “Meet Our People” carousel is under “Meet Our People” heading

```
<div class="two-two-grid_inner_team_slider_container slick-initialized slick-slider">
  <button class="slick-prev slick-arrow" aria-label="Previous" type="button" style="tabIndex=0">
    :before
    "Previous"
  </button>
  <div class="slick-list draggable" style="height: 361px;">
    <div class="slick-track" style="opacity: 1; width: 57592px; transform: translate3d(-2504px, 0px, 0px);">
      <div class="slick-slide slick-cloned" data-slick-index="-1" id="aria-hidden=true" tabIndex=-1 style="width: 2504px;">
        <div class="team_slider" style="width: 100%; display: inline-block;">
          <div class="two-two-grid_child two-two-grid_child_topcontent">
            <div class="two-two-grid_child_copy">
              <h2>
                "Meet Our"
                <br>
                "People"
              </h2>
              <p>Couldn't find what you were looking for this time?</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Page 8: DOM, previous button for “Meet Our People” carousel comes before “Meet Our People” heading in the reading order

This can also be seen on page 8 with the text “Couldn’t find what you were looking for this time?”. The text is in the reading order before the heading that relates to it.

Note that the reading order in this case matches the visual order. However, it is not a logical reading order because it implies that “Couldn’t find what you were looking for this time?” relates to the “Explore Where You Fit Across Our Digital Teams” heading above it rather than the “Join our talent network” heading below it.

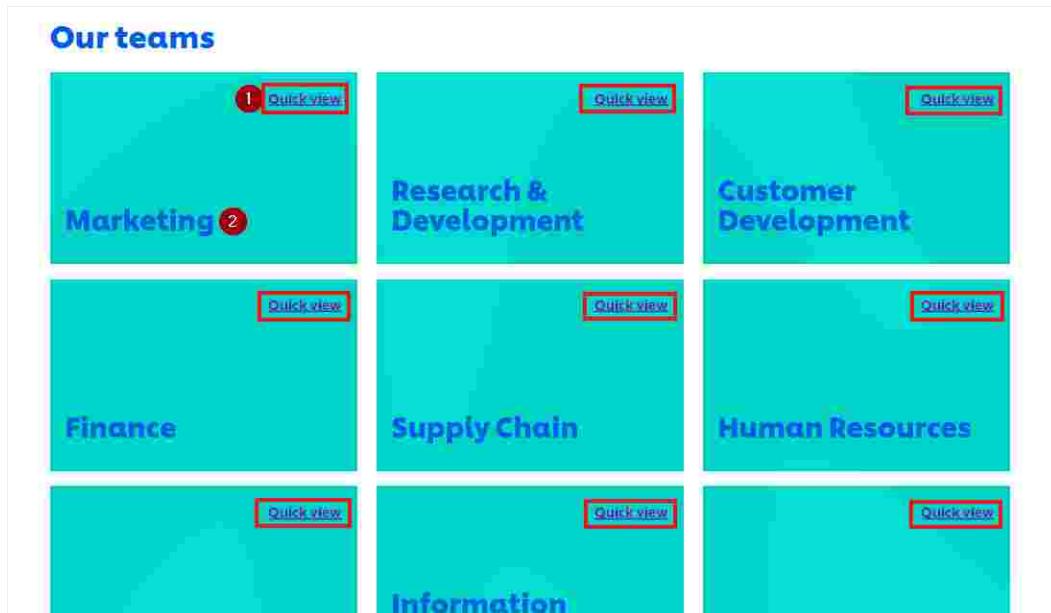


Page 8: "Couldn't find what you were looking for this time?" text above "Join our talent network" heading

```
<section id="join_our_talent" class="digital_content">
  <div class="two-two-grid max-width-wrapper max-width-wrapper--large">
    <p class="text--white">Couldn't find what you were looking for this time?</p>
    <div class="two-two-grid_inner"> <!--
      <h2 class="text--white">
        "Join our "
        <br>
        "talent network"
      </h2>
    -->
```

Page 8: DOM, "Couldn't find what you were looking for this time?" is included in the reading order before the heading "Join our talent network" that relates to it

This can also be seen on page 11 on desktop and iOS with the "Quick view" elements in the cards under "Our teams", such as "Marketing. Each "Quick view" element is included in the reading order before the heading that relates to it.



Page 11: desktop, "Quick view" elements for "Our teams" cards

```
<div class="areas-of-business_grid"> <!--
  <div class="business-area">
    <div class="business-area_cover"> <!--
      <span class="business-area_quickview">Quick view</span>
      <h3 class="business-area_title">Marketing</h3>
    </div>
    <div class="business-area_hidden-section"><!--</div>
  </div>
```

Page 11: DOM, "Quick view" element for "Marketing" comes before the heading that relates to it

Steps to reproduce

1. Observe that the content has a meaningful visual reading order.
2. Inspect the content and review the DOM.
3. Observe that an appropriate reading order has not been programmatically determined.

Recommendation

Assistive technologies navigate content using the order in which elements appear in the DOM. Make sure that the content is programmatically ordered in such a way that it provides an appropriate reading order. It does not always need to exactly match the visuals, but it must provide users with a meaningful reading order. For example:

```
<button aria-label="Close modal">X</button>
<h2>Modal heading</h2>
<p>Modal text content</p>
<button>OK</button>
<button>Cancel</button>
```

For more information, [WCAG - G57 Ordering the content](#), [C57 DOM order match the visual order](#), [The Dark Side of the Grid](#), and [Web Dev - Content Reordering](#).

[Return to priority table](#)

ADR_20590-109 MEDIUM PRIORITY OPEN

Reading order of card disclosure elements includes off-screen elements

Success Criteria 1.3.2: Meaningful Sequence [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 11

Users Affected Vision

Component Link, Plain Text

What happened

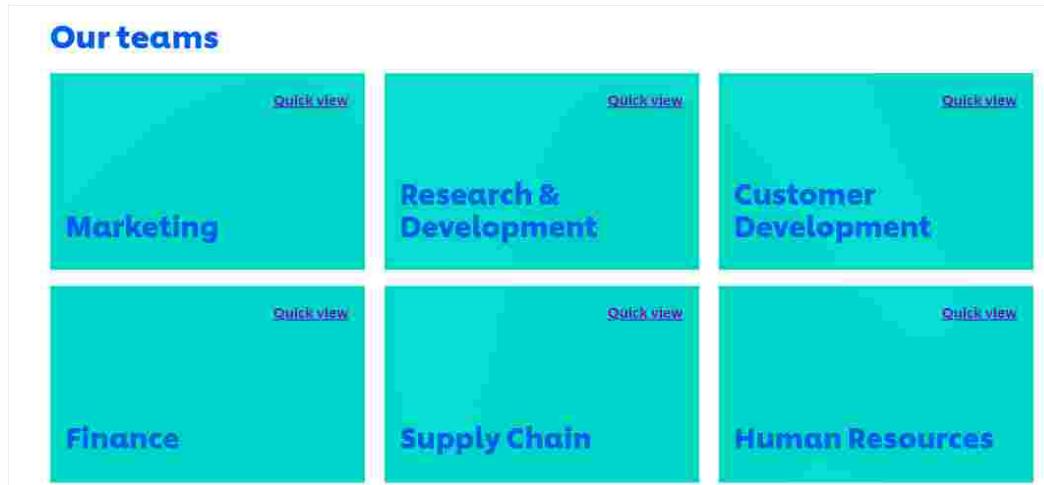
When content has a meaningful order, an appropriate reading order needs to be programmatically determined. This enables those using assistive technology to effectively navigate the content and understand it. Content may otherwise seem unstructured or disorganized.

There are off-screen elements that receive the screen reader cursor. This may particularly disadvantage screen reader users who often navigate by swiping through the elements. The logical reading order is that the screen reader cursor cannot be set to invisible off-screen elements, which is not reflected by the programmatic reading order.

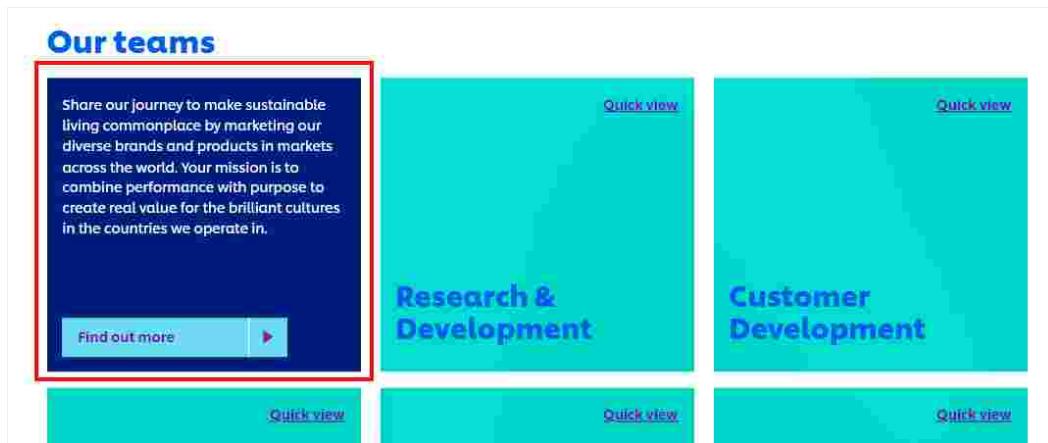
Where it happened

This can be seen on page 11 on desktop and iOS with the cards under “Our teams”. These cards can be activated to disclose further content. This disclosed content is included in the reading order even when the card is collapsed and the content is off screen. Similarly, when the card is expanded, the off-screen “Quick view” element and card heading are included in the reading order.

Note that, on desktop, the cards are triggered with mouse hover or when the links inside the disclosure receive keyboard focus. On iOS, the cards can be activated without a screen reader by tapping or with a screen reader by activating the “Quick view” element or card heading.



Page 11: desktop, “Our teams” cards when collapsed



Page 11: desktop, "Our teams" cards with "Marketing" card expanded

```
<div class="title">Our teams</div>
<div class="areas-of-business_grid"><div>
  <div class="business-area_covers"><div>
    <span class="business-area_quickview">Quick view</span>
    <h3 class="business-area_title">Marketing</h3>
  </div>
  <div class="business-area_hidden-section" style="display:none; position: absolute; left: 0; top: 0; width: 100%; height: 100%; background-color: black; opacity: 0.5; z-index: 100;">
    <p class="business-area_copy">Share our journey to make sustainable living commonplace by marketing our diverse brands and products in markets across the world. Your mission is to combine performance with purpose to create real value for the brilliant cultures in the countries we operate in.</p>
    <a class="business-area_cta button-with-arrow button-with-arrow-teal" data-custom-event="true" data-custom-category="Custom_Event" data-custom-label="Our teams" href="/marketing">Find out more</a>
  </div>
</div>
</div>
```

Page 11: DOM, "Marketing" card disclosure content is exposed to assistive technology when off screen

Quick link element. The entire expanded content area is enclosed in a

Page 11: DOM, "Quick link" element and "Marketing" heading are exposed to assistive technology when off screen

Steps to reproduce

1. Using just your screen reader, swipe through the above.
2. Observe that the screen reader cursor goes onto elements that are not visible on the page.

Recommendation

Make sure that the content is programmatically ordered in such a way that it provides an appropriate reading order. It does not always need to exactly match the visuals, but it must provide users with a meaningful reading order. In this case, make sure that elements that are not visible on the page are not exposed to assistive technologies. Screen reader users should be able to swipe through the page in a meaningful order.

In this case, make sure that off-screen elements are not exposed to assistive technology. Note that we recommend avoiding a design pattern where headings are removed from the screen when content under them is disclosed. This pattern could potentially cause confusion. See also issue ADR_20590 - 108.

Make sure that the off-screen elements are excluded from or included in the focus and reading orders as appropriate.

Removing elements from the focus and reading orders can be achieved with the HTML `hidden` or CSS `display:none`. Both will hide the element from all users. This same can be programmatically achieved by using both `aria-hidden="true"` and `tabindex="-1"` on the same element. For example:

```
<h3><button aria-expanded="false">Marketing</button></h3>
<!-- The content below is hidden until the button is activated -->
<div hidden>
  <p>Share our journey...</p>
  <a href="#">Find out more about marketing at Unilever</a>
</div>
```

However, note that `aria-hidden="true"` and `tabindex="-1"` will not visually remove elements from the page.

Removing these attributes will re-insert the elements into the focus order based on their location in the DOM, allowing users to navigate onto them with a keyboard. Make sure that this is done as appropriate, such as when the user toggles a parent control that shows and hides these interactive elements.

For more information, see [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - Using Tabindex](#), [MDN - aria-hidden](#), and [MDN - hidden](#).

[Return to priority table](#)

ADR_20590-96 MEDIUM PRIORITY OPEN

Colour relied upon to communicate information

Success Criteria 1.4.1: Use of Color [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Button

What happened

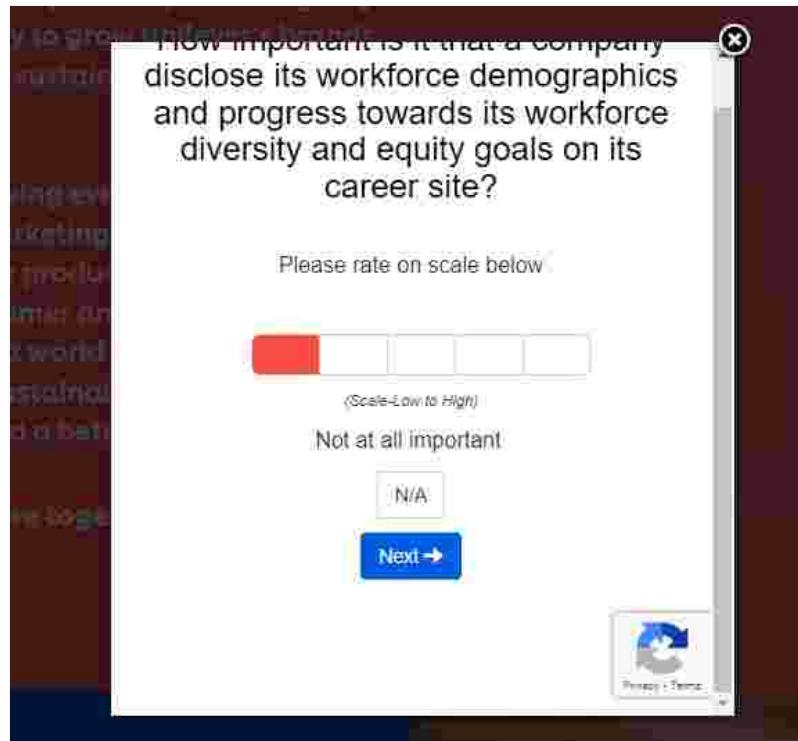
Understanding content and operating interactive elements must not rely on colour. There is information that is visually communicated exclusively through colour. This may particularly disadvantage low vision and colour blind users who may be excluded by the visual reliance on colour.

Where it happened

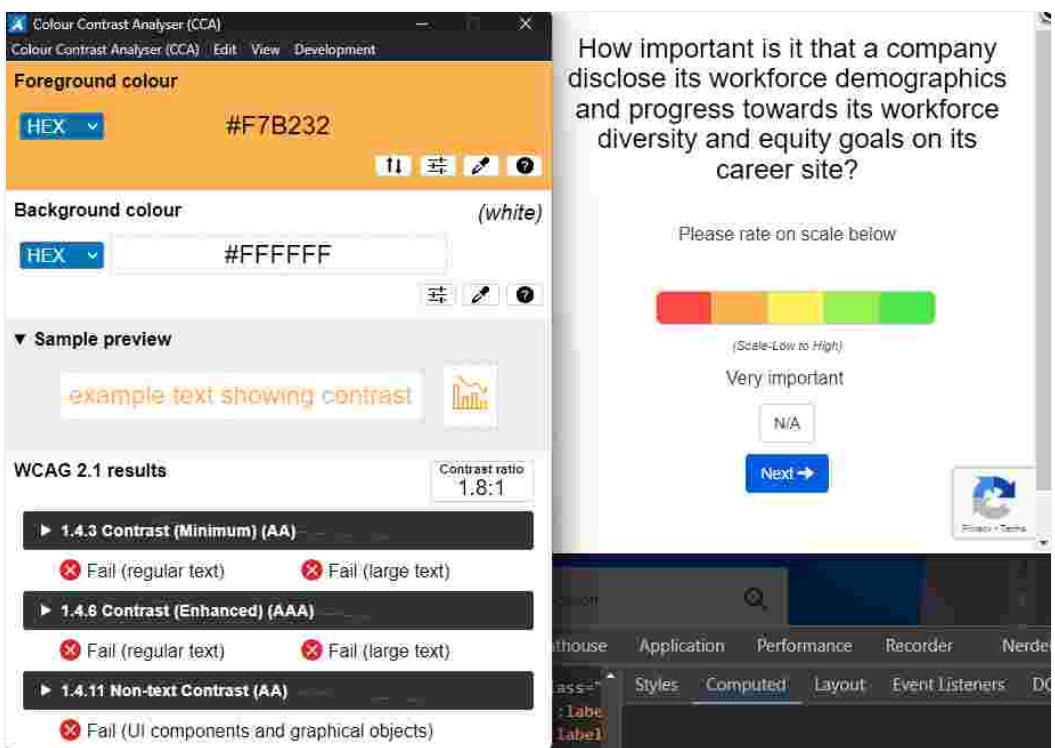
This can be seen on page 13 with the ratings buttons. When a rating button is selected, that button and all the rating buttons lower than it have a coloured background. The coloured backgrounds for the second, third, fourth, and fifth buttons have less than a 3:1 contrast ratio with the default white background of the buttons (#FFFFFF). Therefore, colour is relied on to convey which of these buttons is selected.

The ratios are as follows:

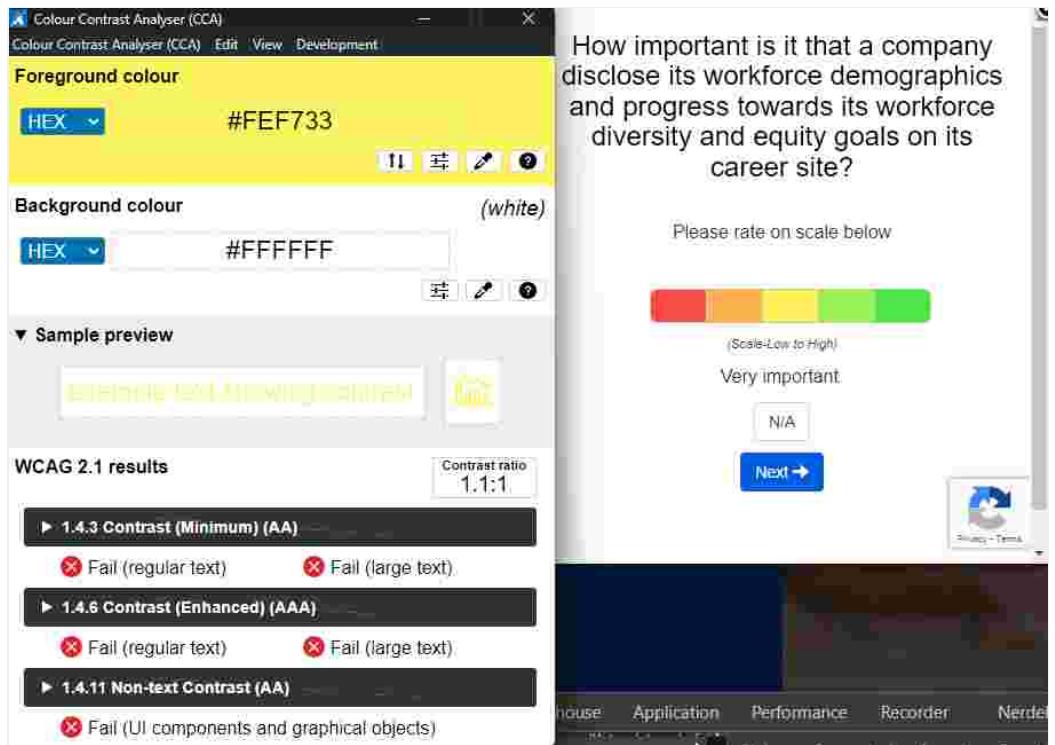
- The orange (#F7B232) has a 1.8:1 contrast ratio on white.
- The yellow (#FEF733) has a 1.1:1 contrast ratio on white.
- The light green (#A2F11D) has a 1.4:1 contrast ratio on white.
- The dark green (#6AE407) has a 1.7:1 contrast ratio on white.



Page 13: desktop, ratings buttons showing red selected state of first button and white unselected states of remaining buttons



Page 13: desktop, Colour Contrast Analyser, orange button has less than 3:1 contrast ratio with white



Page 13: desktop, Colour Contrast Analyser, yellow button has less than 3:1 contrast ratio with white

Colour Contrast Analyser (CCA)

Foreground colour: #A2F11D

Background colour: (white)

Sample preview: example text showing contrast

WCAG 2.1 results: Contrast ratio 1.4:1

- ▶ 1.4.3 Contrast (Minimum) (AA)
 - Fail (regular text)
 - Fail (large text)
- ▶ 1.4.6 Contrast (Enhanced) (AAA)
 - Fail (regular text)
 - Fail (large text)
- ▶ 1.4.11 Non-text Contrast (AA)
 - Fail (UI components and graphical objects)

Please rate on scale below

(Scale-Low to High)

Very important

N/A

Next →

Copy - Test

Page 13: desktop, Colour Contrast Analyser, light green button has less than 3:1 contrast ratio with white

Colour Contrast Analyser (CCA)

Foreground colour: #6AE407

Background colour: (white)

Sample preview: example text showing contrast

WCAG 2.1 results: Contrast ratio 1.7:1

- ▶ 1.4.3 Contrast (Minimum) (AA)
 - Fail (regular text)
 - Fail (large text)
- ▶ 1.4.6 Contrast (Enhanced) (AAA)
 - Fail (regular text)
 - Fail (large text)
- ▶ 1.4.11 Non-text Contrast (AA)
 - Fail (UI components and graphical objects)

Please rate on scale below

(Scale-Low to High)

Very important

N/A

Next →

Copy - Test

Page 13: desktop, Colour Contrast Analyser, dark green button has less than 3:1 contrast ratio with white

Steps to reproduce

- Access the information.
- Observe that colour is used to communicate the information.
- Observe that there is not an alternative way to visually access the information.

Recommendation

Make sure that information is not exclusively visually communicated through colour. Colour can be used to communicate information. However, colour must not be the only visual means of conveying information.

As well, hidden information and ARIA cannot be the only alternative to colour. It is important that information communicated through colour is programmatically determined, such that it is exposed to assistive technology. However, this success criterion requires that there be visual cues that are available to all users.

The simplest solution is providing the information that the colour is communicating through text. For example:

```
<ul>
  <li class="GreenUp">Tesla (up 2%)</li>
  <li class="RedDown">Apple (down 1%)</li>
</ul>
```

Otherwise, graphs can use patterns, textures, and tables to communicate information. For examples, see [Highcharts - Line Chart Accessibility Demo](#) and [GOV - Magna Charta Example](#). States, such as selected, activate, and invalid, can use additional visual cues, such as changed font weight, text decoration, icons, or text. For example:

```
<button class="boldText" role="tab" aria-selected="true"> ... </button>
<button class="standardText" role="tab" aria-selected="false"> ... </button>
```

For more information, see [PSU - Charts & Accessibility](#), [Penn State - Colour Coding](#), and [Princeton - Colour Alone](#).

[Return to priority table](#)

ADR_20590-11 MEDIUM PRIORITY OPEN

Regular link text does not have sufficient contrast

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 03, 06, 07

Users Affected Vision

Component Link

What happened

Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

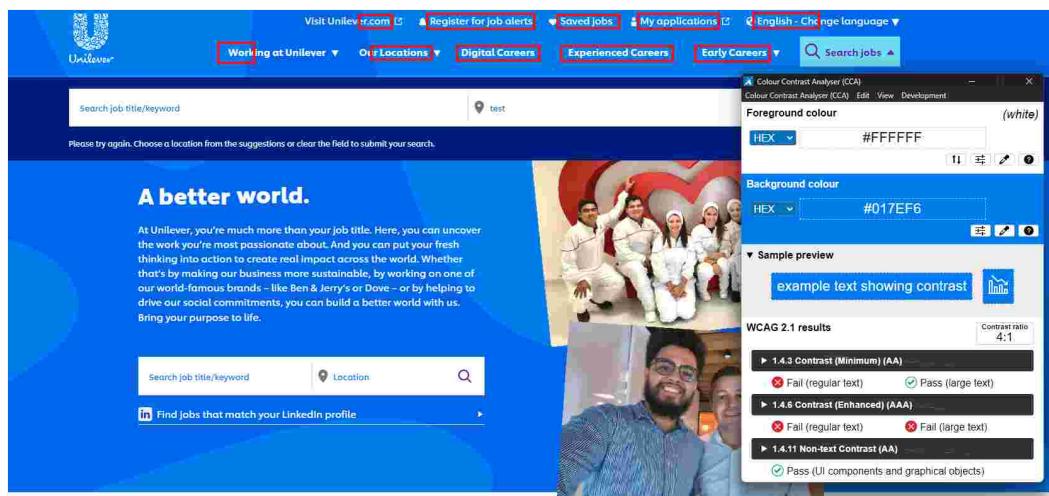
- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular link text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

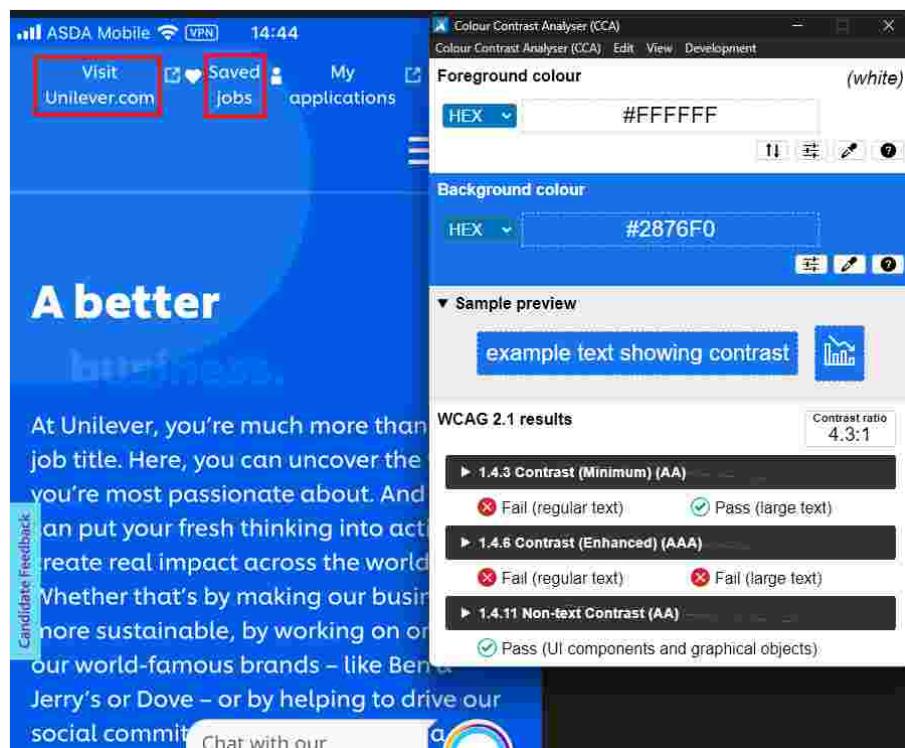
This can be seen on pages 1, 3 and 6, on desktop, with the links and buttons with white text placed over light blue portions of the background. The white text (#FFFFFF) has a 4:1 contrast ratio on the light portions of the blue background (#017EF6). This occurs in the following elements: * On page 1, with the links and buttons with white text in the header navigation, such as “Visit Unilever.com”, “Register for job alerts”, “Working at Unilever”, and so on. * On pages 3 and 6, with “Find jobs that match your LinkedIn profile” link in the hero banner.

Note that the background is animated and the light portions do not always occur in the same place. However, all of the navigation links and buttons with white text are affected.



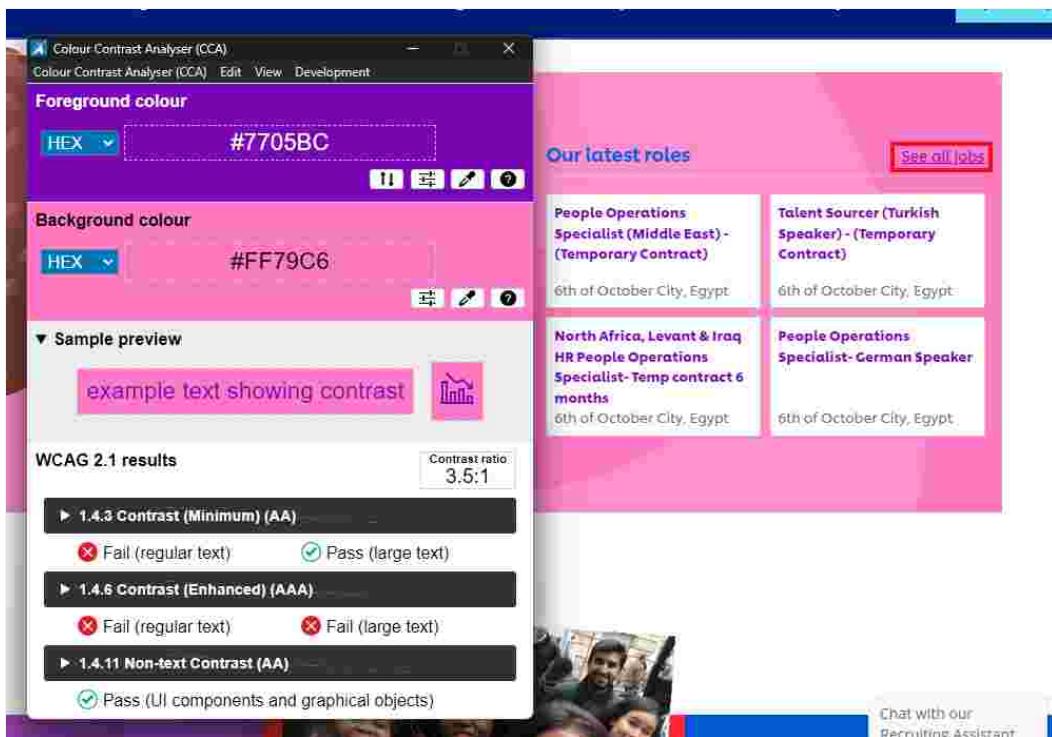
Page 1: desktop, Colour Contrast Analyser, white navigation link and button text does not have sufficient contrast on light portions of background

This can be seen on pages 1 and 3, on iOS, with links with white text placed over light blue portions of the background. The white text (#FFFFFF) has a 4:3 contrast ratio on the light blue background (#2876F0). This occurs in the following elements: * On page 1, with the “Visit Unilever.com” and “Saved jobs” links in the header navigation. * On page 3, with “Find jobs that match your LinkedIn profile” link in the hero banner.



Page 1: iOS, Colour Contrast Analyser, white "Visit Unilever.com" and "Saved jobs" text does not have sufficient contrast on background

This can also be seen on page 3, on desktop and iOS, with the "See all jobs" link in the "Unilever worldwide" section. The purple link text (#7705BC) has a 3.5:1 contrast ratio on the pink background (#FF79C6).



Page 3: desktop, Colour Contrast Analyser, purple "See all jobs" link text does not have sufficient contrast on background

This can also be seen on page 7, on desktop and iOS, with the "here" link in the "Did you know?" section. The blue link text (#035A8F) has a 1.9:1 contrast ratio on the magenta background (#FF087A).

Colour Contrast Analyser (CCA)

Foreground colour: #035A8F

Background colour: #FF087A

WCAG 2.1 results:

- 1.4.3 Contrast (Minimum) (AA): Fail (regular text), Fail (large text)
- 1.4.6 Contrast (Enhanced) (AAA): Fail (regular text), Fail (large text)
- 1.4.11 Non-text Contrast (AA): Fail (UI components and graphical objects)

Page 7: desktop, Colour Contrast Analyser, blue "here" link text does not have sufficient contrast on background

This can also be seen on page 7, on desktop and iOS, with the "Search [location] jobs" button link in the "Netherlands" section. The blue link text (#005EEF) has a 3.5:1 contrast ratio on the light blue background (#77DDF4).

Colour Contrast Analyser (CCA)

Foreground colour: #005EEF

Background colour: #77DDF4

WCAG 2.1 results:

- 1.4.3 Contrast (Minimum) (AA): Fail (regular text), Pass (large text)
- 1.4.6 Contrast (Enhanced) (AAA): Fail (regular text), Fail (large text)
- 1.4.11 Non-text Contrast (AA): Pass (UI components and graphical objects)

Page 7: desktop, Colour Contrast Analyser, blue "Search Rotterdam jobs" link text does not have sufficient contrast on background

Steps to reproduce

1. Observe the link elements.
2. Get the colour values of the text and background colours.
3. Insert these values into a colour contrast analyser.
4. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-54 MEDIUM PRIORITY OPEN

Regular text does not have sufficient contrast

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07, 08, 14

Users Affected Vision

Component Plain Text

What happened

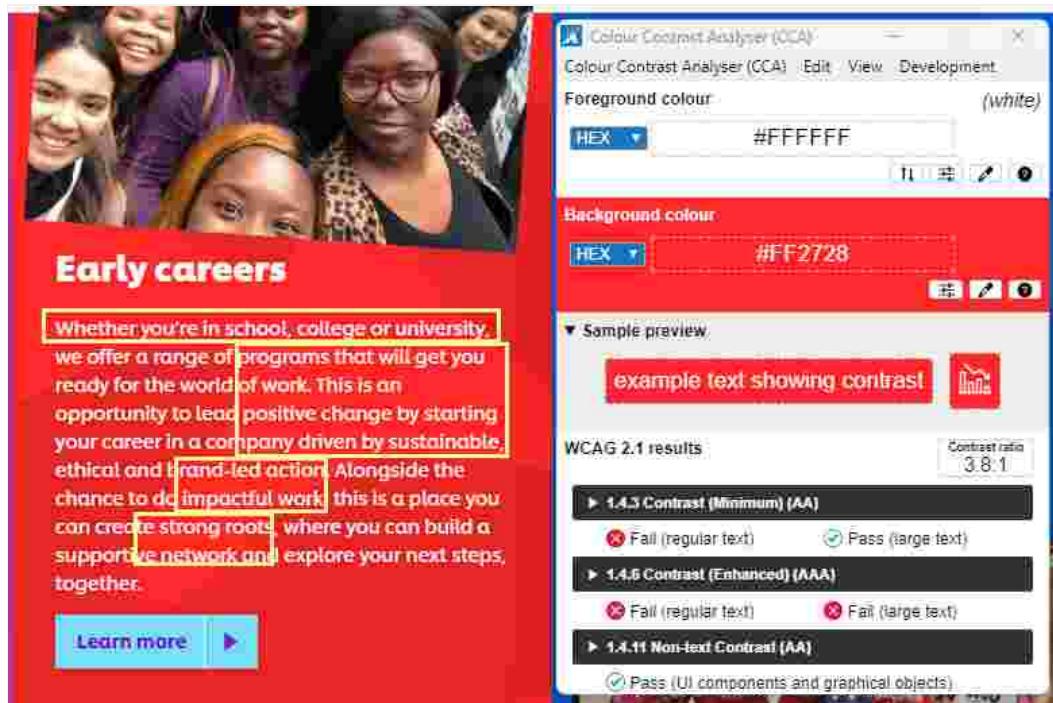
Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on page 3, on desktop and iOS, with the text under the “Early careers” section. The white text (#FFFFFF) has a 3.8:1 contrast ratio on the light portions of the red background (#FF2728).



Page 3: desktop, Colour Contrast Analyser, text under “Early careers” does not have sufficient contrast on light portions of background

This can also be seen on page 7, on desktop and iOS, with the text under the “Did you know?” section. The white text (#FFFFFF) has a 3.8:1 contrast ratio on the light portions of the magenta background (#FF087A).

The screenshot shows the Colour Contrast Analyser (CCA) interface. The foreground color is set to white (#FFFFFF) and the background color is set to #FF087A (pink). In the 'Sample preview' section, there is a box containing text: "Did you know that Unilever Netherlands has a key part in Unilever's heritage? It was the joint venture between Jurgens and Van Den Bergh (Margarine Unie) in the Netherlands and Lever Brothers in the United Kingdom that formed Unilever in 1930? It all started with margarine and soap, but soon after Unilever started expanding its product portfolio and still does to date. Want to learn more about our heritage? Find out more...". Below this, the WCAG 2.1 results show:

Test	Status
1.4.3 Contrast (Minimum) (AA)	Pass (large text)
1.4.6 Contrast (Enhanced) (AAA)	Fail (regular text), Fail (large text)
1.4.11 Non-text Contrast (AA)	Pass (UI components and graphical objects)

Contrast ratio: 3.8:1

Page 7: desktop, Colour Contrast Analyser, text under “Did you know?” does not have sufficient contrast on light portions of background

This can also be seen on page 8, on desktop and iOS, with the text under “Build A Unique Digital Career”. The white text (#FFFFFF) has a 3.2:1 contrast ratio on the red background (#FF544F).

The screenshot shows the Colour Contrast Analyser (CCA) interface. The foreground color is set to white (#FFFFFF) and the background color is set to #FF544F (red). In the 'Sample preview' section, there is a box containing text: "Build A Unique Digital Career.". Below this, the WCAG 2.1 results show:

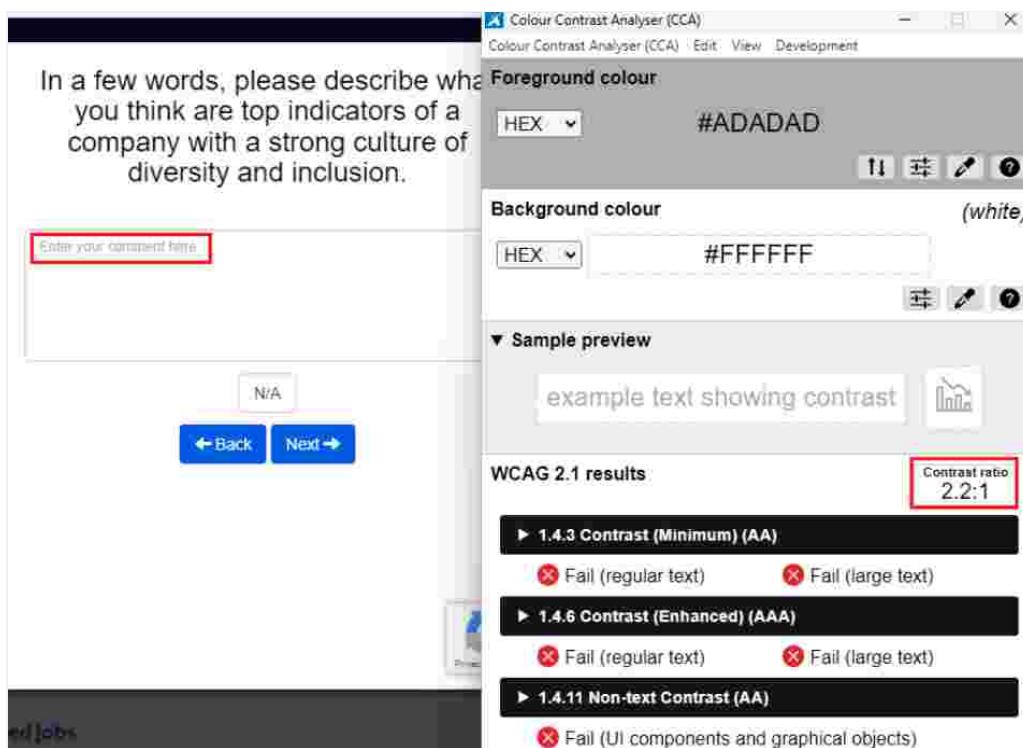
Test	Status
1.4.3 Contrast (Minimum) (AA)	Fail (regular text), Pass (large text)
1.4.6 Contrast (Enhanced) (AAA)	Fail (regular text), Fail (large text)
1.4.11 Non-text Contrast (AA)	Pass (UI components and graphical objects)

Contrast ratio: 3.2:1

Page 8: desktop, Colour Contrast Analyser, text under “Build a Unique Digital Career” does not have sufficient contrast

This can also be seen on page 14, on desktop and iOS, with the “Enter your comment here” placeholder text. The light grey (#ADADAD) placeholder text has an insufficient contrast ratio of 2.2:1 against the white (#FFFFFF) background.

Note that, because the boundary of the input field does not have sufficient contrast, the placeholder text acts as a visual indicator of the presence of the field. See issue ADR_20590 - 40.



Page 14: desktop, Colour Contrast Analyser, placeholder text “Enter your comment here” does not have sufficient contrast with the background

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-60 MEDIUM PRIORITY OPEN

Regular button text does not have sufficient contrast

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Mobile (iOS)

Pages Affected 04, 08

Users Affected Vision

Component Button

What happened

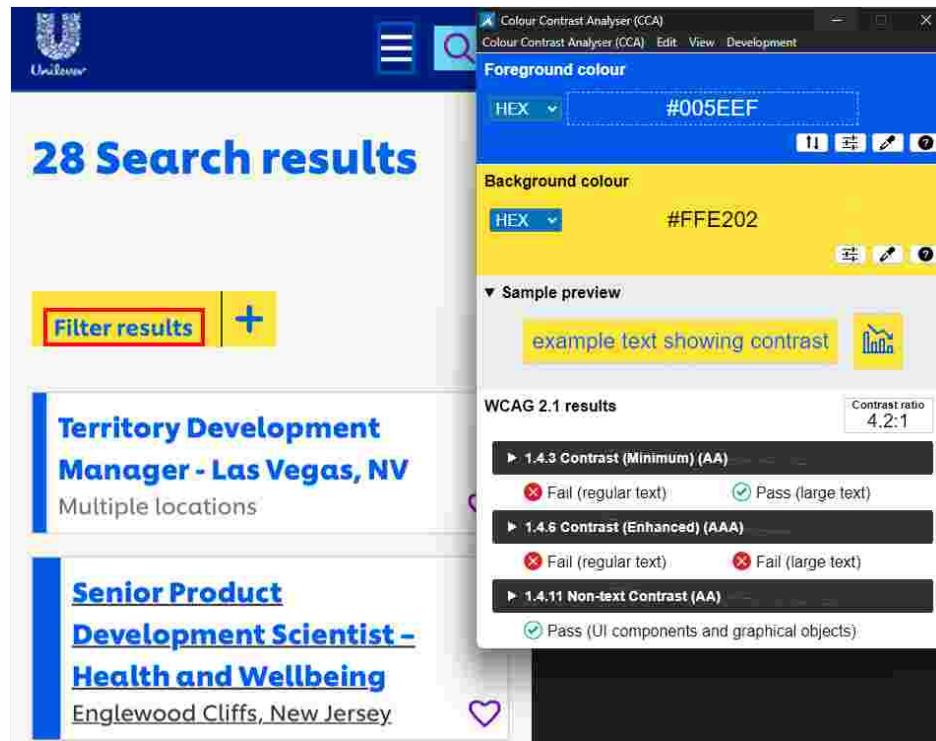
Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular button text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on page 4 on iOS with the “Filter results” button. The blue text (#005EEF) has a 4.2:1 contrast ratio with the yellow background (#FFE202).



Page 4: iOS, Colour Contrast Analyser, “Filter results” button text does not have sufficient contrast

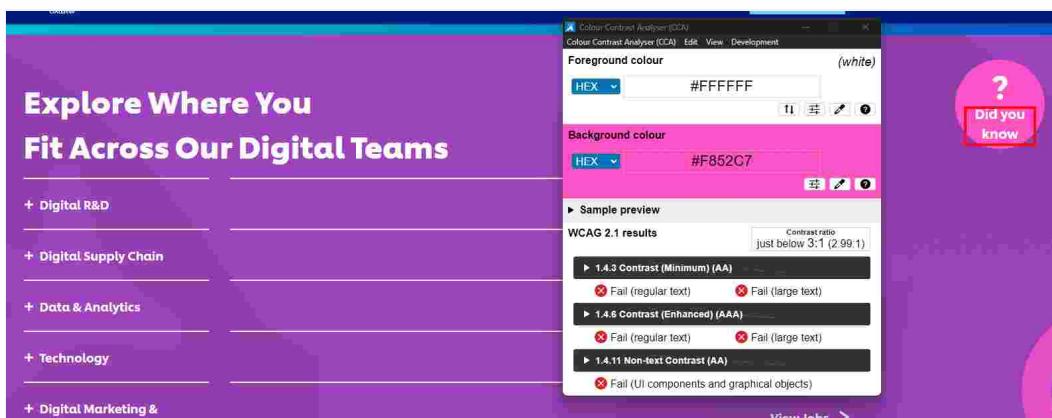
This can also be seen on page 8 on desktop and iOS with the buttons under “Why Choose a Digital Future in Unilever?” such as “Brilliantly different, together”. The dark blue text (#1F36C7) has a contrast ratio of just below 4.5:1 with the teal background.



Page 8: desktop, Colour Contrast Analyser, button text under “Why Choose a Digital Future in Unilever?” does not have sufficient contrast

This can also be seen on page 8 on desktop and iOS with white button text on a magenta background. The white text (#FFFFFF) has a contrast ratio of just below 3:1 with the magenta background (#F852C7). This occurs with the following elements.

- The “Did you know” button under “Explore Where You Fit Across Our Digital Teams”.
- The “Learn more” and “Subscribe” buttons under “Contact us” and “Sign up for job alerts”.



Page 8: desktop, Colour Contrast Analyser, “Did you know” button text does not have sufficient contrast

Steps to reproduce

1. Observe the button elements.
2. Get the colour values of the text and background colours.
3. Insert these values into a colour contrast analyser.
4. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-69 MEDIUM PRIORITY OPEN

Large text does not have sufficient contrast

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07

Users Affected Vision

Component Plain Text

What happened

Informative text must pass colour contrast requirements. For large text, the required contrast ratio is 3:1. Large text is defined as the following.

- At least 18pt or 24px with regular font weight (less than 700).
- At least 14 point or 18.5px with bold font weight (at least 700).

There is large text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

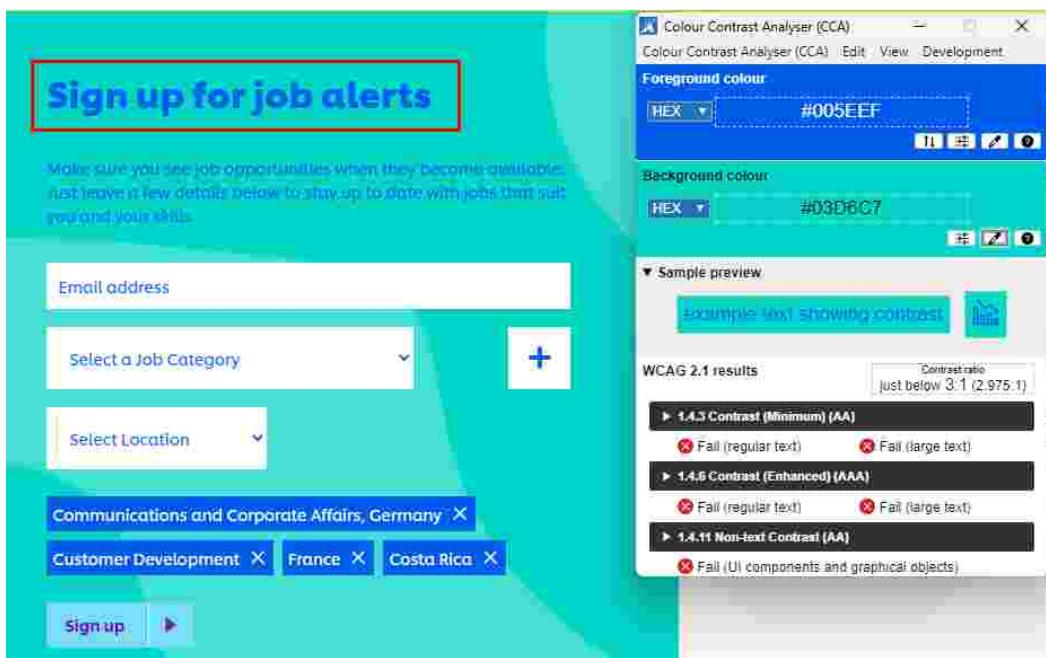
Where it happened

This can be seen on page 3, on desktop and iOS, with the “worldwide” large text in the “Unilever worldwide” section. The blue text (#005EEF) has a contrast ratio of 2.3:1 against the pink background (#FF79C6).



Page 3: Colour Contrast Analyser, example of insufficient contrast on large blue text over a pink background

This can also be seen on pages 3 and 7, on desktop and iOS, with the large blue text placed over a cyan background. The blue text (#005EEF) has a contrast ratio of 2.98:1 against the cyan background (#03D6C7). This occurs in the following elements:- On page 3, with the “Sign up for job alerts” heading that titles the respective section. - On page 7, with the “Netherlands” heading that titles the respective section.



Page 3: Colour Contrast Analyser, example of insufficient contrast on large blue text over a cyan background

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-72 MEDIUM PRIORITY OPEN

Regular text does not have sufficient contrast on two-coloured background (blue text)

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07

Users Affected Vision

Component Plain Text, Heading

What happened

Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on page 3, on desktop and iOS, with the blue text placed over darker pink portions of the “Unilever worldwide” section. The blue text (#005EEF) has a 2.3:1 contrast ratio on the darker portions of the pink background (#FF79C6).

WCAG 2.1 results	Contrast ratio
► 1.4.3 Contrast (Minimum) (AA) Fail (regular text) Fail (large text)	2.3:1
► 1.4.6 Contrast (Enhanced) (AAA) Fail (regular text) Fail (large text)	
► 1.4.11 Non-text Contrast (AA) Fail (UI components and graphical objects)	

Page 3: Colour Contrast Analyser, blue text does not have sufficient contrast on darker pink portions of the background

This can be seen on page 3, on desktop and iOS, with the blue text placed over lighter pink portions of the “Unilever worldwide” section. The blue text (#005EEF) has a 2.6:1 contrast ratio on the lighter portions of the pink background (#FF8DD0).

The screenshot shows a job listing page from the Unilever careers website. It features a large image of a Magnum ice cream bar on the left. The background of the main content area is a darker shade of cyan. Several job descriptions are listed in boxes, with blue text for titles and details. A sidebar on the right contains a 'Colour Contrast Analyser' tool window. The tool shows the foreground color as #005EEF and the background color as #FF8DD0. It displays WCAG 2.1 results with multiple items failing the AA and AAA contrast requirements.

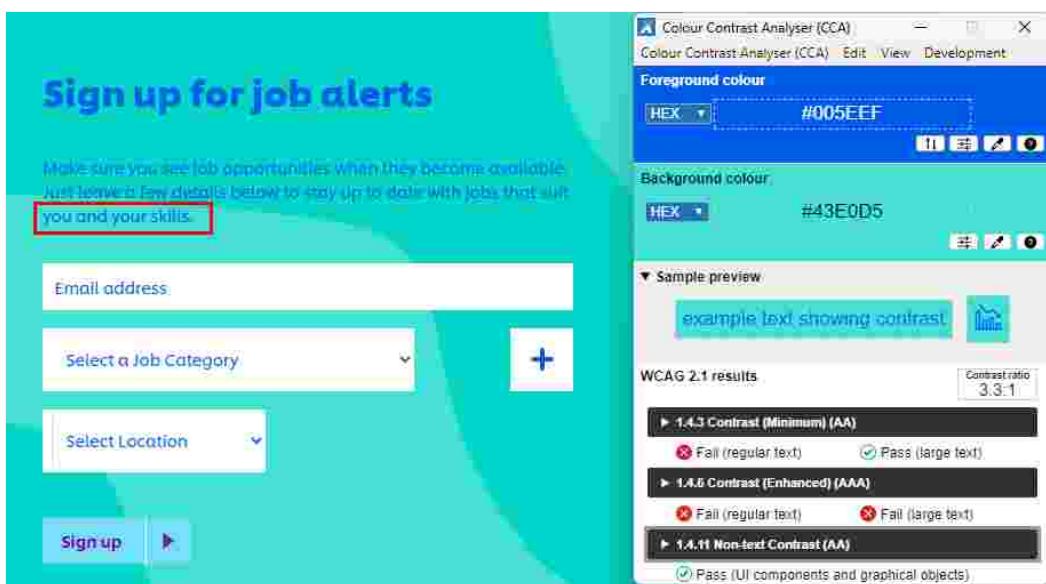
Page 3: Colour Contrast Analyser, blue text does not have sufficient contrast on darker pink portions of the background

This can be seen on pages 3 and 7, on desktop and iOS, with the blue text placed over darker cyan portions of the background of some sections. The blue text (#005EEF) has a 2.98:1 contrast ratio on the darker portions of the cyan background (#03D6C7). This occurs in the following elements: - On page 3, with the text under the “Sign up for job alerts” section. - On page 7, with the text under the “Netherlands” section.

The screenshot shows a sign-up form for job alerts. The background is a darker shade of cyan. The form includes fields for 'Email address', 'Select a Job Category', and 'Select Location'. The 'Email address' field has a red border. The WCAG 2.1 results from the Colour Contrast Analyser show multiple items failing the AA and AAA levels due to insufficient contrast between the blue text and the darker cyan background.

Page 3: Colour Contrast Analyser, blue text does not have sufficient contrast on darker cyan portions of the background

This can be seen on pages 3 and 7, on desktop and iOS, with the blue text placed over lighter cyan portions of the background of some sections. The blue text (#005EEF) has a 3.3:1 contrast ratio on the lighter portions of the cyan background (#43E0D5). This occurs in the following elements: - On page 3, with the text under the “Sign up for job alerts” section. - On page 7, with the text under the “Netherlands” section.



Page 3: Colour Contrast Analyser, blue text does not have sufficient contrast on lighter cyan portions of the background

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-76 MEDIUM PRIORITY OPEN

Regular combobox text does not have sufficient contrast

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 07

Users Affected Vision

Component Button, Combobox

What happened

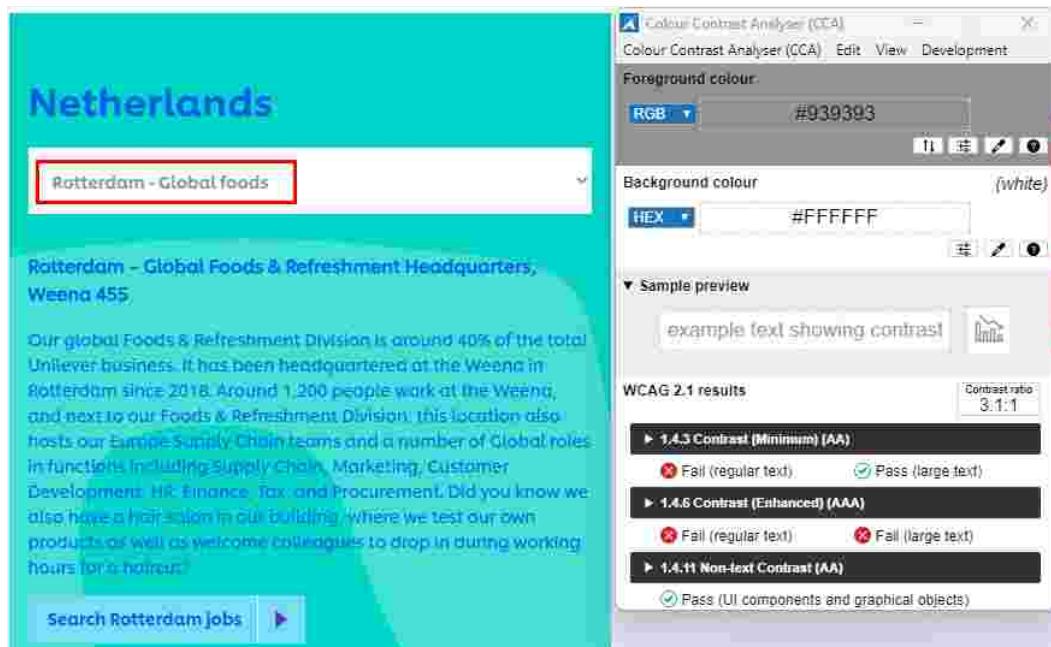
Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular combobox text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

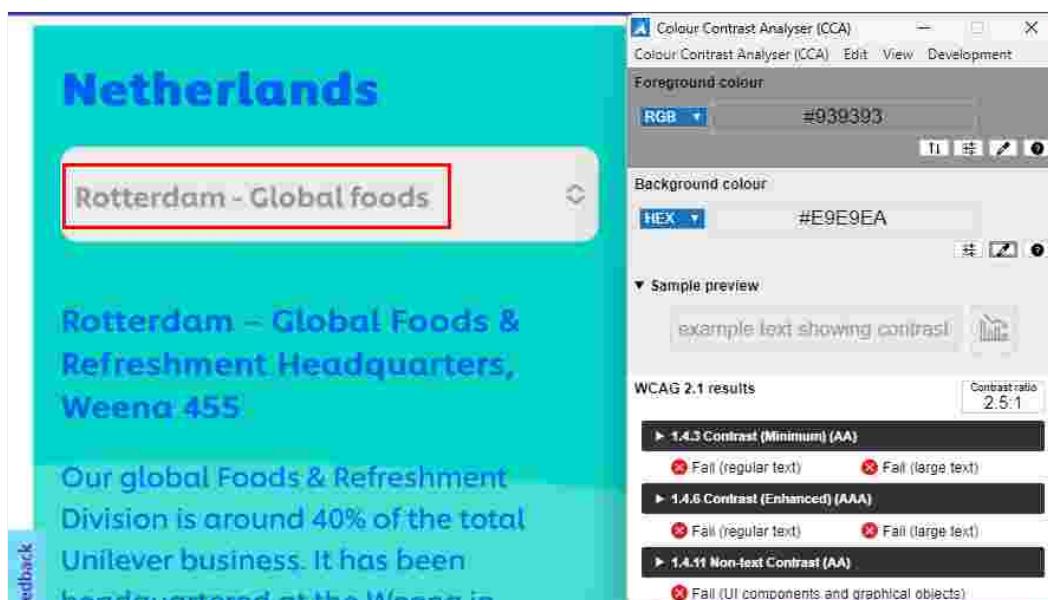
Where it happened

This can be seen on page 7, on desktop, with the combobox under the “Netherlands” section. The grey text (#939393) has a 3.1:1 contrast ratio on the white background (#FFFFFF).



Page 7: Colour Contrast Analyser, grey text does not have sufficient contrast on white background

This can be seen on page 7, on iOS, with the combobox under the “Netherlands” section. The grey text (#939393) has a 2.5:1 contrast ratio on the light grey background (#E9E9EA).



Page 7: Colour Contrast Analyser, grey text does not have sufficient contrast on light grey background

Steps to reproduce

1. Observe the combobox elements.
2. Get the colour values of the text and background colours.
3. Insert these values into a colour contrast analyser.
4. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-82 MEDIUM PRIORITY OPEN

Regular text does not have sufficient contrast on gradient background

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop

Pages Affected 08

Users Affected Vision

Component Plain Text

What happened

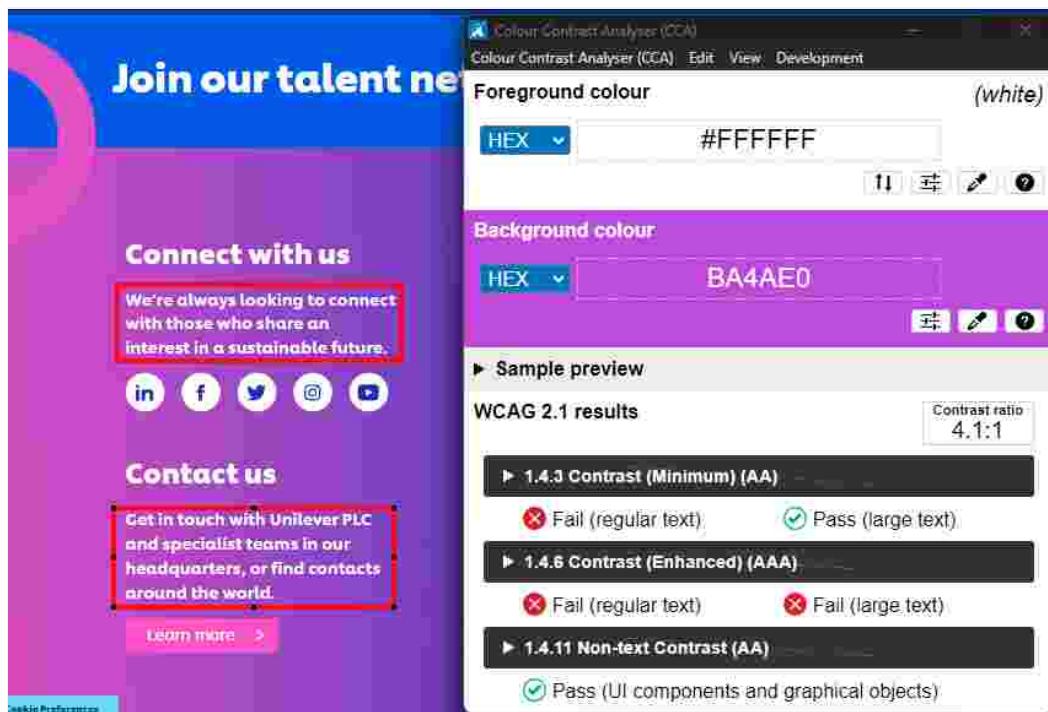
Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

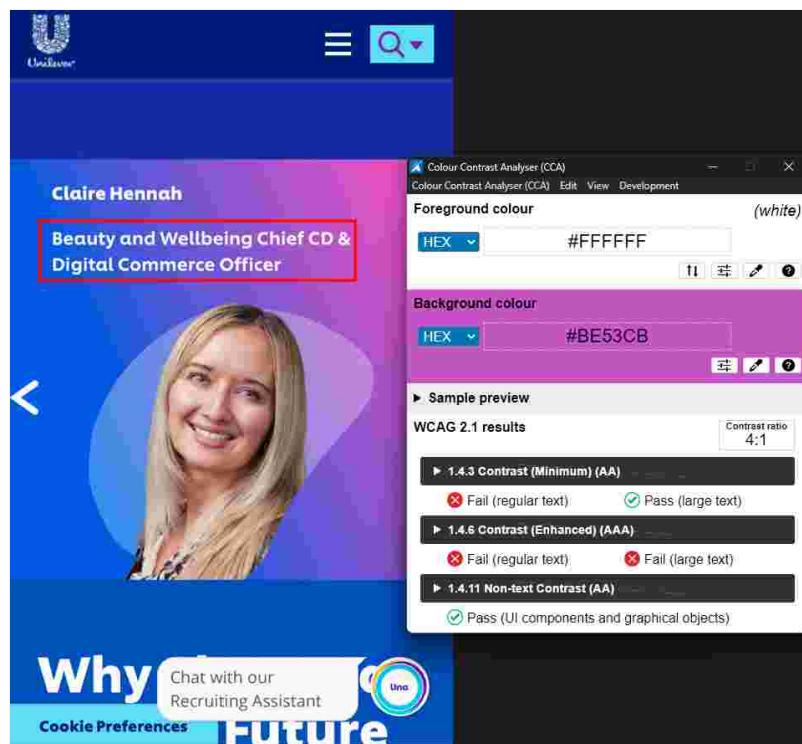
Where it happened

This can be seen on page 8 on desktop with the paragraph text under “Connect with us” and “Contact us”. The background colour is a gradient, and the text does not always have sufficient contrast with the background. For example, the white text (#FFFFFF) has a 4.1:1 contrast ratio on the lighter purple portions of the background at the left edge of the text (#BA4AE0).



Page 8: desktop, Colour Contrast Analyser, text under “Connect with us” and “Contact us” does not have sufficient contrast against portions of background

This can be seen on page 8 on iOS with the job titles in the “Meet Our People” carousel. The background colour is a gradient, and the text does not always have sufficient contrast with the background. For example, the white text “Beauty and Wellbeing Chief CD & Digital Commerce Officer” (#FFFFFF) has a 4.1:1 contrast ratio on the lighter purple portions of the background at the right edge of the text (#BE53CB).



This can also be seen on page 8 on desktop

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-83 MEDIUM PRIORITY OPEN

Regular link text does not have sufficient contrast on gradient background

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop

Pages Affected 08

Users Affected Vision

Component Link

What happened

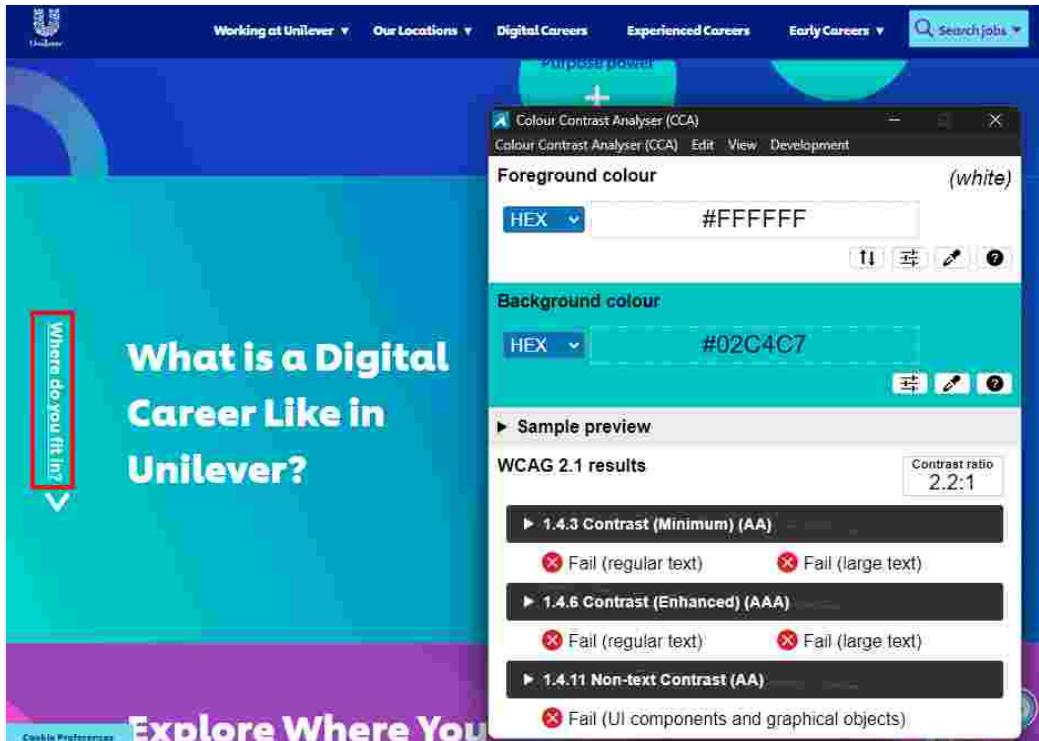
Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular link text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

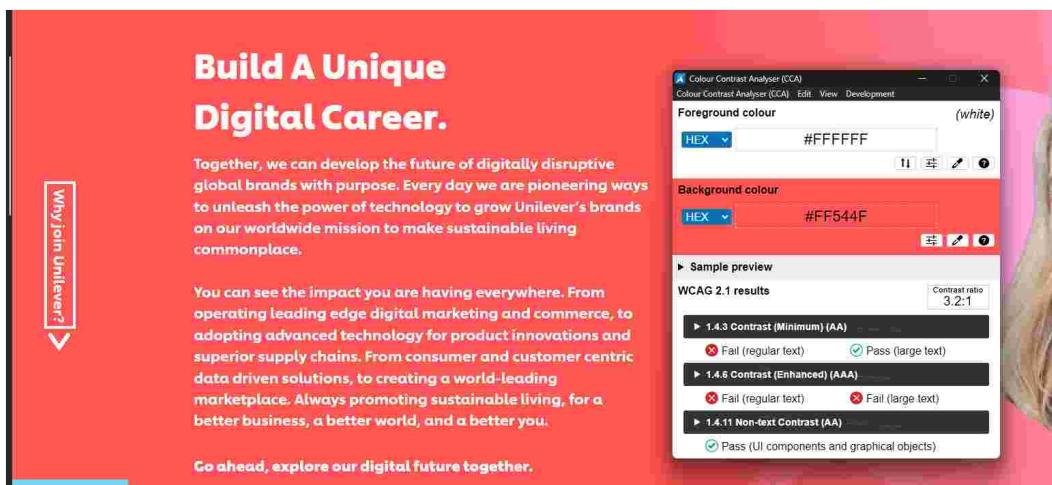
Where it happened

This can be seen on page 8 on desktop with the “Where do you fit in?” same-page link. The link text does not have sufficient contrast with the gradient background. For example, the white text (#FFFFFF) has a 2.2:1 contrast ratio where the cyan background is #03C3C7.



Page 8: desktop, “Where do you fit in?” link does not have sufficient contrast

This can also be seen on page 8 on desktop and iOS with “Why join Unilever?” same-page link. The white text (#FFFFFF) has a 3.2:1 contrast ratio on the red gradient background (#FF544F).



Page 8: desktop, Colour Contrast Analyser, "Why join Unilever?" link does not have sufficient contrast

Steps to reproduce

1. Observe the link elements.
2. Get the colour values of the text and background colours.
3. Insert these values into a colour contrast analyser.
4. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-84 MEDIUM PRIORITY OPEN

Large text does not have sufficient contrast on gradient background

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop

Pages Affected 08

Users Affected Vision

Component Heading

What happened

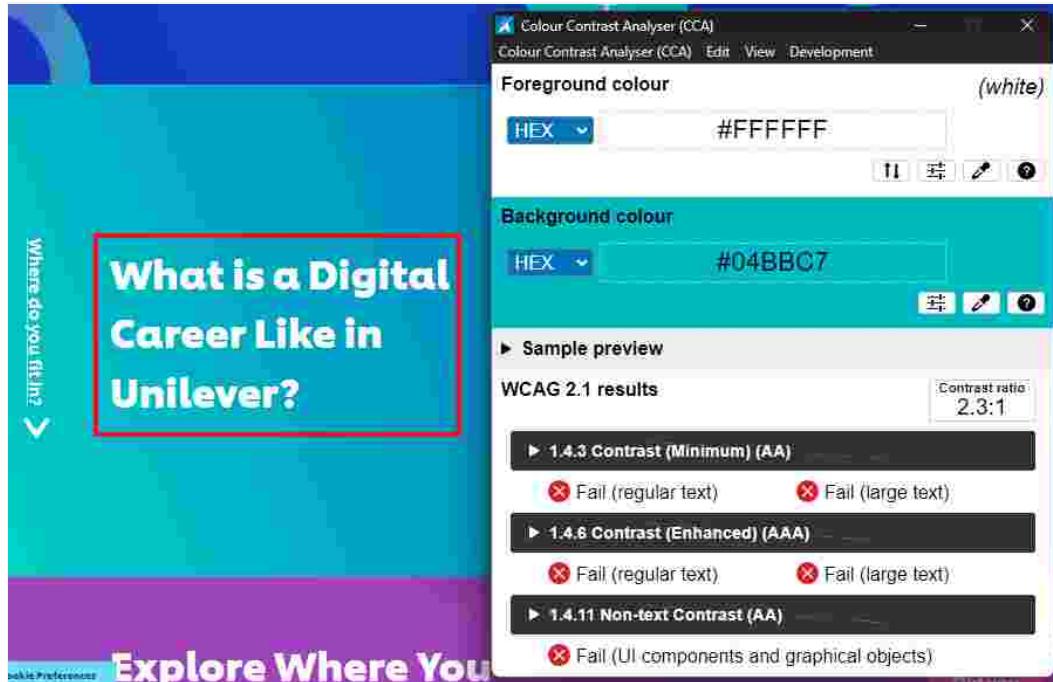
Informative text must pass colour contrast requirements. For large text, the required contrast ratio is 3:1. Large text is defined as the following.

- At least 18pt or 24px with regular font weight (less than 700).
- At least 14 point or 18.5px with bold font weight (at least 700).

There is large text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on page 8 on desktop with the large heading text “What is a Digital Career Like in Unilever?”. The text does not have sufficient contrast with the gradient background. For example, the white text (#FFFFFF) has a 2.3:1 contrast ratio where the cyan background is #04BBC7.



Page 8: desktop, Colour Contrast Analyser, “What is a Digital Career Like in Unilever?” text does not have sufficient contrast on gradient background

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-86 MEDIUM PRIORITY OPEN

Regular button text does not have sufficient contrast on keyboard focus or mouse hover

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 08

Users Affected Vision

Component Button

What happened

Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

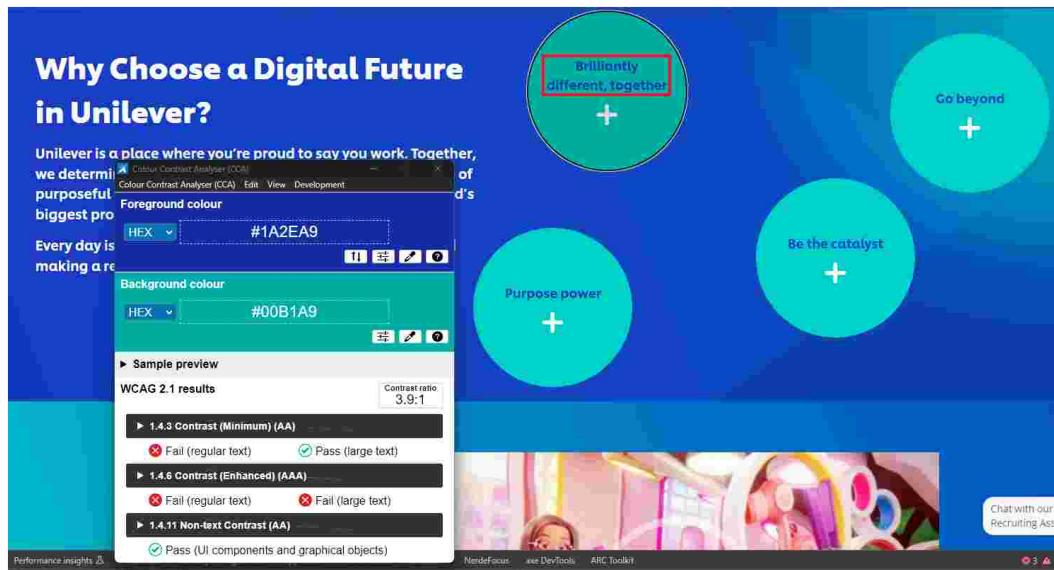
- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular text that does not meet the contrast ratio when receiving mouse hover. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on page 8 on desktop and iOS with the buttons under “Why Choose a Digital Future in Unilever”. On keyboard focus or mouse hover, the dark blue text (#1A2EA9) has a 3.9:1 contrast ratio on the teal background (#00B1A9).

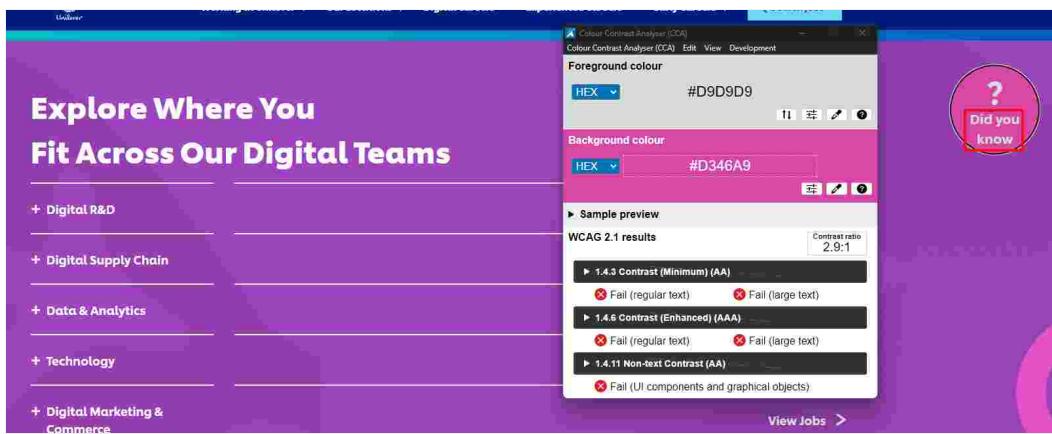
Note that on iOS this occurs when the buttons receive the screen reader cursor or are pressed and held.



Page 8: desktop, Colour Contrast Analyser, “Brilliantly different, together” button does not have sufficient contrast on focus

This can also be seen on page 8 on desktop and iOS with the “Did you know” button under “Explore Where You Fit Across Our Digital Teams”. On keyboard focus or mouse hover, the off-white text (#D9D9D9) has a contrast ratio of 2.9:1 with the pink background (#D346A9).

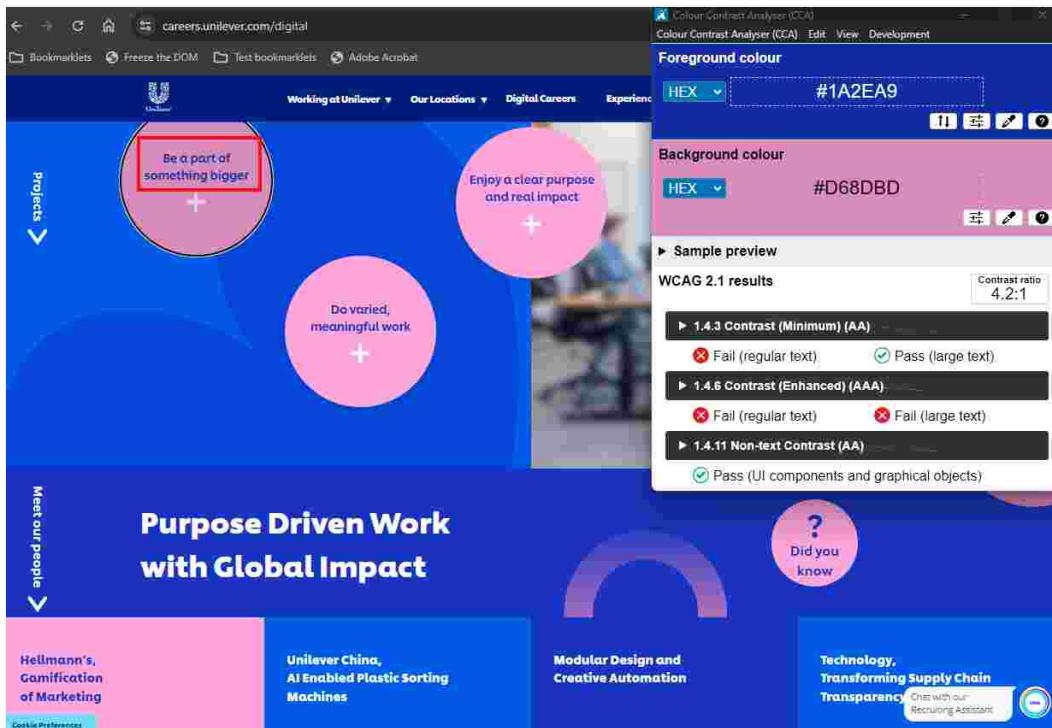
Note that on iOS this occurs when the button receives the screen reader cursor or is pressed and held.



Page 8: desktop, Colour Contrast Analyser, “Did you know” button does not have sufficient contrast on focus

This can also be seen on page 8 on desktop and iOS with the buttons with blue text on a pink background. On keyboard focus or mouse hover, the blue text (#1A2EA9) has a contrast ratio of 4.2:1 with the pink background (#D68DBD). This affects the following elements:

- The buttons under “Why Join Unilever as a Digital Talent?”.
- The “Did you know” button under “Purpose Driven Work with Global Impact”.
- The “Hellmann’s, Gamification of Marketing” button under “Purpose Driven Work with Global Impact”.



Page 8: desktop, Colour Contrast Analyser, “Be a part of something bigger” button does not have sufficient contrast on focus

Steps to reproduce

1. Mouse hover on the above elements.
2. Get the colour values of the text and background colours.
3. Insert these values into a colour contrast analyser.
4. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements.

This includes when interactive elements receive focus and mouse hover. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-89 MEDIUM PRIORITY OPEN

Regular text does not have sufficient contrast on two-coloured background (white text)

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 04, 06, 07, 08, 11

Users Affected Vision

Component Plain Text

What happened

Informative text must pass colour contrast requirements. For regular text, the required contrast ratio is 4.5:1. Regular text is defined as the following.

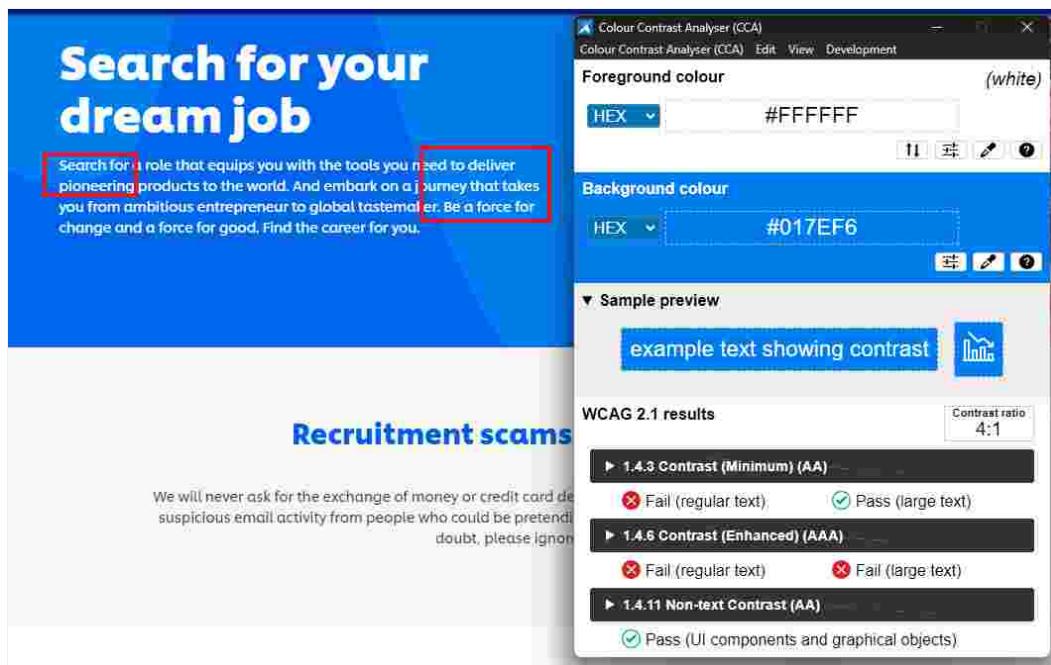
- Less than 18pt or 24px with regular font weight (less than 700).
- Less than 14 point or 18.5px with bold font weight (at least 700).

There is regular text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

This can be seen on pages 3, 4, 6, 7 and 11, on desktop with the text paragraph under the main heading. The white text (#FFFFFF) has a 4:1 contrast ratio on the light portions of the blue background (##017EF6).

Note that the background is animated and the light portions do not always occur in the same place.



Page 4: desktop, Colour Contrast Analyser, text under "Search for your dream job" does not have sufficient contrast on light portions of background

This can also be seen on pages 3, 4, 7 and 11, on iOS with the text paragraph under the main heading. The white text (#FFFFFF) has a 4.3:1 contrast ratio on the light portions of the blue background (##2876F0).

The screenshot shows the Unilever careers website on an iOS device. A modal window from the 'Colour Contrast Analyser (CCA)' is open, overlaid on the page. The text 'Search for your dream job' is highlighted with a red box, indicating it fails the contrast test. The CCA interface shows 'Foreground colour' as white (#FFFFFF) and 'Background colour' as #2876F0. In the 'WCAG 2.1 results' section, the 4.3:1 contrast ratio is shown, with 'Fail (regular text)' checked and 'Pass (large text)' checked.

Page 4: iOS, Colour Contrast Analyser, text under “Search for your dream job” does not have sufficient contrast on light portions of background

This can also be seen on page 8 on iOS with the text in the “Digital Supply Chain” accordion where it overlaps the magenta circle in the background. The white text (#FFFFFF) has a 3.5:1 contrast ratio on the magenta background (#DC4FC7).

The screenshot shows the Unilever careers website on an iOS device. A modal window from the 'Colour Contrast Analyser (CCA)' is open, overlaid on the page. Two words in an accordion ("This" and "to") are highlighted with a red box, indicating they fail the contrast test. The CCA interface shows 'Foreground colour' as white (#FFFFFF) and 'Background colour' as #DC4FC7. In the 'WCAG 2.1 results' section, the 3.5:1 contrast ratio is shown, with both 'Fail (regular text)' and 'Fail (large text)' checked.

Page 8: iOS, Colour Contrast Analyser, portions of the words “This” and “to” in the “Digital Supply Chain” accordion do not have sufficient contrast

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.

3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-103 MEDIUM PRIORITY OPEN

Large text does not have sufficient contrast in video

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Plain Text

What happened

Informative text must pass colour contrast requirements. For large text, the required contrast ratio is 3:1. Large text is defined as the following.

- At least 18pt or 24px with regular font weight (less than 700).
- At least 14 point or 18.5px with bold font weight (at least 700).

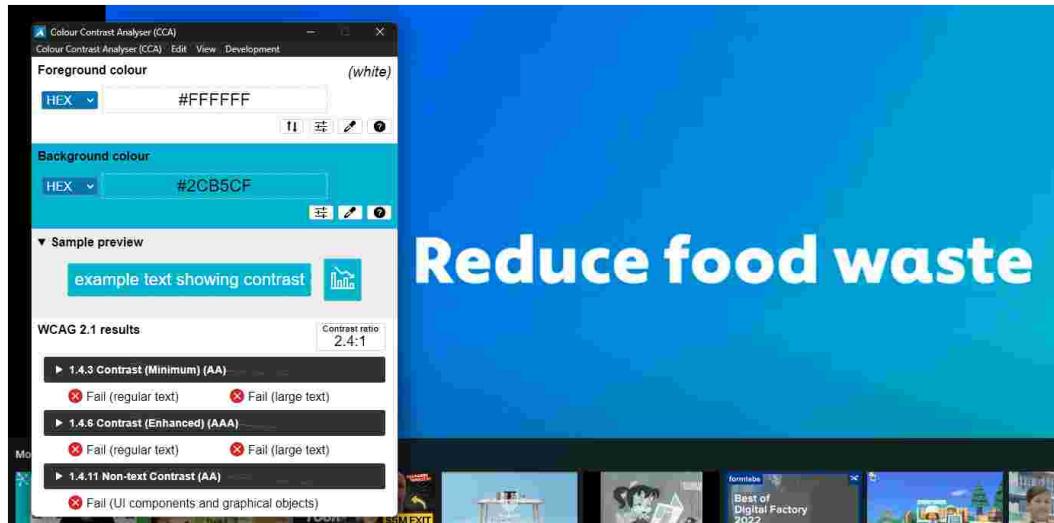
We are unable to get the precise text size. However, there appears to be large text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

Where it happened

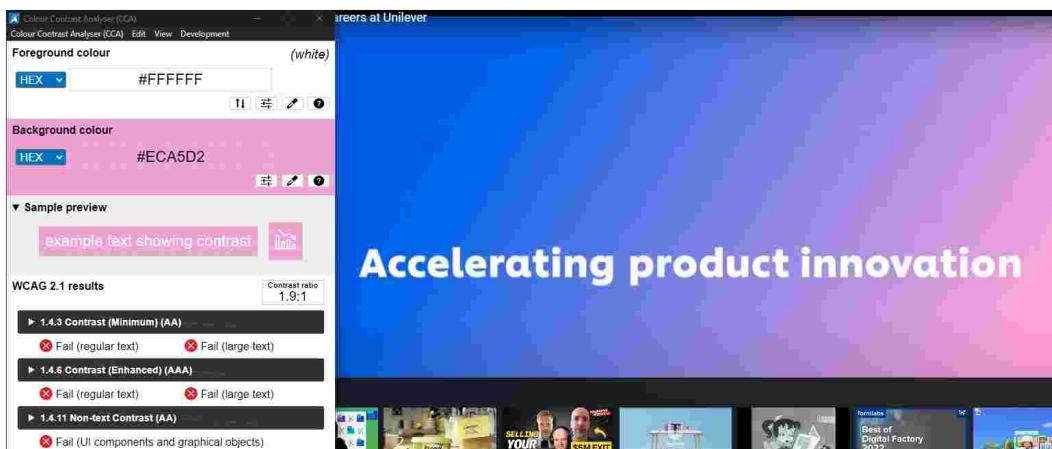
This can be seen on page 13 on desktop and iOS in the “Digital Careers at Unilever” video. The large white text (#FFFFFF) in the video appears on gradient and image backgrounds, and does not consistently have a sufficient contrast against the background.

For example:

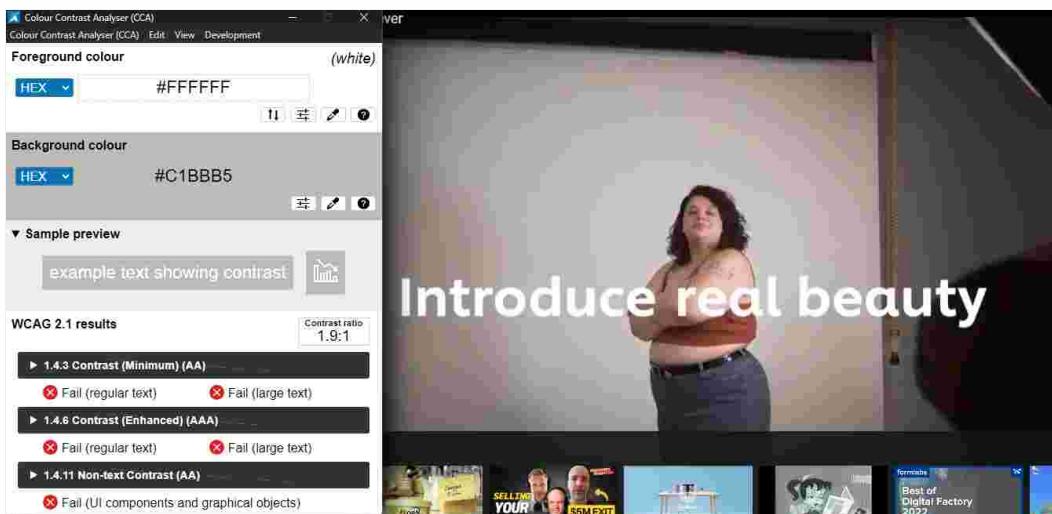
- The text “Reduce food waste” has a 2.4:1 contrast ratio with light blue portions of the background (#2CB5CF).
- The text “Accelerating product innovation” has a 1.9:1 contrast ratio with pink portions of the background (#ECA5D2).
- The text “Introduce real beauty” has a 1.9:1 contrast ratio with grey portions of the background (#C1BBB5).



Page 13: desktop, Colour Contrast Analyser, “Reduce food waste” does not have sufficient contrast



Page 13: desktop, Colour Contrast Analyser, “Accelerating product innovation” does not have sufficient contrast



Page 13: desktop, Colour Contrast Analyser, “Introduce real beauty” does not have sufficient contrast

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-106 MEDIUM PRIORITY OPEN

Large text does not have sufficient contrast on two-coloured background

Success Criteria 1.4.3: Contrast (Minimum) [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 11

Users Affected Vision

Component Plain Text

What happened

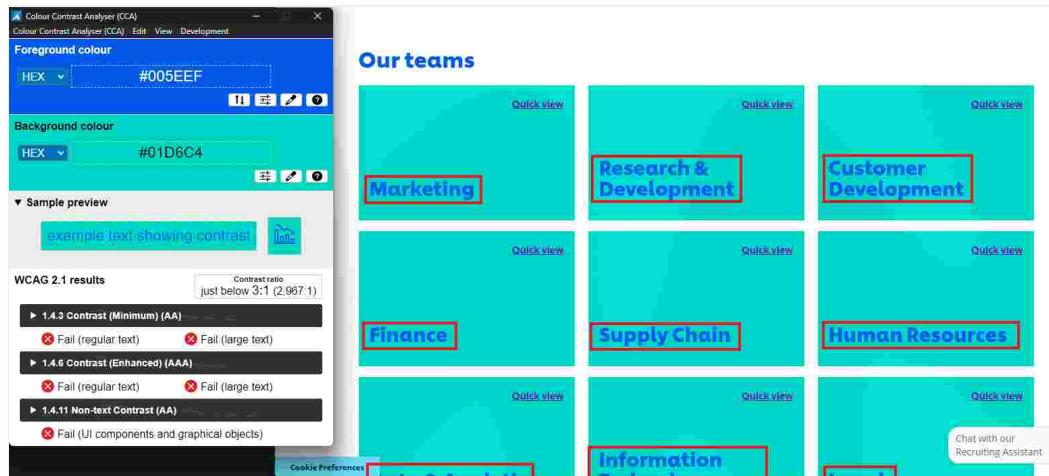
Informative text must pass colour contrast requirements. For large text, the required contrast ratio is 3:1. Large text is defined as the following.

- At least 18pt or 24px with regular font weight (less than 700).
- At least 14 point or 18.5px with bold font weight (at least 700).

There is large text that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the text that fails the colour contrast requirements.

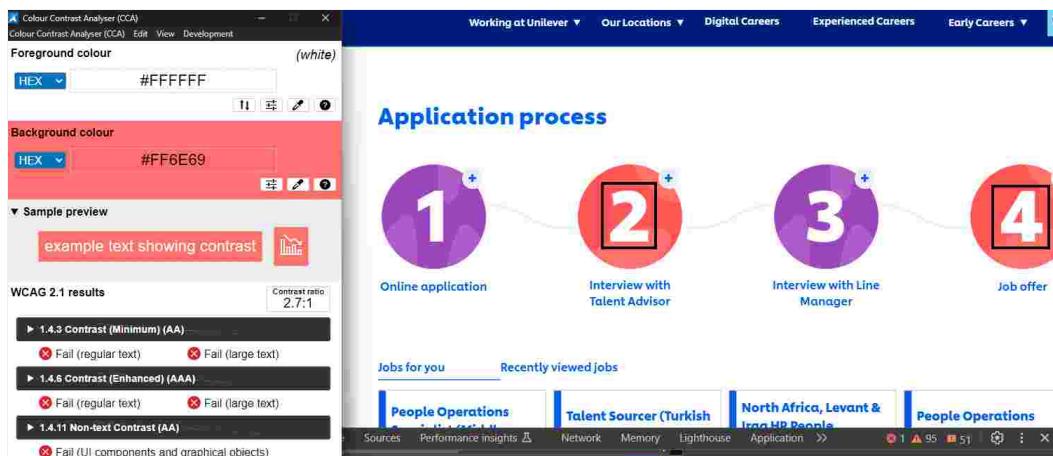
Where it happened

This can be seen on page 11 on desktop and iOS with the heading text of the cards under “Our team”, such as “Marketing”, “Research & Development”, and so on. The dark blue text (#005EEF) has just below a 3:1 contrast ratio (2.967:1) with the darker portions of the cyan background (#01D6C4).



Page 11: desktop, Colour Contrast Analyser, heading text of “Our team” cards does not have sufficient contrast on darker portions of cyan background

This can also be seen on page 11 on desktop and iOS with the numbers “2” and “4” in the steps under “Application process”. The white text (#FFFFFF) has a 2.7:1 contrast ratio on the lighter portions of the coral background (#FF6E69).



Page 11: desktop, Colour Contrast Analyser, “2” and “4” under “Application process” do not have sufficient contrast on lighter portions of coral background

Steps to reproduce

1. Get the colour values of the text and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

Make sure that both regular and bold text pass colour contrast requirements. Adjust the text and background colours accordingly to meet the contrast ratio requirements, and where relevant increase the font-size and font-weight to lower the contrast ratio requirements.

We recommend putting text on solid backgrounds wherever possible to make sure that the text has sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between foreground and background colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-8 MEDIUM PRIORITY OPEN

Zoom causes loss of meaningful content

Success Criteria 1.4.4: Resize text [2.0 AA]

Devices Desktop

Pages Affected 01, 04

Users Affected Vision

Component Link, Button

What happened

Users must be able to access content and functionality at 200% zoom. This is particularly important for low vision users as it can allow them to make content more readable. No loss of content should occur at 200% zoom. There is meaningful or unique content that is lost at 200% zoom.

Where it happened

This can be seen on page 1 with the “Early Careers” expandable button and the “Register for job alerts” link. At 200% zoom, this text is partly overlapped by the “Candidate Feedback” button.

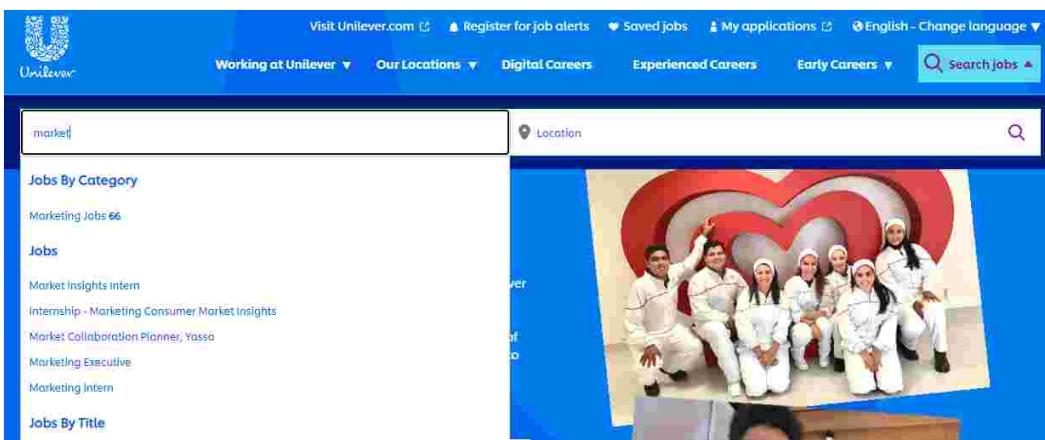


Page 1: desktop, navigation links at default size

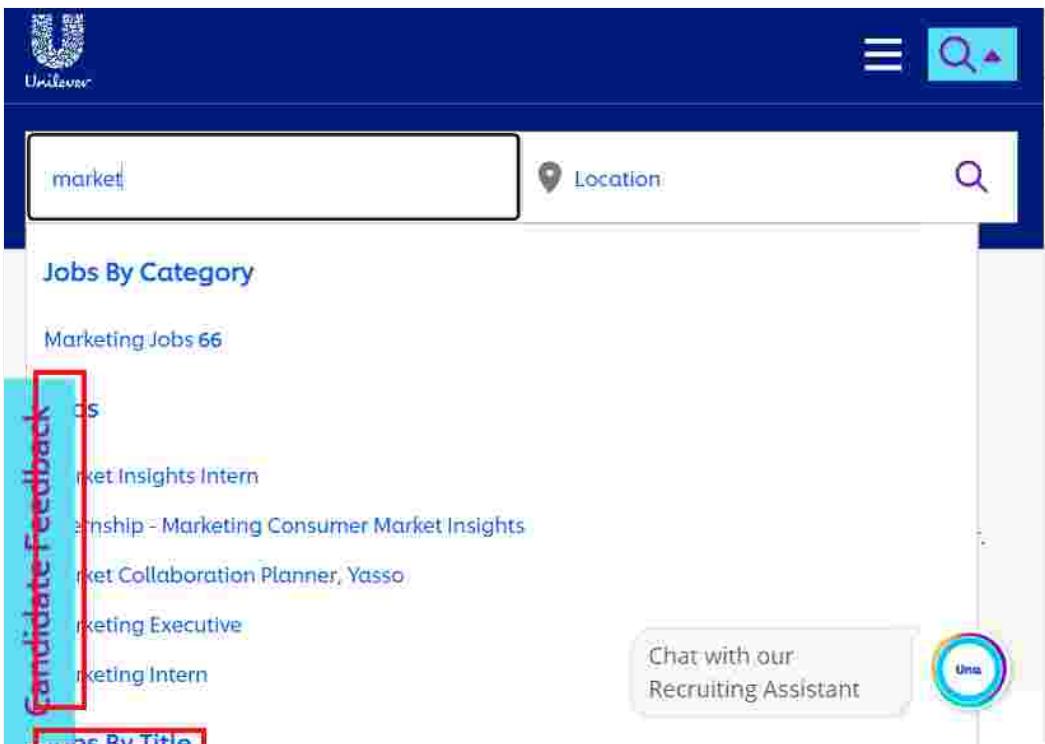


Page 1: desktop, 200% zoom, “Candidate Feedback” button partially overlaps “Early Careers” and “Register for job alerts” text

This can also be seen on page 1 with the job category search panel. At 200% zoom, heading and link text in the “Jobs” section are overlapped by the “Candidate Feedback” button and the “Jobs By Title” heading is cut off.



Page 1: desktop, job category panel at default size

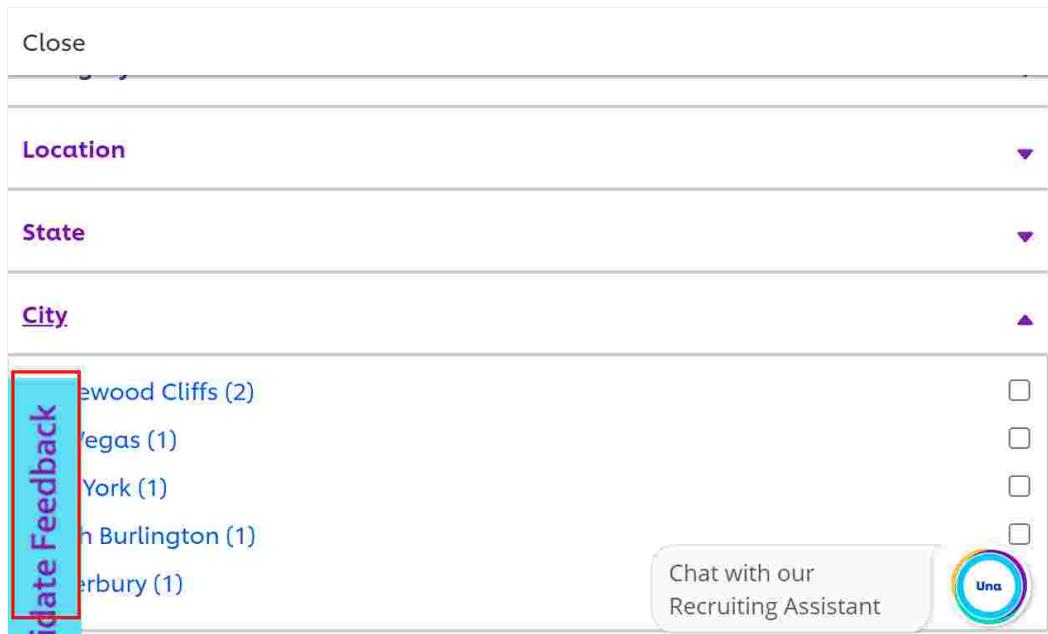


Page 1: desktop, 200% zoom, text is overlapped or cut off in job category panel

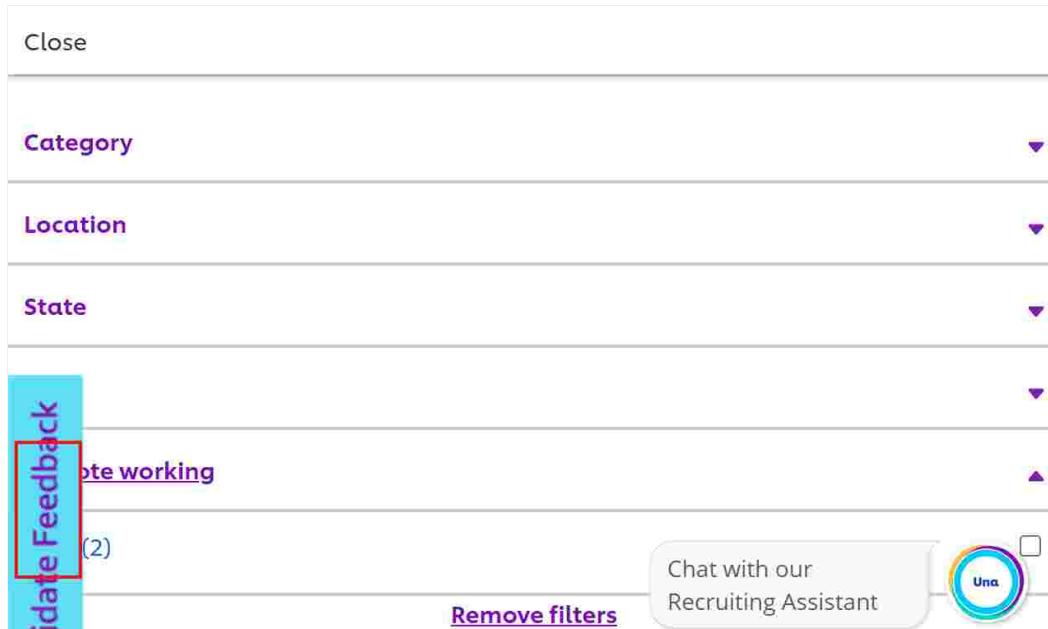
This can also be seen on page 4 with the checkbox labels such as "Englewood Cliffs" in the filters modal when "City" is expanded, and with the "Remote working" button and "True" checkbox label when "City" is collapsed. These elements are partially overlapped by the "Candidate Feedback" triggering button.



Page 4: desktop, search filters at default size



Page 4: desktop, 200% zoom, "Candidate Feedback" button partially overlaps checkbox labels under "City"



Page 4: desktop, 200% zoom, "Candidate Feedback" button partially overlaps "Remote working" button and "True" checkbox label

Steps to reproduce

1. Observe the content at default zoom.
2. Zoom to 200%.
3. Observe that the content is lost
4. Observe that there is not an alternative way to access the same content.

Recommendation

Make sure that content is available at 200% zoom.

Decorative content can be lost. For example, a card pattern comprised of an image and heading. Both elements are wrapped in one anchor tag. The element is understandable through the heading alone. The link can be accessed from just the heading. As a result, the image can be removed when zoom is increased.

Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. In this example, users can access the same functionality.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw`. For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#). As well, use modern, flexible layouts techniques. For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#).

[Return to priority table](#)

ADR_20590-30 MEDIUM PRIORITY OPEN

Footer - Zoom causes loss of meaningful content

Success Criteria 1.4.4: Resize text [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 02

Users Affected Vision

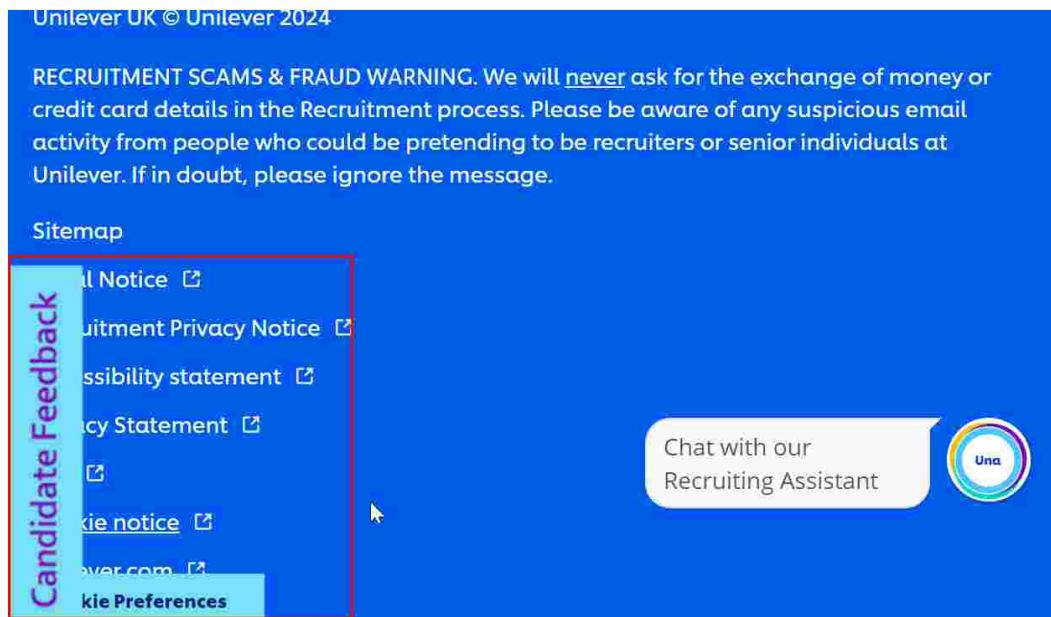
Component Link, Button

What happened

Users must be able to access content and functionality at 200% zoom. This is particularly important for low-vision users as it can allow them to make content more readable. No loss of content should occur at 200% zoom. There is meaningful or unique content that is lost at 200% zoom.

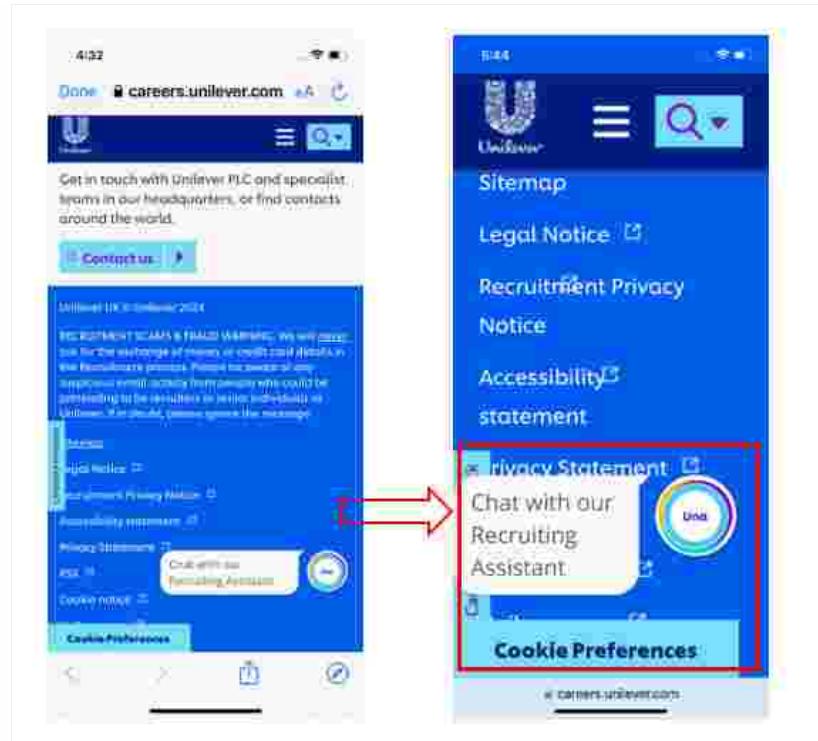
Where it happened

This can be seen on page 2 on desktop, with “Cookie Preferences” button and links such as “Legal Notice”, “Recruitment Privacy Notice”, “Accessibility statement” and so on. At 200% zoom, this button and links are partly overlapped by the “Candidate Feedback” button present in the footer.



Page 2: desktop, the footer section after dynamic text resize. Content is overlapped.

This can be seen on page 2 on iOS, with the “Candidate Feedback” button and links such as “Privacy Statement”, “RSS”, “Cookie notice” and “Unilever.com”. At 200% zoom, this button and links are partly overlapped by the “Una: Chat with our Recruiting Assistant” button present in the footer.



Page 2: iOS, the footer section before and after dynamic text resize. Content is overlapped.

Steps to reproduce

1. Observe the content at default zoom.
2. Zoom to 200%.
3. Observe that the content is lost
4. Observe that there is not an alternative way to access the same content.

Recommendation

Make sure that content is available at 200% zoom.

Decorative content can be lost. For example, a card pattern comprised of an image and heading. Both elements are wrapped in one anchor tag. The element is understandable through the heading alone. The link can be accessed from just the heading. As a result, the image can be removed when zoom is increased.

Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. In this example, users can access the same functionality.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw`. For more information, see [MDN - Relative Units, CSS Tricks - font-size](#). As well, use modern, flexible layouts techniques. For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#).

[Return to priority table](#)

ADR_20590-71 MEDIUM PRIORITY OPEN

Zoom causes loss of functionality

Success Criteria 1.4.4: Resize text [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 08

Users Affected Vision

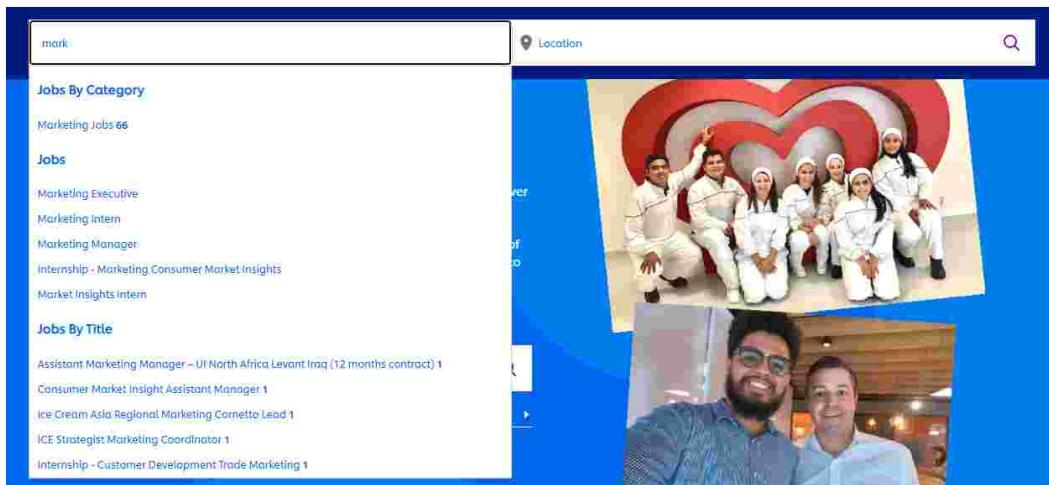
Component Link, Dropdown

What happened

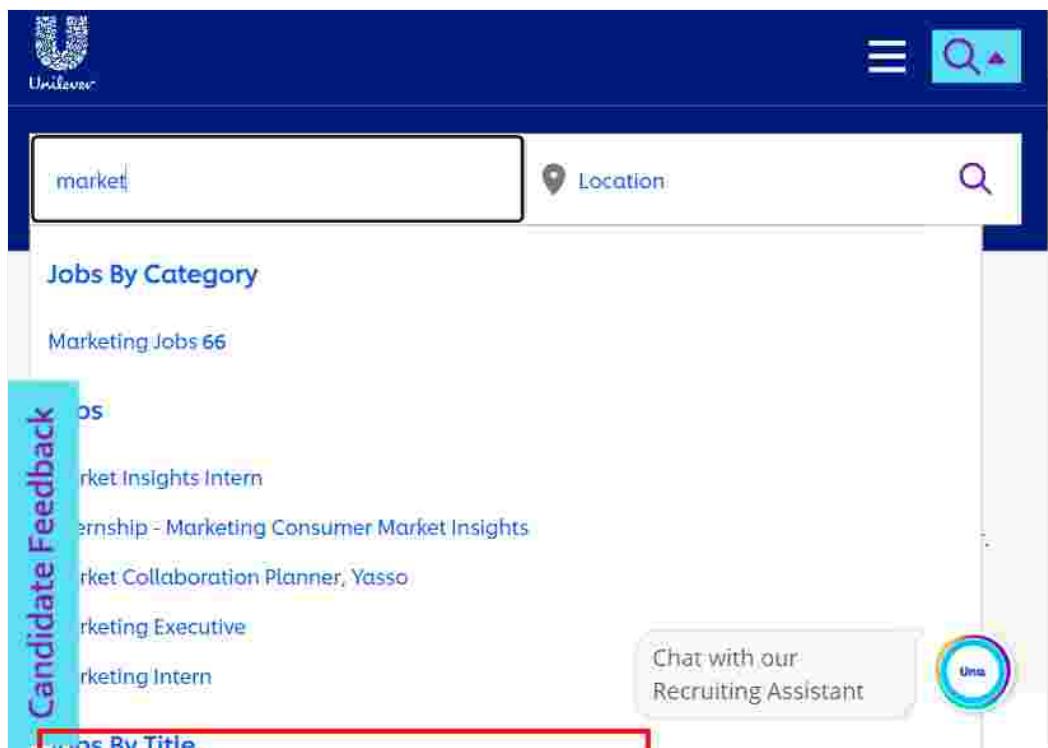
Users must be able to access content and functionality at 200% zoom. This is particularly important for low vision users as it can allow them to make content more readable. No loss of functionality should occur at 200% zoom. There is meaningful or unique functionality that is lost at 200% zoom.

Where it happened

This can be seen on page 1 with the job category search panel. At 200% zoom, links in the panel are off screen and cannot be scrolled onto the screen.



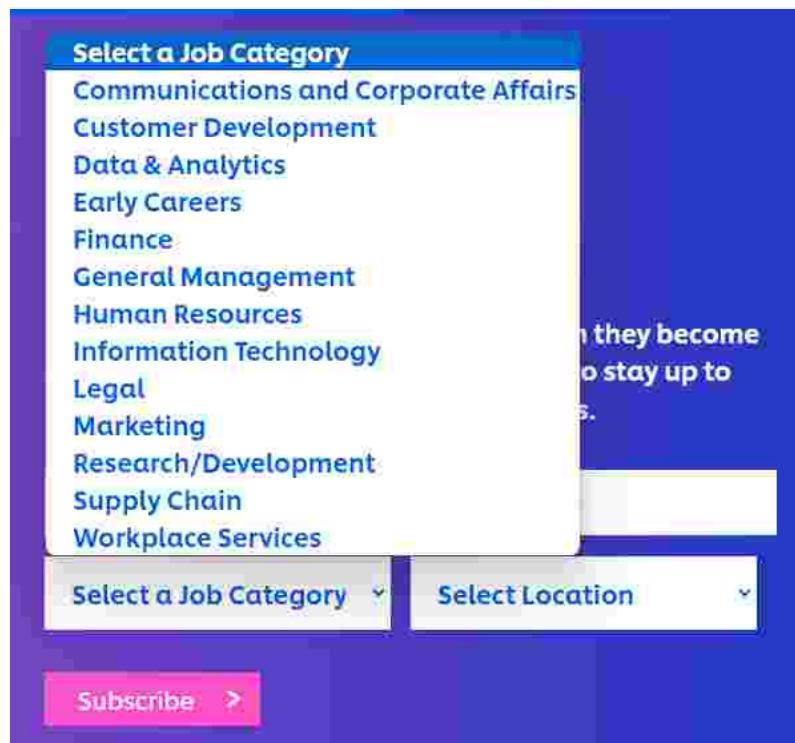
Page 1: desktop, job category panel at default size



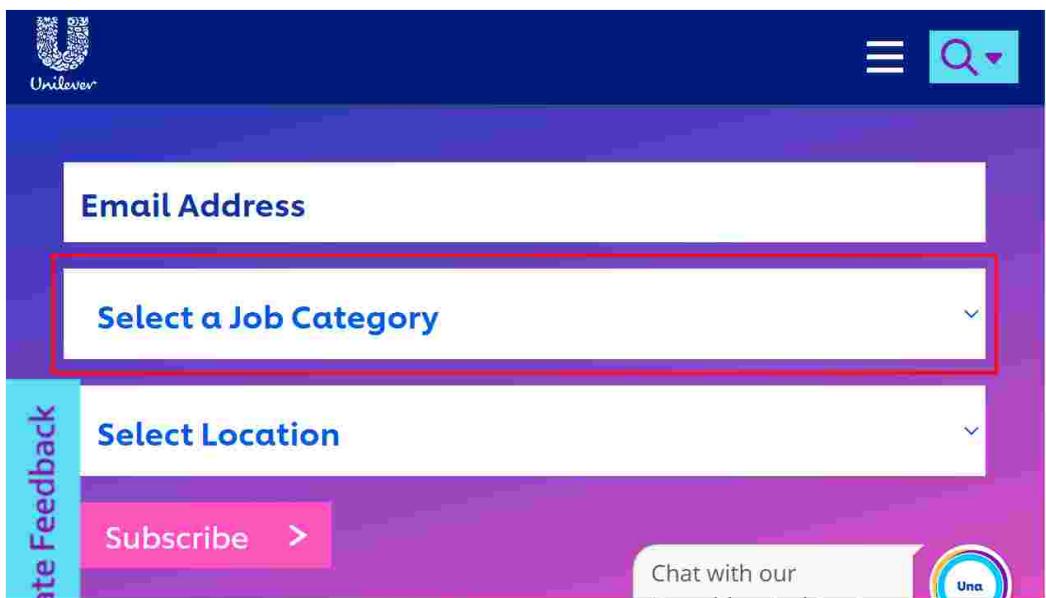
Page 1: desktop, 200% zoom, links are off screen

This can be seen on page 8 on desktop with the “Select a job category” dropdown. At 200% zoom, this element is not operable with the mouse.

Note that we have not been able to determine the cause of this issue. The element remains operable with the keyboard.



Page 8: desktop, “Select a job category” dropdown at default size

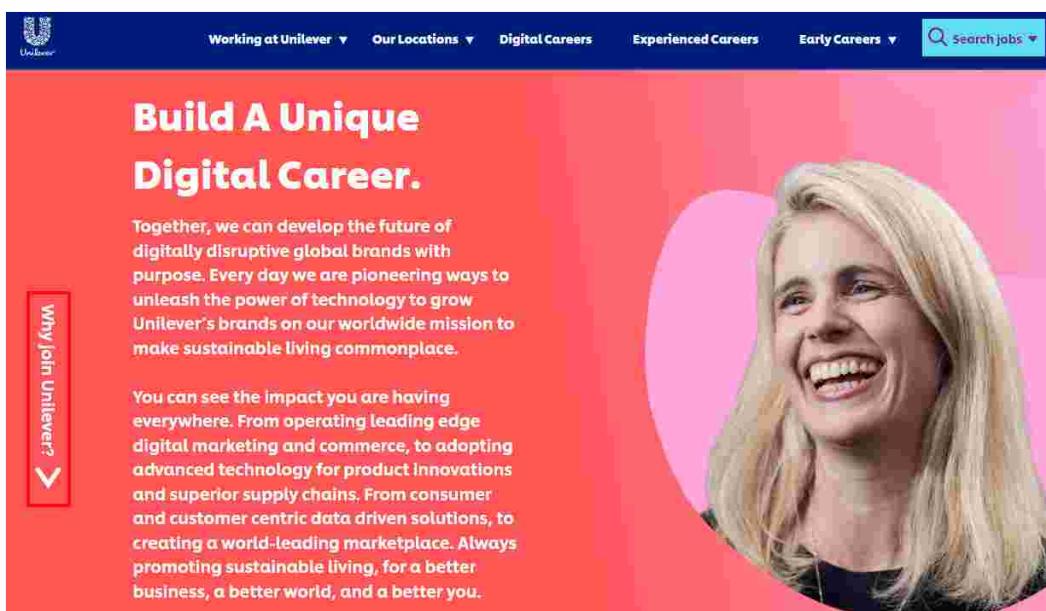


Page 8: desktop, 200% zoom, "Select a job category" dropdown

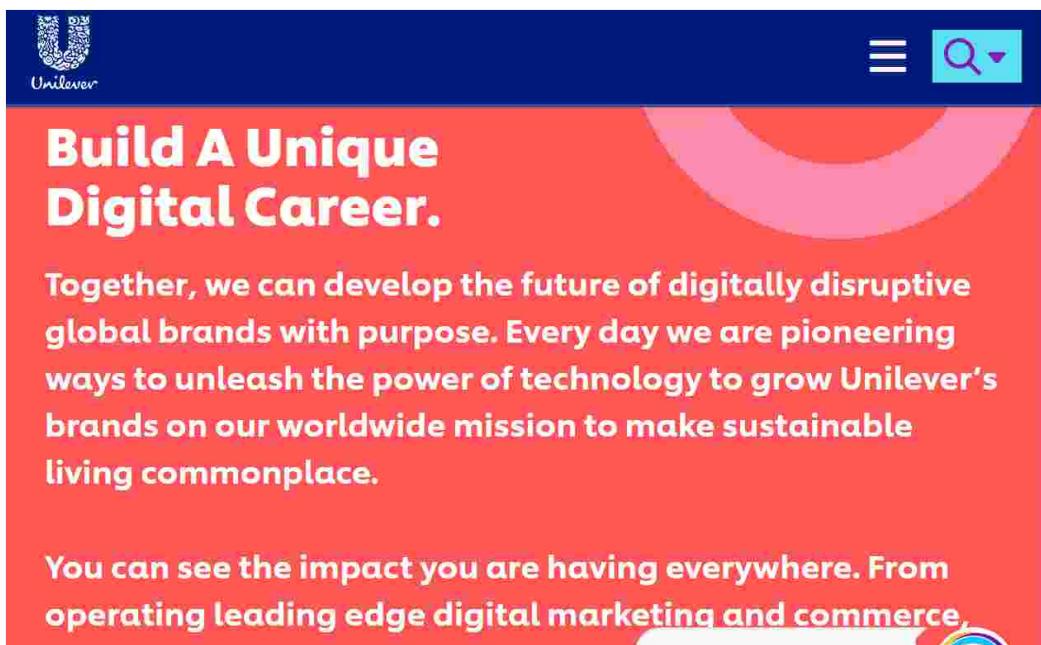
```
<label for="form-field-87203efdf-category">  
    "Job Category"  
    <span class="ico-required-indicator" aria-hidden="true">*</span>  
</label>  
<select class="keyword-category required-keyword-0" data-keyword-list="Form-Field-87203efdf-keywords" data-val="true" data-val-required="requiredKeywords: This field is required." id="form-field-87203efdf-category" name="Category" aria-invalid="false">  
    <option value="Select a Job Category">Select a Job Category</option>  
    <option value="7580730">Communications and Corporate Affairs</option>  
    <option value="8118032">Customer Development</option>  
    <option value="8225216">Data & Analytics</option>  
</select>
```

Page 8: DOM, "Select a job category" dropdown

This can also be seen on page 8 on desktop with the same-page section links such as "Why join Unilever?". These links provide a mechanism to skip sections of the content. At 200% zoom, this functionality is lost.



Page 8: desktop, "Why join Unilever?" link at default size



Page 8: desktop, 200% zoom, "Why join Unilever?" link is lost

Steps to reproduce

1. Observe the functionality at default zoom.
2. Zoom to 200%.
3. Observe that the functionality is lost
4. Observe that there is no alternative way to access the same functionality.

Recommendation

Make sure that functionality is available at 200% zoom.

Decorative content can be lost. For example, a card pattern comprised of an image and heading. Both elements are wrapped in one anchor tag. The element is understandable through the heading alone. The link can be accessed from just the heading. As a result, the image can be removed when zoom is increased.

Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. In this example, users can access the same functionality.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw`. For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#). As well, use modern, flexible layouts techniques. For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#).

[Return to priority table](#)

ADR_20590-63 MEDIUM PRIORITY OPEN

Image contains embedded text

Success Criteria 1.4.5: Images of Text [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 02, 11

Users Affected Vision

Component Image

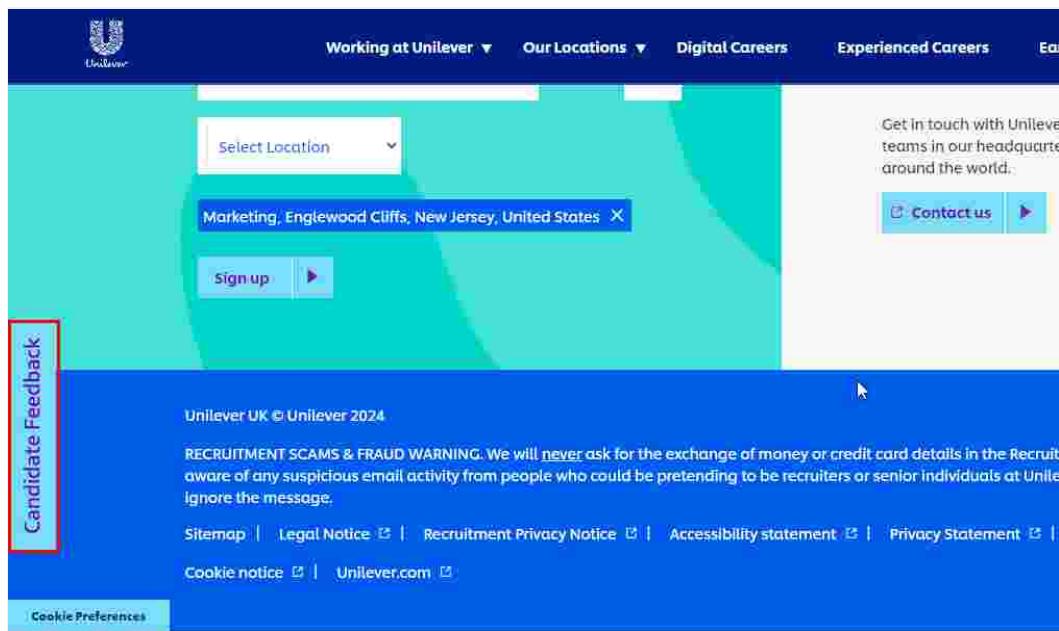
What happened

Text should not be embedded in images. This can make the content inaccessible.

For example, screen readers rely on the text being included in text alternatives, which are commonly not effective for communicating multiple sentences or structured text. As well, the text appears pixelated when accessed with screen magnification software. There are images of text.

Where it happened

This can be seen on page 2 on desktop and iOS, with the “Candidate Feedback” image. This image displays textual content serving as the label of a button.



Page 2: desktop, “Candidate Feedback” Image contains embedded text

```
<script type="text/javascript">...</script>
<script src="https://widget.altru labs.com/main.js" data-altru-widget-id="8831"></script>
<script id="tmp-magic-bullet" data-ally="true" src="https://services.tmpwebeng.com/magicbullet/"></script>
<script type="text/javascript" id="survale.init" src="https://static.survele.com/ext/surveyle.js"></script>
<script type="text/javascript" id="com_survale.init(1695851328E3);></script>
<div>...</div>
<div>
  <div id="survaleSideImageAnchorImgWrap" class="survale-may-animate" style="position: fixed; right: 0; cursor: pointer; background-color: transparent; width: 50px; height: 50px; border-radius: 50%; z-index: 300; font-size: 12px;">
    <button id="survaleSideImageAnchor" style="border-radius: 0px; background: none transparent; min-height: 20px; border: none; width: 50px; class="left">
      
    </button>
  </div>
</div>
<script id="ally-fixes" src="https://services.tmpwebeng.com/magicbullet/a11y/init.js"></script>
<div id="onetrust-consent-sdk">...</div>
<div class="highslide-container" style="padding: 0px; border: none; margin: 0px; position: absolute; left: 0px; top: 0px; width: 100%; height: 100%; direction: ltr;">...</div>
```

Page 2: DOM, "Candidate Feedback" Image is wrapped within button

This can be seen on page 11 on desktop and iOS with the step numbers under "Application process". The numbers are embedded in images.



Page 11: desktop, numbers under "Application process"

The screenshot shows the HTML code for the application process steps. The '1' icon is highlighted with a red box in the developer tools. The CSS for the '1' icon is shown in the right panel, indicating a background image URL and a rendered size of 41x41px. The current source is https://b2cd.talentbrew.com/company/3415/v1/img/application-1.png.

Page 11: DOM, number "1" is an image of text

Steps to reproduce

1. Observe the text.
2. Inspect the text and review the DOM.
3. Note that the text is embedded in an image.

Recommendation

Avoid images with embedded text. Style plain text with CSS. Where desired, position plain text over background images. For example:

```
<div class="backgroundImageCSS">
  Clearance - 50% off
</div>
```

For more information, see [WAI Tutorials - Images of Text](#), [MDN - Styling Text](#), and [MDN - Annotating images](#).

[Return to priority table](#)

ADR_20590-6 MEDIUM PRIORITY OPEN

Reflow causes loss of functionality

Success Criteria 1.4.10: Reflow [2.1 AA]

Devices Desktop

Pages Affected 01, 08

Users Affected Vision

Component Link

What happened

Users must be able to access content and functionality when the page is reflowed.

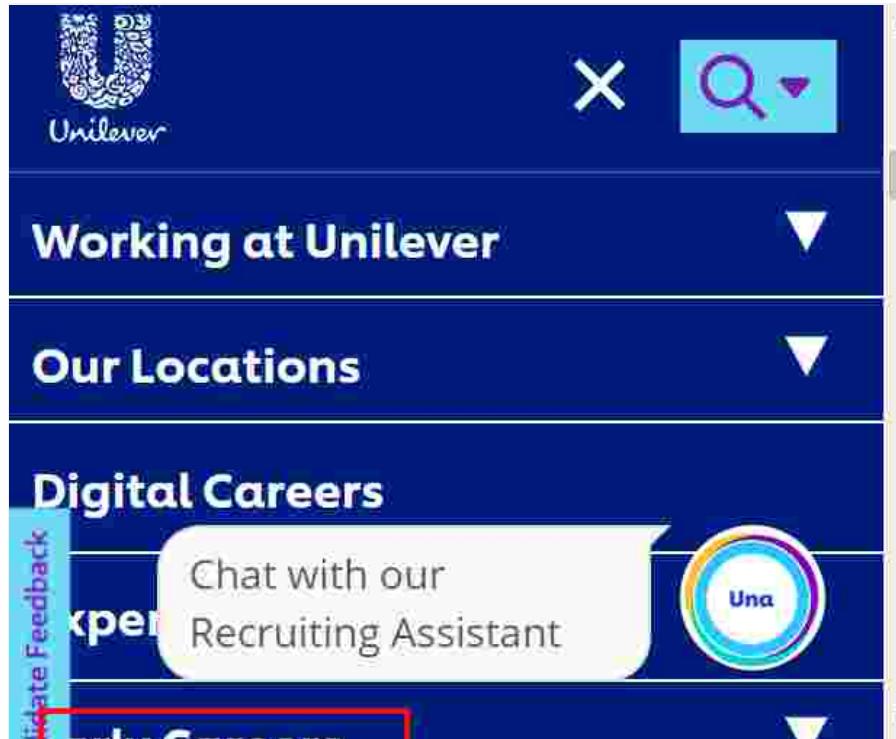
This means when the screen size is 320 x 256px and 100% zoom, or 1280 x 1024px and 400% zoom. This is particularly important for low vision users as it can allow them to make content more readable. There is meaningful or unique functionality that is lost when reflowed.

Where it happened

This can be seen on page 1 with links in the hamburger menu. At 320 x 256px, the links in the “Early Careers” submenu and the “Register for job alerts” link are off screen and cannot be scrolled onto the screen.

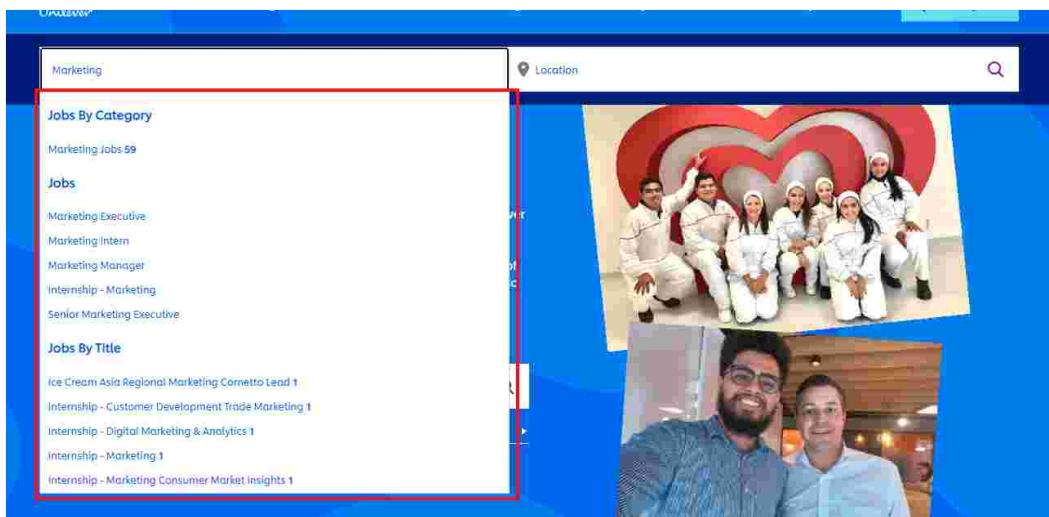


Page 1: desktop, navigation links at default size

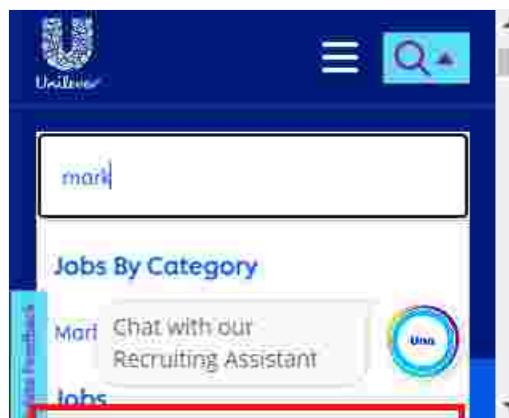


Page 1: desktop, 320 x 256px, links under “Early Careers” are off screen

This can also be seen on page 1 with links in the job category search panel. At 320 x 256px, the panel contains links that are off screen and cannot be scrolled onto the screen.

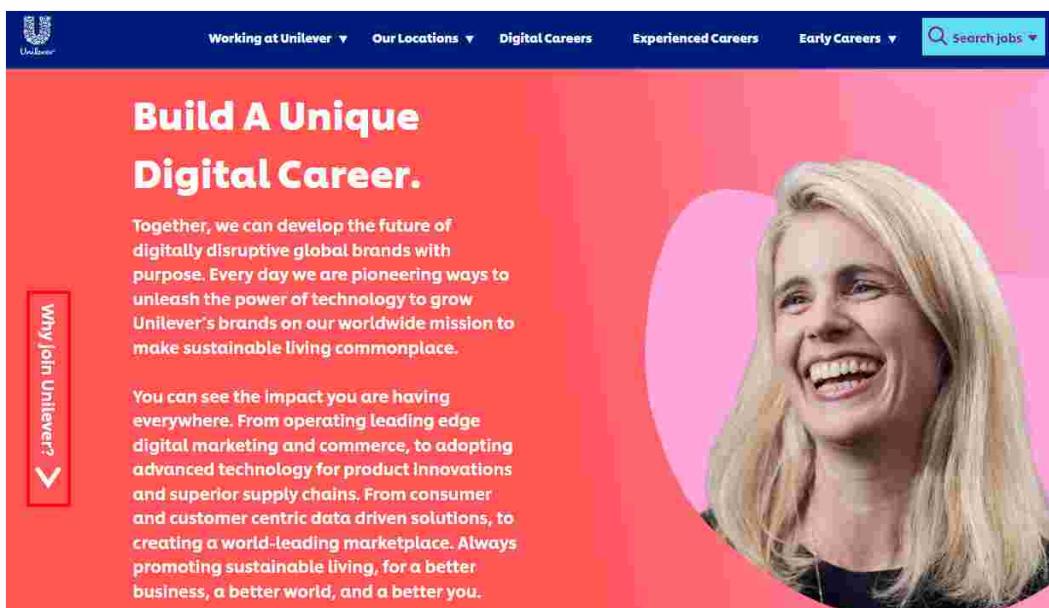


Page 1: desktop, job category panel at default size

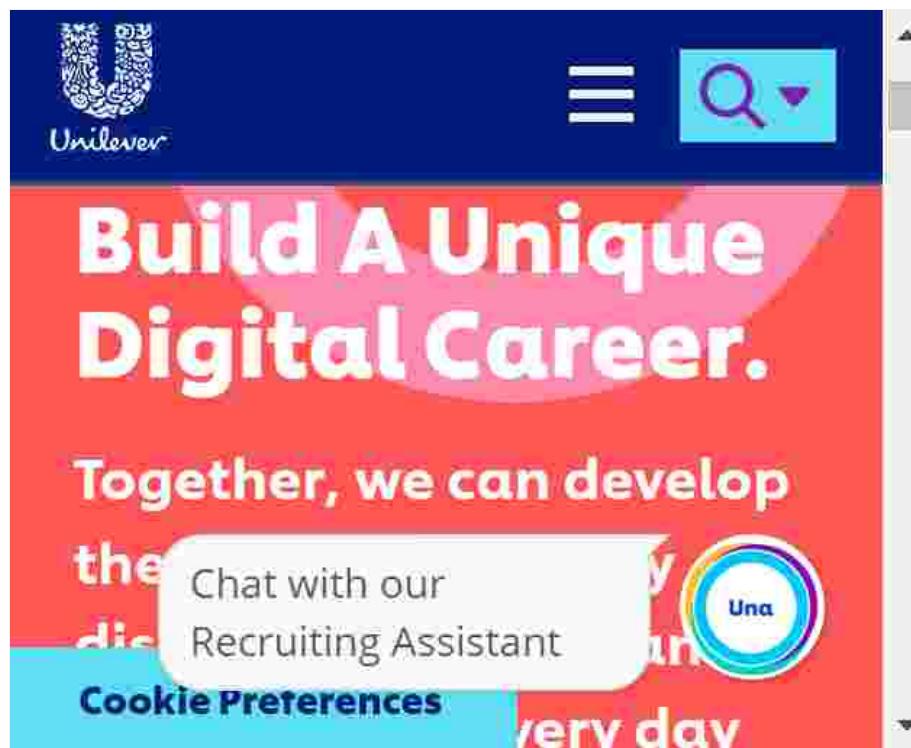


Page 1: desktop, 320 x 256px, links in job category panel are off screen

This can be seen on page 8 with the same-page section links such as "Why join Unilever?". These links provide a mechanism to skip sections of content. At 320 x 256px, these links are lost.



Page 8: desktop, "Why join Unilever?" link at default size



Page 8: desktop, 320 x 256px, "Why join Unilever?" link is lost

Steps to reproduce

1. Observe the functionality at default screen size and zoom.
2. Set the screen size to 320 x 256px and 100% zoom.
3. Alternatively, set the screen size to 1280 x 1024px and 400% zoom.
4. Observe that the functionality is lost
5. Observe that there is no alternative way to access the same functionality.

Recommendation

Make sure that functionality is available when the page is reflowed. When the viewport is adjusted, make sure that the functionality is still available on the page. As well, make sure that users are not required to scroll in multiple directions (such as up and down, as well as left and right) to access the functionality.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw` and modern, flexible layouts techniques to achieve this.

Note that decorative content can be lost, such as images that are not functional or informative. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. If you need to tweak the design to allow users to access the functionality, that is acceptable.

For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#), [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#). As well, testing can be done with [Web Developer Toolbar](#) or browser dev tools, such as [Chrome - Device Mode](#).

[Return to priority table](#)

ADR_20590-7 MEDIUM PRIORITY OPEN

Reflow causes loss of content

Success Criteria 1.4.10: Reflow [2.1 AA]

Devices Desktop

Pages Affected 01, 04, 08

Users Affected Vision

Component Button, Dropdown

What happened

Users must be able to access content and functionality when the page is reflowed.

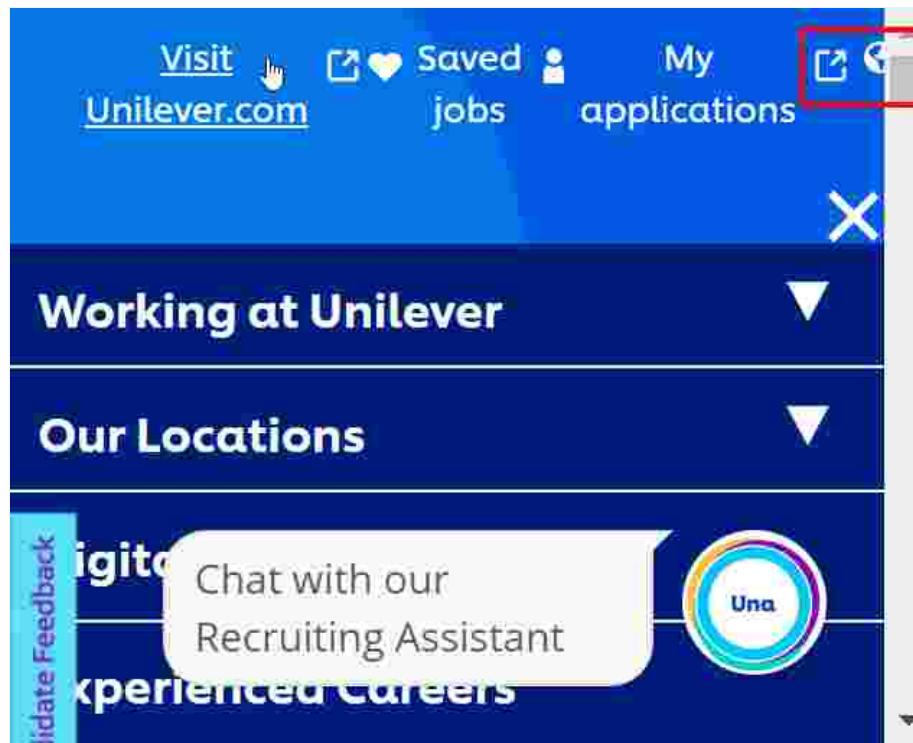
This means when the screen size is 320 x 256px and 100% zoom, or 1280 x 1024px and 400% zoom. This is particularly important for low vision users as it can allow them to make content more readable. There is meaningful or unique content that is lost when reflowed.

Where it happened

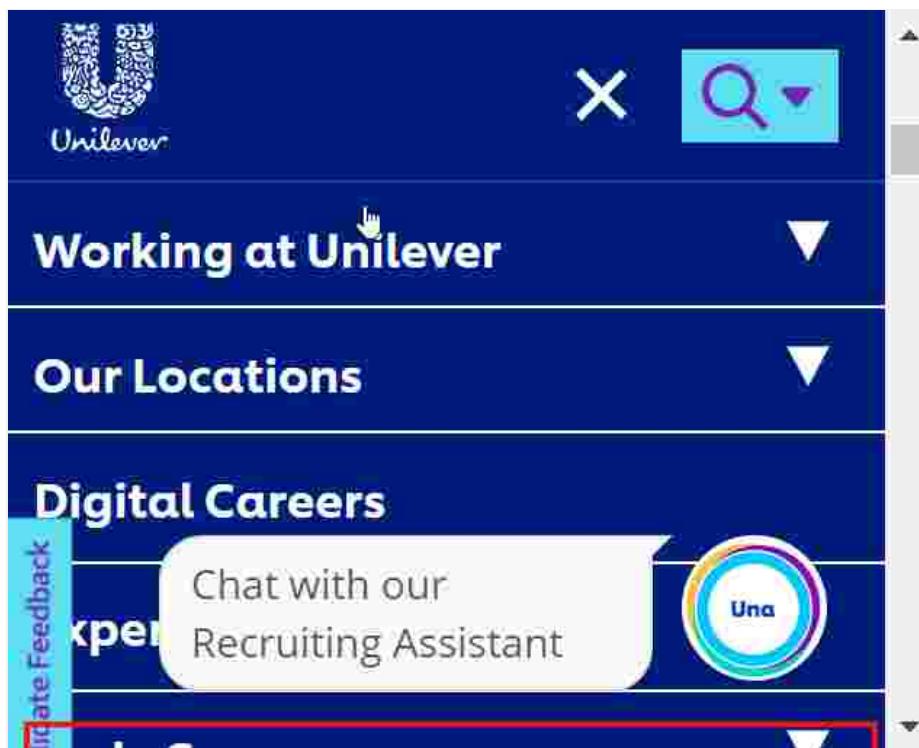
This can be seen on page 1 with the “Change language” button and the “Early Careers” expandable button. At 320 x 256px, the text of the “Change language” button is off screen and the text of the “Early Careers” button is cut off.



Page 1: desktop, page at default size



Page 1: desktop, 320 x 256px, “Change language” button text is off screen

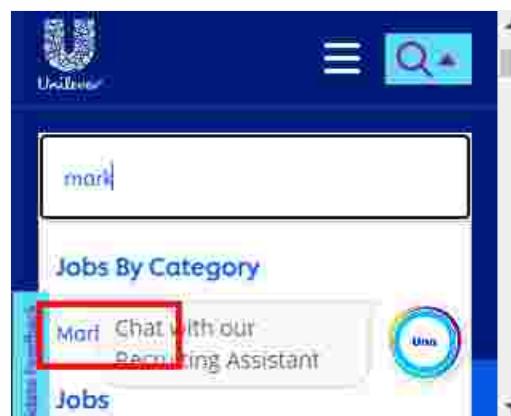


Page 1: desktop, 320 x 256px, "Early Careers" button text is cut off

This can also be seen on page 1 with text in the job category search panel. At 320 x 256px, the text of the "Marketing jobs 66" link is overlapped by the chat widget.



Page 1: desktop, job category search panel at default size



Page 1: desktop, 320 x 256px, "Marketing jobs 66" link is overlapped by chat widget

This can also be seen on page 4 with the “True (2)” checkbox label and the “Remove filters” button in the filters modal. At 320 x 256px, this text is partially overlapped by the “Candidate Feedback” and “Cookie Preferences” triggering buttons, and cannot be scrolled entirely onto the screen.

Filter your search

Category Location State City Remote working Remove filters

True (5)

29 Search results

US Digital Marketing Manager, Ben & Jerry's South Burlington, Vermont	Market Collaboration Planner, Yasso Englewood Cliffs, New Jersey	Associate Brand Manager - Liquid I.V. International Multiple locations
--	---	---

Page 4: desktop, page at default size

Close

State

City

Remote working

true (2) Chat with our Recruiting Assistant

Candidate Feedback

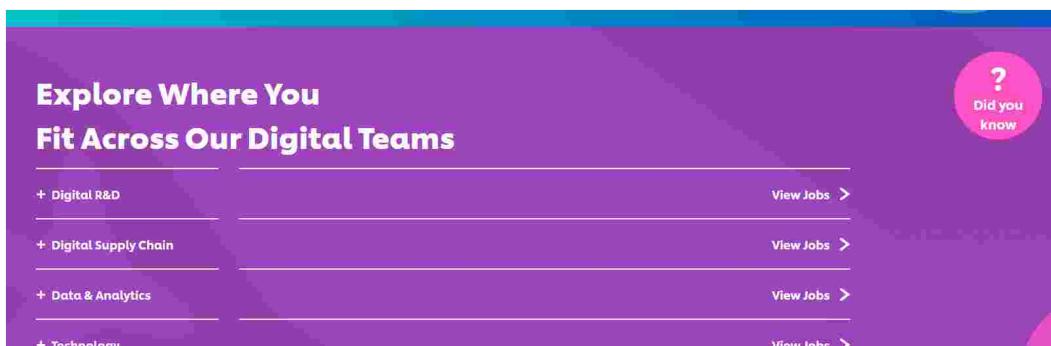
Cookie Preferences

Remove filters

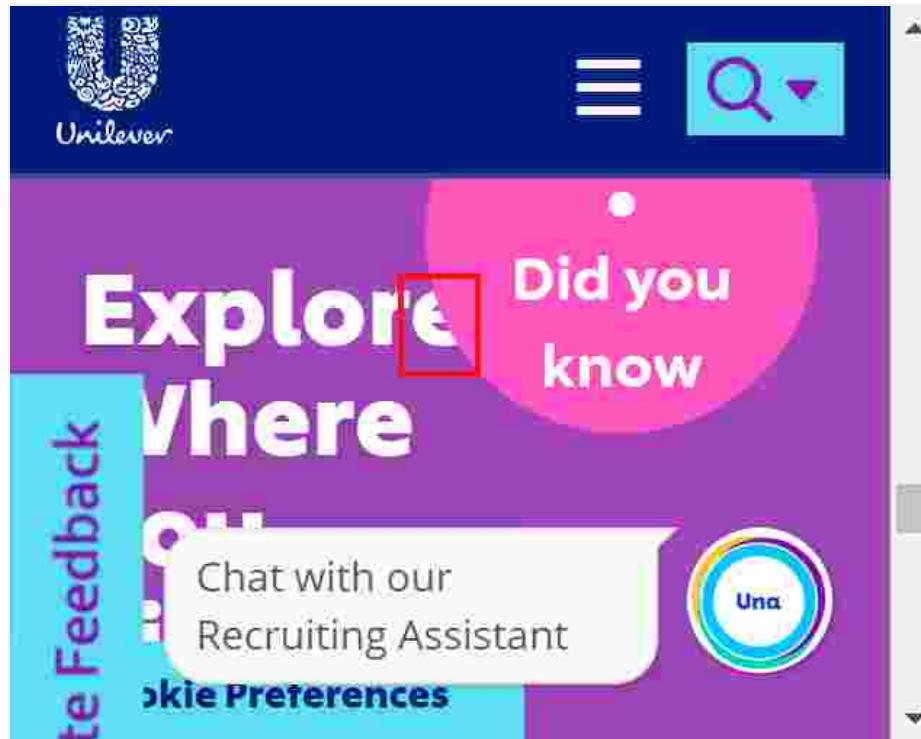
Page 4: desktop, 320 x 256px, “True (2)” checkbox label and “Remove filters” button are partially overlapped

This can also be seen on page 8 with the following elements:

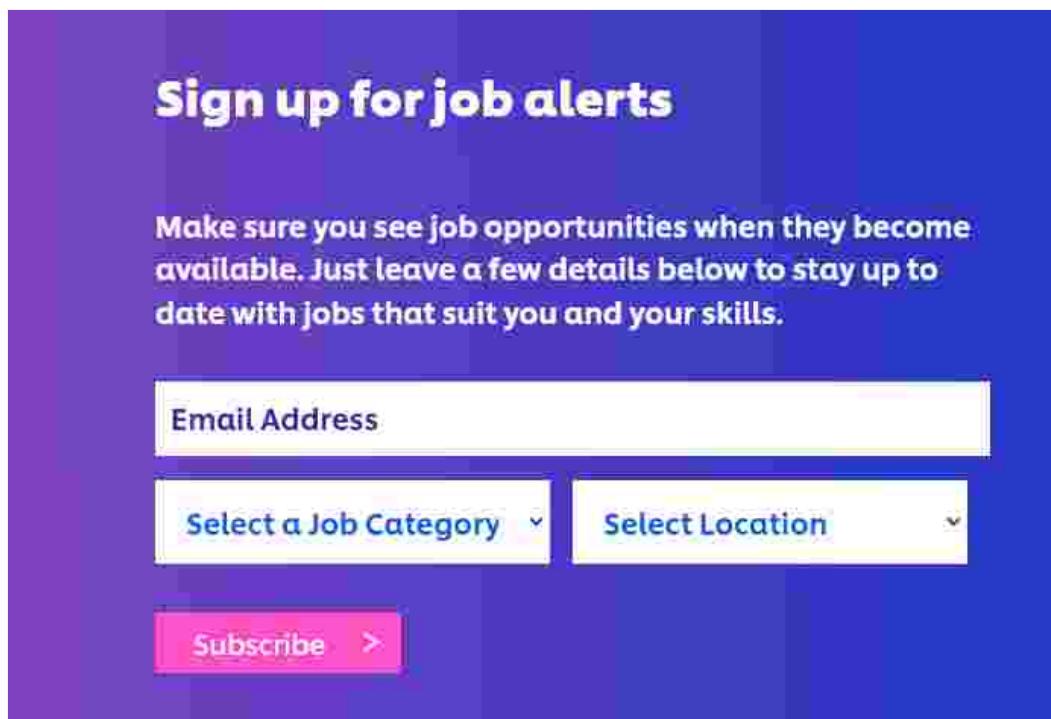
- The “Explore Where You Fit Across Our Digital Teams” heading. At 320 x 256px, the adjacent “Did you know?” button partially overlaps this heading.
- The “Select a Job Category” dropdown in the job alerts section. At 320 x 256px, the text “Select a Job Category” is cut off. Note that, because the element does not have a permanent label, this text functions to identify the element.



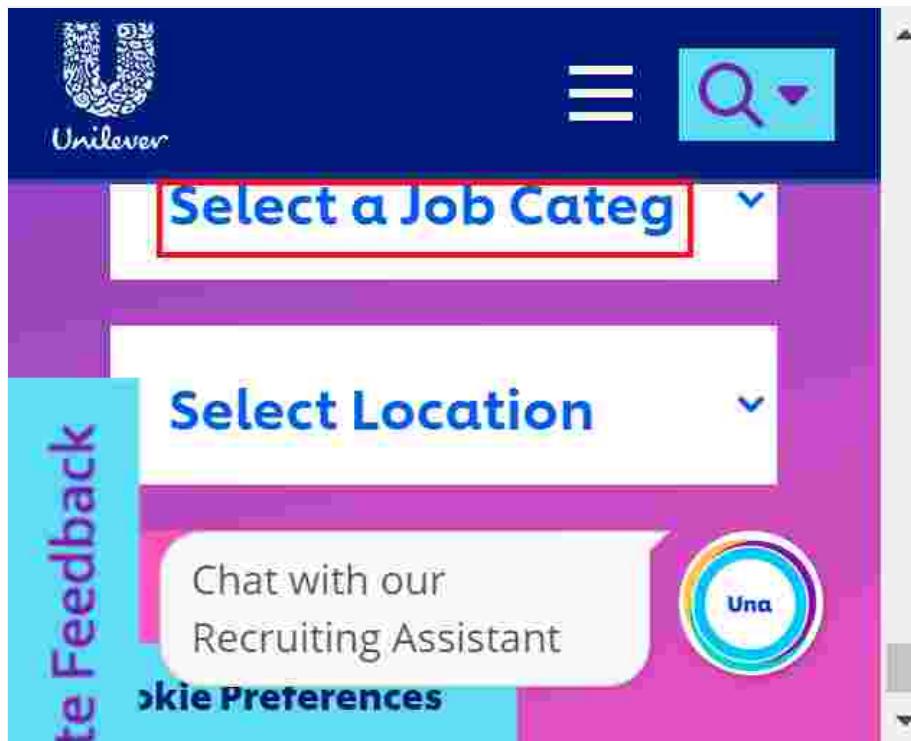
Page 8: desktop, "Explore Where You Fit Across Our Digital Teams" at default size



Page 8: desktop, 320 x 256px, "Did you know?" button overlaps "Explore Where You Fit Across our Digital Teams" heading



Page 8: desktop, "Select a Job Category" dropdown at default size



Page 8: desktop, 320 x 256px, "Select a Job Category" is cut off in dropdown

Steps to reproduce

1. Observe the content at default screen size and zoom.
2. Set the screen size to 320 x 256px and 100% zoom.
3. Alternatively, set the screen size to 1280 x 1024px and 400% zoom.
4. Observe that the meaningful content is lost
5. Observe that there is not an alternative way to access the same content.

Recommendation

Make sure that content is available when the page is reflowed. When the viewport is adjusted, make sure that the content is still available on the page. As well, make sure that users are not required to scroll in multiple directions (such as up and down, as well as left and right) to access the content.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw` and modern, flexible layouts techniques to achieve this.

Note that decorative content can be lost, such as images that are not functional or informative. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. If you need to tweak the design to allow users to access the meaningful content, that is acceptable.

For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#), [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#). As well, testing can be done with [Web Developer Toolbar](#) or browser dev tools, such as [Chrome - Device Mode](#).

[Return to priority table](#)

ADR_20590-20 MEDIUM PRIORITY OPEN

Reflow requires users to scroll in multiple directions (no exceptions)

Success Criteria 1.4.10: Reflow [2.1 AA]

Devices Desktop

Pages Affected 09

Users Affected Vision

Component Modal Dialog

What happened

Users must be able to access content and functionality when the page is reflowed. As well, when the page is reflowed, users should not need to navigate up, down, left, and right. The content should be stacked such users only need to scroll in one direction, such as up and down.

This means when the screen size is 320 x 256px and 100% zoom, or 1280 x 1024px and 400% zoom. This is particularly important for low vision users as it can allow them to make content more readable.

You do not need to reflow content when it would change the meaning of the content. For example, data tables, maps and diagrams, video, games, presentations, and toolbars. However, there are pages without such content that require scrolling in multiple directions when reflowed.

Where it happened

This can be seen on page 9 on desktop, with the “Be a part of something bigger” modal dialog. When the viewport is adjusted, users are required to scroll up, down, left and right, to access button, as well as the plain text contained in them.



Page 9: desktop, adjusted viewport “Be a part of something bigger” modal

Steps to reproduce

1. Observe the functionality at default screen size and zoom.
2. Set the screen size to 320 x 256px and 100% zoom.
3. Alternatively, set the screen size to 1280 x 1024px and 400% zoom.
4. Observe that user is required to scroll in more than one direction.
5. Observe that the content in scope does not meet the exemption criteria.

Recommendation

When the page is reflowed, make sure that the content is stacked such users only need to scroll in one direction. When the viewport is adjusted, make sure that users are not required to scroll in multiple directions (such as up and down, as well as left and right) to access the content and functionality.

As noted, there are some exceptions. You do not need to reflow content when it would change the meaning of the content. For example, a table does not need to be reflowed because the tabular data may not make sense when reflowed. However, any other content must be reflowed.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw` and modern, flexible layouts techniques to achieve this.

Note that decorative content can be lost, such as images that are not functional or informative. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. If you need to tweak the design to mitigate the need for users to scroll in multiple directions, that is acceptable.

For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#), [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#). As well, testing can be done with [Web Developer Toolbar](#) or browser dev tools, such as [Chrome - Device Mode](#).

[Return to priority table](#)

ADR_20590-34 MEDIUM PRIORITY OPEN

Footer - Reflow causes loss of content

Success Criteria 1.4.10: Reflow [2.1 AA]

Devices Desktop

Pages Affected 02

Users Affected Vision

Component Link, Button

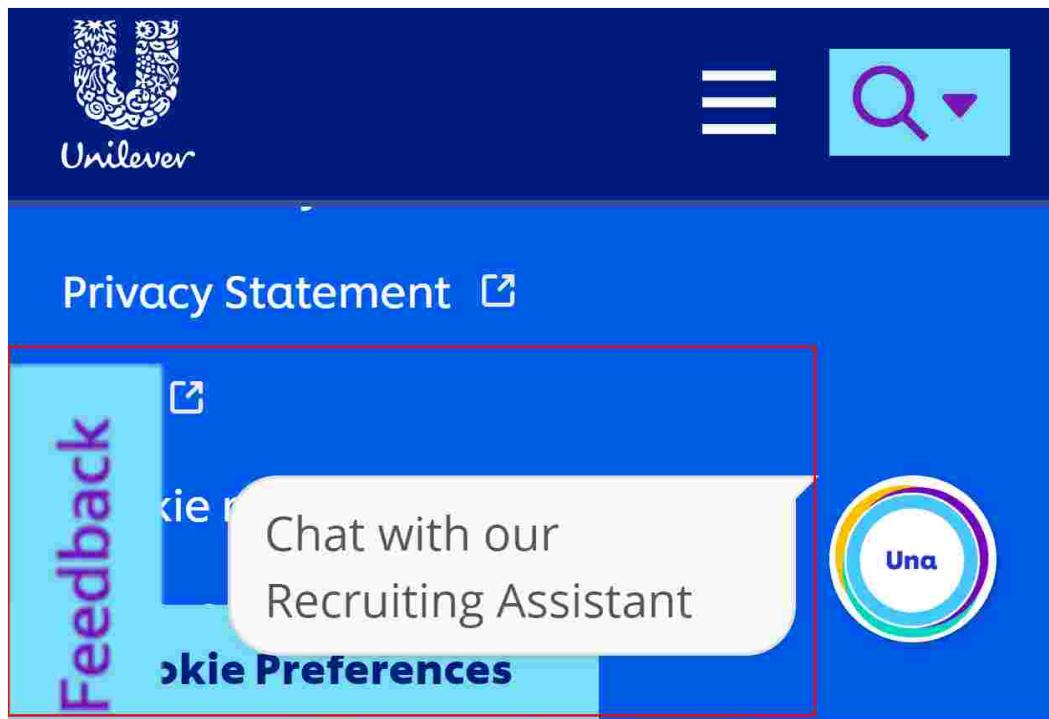
What happened

Users must be able to access content and functionality when the page is reflowed.

This means when the screen size is 320 x 256px and 100% zoom, or 1280 x 1024px and 400% zoom. This is particularly important for low-vision users as it can allow them to make content more readable. There is meaningful or unique content that is lost when reflowed.

Where it happened

This can be seen on page 2 on desktop, with “Cookie Preferences” and “Candidate Feedback” buttons and links such as “RSS”, “Cookie notice” and “Unilever.com”. On changing the viewport, the “Cookie Preferences” button and links are partly overlapped by the “Candidate Feedback” and “Una: Chat with our Recruiting Assistant” buttons present in the footer. The “Candidate Feedback” button is also clipped.



Page 2: desktop, 400% zoom, content is overlapped and lost

Steps to reproduce

1. Observe the content at the default screen size and zoom.
2. Set the screen size to 320 x 256px and 100% zoom.
3. Alternatively, set the screen size to 1280 x 1024px and 400% zoom.
4. Observe that the meaningful content is lost.
5. Observe that there is not an alternative way to access the same content.

Recommendation

Make sure that content is available when the page is reflowed. When the viewport is adjusted, make sure that the content is still available on the page. As well, make sure that users are not required to scroll in multiple directions (such as up and down, as well as left and right) to access the content.

Use relative units, such as `em`, `ex`, `rem`, `%`, `vw` and modern, flexible layouts techniques to achieve this.

Note that decorative content can be lost, such as images that are not functional or informative. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. If you need to tweak the design to allow users to access the meaningful content, that is acceptable.

For more information, see [MDN - Relative Units](#), [CSS Tricks - font-size](#), [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#). As well, testing can be done with [Web Developer Toolbar](#) or browser dev tools, such as [Chrome - Device Mode](#).

[Return to priority table](#)

ADR_20590-12 MEDIUM PRIORITY OPEN

Functional non-text content fails contrast when receiving keyboard focus and mouse hover

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 03, 05, 08

Users Affected Vision

Component Link, Button

What happened

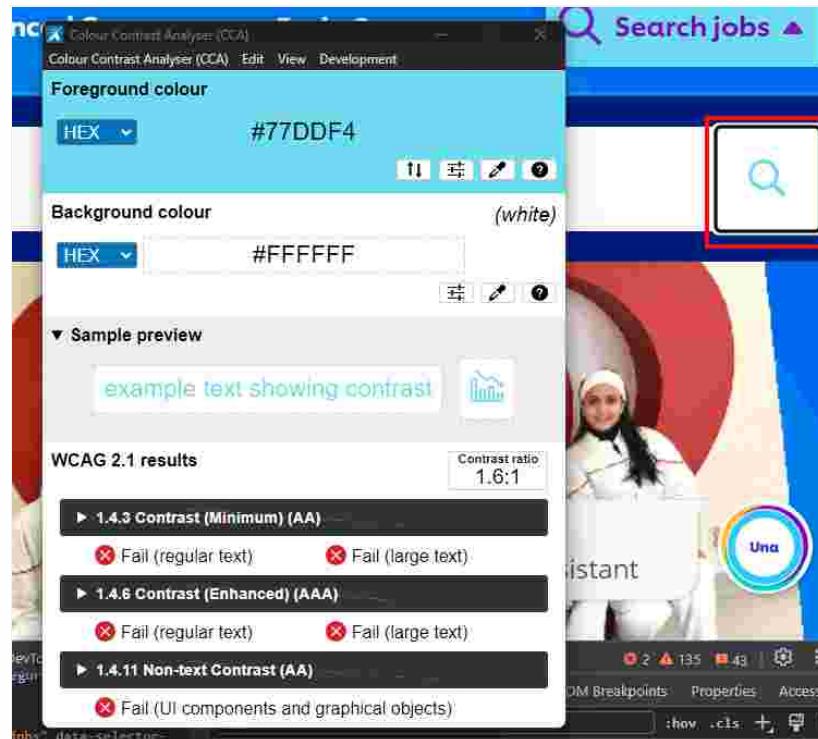
When components have custom styling that causes them to change colour on focus or mouse hover, the colours used must pass colour contrast requirements with adjacent colours. The required contrast ratio is 3:1, unless the element uses default browser CSS.

There are components with custom styling that do not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

Where it happened

This can be seen on page 1 on desktop and iOS with the header search submission button. On keyboard focus or mouse hover, the light blue magnifying glass icon (#77DDF4) has a 1.6:1 contrast ratio on the white background (#FFFFFF).

Note that on iOS this occurs when the button receives the screen reader cursor or is pressed and held.



Page 1: desktop, Colour Contrast Analyser, magnifying glass icon does not have sufficient contrast on focus

This can also be seen on page 3, on desktop, with the social media icon links under the "Connect with us" section. On keyboard focus or mouse hover, the light blue icons (#77DDF4) have a 1.6:1 contrast ratio on the white background (#FFFFFF).

The screenshot shows the Colour Contrast Analyser (CCA) interface. The foreground color is set to #77DDF4 and the background to white (#FFFFFF). The WCAG 2.1 results table indicates a contrast ratio of 1.6:1, which fails both the 1.4.3 Contrast (Minimum) (AA) and 1.4.6 Contrast (Enhanced) (AAA) levels for regular text.

Page 3: desktop, Colour Contrast Analyser, social media icons do not have sufficient contrast on mouse hover

This can also be seen on page 5 on desktop and iOS, with the social media links such as “Twitter”, “facebook” and so on. On keyboard focus or mouse hover, the light blue social media icons (#77DDF4) has a 1.6:1 contrast on the white background (#FFFFFF).

Note that on iOS this occurs when the social media links receive the screen reader cursor or are pressed and held.

The screenshot shows the Colour Contrast Analyser (CCA) tool integrated into a web page. The background is teal (#00B1A9) and the social media icons are off-white (#D9D9D9). The WCAG 2.1 results table indicates a contrast ratio of 1.6:1, failing both the 1.4.3 Contrast (Minimum) (AA) and 1.4.6 Contrast (Enhanced) (AAA) levels for regular text.

Page 5: desktop, Colour Contrast Analyser, social media links does not have sufficient contrast on focus or hover

This can be seen on page 8 on desktop and iOS with the plus signs in the buttons under “Why Choose a Digital Future in Unilever?”. The plus signs communicate that the buttons can be activated to access further information. However, on keyboard focus or mouse hover, the off-white icons (#D9D9D9) have a 1.9:1 contrast ratio on the teal background (#00B1A9).

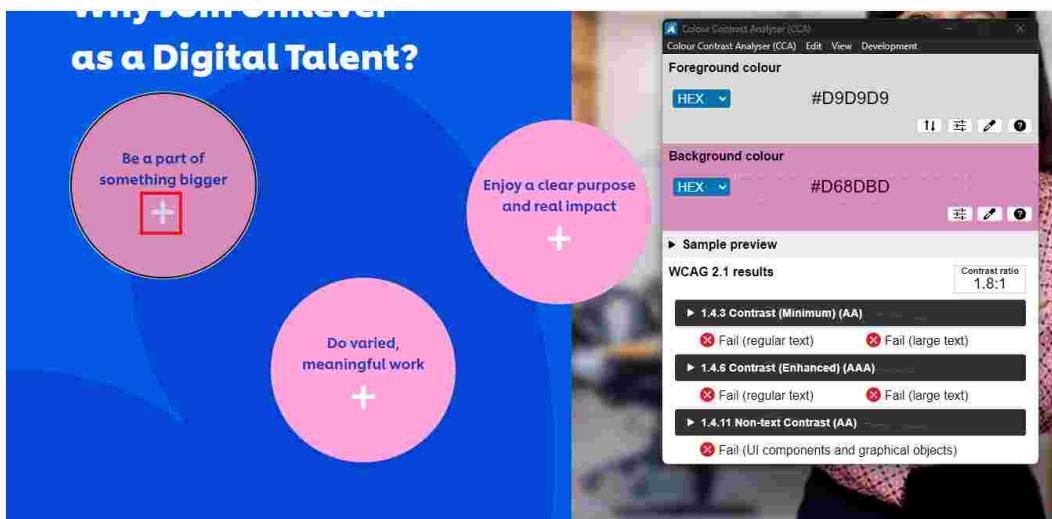
Note that on iOS this occurs when the buttons receive the screen reader cursor or are pressed and held.



Page 8: desktop, Colour Contrast Analyser, plus signs under “Why Choose a Digital Future in Unilever?” do not have sufficient contrast on focus

This can also be seen on page 8 on desktop and iOS with the plus signs in the buttons under “Why Join Unilever as a Digital Talent?”. The plus signs communicate that the buttons can be activated to access further information. However, on keyboard focus or mouse hover, the off-white icons (#D9D9D9) have a 1.8:1 contrast ratio on the pink background (#D68DBD).

Note that on iOS this occurs when the buttons receive the screen reader cursor or are pressed and held.



Page 8: desktop, Colour Contrast Analyser, plus signs under “Why Join Unilever as a Digital Talent?” do not have sufficient contrast on focus

Steps to reproduce

1. Keyboard tab onto/mouse hover on the above component.
2. Observe the custom styling that causes the component to change colour.
3. Get the colour values of the component.
4. Insert these values into a colour contrast analyser.
5. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

When using components with custom focus effects, make sure that the style rules pass 3:1 colour contrast requirements.

Adjust the colours to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative is to use native semantics and default browser stylings, which this success criterion considers to be out of scope. For examples, see [Native Form Elements](#).

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-40 MEDIUM PRIORITY OPEN

Custom interactive element boundary has insufficient contrast

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 13, 14

Users Affected Vision

Component Input Field

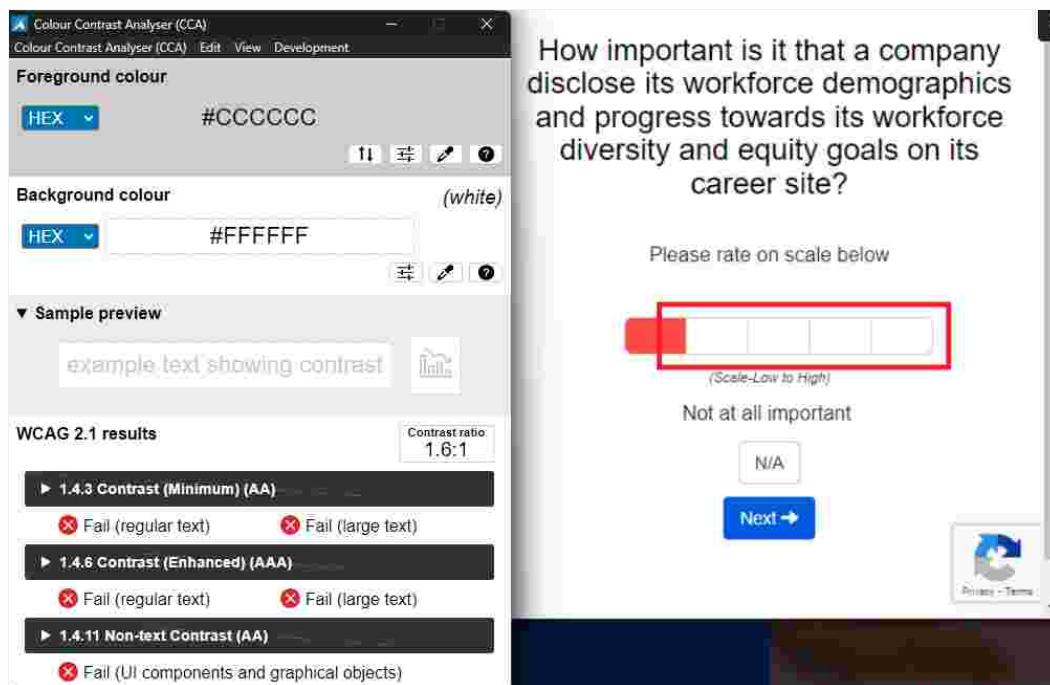
What happened

When custom components do not have internal text, such as custom input fields and checkboxes, the boundary of the hit area must pass colour contrast requirements with adjacent colours. The required contrast ratio is 3:1, unless the element uses default browser CSS.

There are custom elements that do not meet the contrast ratio. This may particularly disadvantage low-vision and color blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

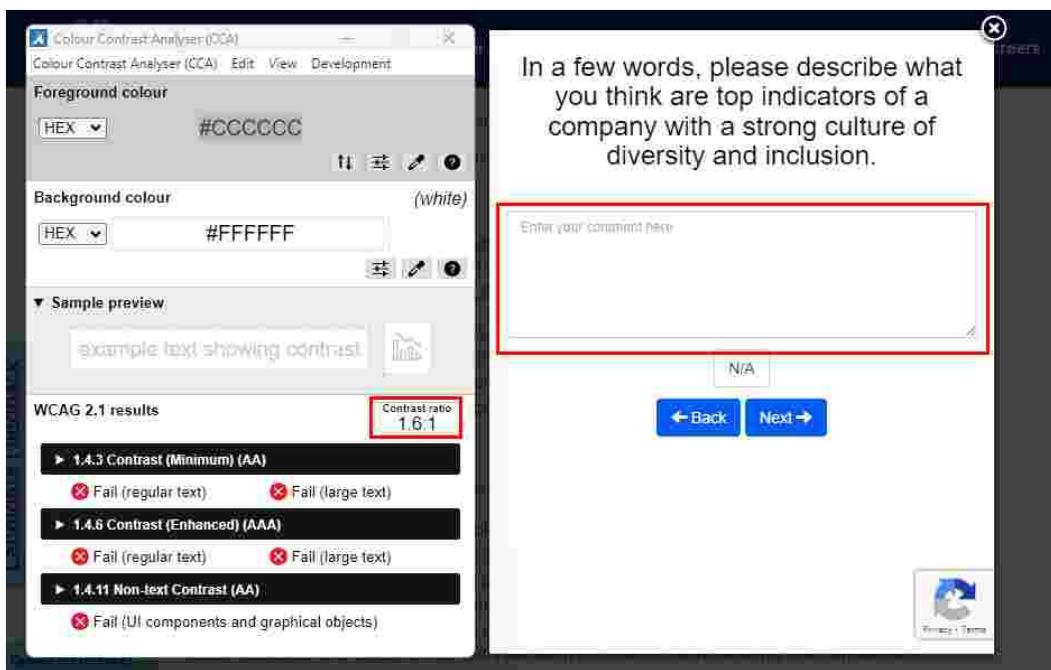
Where it happened

This can be seen on page 13 on desktop and iOS with the unselected ratings buttons when they have a white background. The grey element boundary (#CCCCCC) has an insufficient contrast ratio of 1.6:1 against the white background (#FFFFFF).



Page 13: desktop, Colour Contrast Analyser, ratings buttons with white background do not have sufficient contrast with boundary

This can be seen on page 14 on desktop and iOS, with the boundary of the text field present below the “In a few words, please describe what...” text element. The boundary light grey (#CCCCCC) has an insufficient contrast ratio of 1.6:1 against the white (#FFFFFF) background.



Page 14: desktop, Colour Contrast Analyser, custom interactive element boundary fails contrast requirement

The screenshot shows the DOM structure and the corresponding CSS styles. The CSS code highlights the border style for the input field:

```

    .form-control {
        width: 100%;
        height: 34px;
        padding: 6px 12px;
        background-color: #fff;
        background-image: none;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);
        transition: border-color .15s ease-in-out, box-shadow .15s ease-in-out;
    }

```

Page 14: DOM, input field border with 1px solid light grey CSS border property

Steps to reproduce

1. Get the colour values of the boundary and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.
4. Observe that there is not internal text or some other visual signifier, other than the boundary.

Recommendation

When using custom components that do not have internal text, make sure that the boundaries pass 3:1 colour contrast requirements. When they have internal text, make sure that at least the text passes colour contrast requirements, although it is ideal to make sure that the boundary and the text pass the requirements.

Adjust colours to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative is to use native semantics and default browser stylings, which this success criterion considers to be out of scope. For examples, see [Native Form Elements](#).

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-85 MEDIUM PRIORITY OPEN

Functional non-text content fails contrast

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 08

Users Affected Vision

Component Button

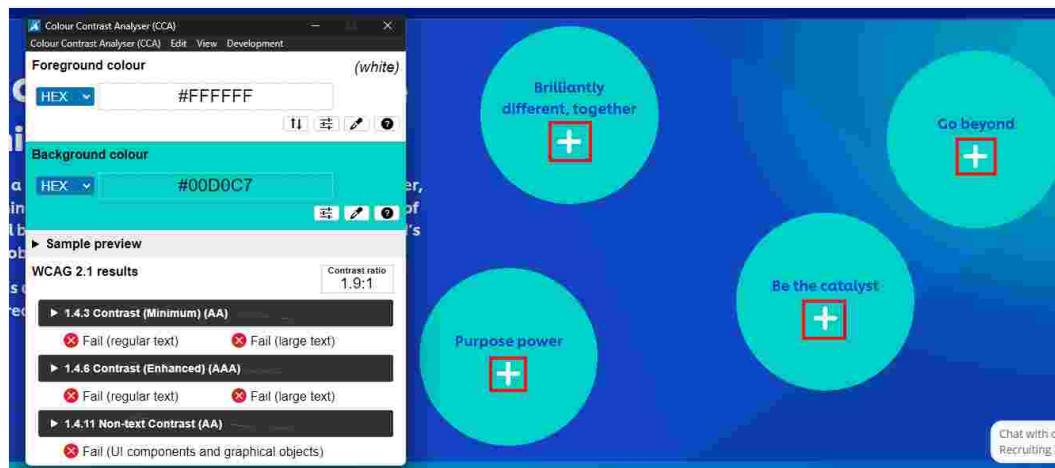
What happened

When non-text content communicates functional that is not available through text, the non-text content must pass colour contrast requirements with adjacent colours. For example, when using icon buttons that have no text, the icon used must pass the colour contrast requirements.

The required contrast ratio is 3:1. There is functional non-text content that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

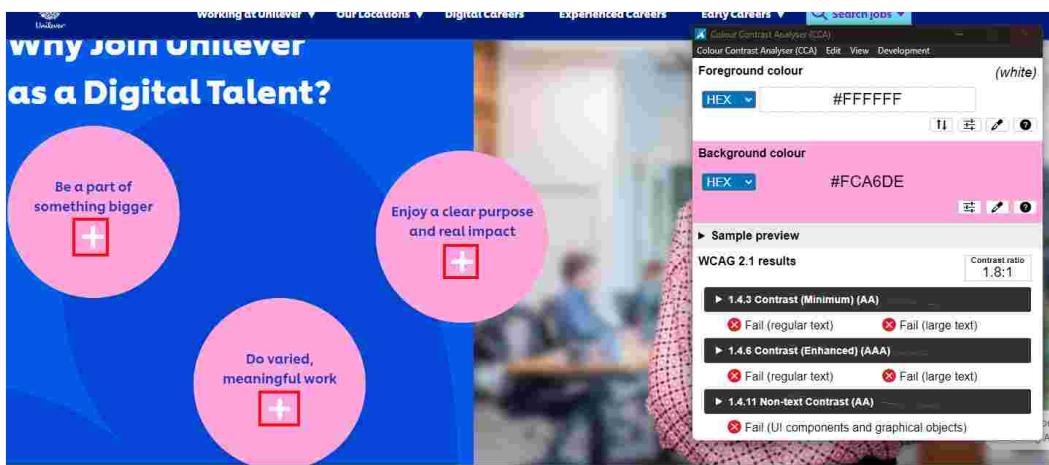
Where it happened

This can be seen on page 8 on desktop and iOS with the plus signs in the buttons under “Why Choose a Digital Future in Unilever?”. The plus signs communicate that the buttons can be activated to access further information. However, the white icons (#FFFFFF) have a 1.9:1 contrast ratio on the teal background (#00D0C7).



Page 8: desktop, Colour Contrast Analyser, plus signs under “Why Choose a Digital Future in Unilever?” do not have sufficient contrast

This can also be seen on page 8 on desktop and iOS with the plus signs in the buttons under “Why Join Unilever as a Digital Talent?”. The plus signs communicate that the buttons can be activated to access further information. However, the white icons (#FFFFFF) have a 1.8:1 contrast ratio on the pink background (#FCA6DE).



Page 8: desktop, Colour Contrast Analyser, plus signs under "Why Join Unilever as a Digital Talent?" do not have sufficient contrast

Steps to reproduce

1. Get the colour values of the functional non-text content.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.
4. Observe that there is not an alternative way to access the same information.

Recommendation

When using functional non-text content, make sure that the colours pass 3:1 colour contrast requirements.

Adjust colours to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative would be providing a button with text on it, assuming that the colour of the text passed the colour contrast requirements.

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-87 MEDIUM PRIORITY OPEN

Functional non-text content fails contrast on background image

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 08

Users Affected Vision

Component Button

What happened

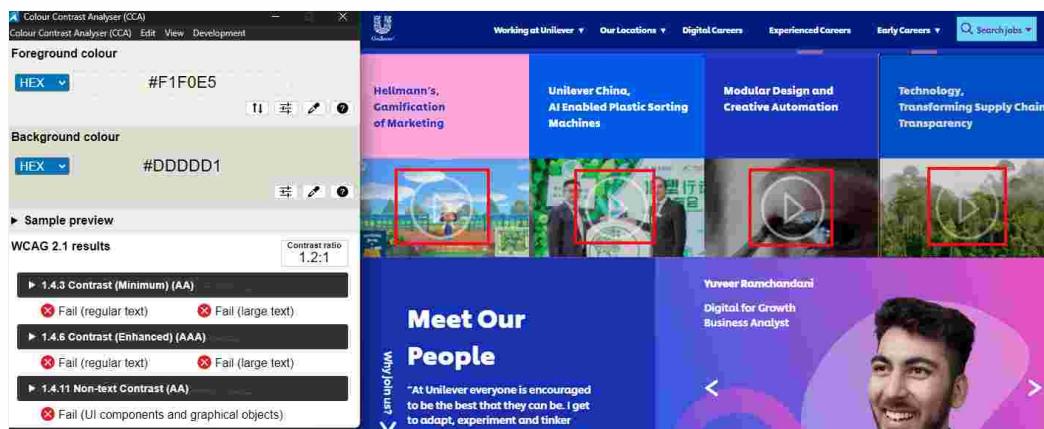
When non-text content communicates functional that is not available through text, the non-text content must pass colour contrast requirements with adjacent colours. For example, when using icon buttons that have no text, the icon used must pass the colour contrast requirements.

The required contrast ratio is 3:1. There is functional non-text content that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

Where it happened

This can be seen on page 8 on desktop and iOS with the play icons in the buttons under “Purpose Driven Work with Global Impact”. The play icons indicate that the buttons will open video modals. However, they do not have sufficient contrast against portions of the background images. For example, for the portion of the “Hellmann’s, Gamification of Marketing” play icon that overlaps the game character’s hat, the off-white foreground (#F1F0E5) has a 1.2:1 contrast ratio with the off-white background (#DDDDDD1).

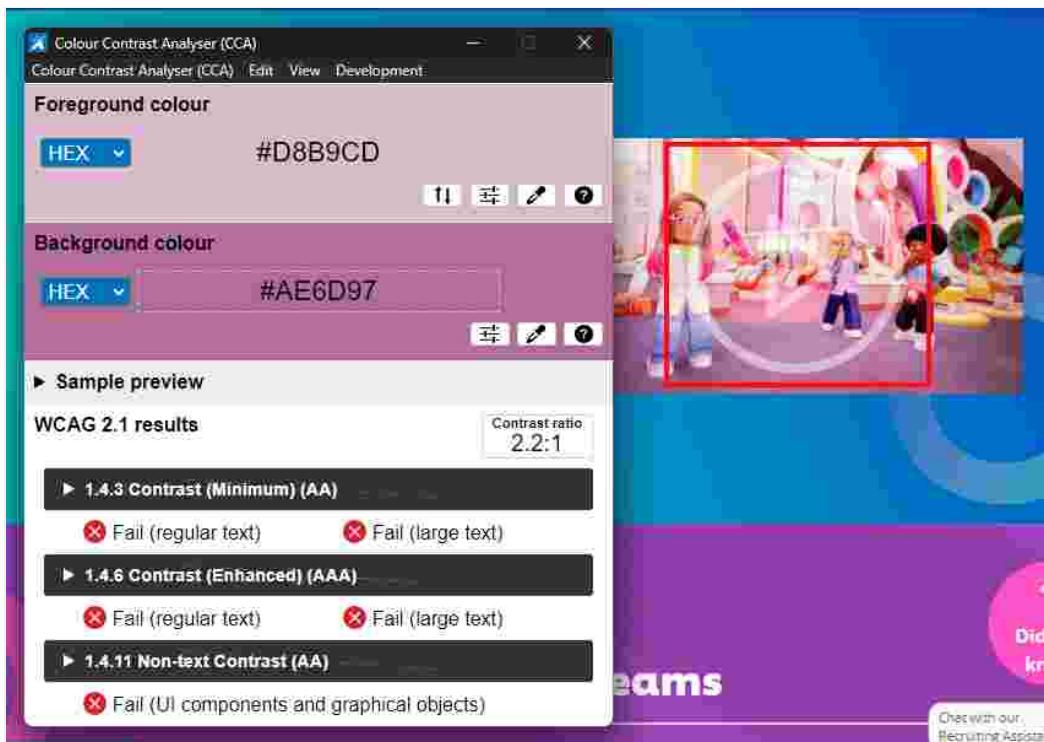
Note that, on keyboard focus or mouse hover, the whole button is slightly darkened, which changes the foreground and background colours. However, portions of each play icon still do not have sufficient contrast with the background image.



Page 8: desktop, Colour Contrast Analyser, play icon for “Hellmann’s, Gamification of Marketing” does not have sufficient contrast on variable coloured background

This can also be seen on page 8 on desktop and iOS with the play icon for the “What is a Digital Career Like in Unilever?” link. The play icon indicates that the button will open a video modal. However, it does not have sufficient contrast against portions of the background image. For example, the bottom portion of the play icon (#D8B9CD) has a 2.2:1 contrast ratio with the purple floor in the image (#AE6D97).

Note that, on keyboard focus or mouse hover, the play icon changes size, which changes the foreground and background colours. However, portions of the play icon still do not have sufficient contrast with the background image.



Page 8: desktop, Colour Contrast Analyser, play icon for "What is a Digital Career Like in Unilever?" link does not have sufficient contrast on variable coloured background

Steps to reproduce

1. Get the colour values of the functional non-text content.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.
4. Observe that there is not an alternative way to access the same information.

Recommendation

When using functional non-text content, make sure that the colours pass 3:1 colour contrast requirements.

Adjust colours to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative would be providing a button with text on it, assuming that the colour of the text passed the colour contrast requirements.

We recommend putting meaningful icons on solid backgrounds wherever possible to make sure that they have sufficient contrast with all of the background, including at different zoom levels and on different devices.

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-95 MEDIUM PRIORITY OPEN

Functional non-text content fails contrast (ratings buttons)

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Button

What happened

When non-text content communicates functional that is not available through text, the non-text content must pass colour contrast requirements with adjacent colours. For example, when using icon buttons that have no text, the icon used must pass the colour contrast requirements.

The required contrast ratio is 3:1. There is functional non-text content that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

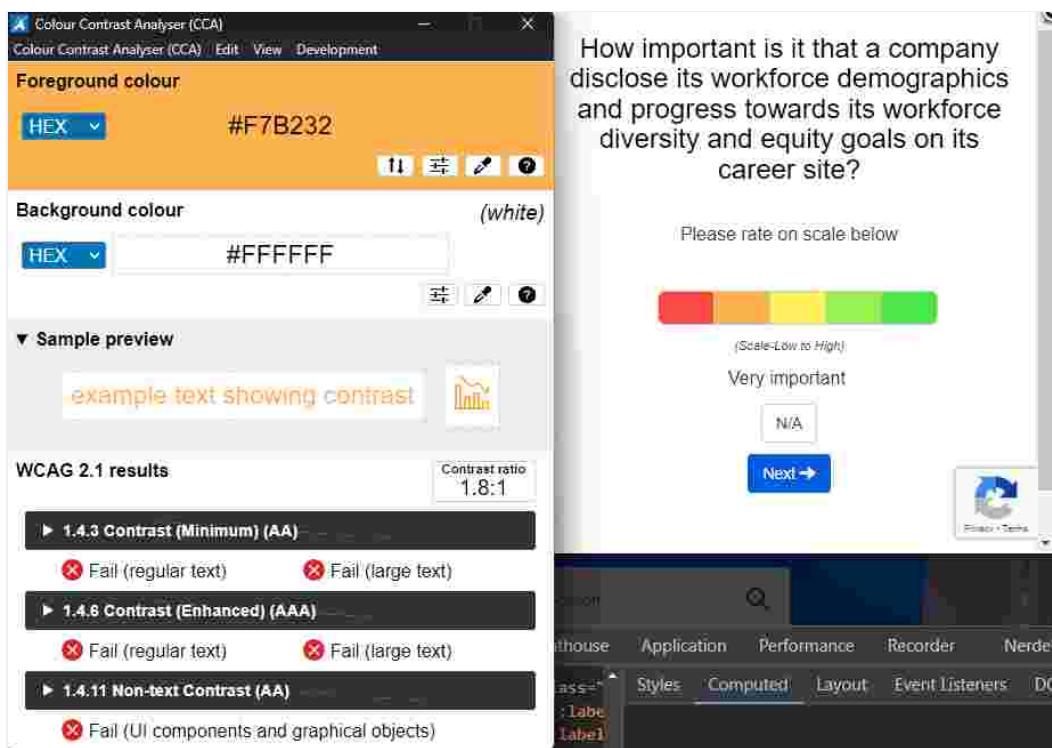
Where it happened

This can be seen on page 13 on desktop and iOS with the second, third, fourth, and fifth ratings buttons when they have a coloured background.

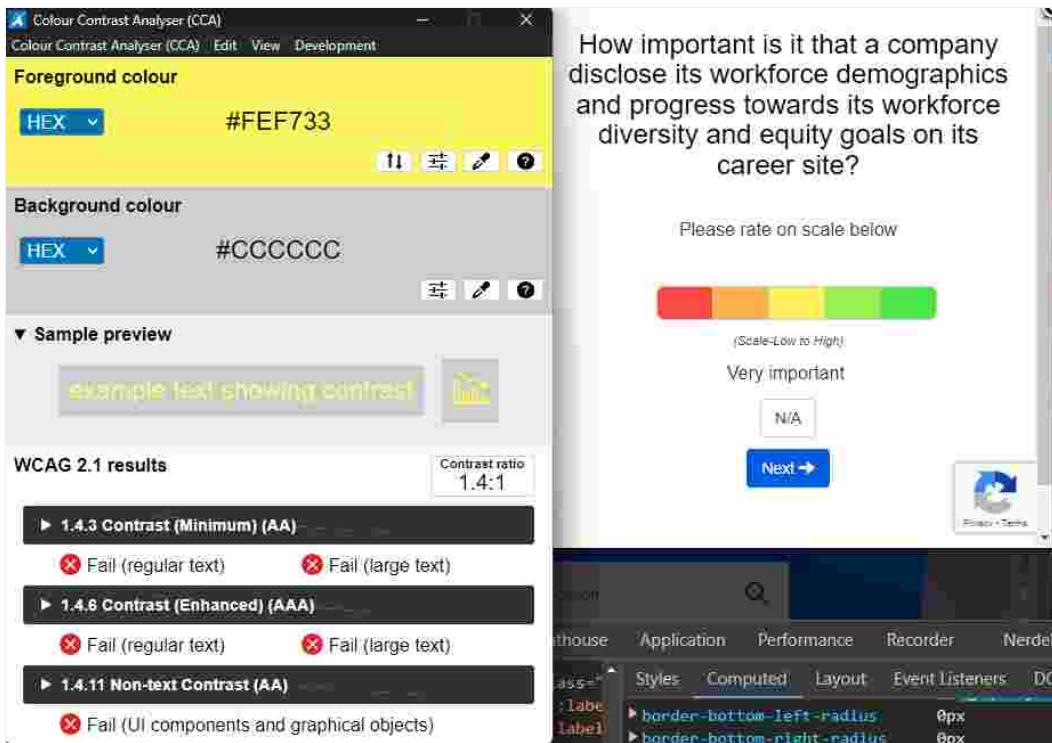
Not that, when a button has mouse hover or is selected, that button and all of the lower ratings buttons have a coloured background.

The following colour combinations do not have sufficient contrast:

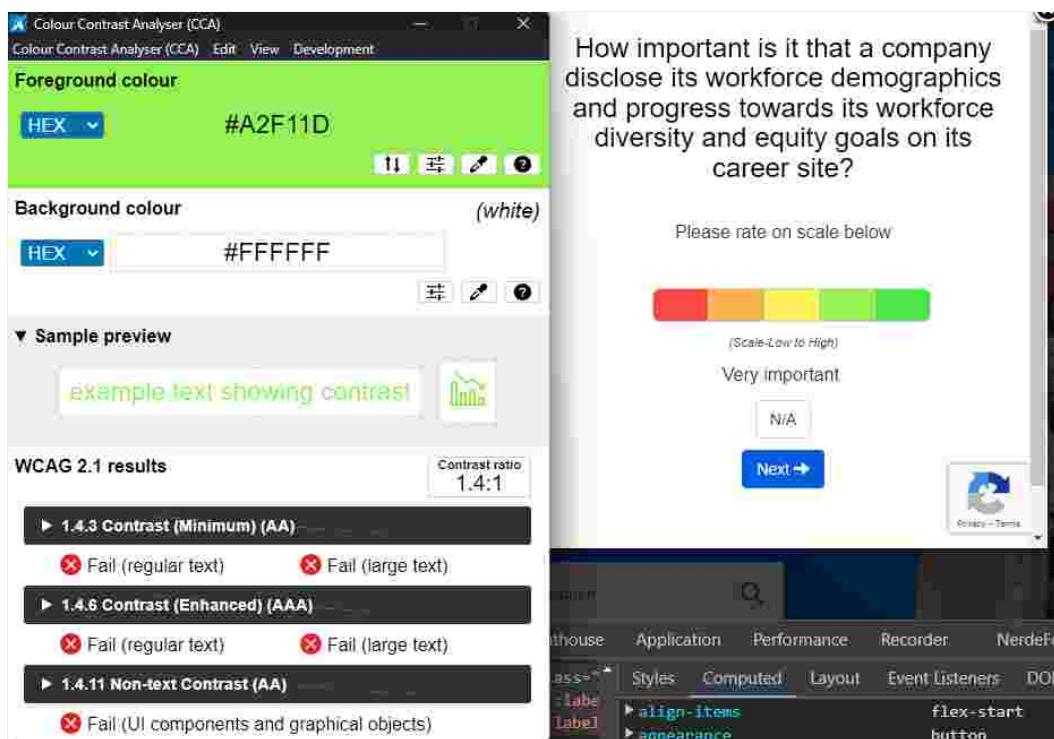
- The orange button background (#F7B232) has a 1.8:1 contrast ratio with the white background (#FFFFFF). Note that this is a higher ratio than between the orange button and the grey boundary.
- The yellow button background (#FEF733) has a 1.4:1 contrast ratio with the grey boundary (#CCCCCC). Note that this is a higher ratio than between the yellow button and the white background.
- The lighter green button background (#A2F11D) has a 1.4:1 contrast ratio with the white background (#FFFFFF). Note that this is a higher ratio than between the orange button and the grey boundary.
- The darker green button background (#6AE407) has a 1.7:1 contrast ratio with the white background (#FFFFFF). Note that this is a higher ratio than between the orange button and the grey boundary.



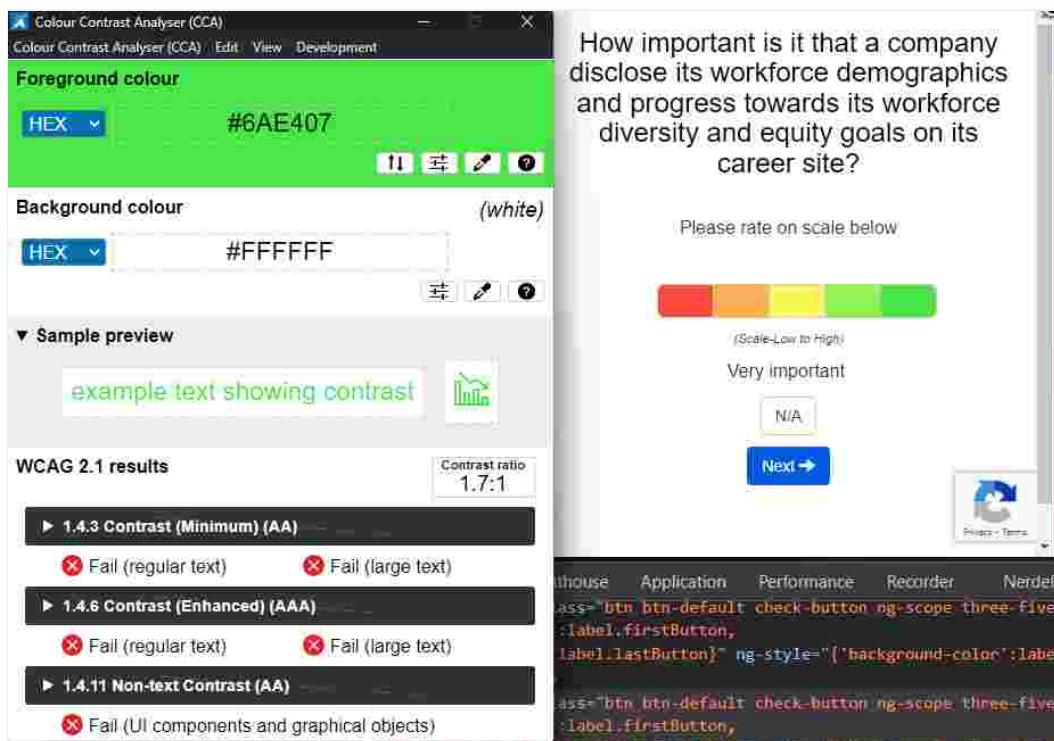
Page 13: desktop, Colour Contrast Analyser, orange button does not have sufficient contrast



Page 13: desktop, Colour Contrast Analyser, yellow button does not have sufficient contrast



Page 13: desktop, Colour Contrast Analyser, light green button does not have sufficient contrast



Page 13: desktop, Colour Contrast Analyser, dark green button does not have sufficient contrast

Steps to reproduce

1. Get the colour values of the functional non-text content.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.
4. Observe that there is not an alternative way to access the same information.

Recommendation

When using functional non-text content, make sure that the colours pass 3:1 colour contrast requirements.

Adjust colours to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative would be providing a button with text on it, assuming that the colour of the text passed the colour contrast requirements.

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-110 MEDIUM PRIORITY OPEN

Custom focus indication fails contrast

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop

Pages Affected 02

Users Affected Vision

Component Button

What happened

When components have custom focus indication, such as a custom **outline**, the indication must pass colour contrast requirements with adjacent colours. The required contrast ratio is 3:1, unless the element uses default browser CSS.

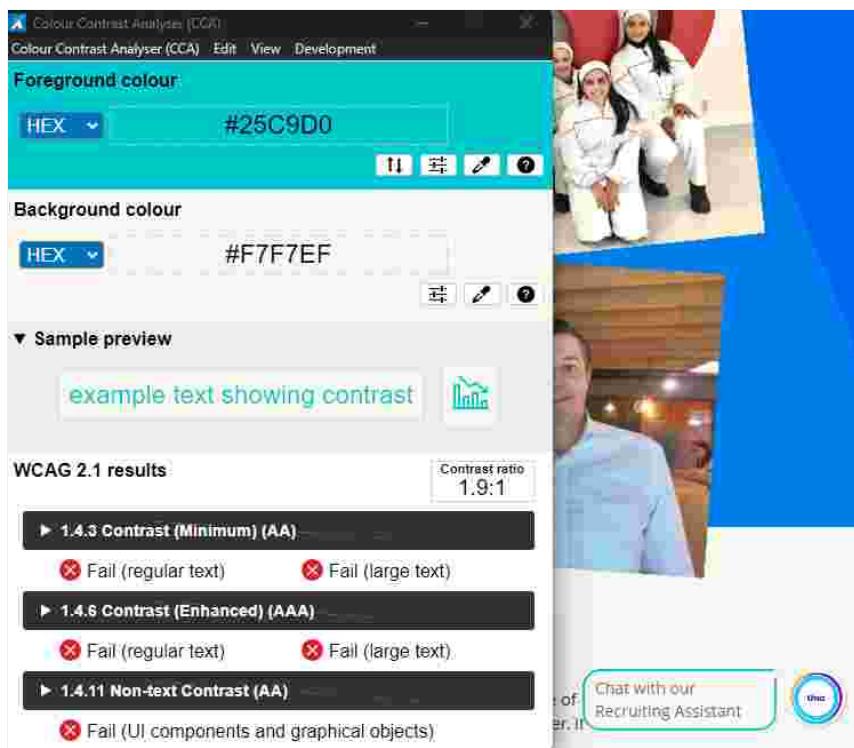
There is custom focus indication that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the focus indication that fails the colour contrast requirements.

Where it happened

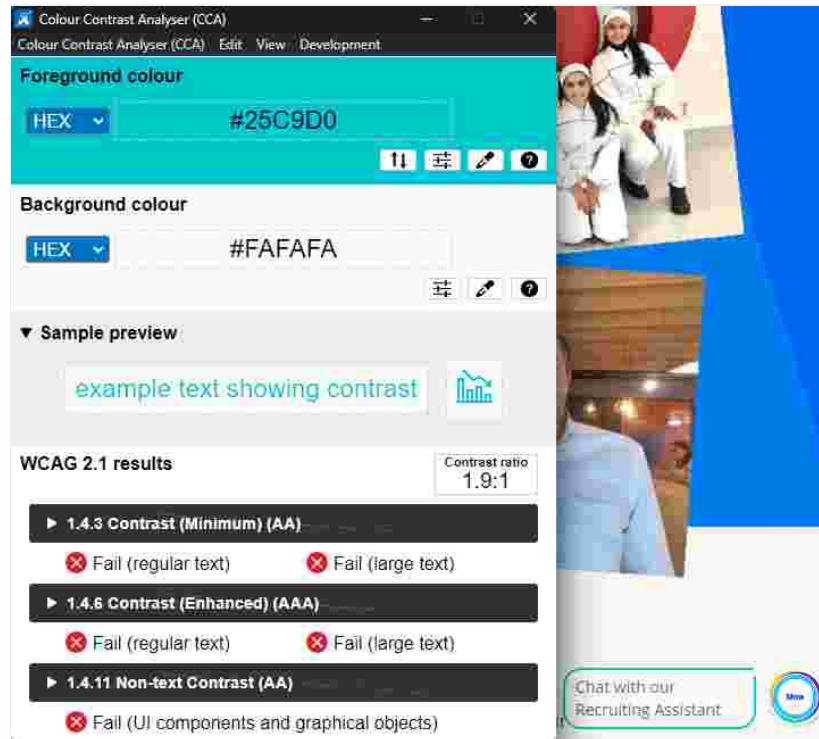
This can be seen on page 2 with the triggering buttons for the chat widget. These elements have a cyan focus indicator (#25C9D0), which does not have sufficient contrast with the interior of the buttons and does not consistently have sufficient contrast with the page background.

For example,

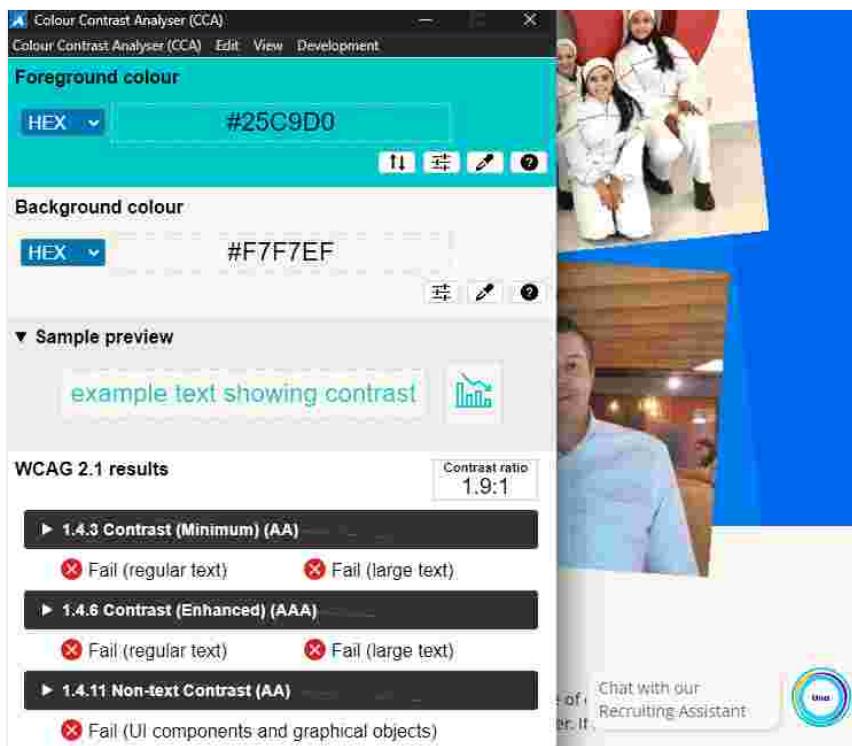
- The focus indicator of the “Chat with our Recruiting Assistant” element has a 1.9:1 contrast ratio with the off-white background of the “Recruitment scams & fraud warning” section on the homepage (#F7F7EF).
- The focus indicator of the “Chat with our Recruiting Assistant” element has a 1.9:1 contrast ratio with the off-white internal background of the element (#FAFAFA).
- The focus indicator of the “Una” element has a 1.9:1 contrast ratio with the off-white background of the “Recruitment scams & fraud warning” section on the homepage (#F7F7EF).
- The focus indicator of the “Una” element has a 2:1 contrast ratio with the off-white internal background of the element adjacent to the focus indicator (#FFFFFF).



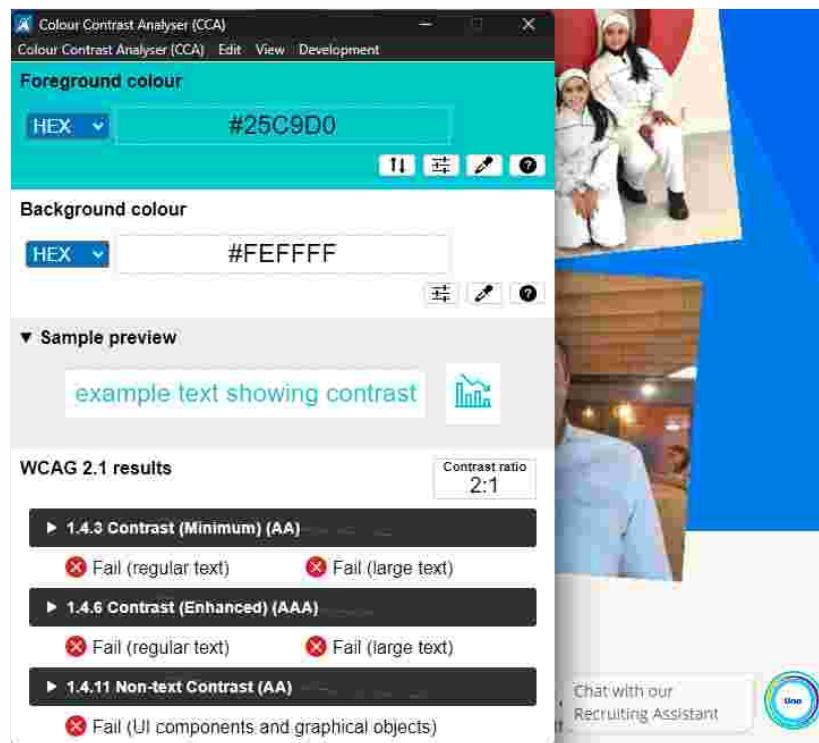
Page 2: desktop, Colour Contrast Analyser, focus indicator of the "Chat with our Recruiting Assistant" element does not have sufficient contrast with background of "Recruitment scams & fraud warning" section on the homepage



Page 2: desktop, Colour Contrast Analyser, focus indicator of the "Chat with our Recruiting Assistant" element does not have sufficient contrast with its internal background



Page 2: desktop, Colour Contrast Analyser, focus indicator of the “Una” element does not have sufficient contrast with the background of the “Recruitment scams & fraud warning” section on the homepage



Page 2: desktop, Colour Contrast Analyser, focus indicator of the “Una” element does not have sufficient contrast with the internal background of the element

Steps to reproduce

1. Get the colour values of the focus indication and background colours.
2. Insert these values into a colour contrast analyser.
3. Observe that the analyser reports that the contrast ratio is not sufficient.

Recommendation

When using custom focus indication, make sure that the indication passes 3:1 colour contrast requirements.

Adjust colours meet the contrast ratio requirements. If you cannot adjust the colours, an alternative is to use native semantics and default browser stylings, which this success criterion considers to be out of scope. For examples, see [Native Form Elements](#).

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-5 MEDIUM PRIORITY OPEN

Text spacing causes content or functionality to overlap

Success Criteria 1.4.12: Text Spacing [2.1 AA]

Devices Desktop

Pages Affected 01

Users Affected Cognitive

Component Plain Text

What happened

Users must be able to adjust certain text properties. This is particularly important for users with dyslexia as it can allow them to make text more readable. No loss of content or functionality should occur when the following style properties are adjusted.

- Line height (line spacing) to at least 1.5 times the font size.
- Spacing following paragraphs to at least 2 times the font size.
- Letter spacing (tracking) to at least 0.12 times the font size.
- Word spacing to at least 0.16 times the font size.

Adjusting these properties causes content or functionality to overlap.

Where it happened

This can be seen on page 1 with the “Search jobs” button text. When text spacing is increased, the chevron icon overlaps with the “s” in “jobs”.



Page 1: desktop, text spacing bookmarklet, chevron overlaps with “jobs” text in “Search jobs” button

Steps to reproduce

1. Navigate to the content.
2. Adjust the style properties with an extension, bookmarklet, or custom stylesheet.
3. Note that the content or functionality overlaps.

Recommendation

Make sure that the style properties are supported. Example techniques are provided by [WCAG Techniques - Allow for text spacing without wrapping](#), [Allow for text spacing override](#), and [Orange Developer Guidelines - Allow Text Spacing](#).

Testing can be done with a bookmarklet, such as [CodePen - Text Spacing Bookmarklet](#) or setting the following CSS rules.

```
* {  
  line-height: 1.5!important;  
  letter-spacing:.12em!important;  
  word-spacing: .16em !important;  
}  
  
p {  
  margin-bottom: 2em!important;  
}
```

As well, use modern, flexible layouts techniques. For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#).

[Return to priority table](#)

ADR_20590-51 MEDIUM PRIORITY OPEN

Text spacing causes content or functionality to be lost

Success Criteria 1.4.12: Text Spacing [2.1 AA]

Devices Desktop

Pages Affected 03, 08

Users Affected Cognitive

Component Heading, Dropdown

What happened

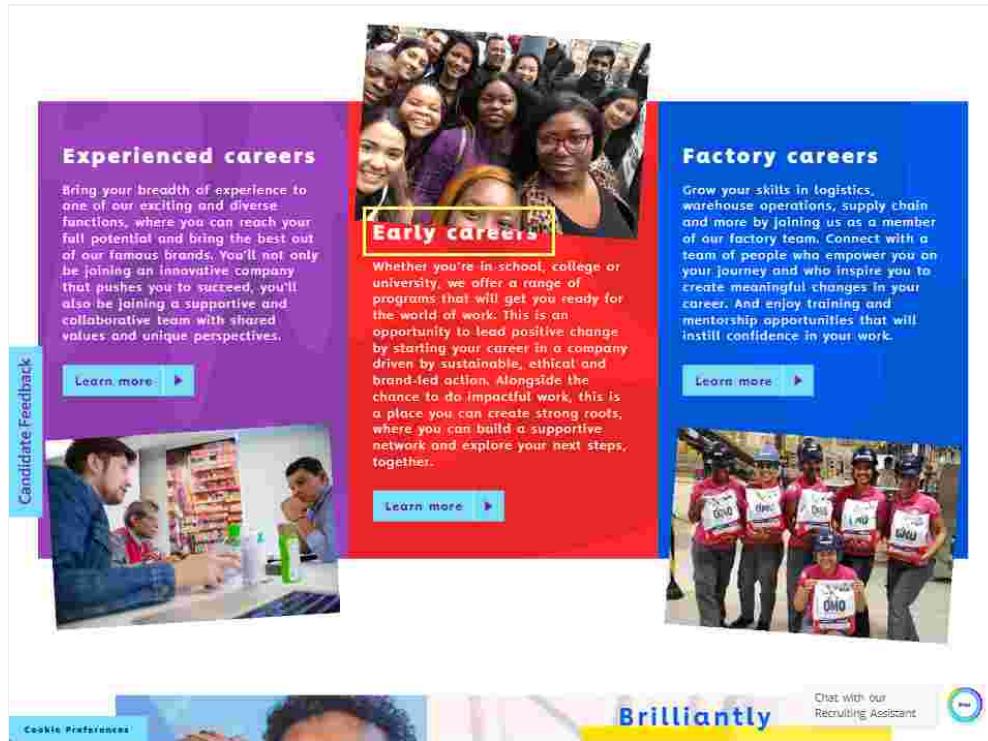
Users must be able to adjust certain text properties. This is particularly important for users with dyslexia as it can allow them to make text more readable. No loss of content or functionality should occur when the following style properties are adjusted.

- Line height (line spacing) to at least 1.5 times the font size.
- Spacing following paragraphs to at least 2 times the font size.
- Letter spacing (tracking) to at least 0.12 times the font size.
- Word spacing to at least 0.16 times the font size.

Adjusting these properties causes content or functionality to be lost.

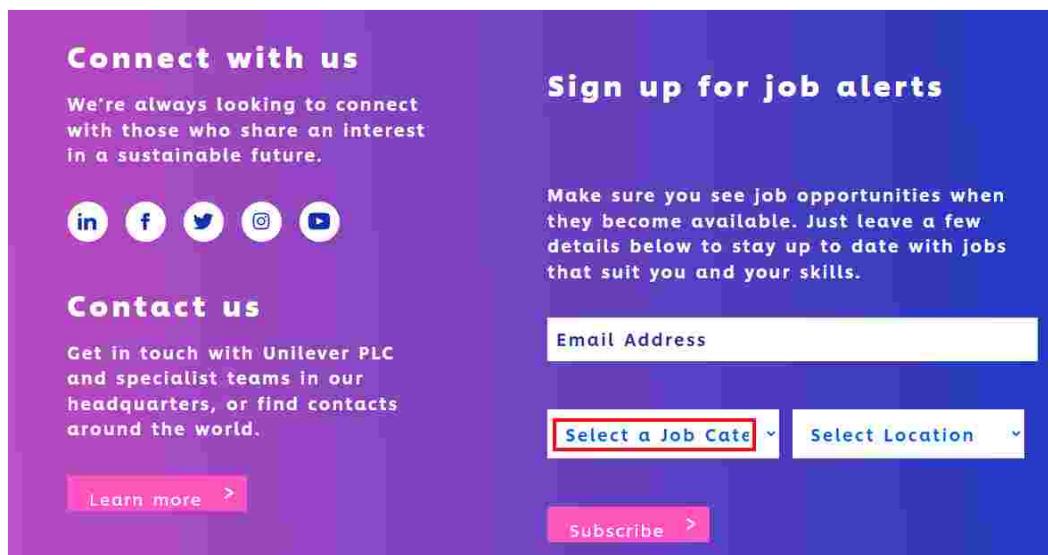
Where it happened

This can be seen on page 3, on desktop. When text spacing is adjusted, the “Early careers” heading is cut off.



Page 3: desktop, text spacing adjusted, “Early careers” heading is cut off

This can be seen on page 8 on desktop. When text spacing is adjusted, the “Select a job category” text is cut off in the job alerts section. Note that, because the dropdown does not have a permanent label, this value functions to identify the element.



Page 8: desktop, text spacing adjusted, "Select a job category" is cut off

Steps to reproduce

1. Navigate on the content.
2. Adjust the style properties with an extension, bookmarklet, or custom stylesheet.
3. Note that the content or functionality is lost.

Recommendation

Make sure that the style properties are supported. Example techniques are provided by [WCAG Techniques - Allow for text spacing without wrapping](#), [Allow for text spacing override](#), and [Orange Developer Guidelines - Allow Text Spacing](#).

Testing can be done with a bookmarklet, such as [CodePen - Text Spacing Bookmarklet](#) or setting the following CSS rules.

```
* {  
    line-height: 1.5!important;  
    letter-spacing:.12em!important;  
    word-spacing: .16em !important;  
}  
  
p {  
    margin-bottom: 2em!important;  
}
```

As well, use modern, flexible layouts techniques. For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), and [CUBE CSS](#).

[Return to priority table](#)

ADR_20590-105 MEDIUM PRIORITY OPEN

Content triggered by mouse hover or keyboard focus is not dismissable

Success Criteria 1.4.13: Content on Hover or Focus [AA]

Devices Desktop

Pages Affected 11

Users Affected Vision

Component Link

What happened

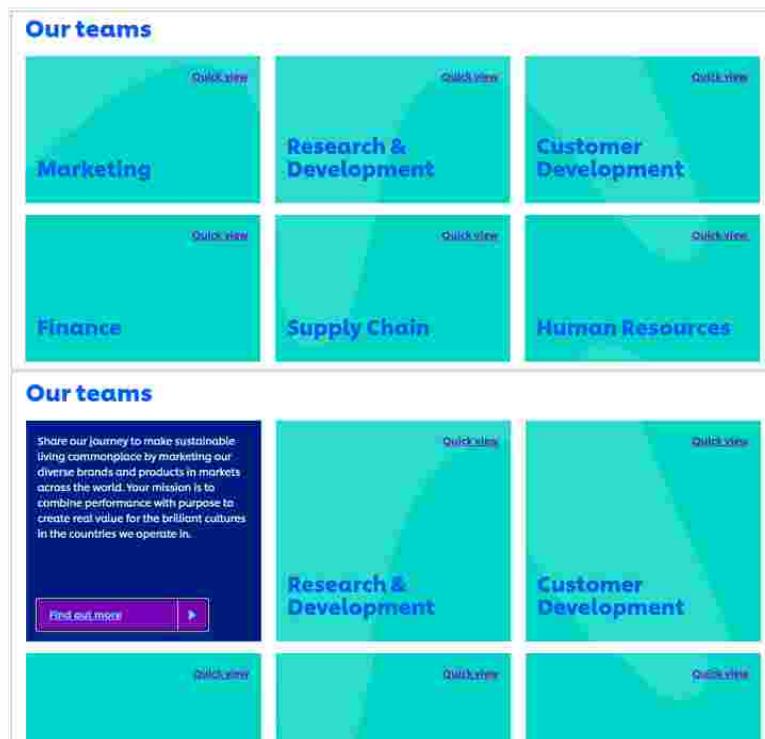
Where content is shown when an element receives pointer hover or keyboard focus, the following must be true.

- When the new content obscures other content, it must be possible to dismiss the content without moving the mouse pointer or keyboard focus.
- Users must be able to move the mouse pointer over the new content without it being dismissed.
- The content must remain persistently visible until the user moves their mouse or keyboard focus onto other elements.

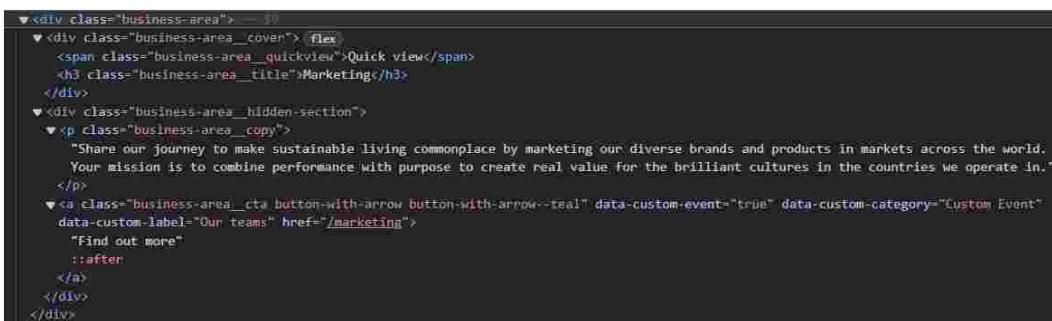
There is additional content on mouse hover or keyboard focus that obscures other content and cannot be dismissed.

Where it happened

This can be seen on page 11, on desktop, with the cards under the “Our teams” section. On mouse hover or keyboard focus, the cards expand further information regarding the respective card, which cannot be dismissed without moving the mouse pointer or keyboard focus. The information entirely overlaps the initial card content.



Page 11: desktop, example of further information presented on keyboard focus for “Marketing” card



The screenshot shows the DOM structure for a 'Marketing' card. It includes a title 'Marketing', a quick view link, a mission statement, and a button labeled 'Find out more'. The code uses classes like 'business-area', 'business-area__cover', 'business-area__title', 'business-area__copy', and 'business-area__cta'.

```
<div class="business-area">
  <div class="business-area__cover">flex
    <span class="business-area__quickview">Quick view</span>
    <h3 class="business-area__title">Marketing</h3>
  </div>
  <div class="business-area__hidden-section">
    <p class="business-area__copy">
      "Share our journey to make sustainable living commonplace by marketing our diverse brands and products in markets across the world. Your mission is to combine performance with purpose to create real value for the brilliant cultures in the countries we operate in."
    </p>
    <a class="business-area__cta button-with-arrow button-with-arrow--teal" data-custom-event="true" data-custom-category="Custom Event" data-custom-label="Our teams" href="/marketing">
      "Find out more"
      ::after
    </a>
  </div>
</div>
```

Page 11: DOM, example of “Marketing” card

Steps to reproduce

1. Navigate onto the trigger element.
2. Observe that additional content is added to the page.
3. Observe that the additional content obscures other content.
4. Observe that you cannot dismiss the additional content without moving the mouse or keyboard focus.

Recommendation

In this case, we recommend avoiding a design pattern where disclosed content obscures the heading that relates to it. This pattern could potentially cause confusion, because relevant information is removed from the page when the additional content is disclosed.

Instead, we recommend a standard accordion pattern where the heading remains visible when it is expanded to disclose further content. For example,

```
<h3><button aria-expanded="false">Marketing</button></h3>
<!-- The content below is hidden until the button is activated -->
<div hidden>
  <p>Share our journey...</p>
  <a href="...">Find out more about marketing at Unilever</a>
</div>
```

If this is not possible, make sure that users can dismiss the additional content. This is best done by allowing users to dismiss the content using Escape.

Alternatively, refactor the component such that it requires activation, such as mouse click or tabbing onto it and activating it with Return or Space. This success criterion considers components that are manually toggled to be out of scope.

For more information, see [W3C - Making content on focus or hover hoverable, dismissible, and persistent](#) and [Inclusive Components - Tooltips and Toggletips](#).

[Return to priority table](#)

ADR_20590-9 MEDIUM PRIORITY OPEN

Same page link does not move focus when activated

Success Criteria 2.1.1: Keyboard [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 08

Users Affected Motor

Component Link

What happened

Interactive elements must be entirely compatible with a keyboard.

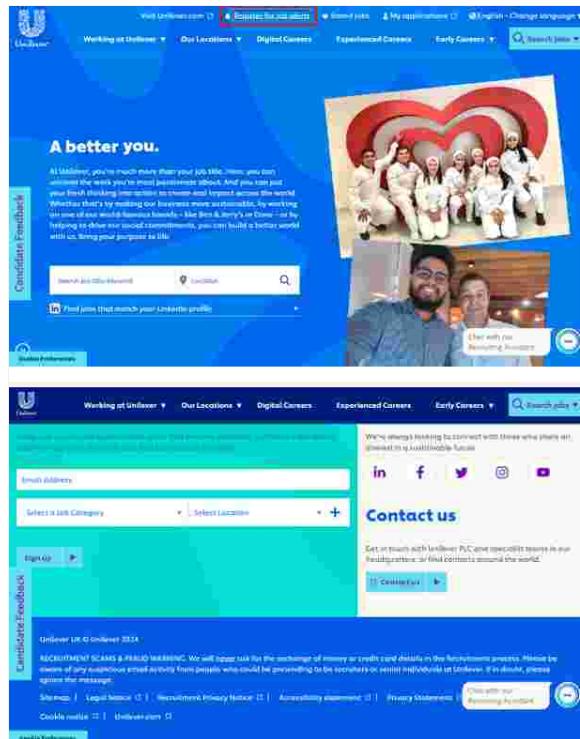
Users should never need to use a mouse. Otherwise, screen reader, keyboard, and switch users, may be unable to control the interactive element. There are same-page links that do not move focus when activated. Instead, focus remains on the same page link. As a result, the same page functionality cannot be accessed with a keyboard.

Note that this has been raised as medium priority, rather than high priority, because it does not block the user journey. As well, users can manually traverse the page to access the elements. However, all users should be able to access the same-page links, and the shortcuts that they provide.

Where it happened

This can be seen on page 1 on desktop and iOS with the “Register for job alerts” link. When this link is activated on desktop, the page scrolls visually to the “Sign up for job alerts” section. However, focus is programmatically on the `<body>` element. Pressing Tab moves the keyboard focus to the “Working at Unilever” link in the header.

On iOS, when the “Register for job alerts” link is activated with a screen reader, the page scrolls visually to the bottom, and the cursor is not placed on an element on the screen.



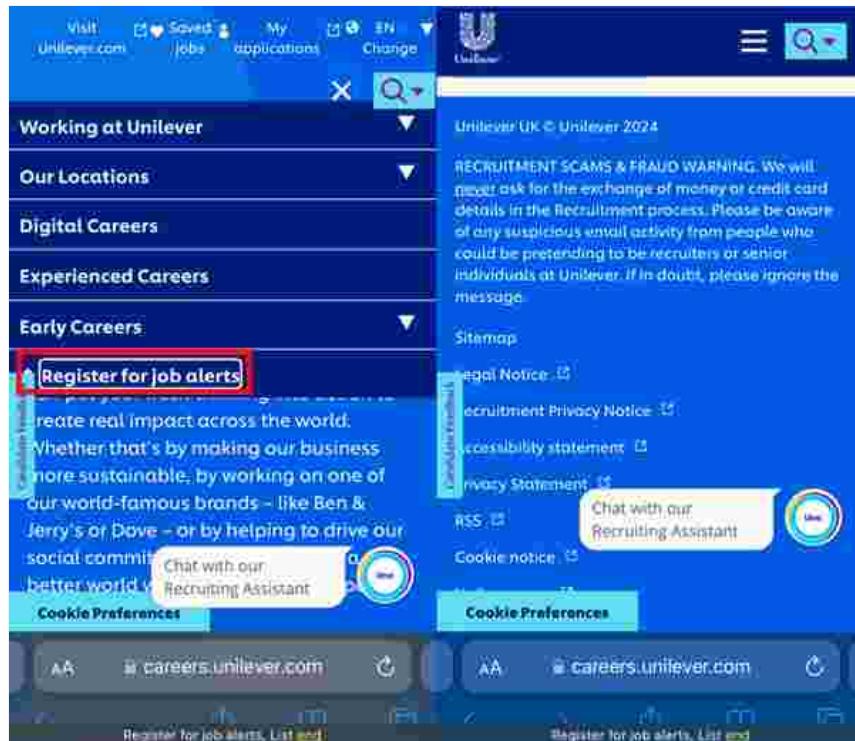
Page 1: desktop, when “Register for job alerts” link is activated, page scrolls visually to “Sign up for job alerts” section, but keyboard focus is not placed in this section

```
X document.activeElement  
body#house.magic-bullet-sally
```

Page 1: Console, document.activeElement query shows that focus is on the body

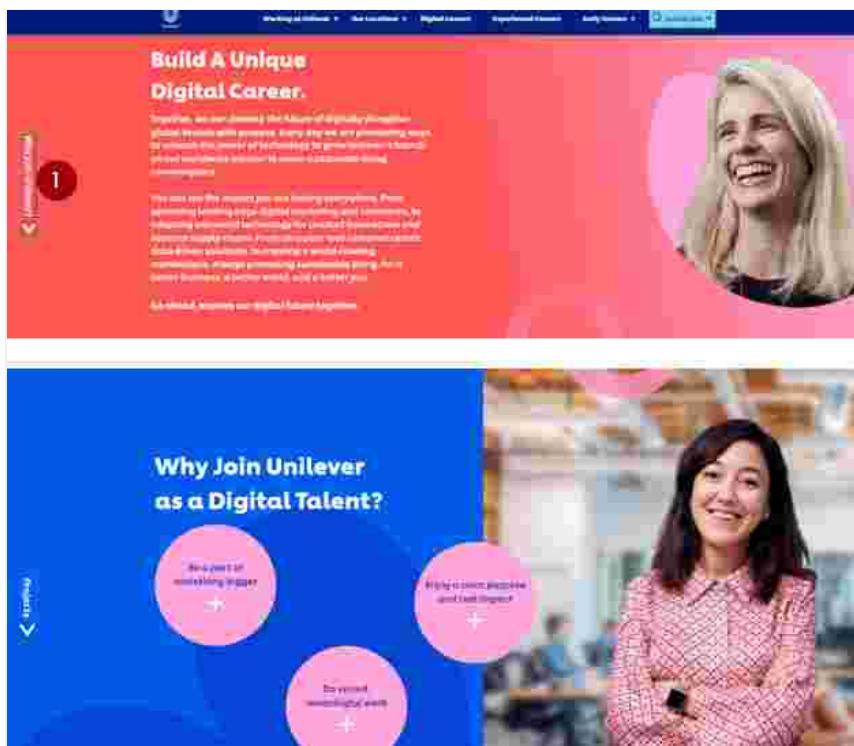
```
<div class="upper-nav" role="list"> #  
  <a class="upper-nav_element external-link" href="https://unilever.com" target="_blank" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Unilever.com" role="listitem" rel="noopener">...</a>  
  <button class="desktop-only desktop-only--inline upper-nav_element upper-nav_element-alerts" id="hide-in-russia" data-custom-event="true" data-custom-category="Custom" data-custom-label="Contact Us" role="listitem">...</button>  
  <:before>  
  "Register for job alerts"  
</button>  
  <a class="upper-nav_element upper-nav_element--saved-jobs" href="/saved-jobs" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Saved jobs" role="listitem">...</a>  
  <a class="upper-nav_element upper-nav_element--login-external-link" href="https://unilever-wd3.myworkdayjobs.com/unilever_Experienced_Professionals/UserHome" target="_blank" data-custom-event="true" data-custom-category="ATS_Login" data-custom-label="ATS Sign-in button" role="listitem" rel="noopener">...</a>
```

Page 1: DOM, "Register for job alerts" link



Page 1: iOS, when "Register for job alerts" link is activated, page scrolls visually to bottom, and screen reader cursor is not placed in "Sign up for job alerts" section

This can also be seen on page 8 on desktop with the in-page links for each section of the page, such as "Why join Unilever?", "Projects", and so on. When these links are activated with the keyboard, the page scrolls visually to the appropriate section. However, focus is visually lost and is programmatically on the `<body>` element. Note that pressing Tab moves the keyboard focus to the next in-page section link, which is appropriate.



Page 8: desktop, when "Why join Unilever?" link is activated, the page scrolls visually to the appropriate section but keyboard focus is not placed in this section

```
X document.activeElement  
body
```

Page 8: Console, document.activeElement query shows that focus is on the body

```
<div class="Vertical">  
  <b><a href="#blue_img" title="jump to Why join Unilever section" class="text--white digital_content__anchor">  
    Why join Unilever?  
  </a></b>  
</div>
```

Page 8: DOM, "Why join Unilever?" link

Steps to reproduce

1. Observe the same page links.
2. Navigate onto the links with a keyboard using Tab.
3. Attempt to activate the links with Return or Space.
4. Observe that keyboard focus does not move.

Recommendation

When receiving keyboard focus, make sure that components respond to keyboard commands. In this case, when the same page links are activated, make sure that focus is moved to the relevant part of the page. This can be done with same-page links or using the **focus** method.

As well, use **tabindex="-1"** to allow elements to receive programmatic focus when they are not naturally in the focus order, such as headings. That is, when you are not trying to set focus onto buttons or links, which should be in the focus order already. For example:

```
<ul>
  <li><a href="#SecA">Jump to Terms Starting with A</a></li>
  ...
</ul>
<h2 id="SecA" tabindex="-1">Terms Starting with A</h2>
<dl>
  <dt>ACME</dt>
  <dd>The company name</dd>
  <dt>Aircrafts</dt>
  <dd>What the company produces</dd>
</dl>
```

For examples, see [MDN - focus method](#) and [Dev - Managing Focus for Accessibility](#). For more information, see [MDN - Keyboard Accessibility](#), [Google Developer - Using Tabindex](#), [Yale - Web Accessibility - Links](#), [University of Washington - Links and buttons](#), and [A11y Project - Valid and Accessible Links](#).

[Return to priority table](#)

ADR_20590-94 MEDIUM PRIORITY OPEN

Interactive element cannot be activated with keyboard

Success Criteria 2.1.1: Keyboard [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12, 13

Users Affected Motor

Component Button

What happened

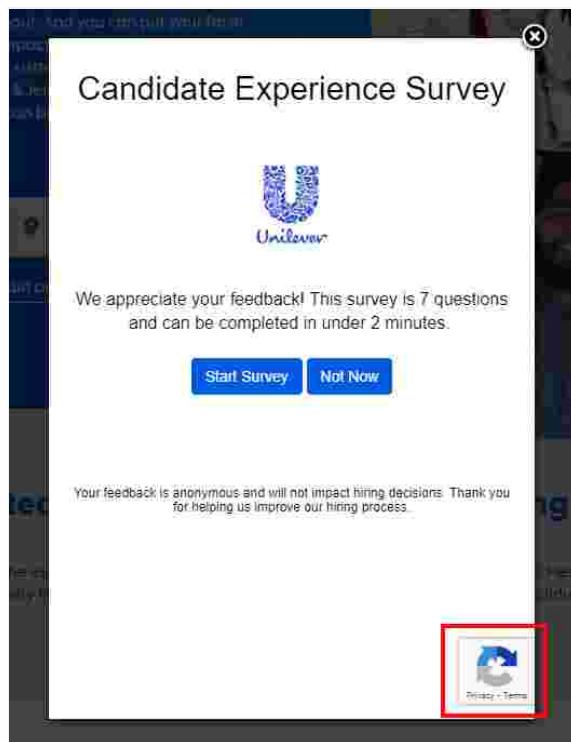
Interactive elements must be entirely compatible with a keyboard.

Users should never need to use a mouse. Otherwise, screen reader, keyboard, and switch users, may be unable to control the interactive element. There are interactive elements that receive keyboard focus, but which cannot be activated with a keyboard. As a result, they cannot be operated with a keyboard.

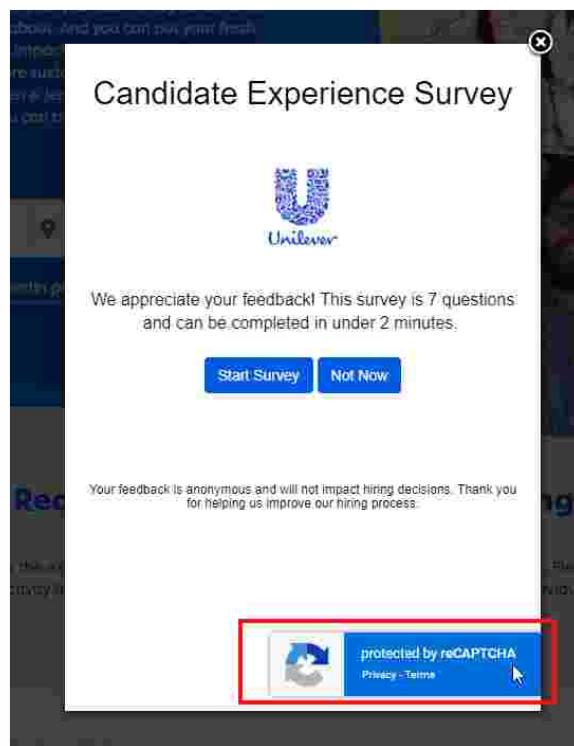
This has been raised as a medium instead of a high priority issue because it does not block progress.

Where it happened

This can be seen on page 12 on desktop and iOS with the “protect by reCAPTCHA” element. On mouse hover, this element expands to disclose the text “protected by reCAPTCHA”. However, this content cannot be accessed with the keyboard or screen reader.



Page 12: desktop, collapsed “protected by reCAPTCHA” element



Page 12: desktop, "protected by reCAPTCHA" element expanded on mouse hover

```
<div class="rc-anchor-logo-large" role="presentation">
  <div class="rc-anchor-logo-img rc-anchor-logo-img-large"></div>
</div>
<div class="rc-anchor-pt">
  <a href="https://www.google.com/intl/en/policies/privacy/" target="_blank">Privacy</a>
  <span aria-hidden="true" role="presentation"> - </span>
  <a href="https://www.google.com/intl/en/policies/terms/" target="_blank">Terms</a>
</div>
<div class="rc-anchor-invisible-text" flex>
  <span>
    "protected by "
    <strong>reCAPTCHA</strong>
  </span>
  <div class="rc-anchor-pt">
    <a href="https://www.google.com/intl/en/policies/privacy/" target="_blank" style>Privacy</a>
    <span aria-hidden="true" role="presentation"> - </span>
    <a href="https://www.google.com/intl/en/policies/terms/" target="_blank" style>Terms</a>
  </div>
</div>
</div>
```

Page 12: DOM, "protected by reCAPTCHA" element

This can also be seen on page 13 on desktop and iOS with the visual labels for the first, third, and fifth ratings buttons. On mouse hover, a label such as "Somewhat important" appears that explains the meaning of the rating level. However, the labels cannot be triggered with a keyboard or mobile screen reader.

Note that the second and fourth ratings buttons do not have labels that appear on hover.

Note that the ratings buttons can be activated with a keyboard or screen reader to submit a rating.

How important is it that a company disclose its workforce demographics and progress towards its workforce diversity and equity goals on its career site?

Please rate on scale below.



Somewhat important

N/A

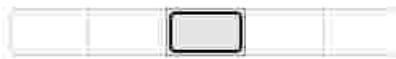
Next ➔



Page 13: "Somewhat important" text for third ratings button appears on hover

How important is it that a company disclose its workforce demographics and progress towards its workforce diversity and equity goals on its career site?

Please rate on scale below



(Scale Low to High)

Not rated

N/A

Next ➔



Page 13: text does not appear when third ratings button has keyboard focus

```
<!-- end ngRepeat: label in q.hilo.labels -->
<button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-width" id="border-radius-first" label.firstButton,
        'border-radius-last':label.lastButton" ng-style="{'background-color':label.bgColor}" style=
<!-- end ngRepeat: label in q.hilo.labels -->
<button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-width" id="border-radius-first" label.firstButton,
        'border-radius-last':label.lastButton" ng-style="{'background-color':label.bgColor}" style=
<!-- end ngRepeat: label in q.hilo.labels -->
<button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button ng-scope three-five-labels-width border-radius-first" label.firstButton,
        'border-radius-last':label.lastButton" ng-style="{'background-color':label.bgColor}" style=
<!-- end ngRepeat: label in q.hilo.labels -->
</div>
</div>
<p class="font-size-10 ng-binding" style="font-style: italic">(Scale low to High)</p>


Somewhat important


</div>
```

Page 13: DOM, “Somewhat important” text

Steps to reproduce

1. Observe the interactive elements.
 2. Navigate onto it with a keyboard using Tab.
 3. Attempt to activate the element with Return or Space.
 4. Observe that the element does not respond.
 5. Observe that there is not an alternative way to access the same information or functionality.

Recommendation

When receiving keyboard focus, make sure that components respond to keyboard commands. For example, make sure that they can be tabbed onto and activated with Return or Space. Alternatively, make sure that they can be adjusted with arrow keys, typed into, or whatever is appropriate for the component.

If you cannot make the component compatible with a keyboard, provide an alternative that gives users access to the same functionality. For example, a date-entry input field can be provided as an alternative to a date picker component.

For more information, see [MDN - Keyboard Accessibility](#), [Keyboard-navigable JavaScript widgets](#), and [WAI ARIA - Developing a Keyboard Interface](#). For practical examples, see [WAI ARIA Practices](#) and [Inclusive Components](#).

[Return to priority table](#)

ADR_20590-45 MEDIUM PRIORITY OPEN

Mechanism for pausing automatically moving content does not work with keyboard

Success Criteria 2.2.2: Pause, Stop, Hide [2.0 A]

Devices Desktop

Pages Affected 03

Users Affected Cognitive

Component Video

What happened

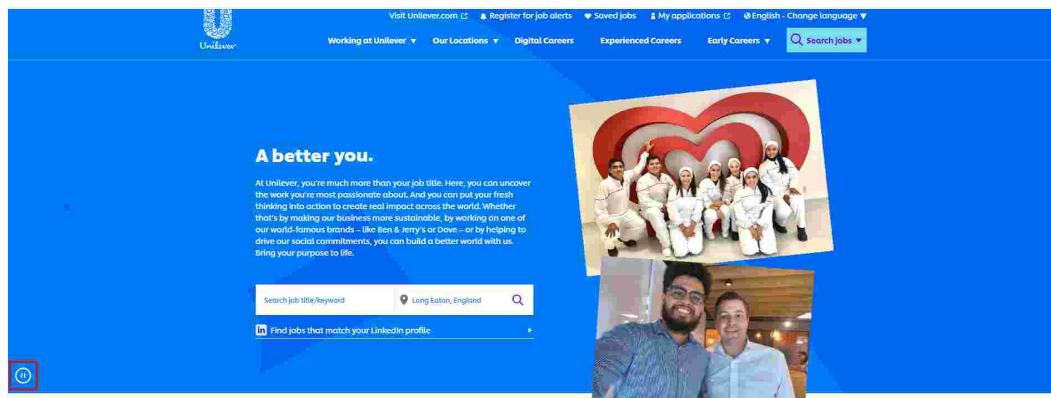
Users must be able to pause, stop, or hide, any content that automatically moves, scrolls, or blinks.

There is automatically moving content that cannot be controlled. This may disadvantage several user groups. For example, the content may move faster than screen readers can announce it, keyboard users may struggle to navigate through the content, and it may be cognitively overloading for various users.

There is a mechanism that seems like it should control the movement. However, the mechanism does not work with a keyboard.

Where it happened

This can be seen on page 3, on desktop, with the auto-playing hero banner video. There is a pause button on the page, which can be activated with a mouse pointer. However, this button does not receive keyboard focus. As a result, there is no pause mechanism provided to pause the video when using keyboard.



Page 3: desktop, video pause button

```
<div style="position: absolute; z-index: -1; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-position: 50% 0%; background-image: none;">
  <video loop muted style="margin: auto; position: absolute; z-index: -1; top: 0%; left: 50%; transform: translate(-50%, 0%); visibility: visible; opacity: 1; width: 1899px; height: auto;"></video>
</div>
<div class="play-pause-button" paused="false"></div>
```

Page 3: DOM, video pause button

Steps to reproduce

1. Navigate onto the above.
2. Observe that the content automatically moves.
3. Observe that there is not a mechanism to control the movement with a keyboard.

Recommendation

Where possible, avoid anything that moves automatically. Where this is not possible, provide a mechanism to control the movement. It is not enough to pause the movement when the content receives mouse hover or keyboard focus. The user must be able to control the movement and navigate the rest of the page.

The most reliable way to pass this success criterion is by providing a play / pause button. For more information, see [WAI Tutorials - Carousel Concepts](#), [Inclusive Components - Content Slider](#), and [CSS Tricks - Accessible Web Animation](#).

As well, where possible, support the reduced motion media query. For more information, see [MDN - Reduced Motion](#) and [CSS Tricks - Reduced Motion Media Query](#).

[Return to priority table](#)

ADR_20590-49 MEDIUM PRIORITY OPEN

Automatically moving content cannot be paused

Success Criteria 2.2.2: Pause, Stop, Hide [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07

Users Affected Cognitive

Component Plain Text

What happened

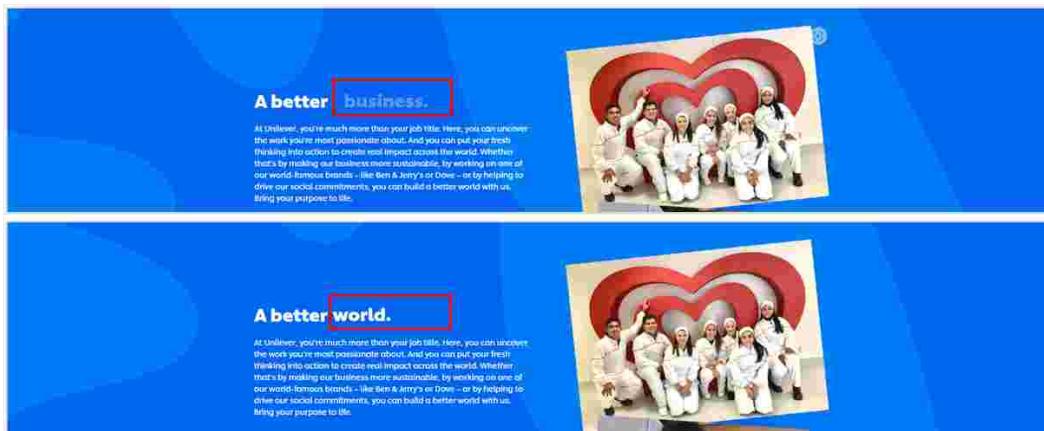
Users must be able to pause, stop, or hide, any content that automatically moves, scrolls, or blinks.

There is automatically moving content that cannot be controlled. This may disadvantage several user groups. For example, the content may move faster than screen readers can announce it, keyboard users may struggle to navigate through the content, and it may be cognitively overloading for various users.

Where it happened

This can be seen on pages 3 and 7, on desktop and iOS, with the automatically moving text over the hero banner of the page. The static text “A better” is followed by moving text which changes between “business.”, “world.” and “you.”. This moving content cannot be paused, stopped or hidden.

Note that, although there is a pause/play button available to pause the moving hero banner background, this does not pause the moving text.



Page 3: desktop, example of moving text in the hero banner

Steps to reproduce

1. Navigate onto the above.
2. Observe that the content automatically moves.
3. Observe that there is not a mechanism to control the movement.

Recommendation

Where possible, avoid anything that moves automatically. Where this is not possible, provide a mechanism to control the movement. It is not enough to pause the movement when the content receives mouse hover or keyboard focus. The user must be able to control the movement and navigate the rest of the page.

The most reliable way to pass this success criterion is by providing a play / pause button. For more information, see [WAI Tutorials - Carousel Concepts, Inclusive Components - Content Slider](#), and [CSS Tricks - Accessible Web Animation](#).

As well, where possible, support the reduced motion media query. For more information, see [MDN - Reduced Motion](#) and [CSS Tricks - Reduced Motion Media Query](#).

[Return to priority table](#)

ADR_20590-18 MEDIUM PRIORITY OPEN

Navigation skip link does not move keyboard focus

Success Criteria 2.4.1: Bypass Blocks [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01

Users Affected Motor

Component Link

What happened

There must be a mechanism for users to bypass repeated blocks of content, such as navigation menus.

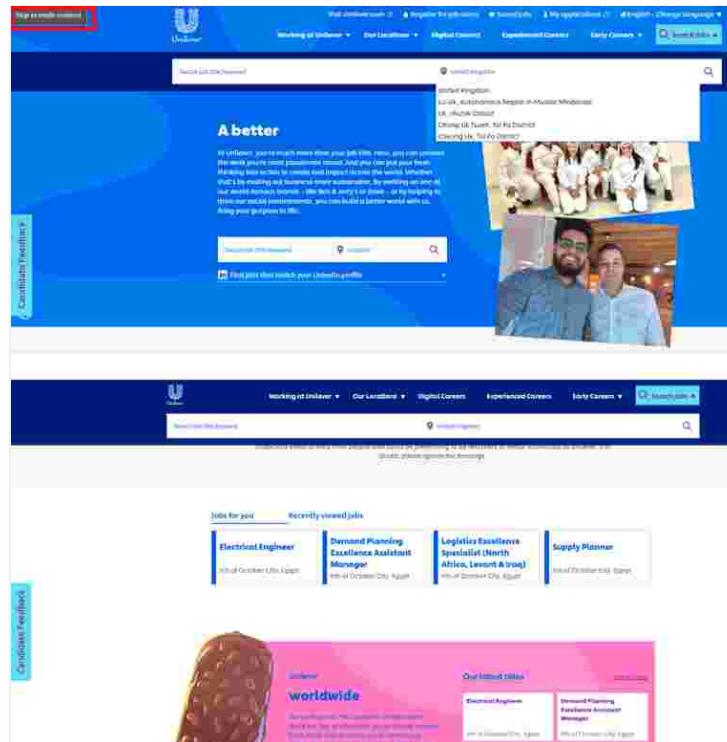
There are skip links. However, after activating them with a keyboard, keyboard focus does not bypass the content and move to the target. It remains on the skip link. This may particularly disadvantage sighted keyboard users. The inoperable skip link may make navigating the pages much more time consuming.

Where it happened

This issue can be seen on page 1 on desktop and iOS with the “Skip to main content” link. When activated with a keyboard on desktop, the link scrolls the page visually but focus is placed on the `<body>` element. On iOS, when the skip to main content link is activated with a screen reader, the page scrolls visually, but the screen reader cursor is placed on the “Unilever” logo at the top of the page.

Note that the destination of the “Skip to main content link” is the `<main>` landmark. However, focus is not placed on this element when the link is activated.

Note that, on desktop, when Tab is pressed after activating the skip link with a keyboard, the focus then moves appropriately to the “Jobs for you” tab.



Page 1: desktop, “Skip to main content” link scrolls the page visually

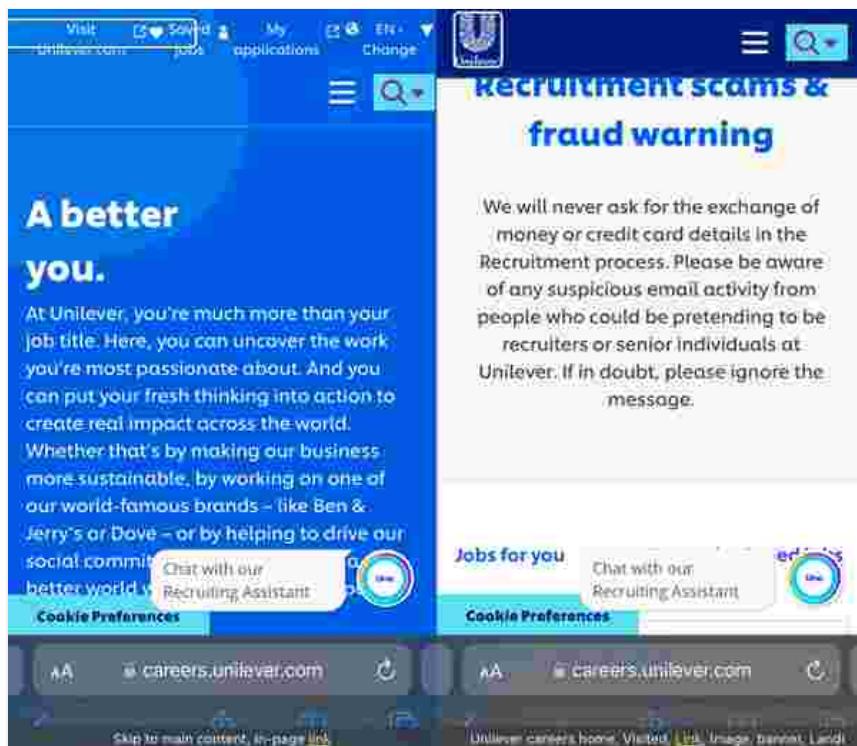


Page 1: Console, document.activeElement query shows focus is on the body

A screenshot of a browser's DOM inspector. A specific link in the header is highlighted with a red box. The link is located within a header element and has the href attribute set to "#content". The text of the link is "Skip to main content".

```
<div id="page" style="padding-top: 72.794px;">
  <a id="skip-nav" class="access-link" href="#content">Skip to main content</a>
  <div class="header-background header-background--video header-background--video--core" data-video-bg="mp4:https://tbcdn.talentbrew.com/company/34155/v1/img/video-poster-99.jpg" data-video-options="posterType: jpg, loop: true, muted: true, position: 50% 0%, autoplay: false" style="position: relative;">
    <div style="position: absolute; z-index: 1; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-position: 50% 0%; background-image: none;"></div>
    <div class="play-pause-button" paused="false"></div>
  </div>
  <header class="core-header sticky">
    <div class="max-width-wrapper">
      <div class="core-header__left">
        <a href="/" class="header-logo">
          
        </a>
      </div>
      <div class="core-header__right">...</div>
    </div>
  </header>
  <div class="hero-content max-width-wrapper max-width-wrapper--narrow">...</div>
</div>
<main id="content">...</main>
<footer>...</footer>
```

Page 1: DOM, "Skip to main content" link is an in-page link with the main landmark as its destination



Page 1: iOS, screen reader cursor is placed on Unilever logo when "Skip to main content" link is activated

Steps to reproduce

1. Navigate around the repeated block of content with the keyboard or a screen reader.
2. Observe the skip link.
3. Activate the skip link with keyboard or a screen reader.
4. Observe that the keyboard focus or screen reader cursor has not been correctly updated.

Recommendation

Make sure that skip links send keyboard focus to the appropriate place. For example, navigation skip links should send focus to the `<main>` container or the first element in the DOM after the navigation menu, such as a heading element. For example:

```
<body>
  <a href="#main">Skip to main content</a>
  <nav role="navigation">
    <ul>
      <li><a href="...">Home</a></li>
      <li><a href="...">About</a></li>
    </ul>
  </nav>
  <main id="main">
  </main>
</body>
```

Skip links can be hidden by default. This can be achieved with a CSS class that visually hides them, but keeps them present in the DOM. However, they must become visible when receiving keyboard focus. This can be achieved using the CSS **focus** pseudoclass.

For more information on skip links, see [a11y Project - Skip Links](#), [WCAG - G1 Adding a link at the top of each page](#) and [Axess Labs - Your Skip Links are Broken](#).

[Return to priority table](#)

ADR_20590-22 MEDIUM PRIORITY OPEN

Background elements receive focus

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12

Users Affected Vision, Motor

Component Modal Dialog

What happened

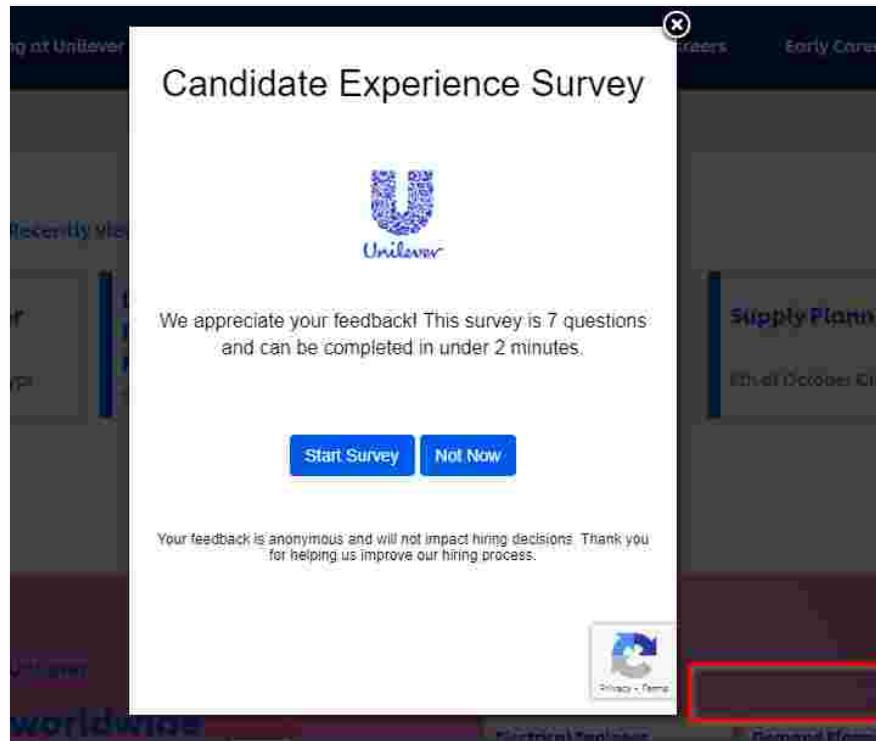
The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are modal dialogs where the background elements can still be accessed with a keyboard. This may particularly disadvantage keyboard users. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 12 on desktop and iOS with “Candidate Experience Survey” modal. When navigating the modal with Tab, background elements can still be accessed with a keyboard when tabbing past the “Close” button.

Similarly on iOS, when swiping through the modal, background elements can still receive screen reader cursor.



Page 12: desktop, background elements receive focus

Steps to reproduce

1. Using just your keyboard, activate the modal dialog.
2. Observe that the background is visually represented as inoperable.
3. Use Tab to navigate around the page.
4. Observe that you can still access the background with a keyboard.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in meaningful order.

In this case, when the modal dialog is open, make sure that focus is set onto the modal dialog. As well, make sure that keyboard focus is restricted to the dialog. This can be achieved with the following.

- Send the focus to the first element on the modal container. This can be done using the JavaScript `.focus()` method.
- Hide all of the other content of the page with `aria-hidden="true"`.
- Remove the interactive elements in the background from the focus order with `tabindex="-1"`.
- Revert the above when the modal is closed. Removing the `aria-hidden` and `tabindex` properties.
- When the dialog is dismissed, use the `.focus()` method to set focus back onto the element that opened it.

The `aria-hidden` and `tabindex` can be supplemented with an inert polyfill. For examples, see [GitHub - WICG - Inert](#), and [inert-polyfill](#).

For more information, see [WAI ARIA - Dialog Example](#), [MDN - aria-hidden](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - DOM Order Matters](#), and [Using Tabindex](#).

[Return to priority table](#)

ADR_20590-56 MEDIUM PRIORITY OPEN

Non-interactive elements receive keyboard focus

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop

Pages Affected 07

Users Affected Motor

Component Plain Text, Heading

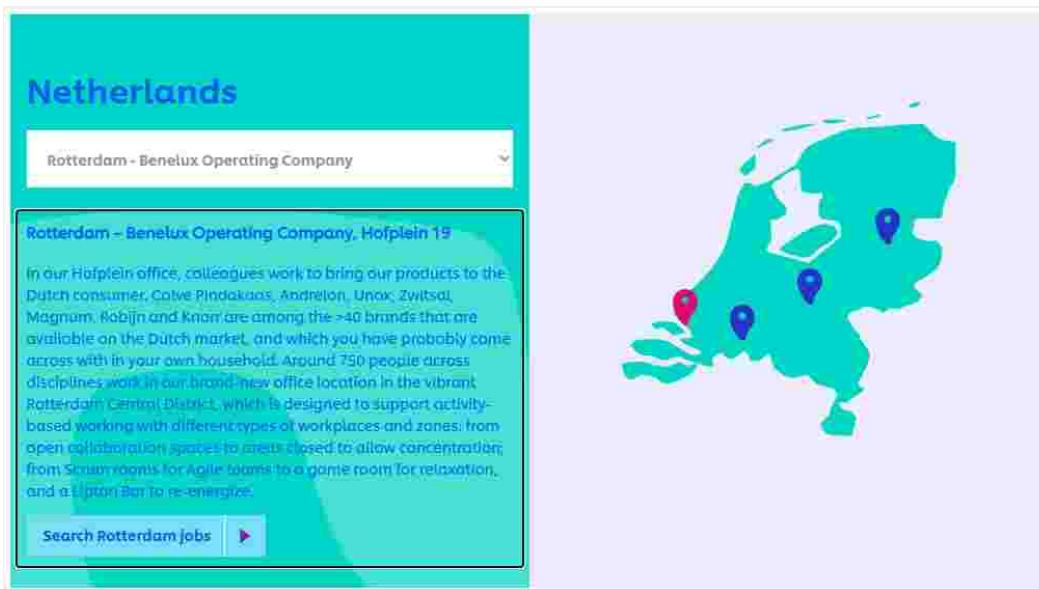
What happened

The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are elements that receive focus that are not interactive. This is illogical as only interactive elements should receive focus. This may particularly disadvantage keyboard users. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 7, on desktop, with the static container which contains the informative text and “Search [location] jobs” link for the location selected under the “Netherlands” section. After selecting a location, the static container becomes focusable due to the use of a `tabindex="0"` attribute.



Page 7: desktop, static container receiving keyboard focus

```

<div class="standalone-map_left"> ...
    <h2 class="standalone-map_header">Netherlands</h2>
    <select class="standalone-map_select" name="Our UK locations" id="uk-locations">@</select>
    <div class="standalone-map_locationn_copy">
        <div class="standalone-map_location_copy_child" data-copy="netherlands1" tabindex="-1">@</div>
        <div class="standalone-map_location_copy_child active" data-copy="netherlands1b" tabindex="0">
            <h3>Rotterdam - Benelux Operating Company, Hofplein 19</h3>
            <p>
                In our Hofplein office, colleagues work to bring our products to the Dutch consumer. Calve Pindakaas, Andrelon, Unox, Zwitsal, Magnum, Robijn and Knorr are among the >40 brands that are available on the Dutch market, and which you have probably come across with in your own household. Around 750 people across disciplines work in our brand-new office location in the vibrant Rotterdam Central District, which is designed to support activity-based working with different types of workplaces and zones: from open collaboration spaces to areas closed to allow concentration; from Scrum rooms for Agile teams to a game room for relaxation, and a Lipton Bar to re-energize.
            </p>
            <p>
                <a href="/search-jobs/Rotterdam%20South%20Holland/34155/A/2750405-2745638-2747830-2747891/51x9225/4x47917/0/2" class="button-with-arrow button-with-arrow-teal" data-custom-event="true" data-custom-category="Custom" data-custom-label="Map - Rotterdam Benelux Operating Company - Clicked">@</a>
            </p>
        </div>
        <div class="standalone-map_location_copy_child" data-copy="netherlands2" tabindex="-1">@</div>
        <div class="standalone-map_location_copy_child" data-copy="netherlands3" tabindex="-1">@</div>
        <div class="standalone-map_location_copy_child" data-copy="netherlands4" tabindex="-1">@</div>
    </div>
</div>

```

Page 7: DOM, static container has tabindex="0"

Steps to reproduce

1. Using just your keyboard, tab through the above.
2. Observe that focus goes onto elements that are not interactive.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in meaningful order.

In this case, make sure that the non-interactive elements do not receive keyboard focus. For example, make sure to avoid using **tabindex="0"** on elements that should not receive focus. If you need the non-interactive elements to be capable of receiving focus through the **focus** method, use **tabindex="-1"**. For example:

```
<h1 tabindex="-1">Main Heading</h1>
```

Using **tabindex="-1"** allows elements to be used as a target for the **focus** method without adding them to the focus order. For more information, see [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#) and [Google Developer - Using Tabindex](#).

[Return to priority table](#)

ADR_20590-57 MEDIUM PRIORITY OPEN

Focus is not managed appropriately when page is dynamically updated

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 04

Users Affected Motor

Component Link, Button

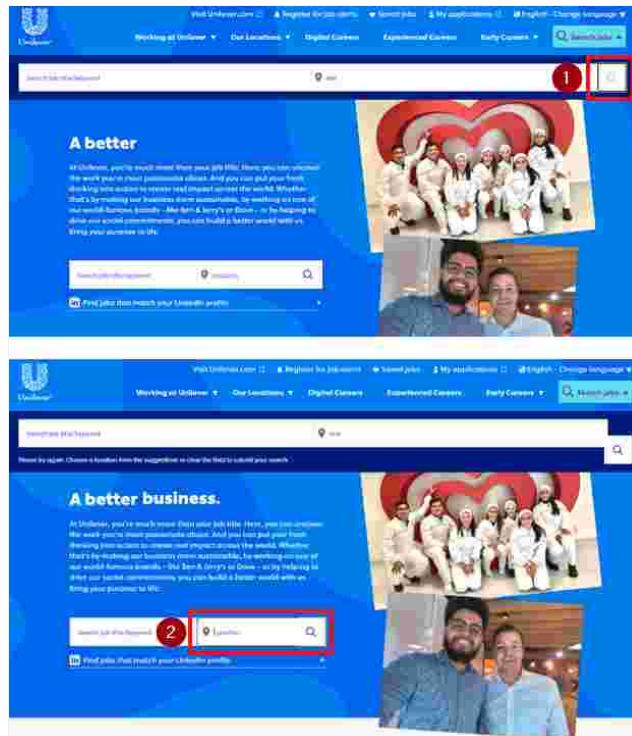
What happened

The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are instances where the focus order is not managed when the page is dynamically updated. This may particularly disadvantage keyboard users who often navigate by tabbing through the interactive elements. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 1 on desktop and iOS when the search button is activated and the “location” field has invalid input. The keyboard focus or screen reader cursor is placed automatically on the “location” input field in the main content, rather than the “location” input field in the header where the error text displays.



Page 1: desktop, when search is submitted with keyboard and location is invalid, keyboard focus is placed on location field in main content instead of in header

```

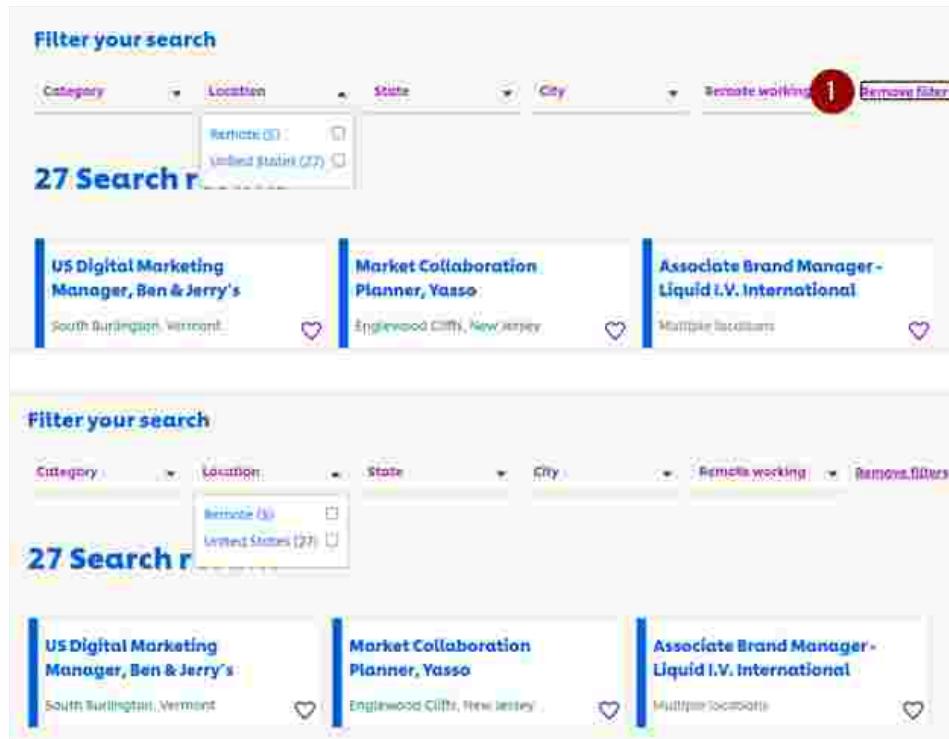
<form>
  <div>
    <label for="search-keyword-a5f2ef227c">Search job title/keywords</label>
    <input type="search" id="search-keyword-a5f2ef227c" class="search-keyword" placeholder="Search job title/keyword" autocomplete="off" name="k" value="" maxlength="150">
    <div class="auto-complete-keyword-popup" tabindex="-1" id="search-keyword-a5f2ef227c-auto-complete-popup" style="position: absolute; z-index: 1;></div>
  </div>
  <div>
    <label for="search-location-a5f2ef227c">Location</label>
    <input type="text" id="search-location-a5f2ef227c" class="search-location mindreader-results-open" placeholder="Location" autocomplete="off" name="l" value="" data-lat="150" data-lon="150" data-kt="data-lt data-kt data-lp" data-ajax-url="/search-jobs/locations?term=" data-country-codes="" data-display-typeahead="true" data-error-message="Please try again. Choose a location from the suggestions or clear the field to submit your search." data-initial-aria-describedby="search-error-1" aria-invalid="false" data-pc-data-current-val="data-gtm-form-interact-field-id="4" style="outline: none; position: absolute; top: 0px; left: 0px; width: 970px; height: 55px; z-index: 0; opacity: 0.5;">
    <input type="text" value="Testour, Gouverneur de Béja" class="search-location display-typeahead" readonly="readonly" style="position: absolute; top: 0px; left: 0px; width: 970px; height: 55px; z-index: 0; opacity: 0.5;">
  </div>
  <div>
    <label for="search-radius-a5f2ef227c">Search radius</label>
    <select class="search-radius" id="search-radius-a5f2ef227c" name="r" disabled="disabled"></select>
  </div>
  <input id="search-org-ids-a5f2ef227c" name="orgIds" type="hidden" value="34155">
  <button id="search-submit-a5f2ef227c" onclick="APP.MODELS.GoogleBot.sendCustomDimensions('Job Search', 'Click', 'Supply Chain - Search Form - Click s', event)">
    <span class="visually-hidden">Search</span>
  </button>
  <p class="search-location-error" aria-hidden="true" id="search-error-1" style="outline: none; display: none;"></p>
</div>
</form>

```

Page 1: DOM, location input field and search submission button

This can also be seen on page 4 on desktop and iOS when the “Remove filters” button is activated. On desktop, the keyboard focus is visually lost and is placed on the **<body>** element. On iOS, the screen reader cursor is placed on an arbitrary element.

Note that, on desktop, if the “Remove filters” button is activated and then Tab is pressed, keyboard focus moves onto the “Category” filter triggering button, which is logical.



Page 4: desktop, keyboard focus is visually lost when “Remove filters” button is activated

```

X document.activeElement
body#search_magic-bullet-all

```

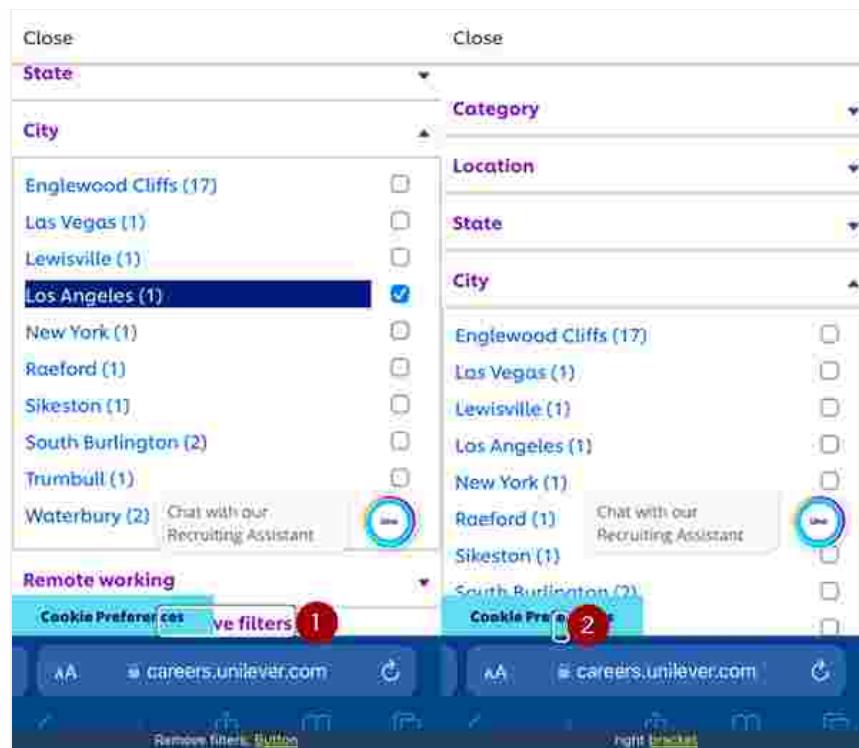
Page 4: Console, document.activeElement query shows that focus is on the body

```

<div data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="5" data-expandable-id="qjjpk3kmnf" style="display: none;" class="expandable"></div>
<div id="search-filter-controls">
  <button type="button" id="search-filter-clear">Remove filters</button>
</div>
</div>
</section>

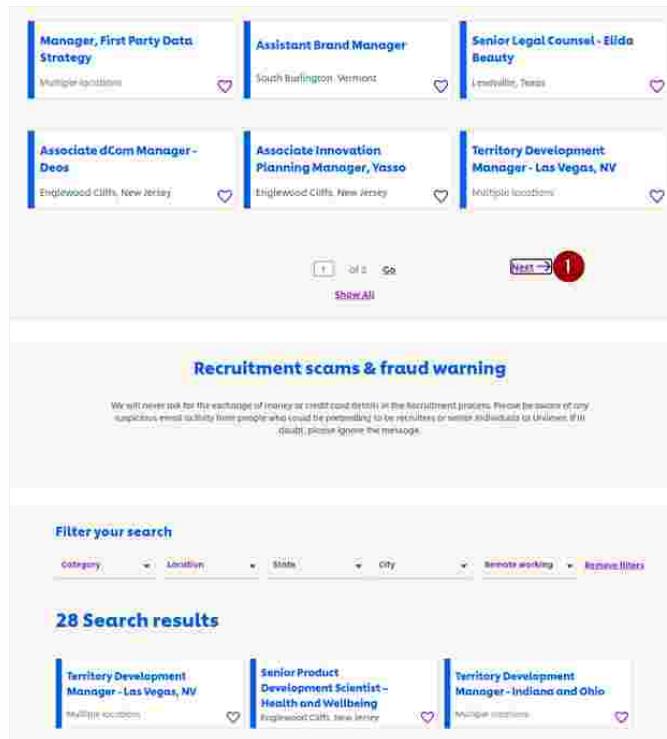
```

Page 4: DOM, remove filters button



Page 4: iOS, screen reader cursor is placed on right bracket for "Trumbull" filter when "Remove filters" button is activated

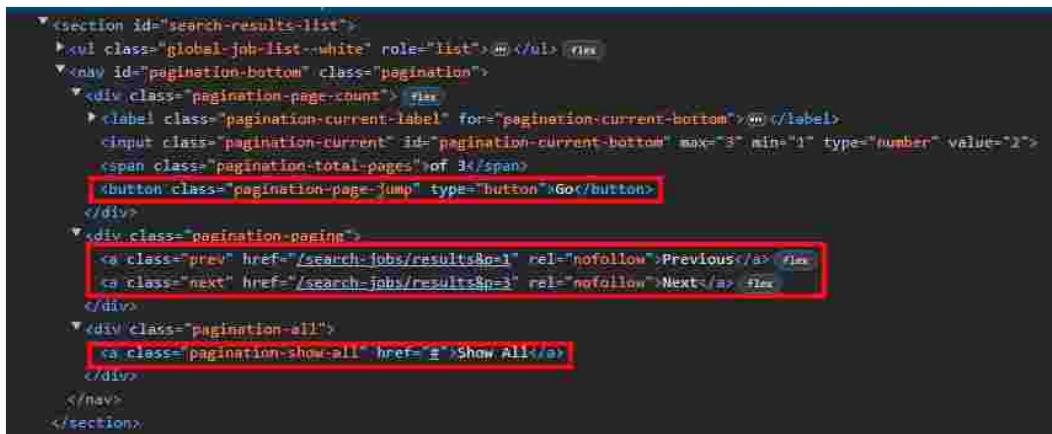
This can also be seen on page 4 on desktop and iOS with the pagination links and buttons ("Previous", "Go", "Next", and "Show all"). On desktop, when one of these elements is activated with the keyboard, focus is visually lost and is placed on the `<body>` element. On iOS, the behaviour is inconsistent but the screen reader cursor either remains on the activated element or is placed on an arbitrary element.



Page 4: desktop, keyboard focus is visually lost when "Next" button is activated

```
X: document.activeElement  
body@search.magic>bullet-ally
```

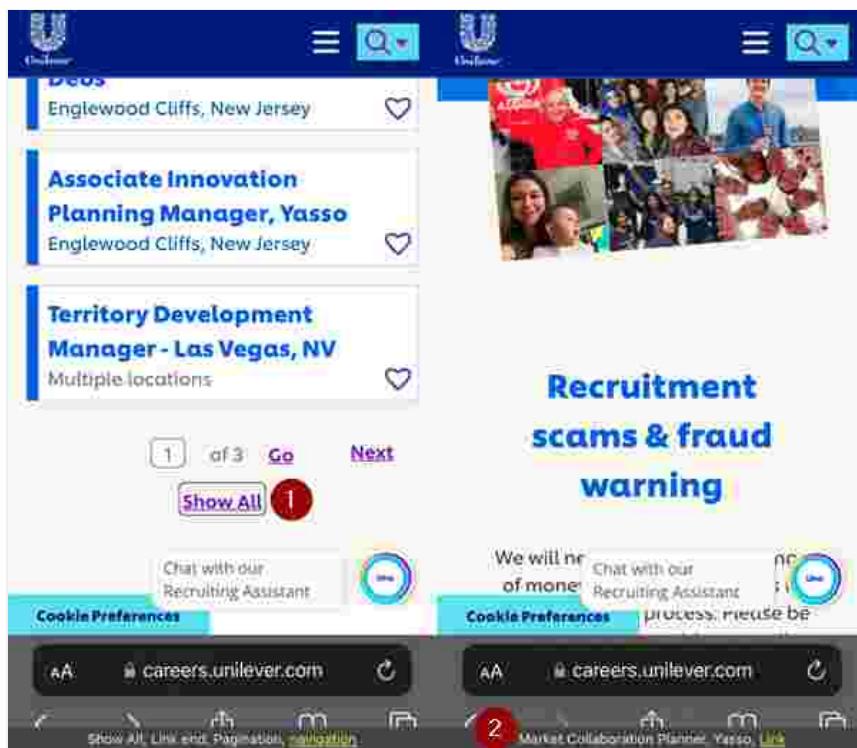
Page 4: Console, document.activeElement query shows that focus is on the body



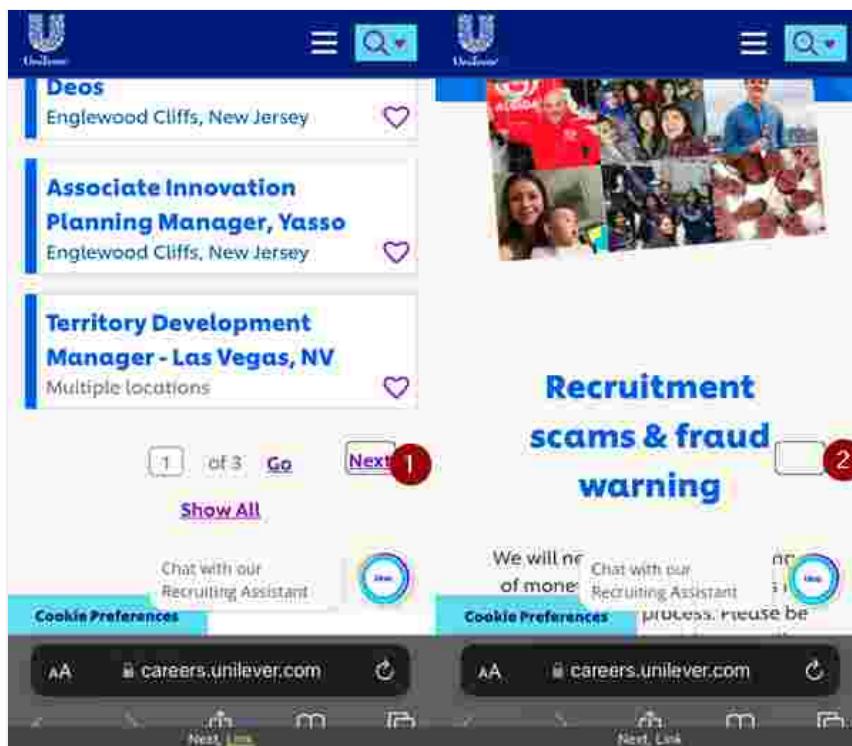
The screenshot displays the DOM structure of a page section. Several elements are highlighted with red boxes:

- A button labeled "pagination-page-jump" with the value "Go".
- A "prev" link with the href "/search-jobs/results&p=1" and rel="nofollow".
- A "next" link with the href "/search-jobs/results&p=3" and rel="nofollow".
- A "Show All" link with the href "#".

Page 4: DOM, the “Previous”, “Go”, “Next”, and “Show all” links or buttons



Page 4: iOS, screen reader cursor is placed on “Market Collaboration Planner, Yasso” link when “Show all” button is activated



Page 4: iOS, screen reader cursor remains on "Next" button when "Next" button is activated

Steps to reproduce

1. Observe that the pages are dynamically updated.
2. Using just your keyboard or a screen reader, navigate through the above.
3. Activate the dynamic functionality.
4. Continue to navigate through the page.
5. Observe that the keyboard focus and screen reader cursor have not been appropriately managed.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in meaningful order.

As well, make sure that focus is managed appropriately. Use the focus method sparingly when the page has a meaningful update. As well, use `tabindex="-1"` to allow elements to receive programmatic focus when they are not naturally in the focus order, such as headings.

When the entire page changes due to SPA functionality, make sure that the focus is managed. For example, when the user selects some "Continue" or "Next" button and the page is updated, set focus on to the heading of the brand new content.

```
<h2 tabindex="-1">SPA Page Heading</h2>
<p>When the below button is pressed, the page updates with new content.</p>
<p>The focus method sets focus on to the new heading.</p>
<button>Next Page</button>
```

For examples, see [MDN - focus method](#) and [Dev - Managing Focus for Accessibility](#).

For more information, see [WCAG - G59 Placing the interactive elements in an order](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - DOM Order Matters](#), and [Using Tabindex](#).

[Return to priority table](#)

ADR_20590-58 MEDIUM PRIORITY OPEN

Focus order includes off-screen interactive elements

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 02, 07

Users Affected Motor

Component Button

What happened

The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

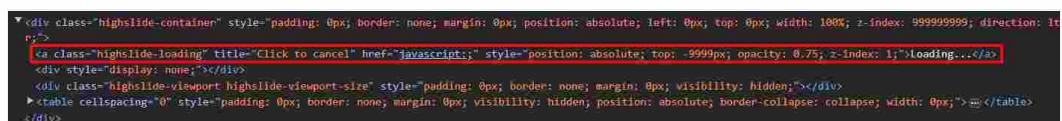
There are off-screen interactive elements that receive focus. This may particularly disadvantage keyboard users who often navigate by tabbing through the interactive elements. That invisible interactive elements receive focus may confuse or frustrate users.

Where it happened

This can be seen on page 2 on desktop and iOS with an off-screen “Loading...” link. This link receives keyboard focus and can be navigated to with a screen reader.



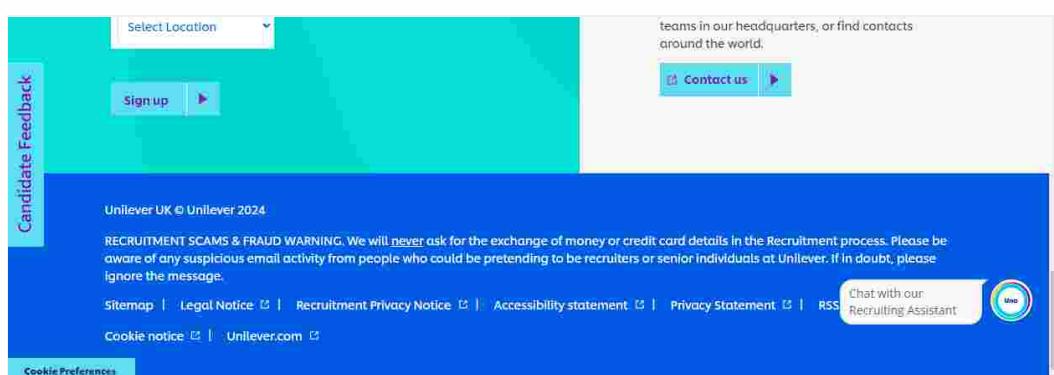
Page 2: desktop, keyboard focus is on off-screen “Loading...” link



Page 2: DOM, off-screen “Loading...” link is focusable

This can also be seen on page 2 on desktop and iOS with the close button for the “Candidate Feedback” modal when it is off screen. The button receives keyboard focus and can be navigated to with a screen reader.

Note that this only occurs after the candidate survey has been opened and then closed.



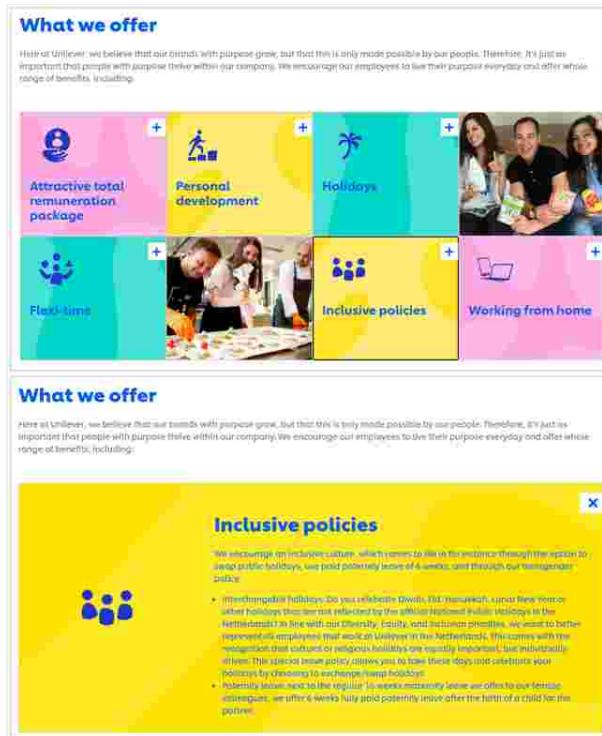
Page 2: desktop, keyboard focus is on off-screen close button

```
<div id="hsId1" style="padding: 0px; border: none; margin: 0px; position: absolute; visibility: visible; width: auto; direction: ltr; opacity: 1; right: 0px; height: auto; top: 0px; z-index: 4;">
    <div style="display: block;">
        <button id="hs-close-button" class="closebutton" title="Close" style="border: none; height: 30px; ">&nbsp;</button>
    </div>
</div>
```

Page 2: DOM, off-screen close button is focusable

This can be seen on page 7, on desktop and iOS, when the “Inclusive policies” panel is expanded. The panel can be expanded by selecting the “Inclusive policies” tile under the “What we offer” section. When expanded, the panel appears over the tiles in this section. However, it is still possible to tab through the off-screen tiles, as well as navigate through them with a screen reader.

Note that only the “Inclusive policies” panel was in scope for this audit. However, this behaviour is likely to also affect the other panels.



Page 7: desktop, “Inclusive policies” panel expands over tiles

```

<div class="interactive-benefits-grid_grid">(flex) +-->
  <div class="interactive-benefits-grid-child one pink">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">one</button> flex
      <div class="interactive-benefits-grid-child_overlay">one</div>
    </div>
  </div>
  <div class="interactive-benefits-grid-child two yellow">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">two</button> flex
      <div class="interactive-benefits-grid-child_overlay">two</div>
    </div>
  </div>
  <div class="interactive-benefits-grid-child three teal">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">three</button> flex
      <div class="interactive-benefits-grid-child_overlay">three</div>
    </div>
  </div>
  <div class="interactive-benefits-grid-child four empty gray">four</div>
  <div class="interactive-benefits-grid-child five teal">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">five</button> flex
      <div class="interactive-benefits-grid-child_overlay">five</div>
    </div>
  </div>
  <div class="interactive-benefits-grid-child six empty gray">six</div>
  <div class="interactive-benefits-grid-child seven yellow">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">seven</button> flex
      <div class="interactive-benefits-grid-child_overlay" style="display: block;">seven</div>
    </div>
  </div>
  <div class="interactive-benefits-grid-child eight pink">
    <div class="interactive-benefits-grid-child_inner">
      <button class="interactive-benefits-grid-child_trigger">eight</button> flex
      <div class="interactive-benefits-grid-child_overlay">eight</div>
    </div>
  </div>
</div>

```

Page 7: DOM, tile buttons are not removed from the focus order when panel is expanded

Steps to reproduce

1. Using just your keyboard, tab through the above.
2. Observe that focus goes onto elements that are not visible on the page.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in meaningful order.

In this case, make sure that off-screen elements do not receive keyboard focus.

If they are not meant to be interactive at any point, remove the elements from the DOM, comment them out, or remove them from the focus order. If they are meant to be interactive at some point, make sure that they are excluded from the focus order and included in the focus order as appropriate.

Removing elements from the focus order can be achieved with the HTML `hidden` or CSS `display:none`. Both will hide the element from all users. This same can be programmatically achieved by using both `aria-hidden="true"` and `tabindex="-1"` on the same element. For example:

```

<button hidden> ... </button>
<button class="visuallyOffScreen" aria-hidden="true" tabindex="-1"> ... </button>

```

However, note that `aria-hidden="true"` and `tabindex="-1"` will not visually remove elements from the page.

Removing these attributes will re-insert the elements into the focus order based on their location in the DOM, allowing users to navigate onto them with a keyboard. Make sure that this is done as appropriate, such as when the user toggles a parent control that shows and hides these interactive elements.

For more information, see [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - Using Tabindex](#), [MDN - aria-hidden](#), and [MDN - hidden](#).

[Return to priority table](#)

ADR_20590-59 MEDIUM PRIORITY OPEN

Focus does not return to triggering element

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 02, 07

Users Affected Motor

Component Button

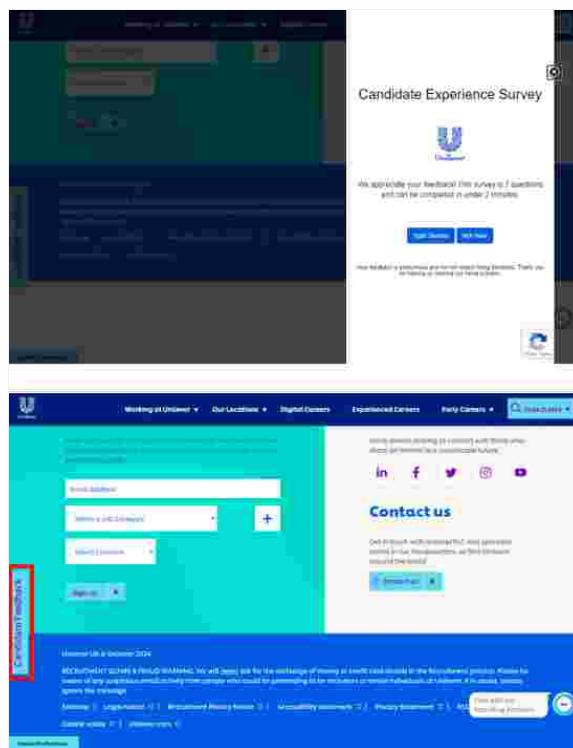
What happened

The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are elements that control new content being opened and closed. When the new content is closed, the focus does not return to the element that triggered the content. This may particularly disadvantage keyboard users. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 2 on desktop and iOS with the “Candidate Feedback” modal triggering button. When the survey modal is closed with the keyboard or screen reader, focus does not return to the triggering “Candidate Feedback” button.

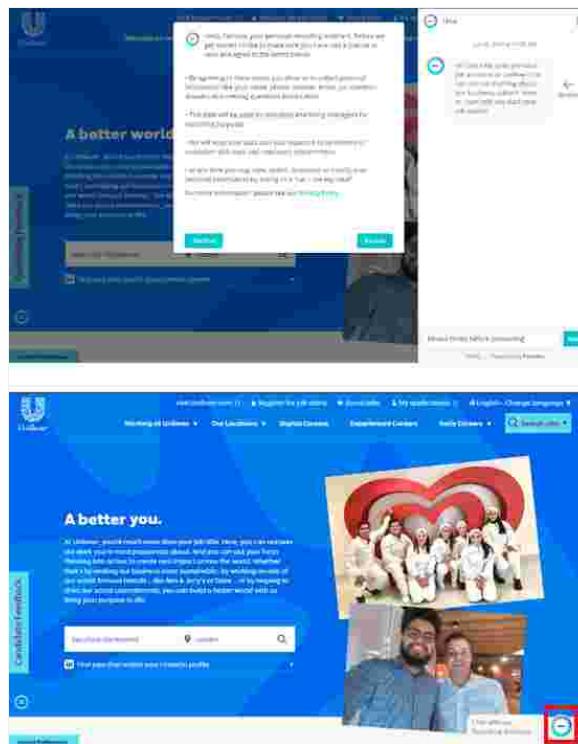


Page 2: desktop, focus does not return to trigger when “Candidate Feedback” modal is dismissed

```
X document.activeElement
body#home.magic-bullet-ally
```

Page 2: DevTools Console, focus is on the body element

This can also be seen on page 2 on desktop and iOS with the chat widget triggering buttons. When the widget is closed with the keyboard or screen reader, focus does not return to the button that was used to trigger it.



Page 2: desktop, focus does not return to trigger when chat widget is dismissed

```
x document.activeElement  
body#home.magic-bullet-ally
```

Page 2: DevTools Console, focus is on the body element

This can be seen on page 7, on desktop and iOS, when the “Inclusive policies” panel is expanded. The panel can be expanded by selecting the “Inclusive policies” tile under the “What we offer” section. When expanded, the panel appears over the tiles in this section. When dismissing the panel, focus does not return to the triggering tile.

Note that only the “Inclusive policies” panel was in scope for this audit. However, this behaviour is likely to also affect the other panels.

The screenshot shows a section titled "What we offer" with a yellow header. Below it is a grid of six items, each with an icon and a title:

- Inclusive policies
- Attractive total remuneration package
- Personal development
- Holidays
- Flexi-time
- Inclusive policies
- Working from home

Page 7: desktop, focus does not return to trigger when panel is dismissed

```
<button class="interactive-benefits-grid-child__close-button" tabindex="0"></button>
> document.activeElement
<body id="unilever-netherlands" class="content-pages full-header-theme magic-bullet-ally" data-site-geolocation="true" data-location-geotype="none" data-geolocation-url="/search-jobs/GetSearchRequestGeolocation"></body>
```

Page 7: DevTools Console, focus does not return to trigger when panel is dismissed

Steps to reproduce

1. Using just your keyboard, tab onto the above elements and activate them.
2. Observe that new content can be added and removed from the page.
3. Using just your keyboard, close or dismiss the new content.
4. Observe that focus does not return to the element that triggered the content.

Recommendation

Make sure that focus returns to the element that triggered the content. This can be achieved with the **focus** method.

For examples, see [MDN - focus method](#) and [Dev - Managing Focus for Accessibility](#).

For more information, see [WCAG - G59 Placing the interactive elements in an order](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - DOM Order Matters](#), and [Using Tabindex](#).

[Return to priority table](#)

ADR_20590-65 MEDIUM PRIORITY OPEN

Focus not set onto expanded content

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 07

Users Affected Motor

Component Not Applicable

What happened

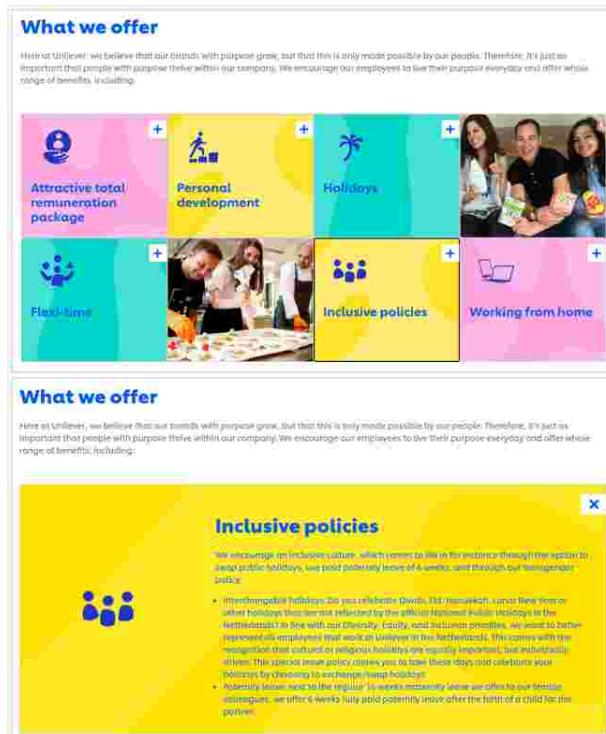
The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are instances where the focus is not set onto the component when it is opened. Instead, keyboard focus remains in the triggering element in the background. This may particularly disadvantage keyboard users. The illogical focus order may make the modal dialog challenging to access and operate.

Where it happened

This can be seen on page 7, on desktop and iOS, when expanding the “Inclusive policies” panel. The panel can be expanded by selecting the “Inclusive policies” tile under the “What we offer” section. When expanded, the panel appears over the tiles in this section. However, focus is not set onto the panel. Instead, it remains on the triggering button in the background.

Note that only the “Inclusive policies” panel was in scope for this audit. However, this behaviour is likely to also affect the other panels.



Page 7: desktop, “Inclusive policies” panel expands over tiles



The screenshot shows the DevTools Console with the following DOM structure:

```
><button class="interactive-benefits-grid-child-trigger">...</button> (flex)
  <button class="interactive-benefits-grid-child-trigger">...</button> (flex)
    <button class="interactive-benefits-grid-child-trigger"> (flex)
      <div class="interactive-benefits-grid-child-icon"></div>
      <div class="interactive-benefits-grid-child-title">
        <h3>Inclusive policies</h3>
      </div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
              > document.activeElement
                <button class="interactive-benefits-grid-child-trigger"> (flex)
                  <div class="interactive-benefits-grid-child-icon"></div>
                  <div class="interactive-benefits-grid-child-title">
                    <h3>Inclusive policies</h3>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Page 7: DevTools Console, focus is not set onto expanded panel

Steps to reproduce

1. Using just your keyboard, activate the panel.
2. Use Tab to navigate around the page.
3. Observe that focus is not automatically set onto the dialog.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in a meaningful order.

In this case, when the panel is open, make sure that focus is set onto the expanded panel. This can be achieved with the `focus` method. As well, if you need the non-interactive elements to be capable of receiving focus through the `focus` method, use `tabindex="-1"`.

For more information, see [MDN - focus method](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), and [Google Developer - Using Tabindex](#).

[Return to priority table](#)

ADR_20590-68 MEDIUM PRIORITY OPEN

Focus is not managed appropriately when content is removed from the page

Success Criteria 2.4.3: Focus Order [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03

Users Affected Motor

Component Button

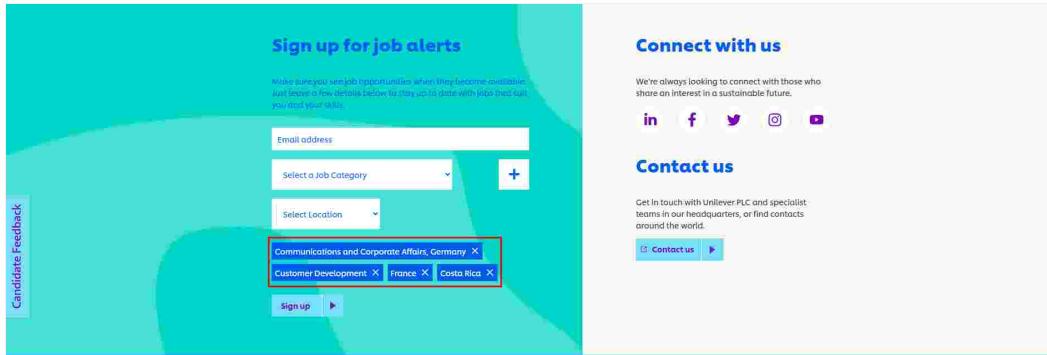
What happened

The keyboard focus order must be meaningful. It does not need to mirror the visual order. However, interactive elements must receive focus in a way that makes some sort of sense. The focus order must not be totally illogical or make the operability of components more challenging.

There are instances where the focus order is not managed when content is removed from the page. This may particularly disadvantage keyboard users who often navigate by tabbing through the interactive elements. The illogical focus order may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 3, on desktop and iOS, with the list of job categories/locations selected in the “Sign up for job alerts” section. When a filter is removed with the keyboard on desktop, the keyboard focus is visually lost and programmatically placed on the `<body>`. On iOS, the screen reader cursor is placed on an arbitrary element such as the “Email address” field.



Page 3: desktop, list of categories/locations selected in the “Sign up for job alerts” section

```
<ul class="keyword-selected" data-keyword-id="form-field-389fdacd7d-keywords" data-list-name="JobAlertCategoryLocation" aria-labelledby="selected-keywords-1">
  <li data-category-facet-type="1" data-category-name="Data & Analytics" data-category-term="0225216" data-custom-facet-name="ALL" data-custom-facet-type="data-location-facet" data-location-name="data-location-term" class="deleting">
    <span class="KeywordText">Data & Analytics</span>
    <a aria-label="Remove Data & Analytics" class="keyword-remove" href="#">
      "Remove"
    </a>
  </li>
```

Page 3: DOM, “Data & Analytics” filter

```
> <a aria-label="Remove Communications and Corporate Affairs, Germany" class="keyword-remove" href="#">≡</a>
  > <a aria-label="Remove Customer Development" class="keyword-remove" href="#">≡</a>
  > <a aria-label="Remove France" class="keyword-remove" href="#">≡</a>
> document.activeElement
< <body id="home" data-site-geolocation="true" data-location-geotype="none" data-geolocation-url="/search-jobs/GetSearchRequestGeoLocation" class="magic-bullet-ally">≡</body>
```

Page 3: DevTools Console, focus is not managed appropriately when content is removed



Page 3: iOS, screen reader cursor is placed on "Email address" field when "Data & Analytics" filter is removed

Steps to reproduce

1. Observe that content can be dynamically removed from the page.
2. Using just your keyboard, tab through the above.
3. Remove the content from the page.
4. Continue to tab through the page.
5. Observe that the keyboard focus has not been appropriately managed.

Recommendation

The focus order is determined by the location of focusable elements in the Document Object Model (DOM). Make sure that the interactive elements are placed in the DOM in a logical sequence that allows keyboard users to tab through the page in meaningful order.

In this case, make sure that focus is managed appropriately. Use the focus method sparingly when the page has a meaningful update. For example, if the user selects a button and the content is removed from the DOM, set focus onto a neighbouring element. For example:

```
// when this card is deleted ... //
<h2>Example Card 1</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam nec eleifend purus...
</p>
<button>Remove Card 1</button>

// ... focus can be set onto this neighbouring card //
<h2>Example Card 2</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam nec eleifend purus...
</p>
<button>Remove Card 2</button>
```

For examples, see [MDN - focus method](#) and [Dev - Managing Focus for Accessibility](#).

For more information, see [WCAG - G59 Placing the interactive elements in an order](#), [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), [Keyboard Navigable JavaScript widgets](#), [Google Developer - DOM Order Matters](#), and [Using Tabindex](#).

[Return to priority table](#)

ADR_20590-78 MEDIUM PRIORITY OPEN

Links not self-explanatory or disambiguated by context.

Success Criteria 2.4.4: Link Purpose (In Context) [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 08

Users Affected Cognitive

Component Link

What happened

Links must be self-explanatory or be disambiguated by their surrounding context.

There are links that do not meet this requirement. This may particularly disadvantage users that find navigating the web cognitively challenging. As well, it is common for screen reader users to navigate pages through lists of elements on the page, including lists of links. The use of vague links can make this navigation method challenging.

Where it happened

This can be seen on page 8 with the “View jobs” links in the “Explore Where You Fit Across Our Digital Teams”. The specific purpose of each link cannot be determined from its programmatically determined link context. Note that the accordion headings such as “Digital R&D”, “Digital Supply Chain”, and so on, are not marked up as headings, which prevents them from providing a context for the links.



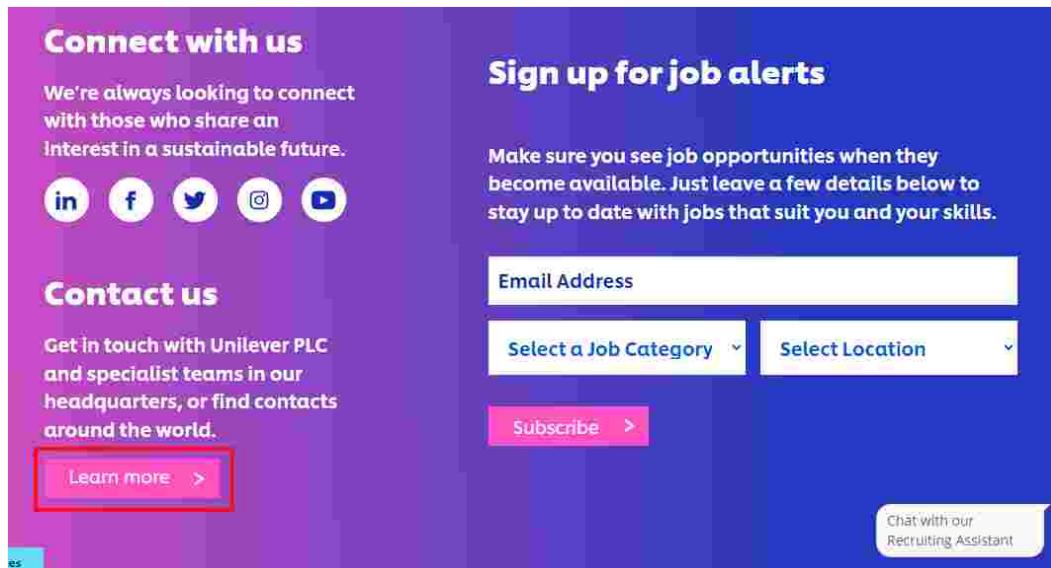
Page 8: desktop, “View Jobs” links

The screenshot shows the DOM structure for the 'Digital R&D' section. It includes an accordion title, content, and a right-side panel with text and another accordion. A red box highlights the 'View Jobs' link located within the right-side panel's content area. The link is an anchor tag with the href attribute set to '#digitalRd' and the class 'jobLink'.

```
<div class="accordionLeft">
  <div>
    <a href="#accordion1" aria-expanded="false" aria-controls="accordion1" class="accordion-title accordionTitle js-accordionTrigger">
      <span>Digital R&D</span>
    </a>
  </div>
  <div class="accordion-content left accordionItem is-collapsed" aria-hidden="true"></div>
</div>
<div class="accordionRight">
  <div>
    <div class="accordion-content right accordionItem is-collapsed" id="accordion1" aria-hidden="true">
      <p>Our R&D teams believe that innovation occurs in truly connected environments where creative ideas are encouraged. We have more than 5,000 research and development experts working across 60 countries, including in six global labs and ten regional hubs, working to create tomorrow's innovation through the power of consumer insights, best in class research, and technology and data.</p>
      <p>People capabilities: innovation mindset, agile, E2E consumer experience creation.</p>
    </div>
    <div>
      <a href="#digitalRd" class="jobLink">
        <span>View Jobs</span>
        <img alt="https://tbohm.talentbrew.com/company/34155/img/graphics/right_arrow.svg" alt="right arrow" data-loading="lazy">
      </a>
    </div>
  </div>
</div>
```

Page 8: DOM, purpose of “View jobs” link for “Digital R&D” cannot be determined from its programmatically determined link context

This can also be seen on page 8 with the “Learn more” link under “Contact us”. This link goes to the Careers home page. This purpose cannot be determined from the link in its programmatically determined context.



Page 8: “Learn more” link

```
<div class="contact">
  <h3>Contact us</h3>
  <div>
    "Get in touch with Unilever PLC and specialist teams in our headquarters, or find contacts around the world."
  </div>
  <a href="/search-jobs" class="button-with-arrow button-with-arrow-right" data-custom-event="true" data-custom-category="Custom" data-custom-label="About Us - Search Jobs">
    "Learn more"
  </a>
</div>
```

The image shows the DOM structure of the 'Learn more' link. The link is highlighted with a red box in the browser's developer tools. The code snippet shows the HTML structure, including the 'a' tag with its attributes and the surrounding div and h3 tags.

Page 8: DOM, purpose of link is not communicated by its text or context

Steps to reproduce

1. Observe that the links are not self-explanatory.
2. Observe that the links are disambiguated by their surrounding context.

Recommendation

Make links self-explanatory. Update the link text to communicate the exact purpose or destination of the link. For example:

```
<a href="...">Read More about our quarterly earnings</a>
```

When this is not possible, provide context that disambiguates links. Example solutions include putting a link in a paragraph tag or table cell with preceding information that communicates the purpose or definition of the link. For example:

```
<p>Our quarterly earnings, <a href="...">Read More</a></p>
```

Another solution is providing a preceding heading or associated table header that communicates the purpose or definition of the link. For example:

```
<h2>Our quarterly earnings</h2>
<a href="...">Read More</a>
```

```
<th scope="row">Account Ending 1234</th>
...
<th scope="col">Account Action</th>
...
<td><a href="...">Delete</a></td>
```

If you cannot do any of the above, you can manually associate the link with some context or manually add it onto the link. For example:

```
<p id="context">Our quarterly earnings</p>
<a href="..." aria-describedby="context">Read More</a>
<a href="..." aria-label="Read More about Our quarterly earnings">Read More</a>
```

For more information, see [Yale - Web Accessibility - Links](#), [University of Washington - Links and buttons](#), [A11y Project - Valid and Accessible Links](#).

[Return to priority table](#)

ADR_20590-52 MEDIUM PRIORITY OPEN

Interactive elements have no visible focus indicator

Success Criteria 2.4.7: Focus Visible [2.0 AA]

Devices Desktop

Pages Affected 03, 08

Users Affected Vision, Motor

Component Button

What happened

When they receive keyboard focus, interactive elements must provide a visual cue.

There are interactive elements that receive focus but which do not provide any visual cue. There is no visual difference between when the element is not receiving keyboard focus and when the element is receiving keyboard focus. Keyboard users may struggle to track their position on the page.

Where it happened

This can be seen on page 3, on desktop, with the “Previous” and “Next” scroll buttons under the “Our brands” section. These controls do not have a visible focus indicator due to the CSS property `outline: 0`.



Page 3: desktop, “Previous” and “Next” scroll buttons

```
<section class="slider-of-logos_max-width-wrapper_px-x-width-wrapper-narrow">
  <h2 class="slider-of-logos__header">Our brands</h2>
  <p class="slider-of-logos__copy"><a href="#">View all</a></p>
  <div class="slider-of-logos_slick_global-slick-slider-styling slick-initialized slick-slider">
    <button class="slick-prev slick-arrow" aria-label="Previous" type="button" style="tabindex=0">Previous</button>
    <div class="slick-list dragger"><a href="#"></a></div>
    <button class="slick-next slick-arrow" aria-label="Next" type="button" style="tabindex=0">Next</button>
  </div>
  <div class="slider-of-logos_cta-button"><a href="#">Learn more</a></div>
</section>
```

```
.slick-prev:hover, .slick-prev:focus, .slick-next:hover, .slick-next:focus {
  color: transparent;
  outline: 0;
  background: transparent;
}
.slick-list {
  background-color: transparent;
  background-size: contain;
  background-repeat: no-repeat;
  background-position: center;
}
```

Page 3: DOM, “Previous” and “Next” scroll buttons

This can also be seen on page 8 on desktop with the previous and next buttons for the “Meet Our People” carousel and the “What is a Digital Career Like in Unilever” video link. These elements do not have visible keyboard focus indication due to the CSS property `outline: 0`.



Page 8: "Meet Our People" carousel previous and next buttons do not have visible focus indication

The screenshot shows the DOM structure and associated CSS styles for the 'Previous' button. The DOM includes elements like the main container, a slick slider container, and the 'slick-prev' button itself. The CSS styles define the button's dimensions and appearance, including its outline when focused.

```
<div class="two-two-grid max-width-wrapper max-width-wrapper--carrousel">
  <div class="two-two-grid__inner two-two-grid__inner--sliders-container slick-initialized slick-slider">
    <button class="slick-prev slick-arrow" aria-label="Previous" type="button" style="tabIndex: 0;"><::before> <span>Previous</span></button>
    <div class="slick-list draggable" style="height: 460px;"><div></div></div>
    <button class="slick-next slick-arrow" aria-label="Next" type="button" style="tabIndex: 0;"><::after> <span>Next</span></button>
  </div>
```

```
width: 36px;
height: 36px;

outline: 0;
background-color: transparent;

outline: 2px solid #000;
outline-offset: 2px;

width: 36px;
height: 36px;
```

Page 8: DOM, "Previous" button has outline: 0 on focus

Steps to reproduce

1. Use a keyboard to navigate through the above.
2. Observe that there is no visual cue when the elements receive keyboard focus.

Recommendation

Make sure that elements provide a visual cue when they receive keyboard focus. This can be achieved using the CSS `focus` pseudoclass and `outline` property. For more information, see [CSS Tricks - Focusing on Focus Styles](#), [GOV - Understanding focus state styles](#), and [Web Dev - Style Focus](#).

Another solution is making sure that default browser indication is supported. This is sometimes undone with CSS reset sheets. Do not overrule browser focus indication without implementing a custom focus indication. For example, do not use `outline: none`. For more information, see [a11y Project - Never Remove CSS Outlines](#).

[Return to priority table](#)

ADR_20590-66 MEDIUM PRIORITY OPEN

Focused element is completely hidden by chat popup which cannot be dismissed

Success Criteria 2.4.11: Focus Not Obscured (Minimum) [2.2 AA]

Devices Desktop

Pages Affected 08

Users Affected Motor

Component Link, Button

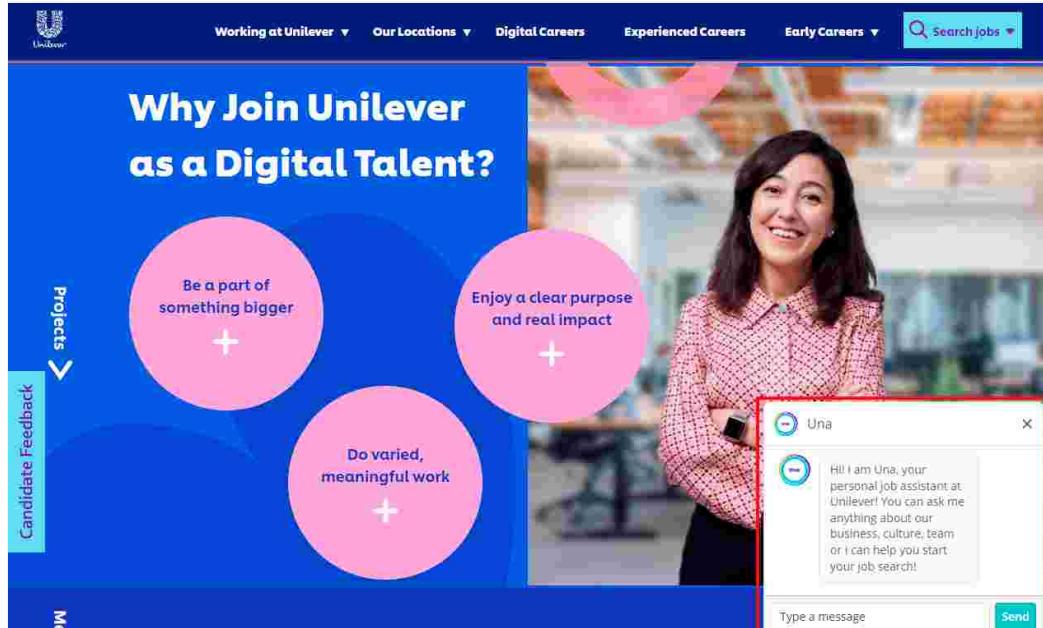
What happened

When an interactive element receives focus, the element and its focus indicator must not be completely hidden by author-created content, such as sticky footers, sticky headers and non-modal dialogs.

There is an interactive element that is completely hidden by an author-created chat popup when it receives focus. This chat popup cannot be closed without moving focus. This may disadvantage keyboard users, who may struggle to track their position on the page.

Where it happened

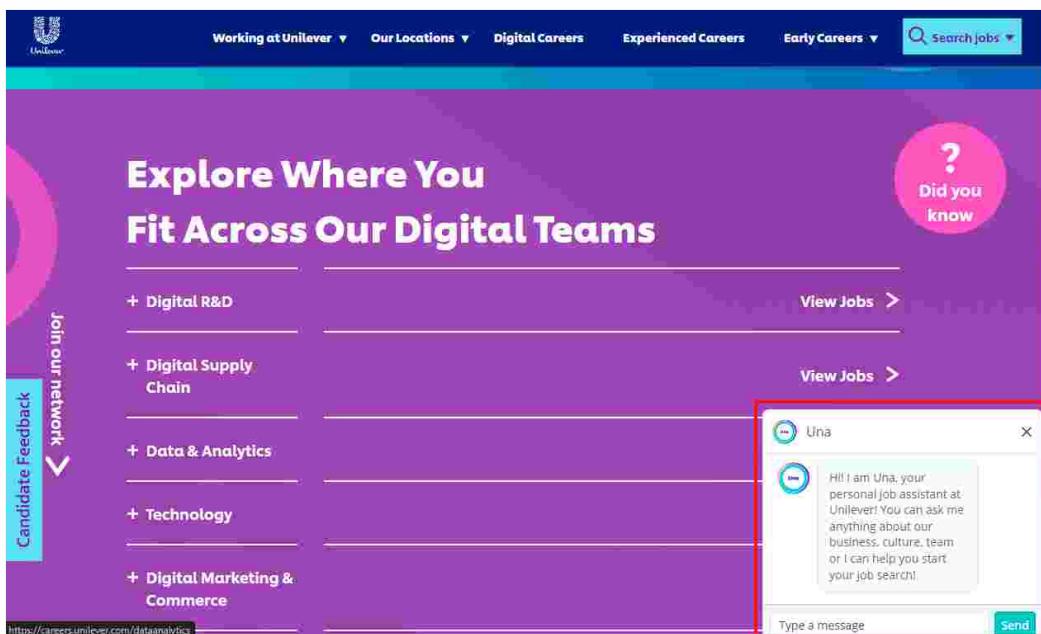
This can be seen on page 8 with the chat popup. When the chat is open, the popup obscures part of the page content, but elements behind it can still receive keyboard focus. The chat popup cannot be closed without moving focus. For example, the “Did you know” button under “Purpose Driven Work with Global Impact” and the “View Jobs” links for “Data & Analytics”, “Technology”, and “Digital Marketing & Commerce” in the “Explore Where You Fit...” accordion are fully obscured by the chat popup when they receive focus.



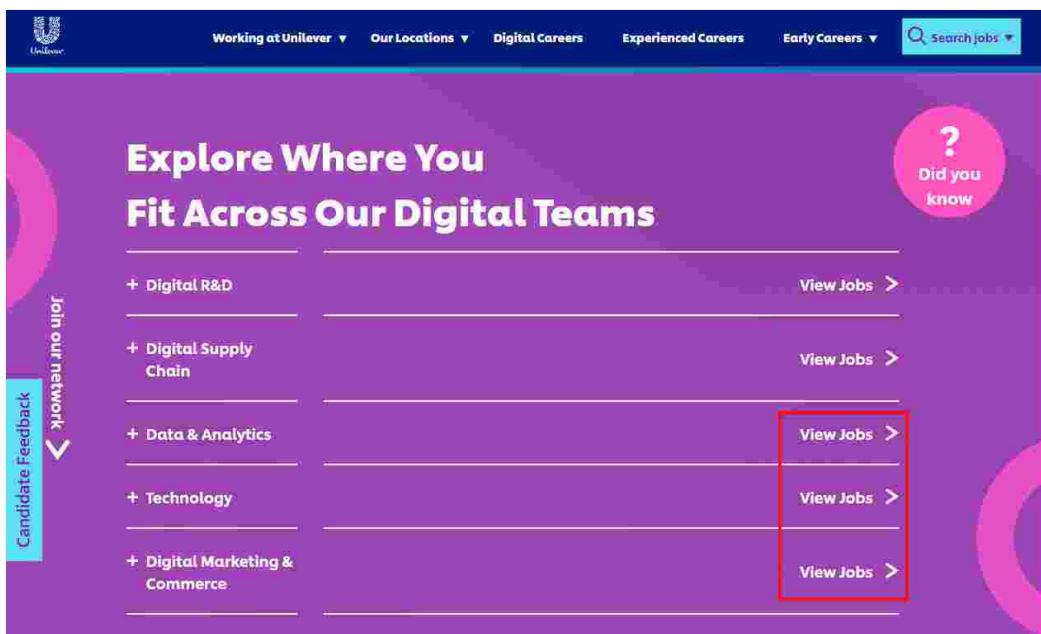
Page 8: desktop, “Did you know?” button under “Purpose Driven Work with Global Impact” is obscured by chat pop-up when it has focus



Page 8: desktop, “Did you know?” button under “Purpose Driven Work with Global Impact” when focus is not obscured



Page 8: desktop, “View Jobs” link for “Data & Analytics” accordion is fully obscured by the chat popup when it receives focus



Page 8: desktop, “View jobs” links in accordion when focus is not obscured

Steps to reproduce

1. Open the chat popup, if it is not already open.
2. Use a keyboard to navigate through the above.
3. Observe that when the interactive element receives keyboard focus, the interactive element and its focus indicator are completely hidden by the chat popup.

Recommendation

Ensure that the interactive element and its focus indicator are visible when the interactive element receives focus. To pass this success criterion, the interactive element and its focus indicator must be at least partially visible. However, for best practice, where possible ensure that it is fully visible.

In this instance, allow the user to close the chat popup without having to move focus. This could be achieved by pressing **Esc**. Ensure that the focus remains on the interactive element.

Another option would be to require the user to dismiss the chat popup before interacting with the other content on the page. That is, to change the non-modal overlay to a modal dialog. Then the chat popup would not hide an element when it receives focus, because elements behind it could not receive focus.

For more information, see [WAI ARIA - Dialog Example](#).

[Return to priority table](#)

ADR_20590-15 MEDIUM PRIORITY OPEN

Accessible name does not include visual name

Success Criteria 2.5.3: Label in Name [2.1 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 08

Users Affected Vision

Component Button

What happened

The accessible name of interactive elements must include the name that is visually communicated by on-screen text. This can otherwise particularly disadvantage voice recognition users, who use a say-what-you see interaction model. There are interactive elements where the accessible name does not include the visual name.

Where it happened

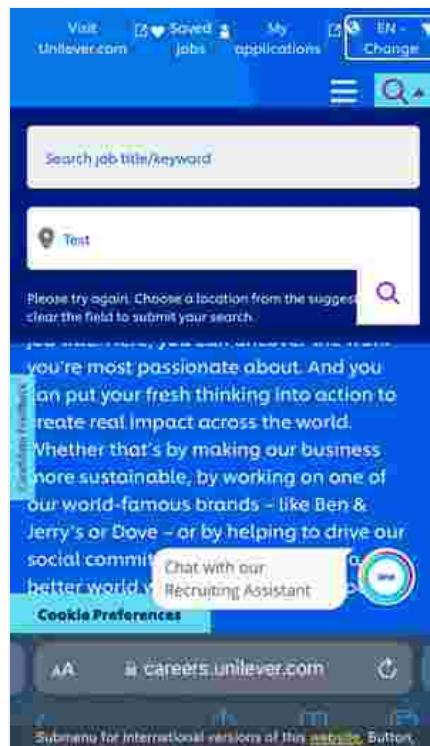
This can be seen on page 1 on desktop and iOS with the “Change language” button. On desktop, this button has a visual label that consists of the currently selected language followed by “Change language”, such as “English - Change language”. On iOS, the button has a condensed name such as “EN - Change”. On both devices, however, the accessible name of the element is “Submenu for international version of this website”, which does not contain the visual label.



Page 1: desktop, “English - Change language” button

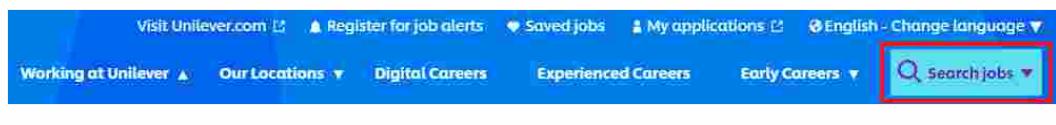


Page 1: DOM, “English - Change language” button has accessible name of “Submenu for international version of this website” determined with an `aria-label`



Page 1: iOS, “EN - Change” button has accessible name of “Submenu for international version of this website”

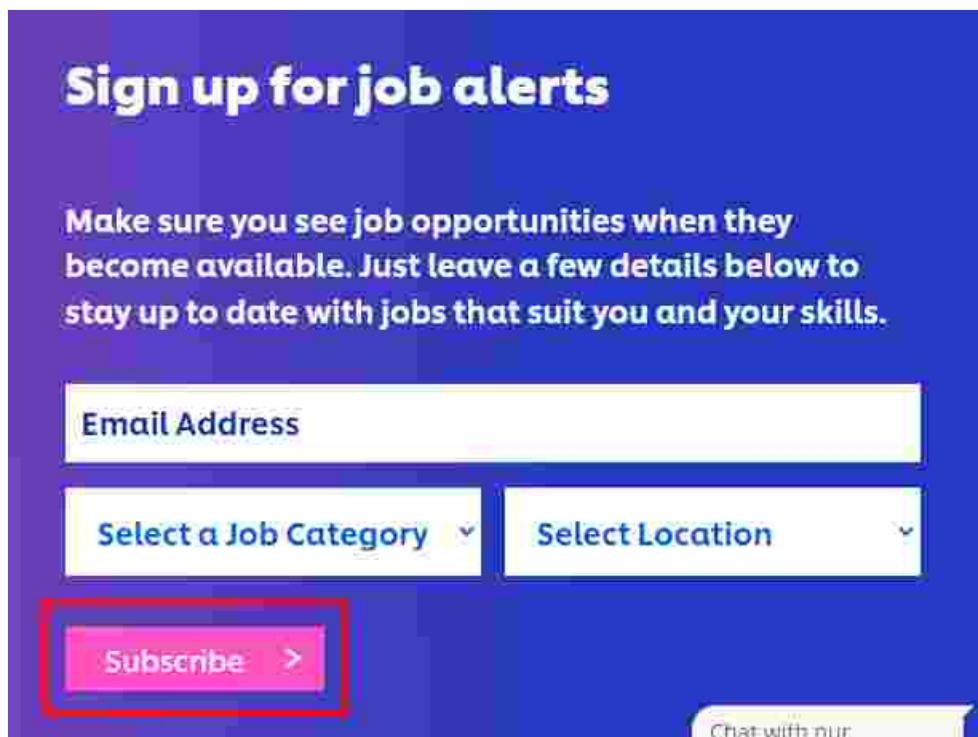
This can also be seen on page 1 on desktop with the “Search jobs” button. The visual label of this button is “Search jobs”. However, its accessible name is “Open search form”, which does not contain the visual label.



```
<button class="toggle-search-form-trigger" aria-expanded="false" aria-label="Open search form" data-custom-event="true" data-custom-category="Custom_Event" data-custom-label="Magnifying glass icon">
  :before
  <span>Search jobs</span>
  :after
</button>
```

Page 1: desktop, “Search jobs” button

This can also be seen on page 8 on desktop and iOS with the “Subscribe” button. The button has an accessible name of “Sign Up for Job Alerts”, which has been determined with an **aria-label**. This name does not contain the visual text “Subscribe”.



Page 8: desktop, "Subscribe" button has visible text "Subscribe"

```
<div class="form-field submit">
  <button type="submit" data-cs="true" data-custom-event="true" data-custom-category="Subscribe" data-custom-label="Digital - Sign Up for Job Alerts - Subscribe button" aria-label="Sign Up for Job Alerts">Subscribe >
</button>
</div>
<div class="form-field form-message"></div>
</form>
</div>
</div>
```

Page 8: DOM, "Subscribe" button has an accessible name of "Sign Up for Job Alerts"

Steps to reproduce

1. Observe the visual name of the element communicated by on-screen text.
2. Inspect the element and review the accessible name.
3. Observe that the accessible name does not include the visual name.

Recommendation

Make sure that the accessible name includes the visual name. This is best achieved by creating the accessible name by referencing on-screen text.

Where a button contains visible text, avoid overriding this text with an `aria-label`. For example,

```
<button aria-expanded="false">English – Change language</button>
```

Otherwise, elements can reference on-screen text with `label` or `aria-labelledby`. Make sure that the `for` and `id` match correctly. For example:

```
<label for="firstName">First Name</label>
<input id="firstName" type="text" autocomplete="given-name">
```

Note that `aria-labelledby` is useful when you need to construct the label using multiple elements. For example:

```
<span id="firstPart">Last</span>
<span id="secondPart">Name</span>
<input aria-labelledby="firstPart secondPart" type="text" autocomplete="family-name">
```

If you cannot reference the visual name, make sure that you manually update the accessible name to include the visual name. Although not strictly required, it is best practice to have the visual name at the start of the accessible name and with the words in the same order.

For more information, see [WAI Tutorials - Labelling Controls](#), [WCAG - G208 Including the text of the visible label](#), and [G211 Matching the accessible name](#).

[Return to priority table](#)

ADR_20590-39 MEDIUM PRIORITY OPEN

Target size for pointer input does not meet minimum size requirements

Success Criteria 2.5.8: Target Size (Minimum) [2.2 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 05

Users Affected Motor

Component Link

What happened

All pointer input targets must have an area of at least 24 by 24 CSS pixels unless:

- Targets that are under 24 by 24 CSS pixels are positioned so that a 24 CSS pixel diameter circle centred on each target would not intersect with another target or another circle.
- The function can be achieved through a different control on the same page which meets this requirement.
- The target is in a block of text or similarly constrained by the line-height of non-target text.
- The size of the target is not modified by the author and is determined by the user agent.
- A particular presentation of the target is essential or is legally required for the information being conveyed.

There are targets which do not have the required area and are not exempt. This may particularly disadvantage users with mobility impairments such as hand tremors, users who have difficulty with fine motor movements, or those using a device where the touch screen is the primary mode of interaction.

Where it happened

This can be found on page 5 on desktop and iOS, with the “Culture for growth”, “Every Voice matters” and so on links present under “What We Can Offer You” section. These links do not meet minimum target size requirement of 24 by 24 CSS pixels.

What We Can Offer You

Culture for Growth | Top Notch Employee Health & Well Being Benefits | Every Voice Matters | Global Reach | Life at Unilever | Careers with Purpose | World Class Career Development Programs | Check Out Our Space | Focus On Sustainability

Pay: The pay range for this position is \$110,700 to \$166,000. Unilever takes into consideration a wide range of factors that are utilized in making compensation decisions including, but not limited to, skill sets, experience and training, licensure and certifications, qualifications and education, and other business and organizational needs.

Bonus: This position is bonus eligible.

Long-Term Incentive (LTI): This position is LTI eligible.

Page 5: desktop, the interactive elements such as “Culture for growth”, “Every Voice matters” and so on links

The developer tools panel shows the following CSS for the highlighted link:

```
span {
    font-size: 1em;
    line-height: 22.4px;
    padding-right: 15px;
    position: relative;
    color: #0070C0;
    text-decoration-color: #0070C0;
    text-decoration-line: none;
    text-decoration-style: solid;
    text-decoration-thickness: 1px;
    text-decoration-width: 0px;
    text-size-adjust: 100%;
    transition-behavior: normal;
    transition-delay: 0s;
    transition-duration: 0.3s;
    transition-property: all;
    transition-timing-function: ease-in-out;
}
```

Page 5: DOM, the line height for “Culture for growth” link is 22.4 pixel

Steps to reproduce

1. Observe the pointer input target.
2. Inspect the target in the browser CSS panel.
3. Observe that the area of the target is not a minimum of 24 by 24 CSS pixels and is not exempt.

Recommendation

Ensure that all targets are a minimum of 24 by 24 pixels in size. For best practice, where possible, ensure all targets are a minimum of 44 x 44 pixels in size.

For more information, see [Using min-height and min-width to ensure sufficient target spacing](#) and [Accessible tap targets](#).

[Return to priority table](#)

ADR_20590-16 MEDIUM PRIORITY OPEN

Text content has no lang attribute

Success Criteria 3.1.2: Language of Parts [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 01

Users Affected Vision

Component Plain Text

What happened

Knowing the language of content enables assistive technology to communicate content accurately. It is used by screen readers, Braille displays, and other text-to-speech programs. There are text snippets that do not have an appropriate `lang` attribute. Assistive technology may miscommunicate content.

Where it happened

This can be seen on page 1 on desktop and iOS with the languages in the language selector, such as “Português”. Each language is specified in the corresponding language, but the links do not have `lang` attributes.



Page 1: desktop, languages in language selector are not in English



Page 1: DOM, language links do not have lang attributes

Steps to reproduce

1. Inspect the text that is in a different language
2. Note that there is not an appropriate `lang` attribute

Recommendation

Specify the appropriate `lang` attribute for any text that is in a different language to the rest of the page. For example:

```
<p>
The French multinational investment bank
<span lang="fr">Société Générale</span>
has headquarters in Paris.
</p>
```

When the text uses a specific dialect, use the correct `lang` subtype. For example `lang="zh-hant"` for Traditional Chinese and `lang="zh-tw"` for Taiwanese. This is particularly important when the dialects are not mutually intelligible.

If this is not possible, you can instead translate the content.

For a list of 2-letter language codes, see [Wikipedia - List of ISO 639-codes](#). As well, you can use [GitHub - BCP47 language subtag lookup](#) to search through the various language codes. For more general information, see [W3 - Language declarations](#) and [W3 - Choosing a Language Tag](#).

[Return to priority table](#)

ADR_20590-64 MEDIUM PRIORITY OPEN

Keyboard focus is moved to another element when element is adjusted

Success Criteria 3.2.2: On Input [2.0 A]

Devices Desktop

Pages Affected 07

Users Affected Motor

Component Combobox

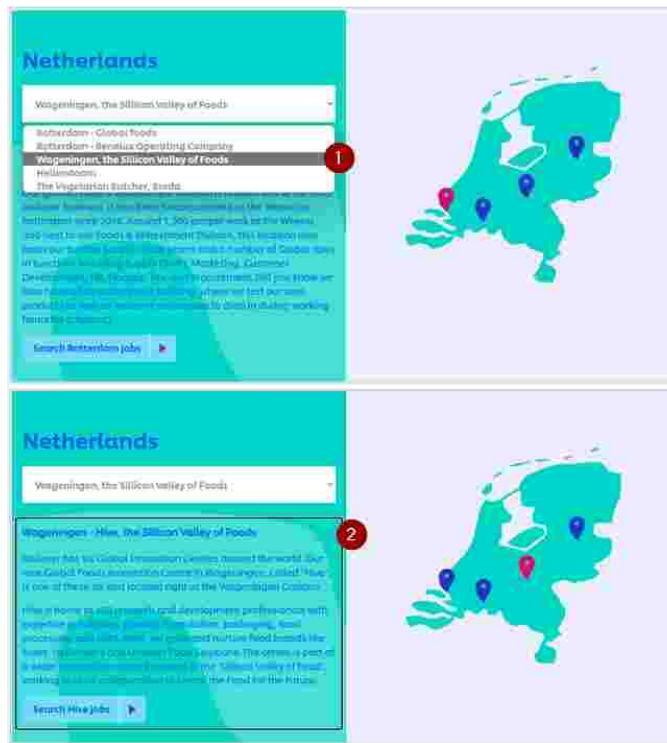
What happened

When interactive elements are adjusted, they must not automatically trigger any unexpected behaviour.

There are elements that move focus elsewhere when they are adjusted. This may particularly disadvantage keyboard users who often navigate by tabbing through the interactive elements. The automatically activating elements may make the page difficult to understand and interact with.

Where it happened

This can be seen on page 7, on desktop, with the location selector under the “Netherlands” section. When selecting a location on the combobox, focus is moved to the content below it.



Page 7: desktop, focus is moved to text description when selecting a location

Steps to reproduce

1. Using just your keyboard, tab through the above.
2. Adjust the interactive elements.
3. Observe that focus is moved when the elements are adjusted.

Recommendation

Navigating and controlling webpages with a keyboard should be logical and predictable. Make sure that elements do not automatically move focus when they are adjusted. Avoid triggering functionality using the `change` event.

Make sure that elements do not automatically activate when they are adjusted. For example, make sure that checking a checkbox, entering text into a text field, or changing the selected option on a dropdown, does not automatically trigger functionality.

Avoid moving focus manually unless you have a good reason. For example, you might move focus onto a modal dialog when it is opened. However, valid uses cases like this are rare. As a general rule, focus should not be overridden or manually moved to another element when the user is simply navigating through the page.

Instead, before triggering functionality, wait until the user manually activates the functionality. Use events such as `click` and `keydown`. Activation should generally be done with Space or Return. This is the standard behaviour associated with `click` events on standard elements, such as `<button>` elements.

When working with custom components, expected keyboard commands are provided by the [WAI ARIA Practices](#). For more information, see [Yale University - Web Accessibility - Focus and Keyboard Operability](#), [MDN - Keyboard](#), and [Keyboard Navigable JavaScript widgets](#).

[Return to priority table](#)

ADR_20590-74 MEDIUM PRIORITY OPEN

Components with the same functionality are inconsistently identified

Success Criteria 3.2.4: Consistent Identification [2.0 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 08

Users Affected Cognitive

Component Button

What happened

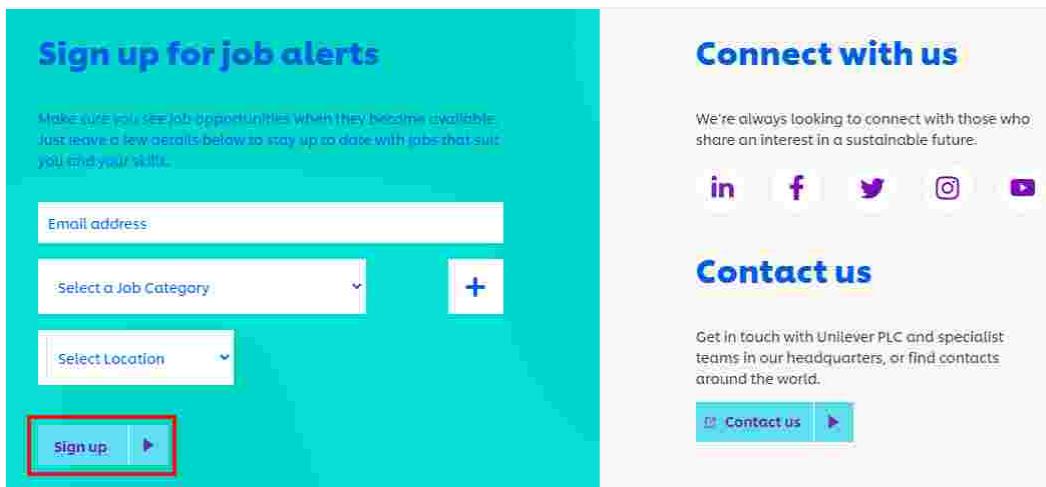
When a collection of webpages share a common purpose and components with the same functionality appear multiple times in the collection, these components need to be consistently identified. This includes the visual label, text alternatives for functional images, and accessible name.

There are repeated components that are inconsistently identified. This may particularly disadvantage users that find navigating the web cognitively challenging. As well, it may significantly impact those that interface with the accessibility properties of elements, such as screen reader and voice recognition users.

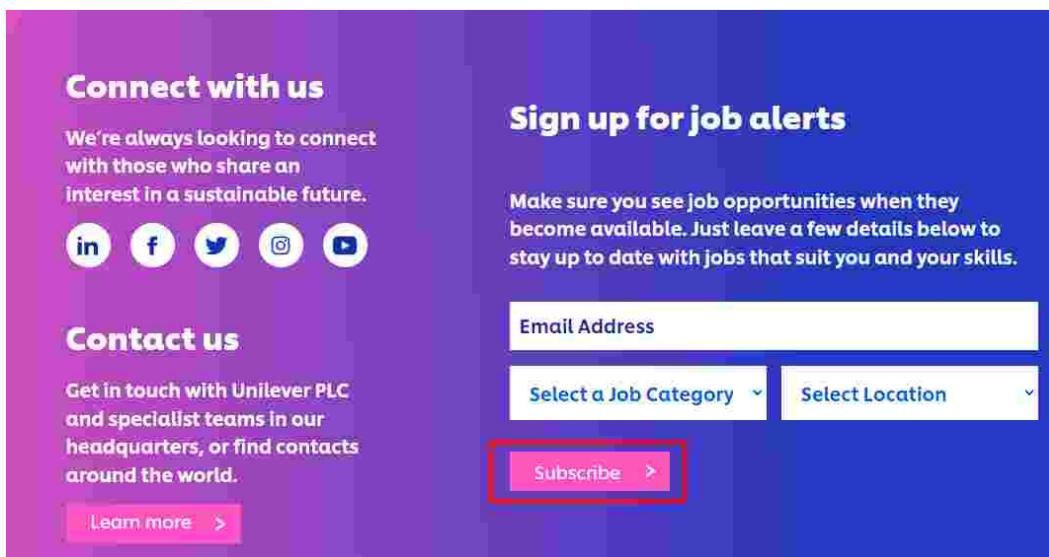
Users may struggle to use the components. The inconsistent identification may confuse or misguide users.

Where it happened

This can be seen on pages 1 and 8. On page 1, the button in the “Sign up for job alerts” section is identified as “Sign up”, while on page 8 it is identified as “Subscribe”. However, the buttons both have the functionality of signing up for a job alert.



Page 1: desktop, “Sign up” button in “Sign up for job alerts” section



Page 8: desktop, "Subscribe" button in "Sign up for job alerts" section

Steps to reproduce

1. Observe that there is a collection of webpages that share a common purpose.
2. Observe that components with the same functionality are used multiple times in the collection.
3. Observe that the components are inconsistently identified.

Recommendation

When components have the same functionality, make sure that the visual labels, text alternatives for functional images, and accessible names, are consistent. They do not need to be identical when there is slightly different functionality, but there should be a consistent approach.

Whatever labelling method is used, such as `label`, `aria-label`, `aria-labelledby`, `aria-describedby`, and `alt`, make sure that the values used with these are consistent when the underlying functionality has a common purpose.

For more information, see [MDN - G197 Using labels, names, and text alternatives consistently for content that has the same functionality](#)

[Return to priority table](#)

ADR_20590-75 MEDIUM PRIORITY OPEN

Programmatic order of help is inconsistent within a set of pages

Success Criteria 3.2.6: Consistent Help [2.2 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 08

Users Affected Cognitive

Component Link, Heading

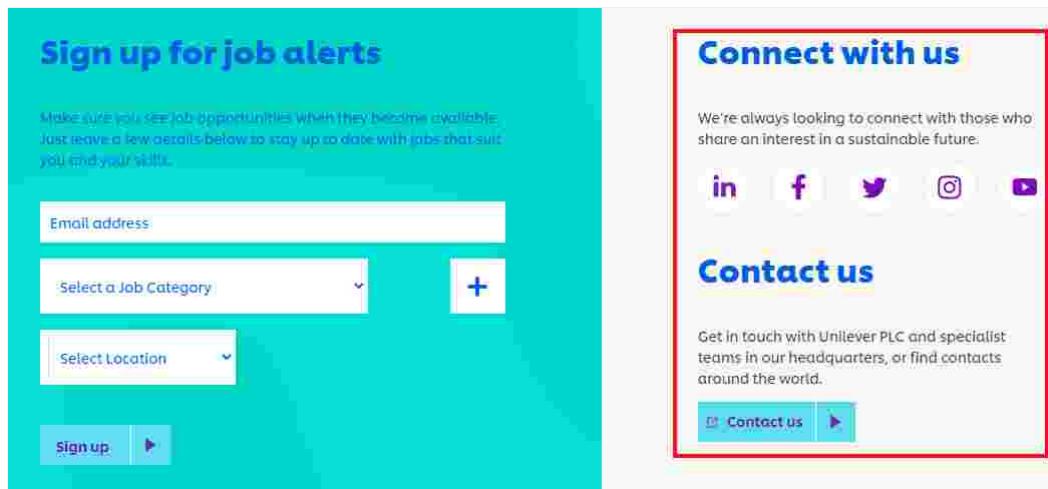
What happened

When there are help mechanisms on the page and they are repeated within a set of pages, they must occur programmatically in the same relative order to other page content. Help mechanisms can include: a phone number or email address, a messaging system, a chat client, a contact form or social media channel, frequently asked questions, support pages or automated chatbot systems.

There are help mechanisms that are repeated across a set of pages, but they do not occur in the same relative order. This may particularly disadvantage users who find navigating the web cognitively challenging.

Where it happened

This can be found on pages 1 and 8 with the “Connect with us” and “Contact us” sections, which function as a help mechanism. On page 1, these sections occur programmatically after the “Sign up for job alerts” section. On page 8, they occur programmatically before the “Sign up for job alerts” section.



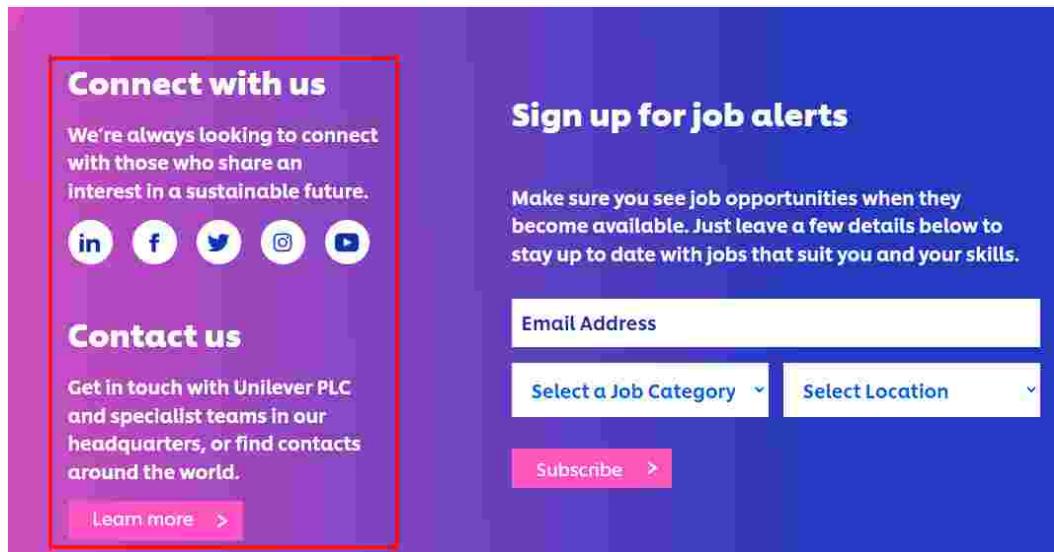
Page 1: “Connect with us” and “Contact us” sections follow “Sign up for job alerts”

```

<h2 class="form-display-heading">Sign up for job alerts</h2>
<p class="form-instructions"><a href="#">Edit</a></p>
<p class="form-field required"><input type="text" value="Unilever PLC" name="JobTitle" data-ajax-category="Keywords" data-available-categories="displayname" data-ajax-location="/keywords/getAvailableCategories?displayname=&data-all-jobs-text>All Jobs" data-birth-required="false" data-id="34355" data-multiple-select-enabled="false" data-site-id="0" data-includingAllCategory="true" /></p>
<p class="form-field confirm-email"><input type="text" value="Unilever PLC" name="Email" data-ajax-category="Email" data-available-categories="displayname" data-ajax-location="/keywords/getAvailableCategories?displayname=&data-all-jobs-text>All Jobs" data-birth-required="false" data-id="34355" data-multiple-select-enabled="false" data-site-id="0" data-includingAllCategory="true" /></p>
<input id="job-type-#4941aid2cf" name="JobType" type="hidden" value="JobAlert">
<input id="module-name-#4941aid7cf" name="ModuleName" type="hidden" value="Job Alert Form">
<input id="job-id-#4941aid2cf" name="JobID" type="hidden" value="34355">
<input id="job-organization-id-#4941aid2cf" name="OrganizationID" type="hidden" value="34355">
<input id="redirect:url-#4941aid2cf" name="RedirectURL" type="hidden" value="34355">
<input id="page:url-#4941aid2cf" name="PageURL" type="hidden" value="careers.unilever.com:80/>
<input id="page:name-#4941aid2cf" name="PageName" type="hidden" value="Homepage">
<input id="page-type-#4941aid2cf" name="PageType" type="hidden" value="Home">
<p class="form-field submit"><input type="submit" value="Submit" /></p>
<p class="form-field form-message"><a href="#">Cancel</a></p>

```

Page 1: DOM, “Connect with us” and “Contact us” sections follow “Sign up for job alerts”



Page 8: “Connect with us” and “Contact us” sections precede “Sign up for job alerts”

```

<div class="two-two-grid__child--right">
  <div class="connect">
    <h3>Connect with us</h3>
    <p>We're always looking to connect with those who share an interest in a sustainable future.</p>
    <ul class="social_media"><li><a href="#">LinkedIn</a></li><li><a href="#">Facebook</a></li><li><a href="#">Twitter</a></li><li><a href="#">Instagram</a></li><li><a href="#">YouTube</a></li></ul>
  </div>
  <div class="contact">
    <h3>Contact us</h3>
    <p>Get in touch with Unilever PLC and specialist teams in our headquarters, or find contacts around the world.</p>
    <a href="#">Learn more ></a>
  </div>
</div>
<div class="two-two-grid__child--left">
  <div class="job-alert-form-background">
    <form id="form-893554000" class="data-form job-alert-form" action="/Form/submit" data-success-message="Your subscription was submitted successfully." data-error-message="There was an error processing your request." data-close-text="Close" data-form-type="Job Alert" novalidate="novalidate">
      <h2 class="form-display-heading">Sign up for job alerts</h2>
      <p class="form-instructions"><a href="#">Edit</a></p>
      <p class="form-field required"><input type="text" value="Unilever PLC" name="JobTitle" data-ajax-category="Keywords" data-available-categories="displayname" data-ajax-location="/keywords/getAvailableCategories?displayname=&data-all-jobs-text>All Jobs" data-birth-required="false" data-id="34355" data-multiple-select-enabled="false" data-site-id="0" data-includingAllCategory="true" /></p>
      <input id="job-type-#4941aid2cf" name="JobType" type="hidden" value="JobAlert">
      <input id="module-name-#4941aid7cf" name="ModuleName" type="hidden" value="Job Alert Form">
      <input id="job-id-#4941aid2cf" name="JobID" type="hidden" value="34355">
      <input id="job-organization-id-#4941aid2cf" name="OrganizationID" type="hidden" value="34355">
      <input id="redirect:url-#4941aid2cf" name="RedirectURL" type="hidden" value="34355">
      <input id="page:url-#4941aid2cf" name="PageURL" type="hidden" value="careers.unilever.com:80/>
      <input id="page:name-#4941aid2cf" name="PageName" type="hidden" value="Homepage">
      <input id="page-type-#4941aid2cf" name="PageType" type="hidden" value="Home">
      <p class="form-field submit"><input type="submit" value="Submit" /></p>
      <p class="form-field form-message"><a href="#">Cancel</a></p>
    </form>
  </div>

```

Page 8: DOM, “Connect with us” and “Contact us” sections precede “Sign up for job alerts”

Steps to reproduce

1. Observe that there is a help mechanism repeated within a set of pages.
2. On the pages that have the help mechanism, observe that it is not in the same relative order programmatically.

Recommendation

Ensure that the help mechanism repeated within a set of pages is in the same relative order programmatically across the pages.

Assistive technologies navigate content using the order in which elements appear in the DOM. Make sure that the content is in the same relative order programmatically, and is ordered in such a way that it provides an appropriate reading order.

For more information, see [Provide a contact-us link in a consistent location](#), [WCAG - G57 Ordering the content](#) and [Web Dev - Content Reordering](#).

[Return to priority table](#)

ADR_20590-55 MEDIUM PRIORITY OPEN

Interactive elements do not have a permanent label

Success Criteria 3.3.2: Labels or Instructions [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07, 08

Users Affected Cognitive

Component Input Field, Dropdown

What happened

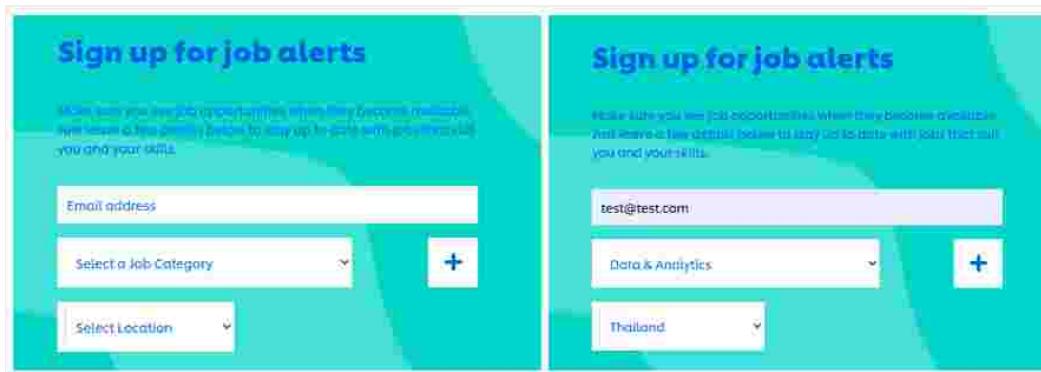
Interactive elements must have labels. In the context of this success criterion, the label is the visual signifier that communicates the nature of the form field. These visual signifiers must appropriately communicate the purpose of the form fields.

There are elements do not have a permanent label. This may particularly disadvantage users that find navigating the web cognitively challenging. Users may struggle to understand the purpose and operation of the interactive elements. The temporary labels may confuse or misguide users.

Where it happened

This can be seen on pages 3 and 8, on desktop and iOS, with the “Email address”, “Select a Job Category” and “Select Location” fields in the “Sign up for job alerts” section. The fields have no permanent label.

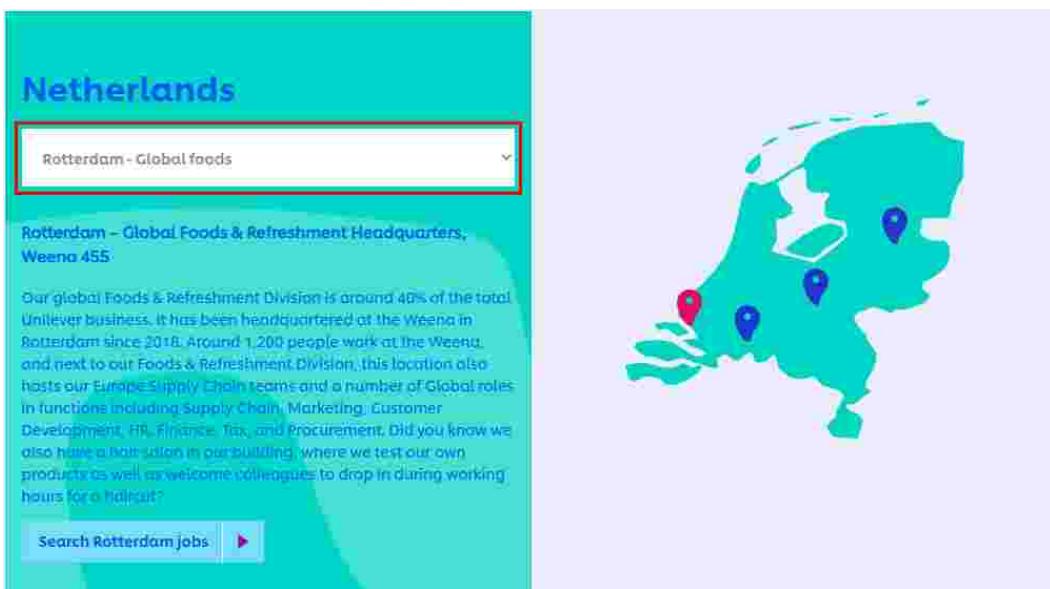
Note that the “Email address” field is visually labelled by the placeholder text “Email address”, which is not permanently available. The “Select a Job Category” and “Select Location” field are visually labelled by the default option in the dropdown, which is only visually available when selected or when the dropdown is expanded.



Page 3: desktop, “Email address”, “Select a Job Category” and “Select Location” fields have no permanent label

This can also be seen on page 7, on desktop and iOS, with the location selector under the “Netherlands” section. The dropdown has no permanent label.

Note that the element is visually labelled only by the currently selected value, such as “Rotterdam - Global foods”.



Page 7: desktop, location selector under "Netherlands" has no permanent label

Steps to reproduce

1. Observe the functionality associated with the interactive elements.
2. Observe the label that is temporarily available to users.
3. Observe that there is no visual signifier that describes the functionality and is permanently available.

Recommendation

Use clear and descriptive labels. Make sure that the labels are visible and permanent.

Make sure that you do not rely on placeholder text. In this context of this success criterion, the visual signifier must be available to all users for as long as the input field is operable. Placeholder text does not meet this requirement as it is not visible once the user has entered some text.

Make sure that the label is not hidden. The label must not only be available as hidden text or through the accessible name of elements

As well, this success criterion allows you to label elements with adjacent elements. For example, the success criterion allows an input field to be labelled by a search button next to the field. In this context, the search button is the visual signifier that communicates the nature of the interactive element.

For more information, see [WCAG - G131 Providing descriptive labels](#), [H71 Providing a description for groups of form controls](#), and [G167 Using an adjacent button](#).

[Return to priority table](#)

ADR_20590-14 MEDIUM PRIORITY OPEN

Interactive element has no role

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01

Users Affected Vision

Component Link, Button

What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

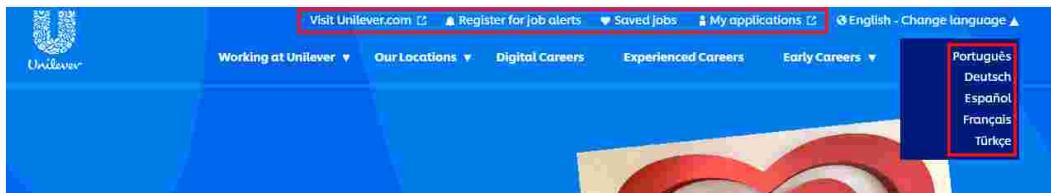
There are interactive elements that do not have an accessible role. This may particularly disadvantage users who interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the role has not been programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 1 on desktop with the following elements, which do not have appropriate interactive roles because their role has been overridden with `role="listitem"`:

- The links and buttons in the top row of the navigation beginning “Visit Unilever.com”, “Register for job alerts”, and so on.
- The language links in the “Change language” panel.
- The links and buttons in the second row of the navigation, beginning “Working at Unilever”, “Our Locations”, and so on.
- The links in the submenus, such as “Working at Unilever”, “Equity, Diversity & Inclusion”, and so on.

Note that this does not include the “Change language” or “Search jobs” buttons, which do have appropriate roles.



Page 1: desktop, top row of navigation links and buttons, and language links in “Change language” panel

```
<div class="core-header__right"></div>
<div class="upper-nav" role="list"></div>
<a class="upper-nav__element external-link" href="https://unilever.com" target="_blank" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Unilever.com" role="listitem" rel="noreferrer">
  Visit Unilever.com
  <span class="visually-hidden">opens in a new window</span>
</a>
<button class="desktop-only desktop-only--inline upper-nav__element upper-nav__element--alerts" id="hide-in-russia" data-custom-event="true" data-custom-category="Custom" data-custom-label="Contact Us" role="listitem">@</button>
<a class="upper-nav__element upper-nav__element--saved-jobs" href="/saved-jobs" data-custom-event="true" data-custom-category="Custom" data-custom-label="Navigation - Saved jobs" role="listitem">@</a>
<a class="upper-nav__element upper-nav__element--login external-link" href="https://unilever.wd3.myworkdayjobs.com/Unilever_Experienced_Professionals/UserHome" target="_blank" data-custom-event="true" data-custom-category="ATS.Sign-in button" data-custom-label="ATS.Sign-in button" role="listitem" rel="noreferrer">@</a>
<button class="upper-nav__element upper-nav__element--language-switcher" aria-expanded="false" aria-label="Submenu for international versions of this website">@</button>
<div class="language-switcher-list" role="list" style="display: none;">
  <a href="https://careers.unilever.com/en/" role="listitem">Português</a>
  <a href="https://karriere.unilever.de/" role="listitem">Deutsch</a>
  <a href="https://careers.unilever.es/" role="listitem">Español</a>
  <a href="https://careers.unilever.fr/" role="listitem">Français</a>
  <a href="https://careers.unilever.com/pt/" role="listitem">Türkçe</a>
</div>
</div>
```

Page 1: DOM, roles of top row of navigation links and buttons and language links have been overridden with `role="listitem"`



Page 1: desktop, second row of navigation links and buttons and “Working at Unilever” submenu links

```
<nav class="main-navigation" role="list"></nav>
<button class="main-navigation__link sub-links-trigger-open" aria-expanded="true" data-submenu-trigger="1" role="listitem" aria-label="Working at Unilever... Top
  <span> submenu </span></button>
<div class="main-navigation__sub-links" data-submenu-target="1" role="listitem" style="display: block;">
  <div class="max-width-wrapper" role="list"><a href="/working-at-unilever" role="listitem" data-custom-event="true"></a>
  <a href="/diversity-and-inclusion" role="listitem" data-custom-event="true" data-custom-category="Custom" data-custom-label="Home - D&I"></a>
  <a href="/benefits-learning-wellbeing" role="listitem" data-custom-label="Benefits, Learning & Wellbeing"></a>
</div>
</div>
```

Page 1: DOM, roles of “Working at Unilever” button and “Working at Unilever” submenu links have been overridden with role=”listitem”

This can also be found on page 1 on iOS with the following elements, which do not have appropriate interactive roles:

- The “Visit Unilever.com”, “Saved jobs”, and “My application” links.
- All of the links and buttons inside the hamburger menu.



Page 1: iOS, “Working at Unilever” button in hamburger menu does not have a role

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the role is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate role. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` which have an accessible role by default. To add roles onto custom components, use the `role` attribute with an appropriate value.

Avoid overriding native link and button roles with other roles, such as list semantics. See issue ADR_20590 - 13.

For example:

```
<button>Submit</button>
<span tabindex="0" role="button">Submit</span>

<a href="...">Homepage</a>
<span tabindex="0" role="link">Homepage</span>
```

For a list of the ARIA roles available, see [ARIA Specification - Role Definitions](#) and [MDN - Using ARIA Roles](#).

For more information, see [WAI-ARIA Practices](#), [Inclusive Components](#), [a11y Project](#), and [Accessibility Developer Guide](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-17 MEDIUM PRIORITY OPEN

Missing or malformed ARIA on editable combobox (with autocomplete)

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop

Pages Affected 01

Users Affected Vision

Component Combobox

What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are comboboxes that are not correctly programmatically determined. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. The malformed semantics may confuse or mislead users.

Where it happened

This can be seen on page 1 on desktop with the location combobox.

- The `<input>` element does not have `role="combobox"`.
- The `<input>` element does not use `aria-expanded` to determine its expanded or collapsed state.
- The currently active option is not determined using `aria-activedescendant`.
- The listbox `` does not have `role="listbox"`.
- The list options do not have `role="option"`.

Note that, instead of giving the currently active option programmatic focus with `aria-activedescendant`, the current option is presented to assistive technology with a live region. This prevents the role of the option elements from being presented to assistive technology.



Page 1: the location combobox

```
<label for="search-location-a5a4e0bc5">Location</label>
<input type="text" id="search-location-a5a4e0bc5" class="search-location mind-reader-results-open" placeholder="location" autocomplete="off" name="l"
value="" maxlength="150" data-lat="54.7584" data-lng="-2.0953" data-kt="1" data-lp="263516" data-ojs-url="/search-jobs/locations?term=" data-
country-codes data-display-typeahead="true" data-error-message="Please try again. Choose a location from the suggestions or clear the field to submit your
search." data-init-val aria-describedby="search-error-1" aria-invalid="false" data-pc="1" data-current-val="United Kingdom" style="z-index: 1;" data-gtm-form-
interact-field-id="0"> =#a
</p>
```

Page 1: DOM, the combobox does not have `role="combobox"`, does not use `aria-expanded`, and does not use `aria-activedescendant` to determine the active option

```

<span role="status" class="mindreader-status" aria-live="polite" id="search-location-af5a4e0bc5-mindreader-status">United Kingdom</span>
<ul class="mindreader-results" id="search-location-af5a4e0bc5-mindreader" style="width: 576px; left: 927px; top: 172px;">
  <li>
    <a href="#" data-target-id="#search-location-af5a4e0bc5" data-lat="39.76" data-lon="-98.5" data-ip="6252001" data-it="2" data-pc class="United States">United States</a>
  </li>
  <li>
    <a href="#" data-target-id="#search-location-af5a4e0bc5" data-lat="54.7584" data-lon="-2.69531" data-ip="2635167" data-it="2" data-pc class="active United Kingdom">United Kingdom</a>
  </li>
  <li>
    <a href="#" data-target-id="#search-location-af5a4e0bc5" data-lat="23.75" data-lon="54.5" data-ip="298557" data-it="2" data-pc class="United Arab Emirates">United Arab Emirates</a>
  </li>
  <li>
    <a href="#" data-target-id="#search-location-af5a4e0bc5" data-lat="18.34829" data-lon="-64.99348" data-ip="4796775" data-it="2" data-pc class="U.S. Virgin Islands">U.S. Virgin Islands</a>
  </li>
  <li>
    <a href="#" data-target-id="#search-location-af5a4e0bc5" data-lat="-3.36667" data-lon="36.85" data-ip="149598-1e1322-161327-11003922-149155" data-it="4" data-pc class="Us-River, Arusha">Us-River, Arusha</a>
  </li>
</ul>

```

Page 1: DOM, listbox element does not have role="listbox", options do not have role="option", and currently active option is determined with a live region

Steps to reproduce

1. Observe the combobox component.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the combobox is not correctly programmatically determined.

Recommendation

Make sure that combobox components are programmatically determined. If possible, use a native `<select>` element. These have out-of-the-box accessibility. If this is not possible, create a combobox with ARIA.

On the `<input>` element of the combobox, use `role="combobox"`. Make sure that the `<input>` is labelled. This can be done with `<label>` or `aria-labelledby`. For more information, see [WAI Tutorials - Labelling Controls](#).

Use `aria-expanded` to differentiate when the listbox is expanded and collapsed. Use the attribute `aria-controls` to identify the element that serves as the popup. Set the `aria-autocomplete` attribute to the value that corresponds to the provided behaviour, such as `aria-autocomplete="inline"`, `aria-autocomplete="list"` or `aria-autocomplete="both"`.

If the combobox popup element has a role other than `listbox`, use the `aria-haspopup` attribute with a value that corresponds to the role of its popup.

On the listbox, use `role="listbox"`. Where possible, expand a list container such as ``. Provide a label for the listbox via `aria-label`. On the options, use `role="option"`. Where possible, expand ``. Use `aria-selected` to differentiate the selected from the unselected options. For example:

```

<label for="ex">Country Search</label>
<input id="ex" type="search" role="combobox"
       aria-activedescendant="01" aria-autocomplete="list"
       aria-expanded="true" aria-controls="results" value="United Kingdom">
<ul id="results" role="listbox" aria-label="Countries List">
  <li id="01" role="option" aria-selected="true">United Kingdom</li>
  <li id="02" role="option">United States</li>
  ...
</ul>

```

Use `aria-activedescendant` to programmatically determine the active option. This attribute uses the `id` of the active option. This is not strictly required for conformance. However, we strongly encourage you to use this attribute.

For more information, see [WAI ARIA Practices - Combobox](#) and [24a11y - Select Your Poison](#).

Make sure that the component can be controlled with a keyboard. The standard approach is that users tab onto the input, type into it, and use Arrow Keys to adjust the listbox. This can be done in parallel with the aforementioned `aria-activedescendant` and `aria-selected`.

For more information, see [Managing focus using aria-activedescendant](#) and [MDN - Keyboard-navigable widgets](#).

[Return to priority table](#)

ADR_20590-24 MEDIUM PRIORITY OPEN

Expandable element is not programmatically identified

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 01, 11

Users Affected Vision

Component Button

What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are expandable elements that are not programmatically determined as such. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the state is not programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 1 on desktop and iOS with the job category search field, which triggers a panel containing related job links when it receives input. The `<input>` element does not have `aria-expanded`.



Page 1: desktop, job category search field with panel

```
<p style="position: relative;">
  <label for="search-keyword-ac2f2c87ce">Search job title/Keyword</label>
  <input type="search" id="search-keyword-ac2f2c87ce" class="search-keyword" placeholder="Search job title/keyword" autocomplete="off" name="k" value="" maxlength="150" data-current-val="market">
  <div class="auto-complete-keyword-popup" tabindex="-1" id="search-keyword-ac2f2c87ce-auto-complete-popup" style="position: absolute; display: Block; width: 576px; left: 0px; top: 55px; z-index: 10001;">
    <div class="auto-complete-keyword-popup-section" data-selector-name="autocomplete" data-post-module="true" data-ajax-url="//careers.unilever.com/module/postmodule" data-module-name="Search form autocomplete" data-module-type="AutoComplete">
      <h2 class="expandable-parent">Jobs By Category</h2>
      <ul>
        <li>
          <a href="/category/marketing-jobs/34155/8110080/1" data-count="66" data-category="Marketing" data-percent="100%" data-quartile="4">...</a>
        </li>
      </ul>
      <h2 class="expandable-parent">Jobs</h2>
      <ul>...</ul>
      <h2 class="expandable-parent">Jobs By Title</h2>
      <ul>...</ul>
    </div>
  </div>
</p>
<p>...</p>
<p>...</p>
<input id="search-org-ids-ac2f2c87ce" name="orgIds" type="hidden" value="34155">
<button id="search-submit-ac2f2c87ce" onclick="APP.MODELS.GoogleBot.sendCustomDimensions('Job Search', 'Click', 'Supply Chain - Search Form - Clicks', 'event')">...</button>
<n class="search-location-error" aria-hidden="true" id="search-error-1" style="outline: 0 !important"></n>
```

Page 1: DOM, job category search field does not have aria-expanded

This can be found on page 1 on iOS with the expandable buttons in the hamburger menu. The elements do not have a programmatically determined expanded or collapsed state.

Note that this may be caused by the overridden button roles. See issue ADR_20590 - 14.

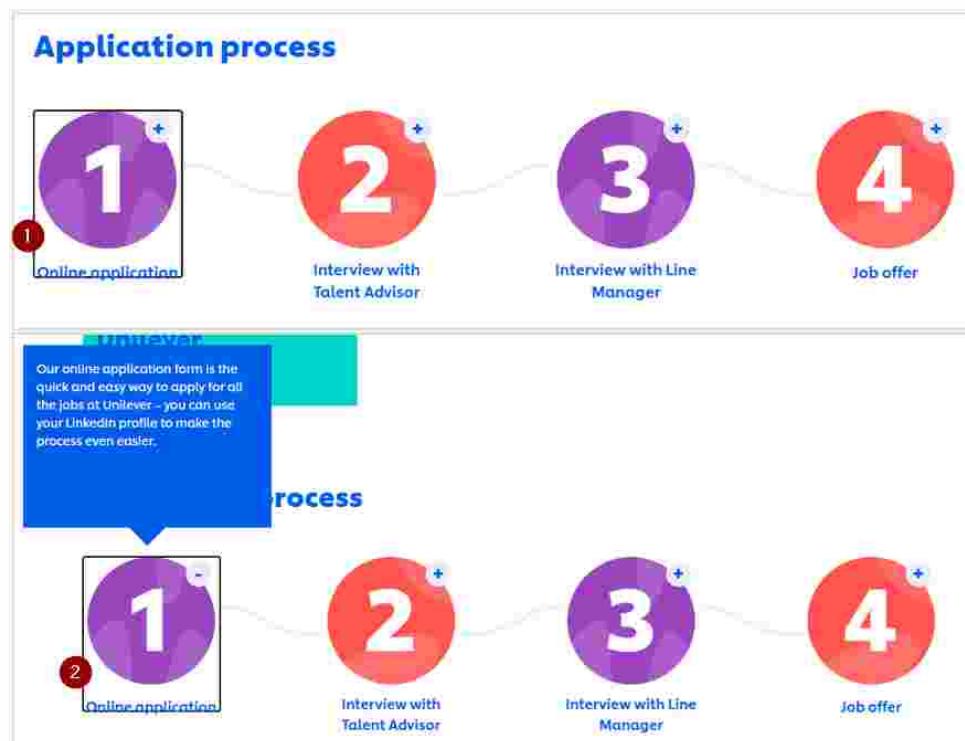


Page 1: iOS, "Working at Unilever" button does not have expandable state

```
<div class="main-navigation" role="list" style="display: block;">
  <button class="main-navigation__link sub-links-trigger" aria-expanded="false" data-submenu-trigger="1" role="listitem" aria-label="Working at Unilever. Toggles submenu">
    Working at Unilever
    <div>
    </div>
  </button>
  <div class="main-navigation__sub-links" data-submenu-target="1" role="listitem">
    <a href="/working-at-unilever" role="listitem" data-custom-event="true">Working at Unilever</a>
    <a href="/diversity-and-inclusion" role="listitem" data-custom-event="true" data-custom-category="Custom" data-custom-label="None = DM">Equity, Diversity & Inclusion</a>
    <a href="/benefits-learning-wellbeing" role="listitem">Benefits, Learning & Wellbeing</a>
  </div>
</div>
```

Page 1: DOM, "Working at Unilever" button has an aria-expanded attribute but the button role is overridden with a list item role

This can also be seen on page 11, on desktop and iOS, with the expandable content under the “Application process” section. The “Online application”, “Interview with Talent Advisor”, and so on expandable buttons do not have a programmatically determined expanded or collapsed state.



Page 11: desktop, example of expandable “Online application” button

```

<section class="application-process max-width-wrapper max-width-wrapper--narrow">
  <h2 class="application-process__header">Application process</h2>
  <div class="application-process__grid">flex
    <div class="application-process__step one">
      <button class="application-process__number-button one" data-custom-event="true" data-custom-category="Custom_Event" data-custom-label="Application process">flex
        <span class="application-process__number-button_copy">Online application</span> flex
        ::after flex
      </button>
      <div class="application-process__hidden-content">@</div>
    </div>
    <div class="application-process__line one"></div>
    <div class="application-process__step two">@</div>
    <div class="application-process__line two"></div>
    <div class="application-process__step three">@</div>
    <div class="application-process__line three"></div>
    <div class="application-process__step four">@</div>
    </div>
  </section>

```

Page 11: DOM, example of expandable “Online application” button with no expandable attribute

Steps to reproduce

1. Observe the expandable interactive elements.
2. Navigate onto the expandable elements with a screen reader.
3. Review the screen reader output.
4. Observe that the expandable state is not programmatically determined.
5. Observe that the state is not available through text.

Recommendation

Make sure that the expandable state is programmatically determined.

Use **aria-expanded** and update the **true** and **false** value appropriately. Make sure that the attribute is on the interactive elements themselves, rather than parent or child elements.

For the expanded state to work as expected, make sure that the elements have appropriate programmatically determined roles.

For example:

```
<button aria-expanded="true">This is expanded</button>
<button aria-expanded="false">This is collapsed</button>
<span role="button" aria-expanded="true" tabindex="0">This is expanded</span>
<span role="button" aria-expanded="false" tabindex="0">This is collapsed</span>
```

For more information, see [WAI-ARIA Practices - Accordion](#), [WAI ARIA - Disclosure](#), and [ADG - Marking elements expandable](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-43 MEDIUM PRIORITY OPEN

ARIA has been defined on the incorrect element

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 04

Users Affected Vision

Component Button

What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are elements which have attempted to use ARIA. However, the ARIA is malformed. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. The malformed semantics may confuse or mislead users.

Where it happened

This can be seen on page 4 on desktop and iOS with the filter triggering buttons such as “Category”, “Location”, and so on. These buttons expand to disclose panels with filtering checkboxes. However, **aria-expanded** is defined on the **** for the disclosed panel, rather than on the triggering button.

Note that the value of the **aria-expanded** attribute also does not consistently match the expanded or collapsed state of the element.

Note that this issue only occurs after ticking a filter checkbox. When the page is initially loaded and no checkboxes are ticked, the **aria-expanded** attribute is appropriately placed on the triggering buttons.



Page 4: desktop, filter triggering buttons with “Category” expanded

```
<div>
  <h2>Filter your search</h2>
  <section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="1" data-expandable-id="#dzewykmqum" class="expandable">
    <button class="expandable-parent expandable-child-open" id="category-toggle">
      "Category"
      ::after
    </button>
    <ul class="search-filter-list expandable-childlist-open" aria-hidden="false" aria-expanded="true"></ul>
  </section>
  <section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="2" data-expandable-id="#99urqw3v3w" class="expandable">
    <button class="expandable-parent" id="country-toggle">
      "Location"
      ::after
    </button>
    <ul class="search-filter-list" aria-hidden="false" aria-expanded="true">
      <li></li> <li></li>
      <li></li> <li></li>
    </ul>
  </section>
</div>
```

Page 4: DOM, aria-expanded attribute is on the disclosed content instead of the triggering buttons, and the value does not update appropriately

Steps to reproduce

1. Tick at least one checkbox in the search filters.
2. Observe the interactive elements.
3. Inspect the elements and review the DOM.
4. Review the computed accessibility properties.
5. Observe that the ARIA is malformed.

Recommendation

Make sure that the ARIA is placed on the correct element. Most ARIA is used on interactive elements. The attributes must be on the interactive elements themselves, rather than parent or child elements. For example:

```
<button aria-expanded="true">This is expanded</button>
<span role="button" aria-expanded="true" tabindex="0">This is expanded</span>
```

For more information, see [WAI-ARIA Practices](#), [Inclusive Components](#), [a11y Project](#), and [Accessibility Developer Guide](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-44 MEDIUM PRIORITY OPEN

Video element with no accessible name

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop

Pages Affected 03

Users Affected Vision

Component Video

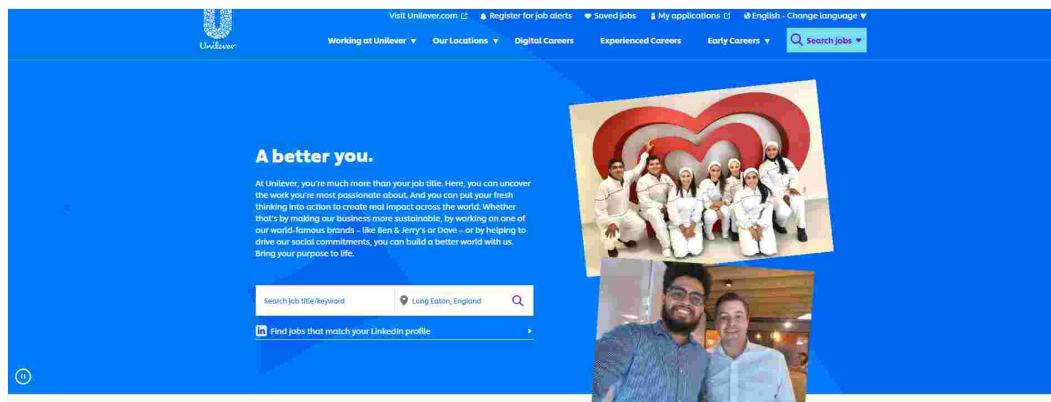
What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

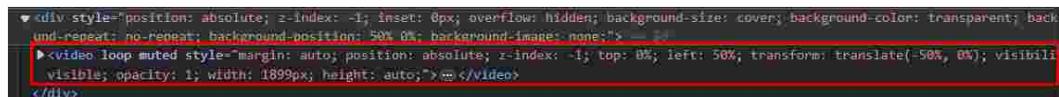
There are interactive elements that do not have an accessible name. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the accessible name is not programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 3, on desktop, with the hero banner video. The `<video>` element does not have an accessible name.



Page 3: desktop, hero banner video



Page 3: DOM, hero banner video

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the name is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate name. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` and provide them with an accessible name through labels or internal text. For example:

```
<label for="firstName">First Name</label>
<input id="firstName" type="text" autocomplete="given-name">
<button>Submit</button>
```

When there is no visible text and you cannot add text to the page, use **aria-label** to manually name elements. For example:

```
<button class="closeIcon" aria-label="Close Modal Dialog"></button>
<a href=".../example.pdf" download class="pdfIcon" aria-label="Download PDF report">
</a>
```

For more information, see [WAI Tutorials - Labelling Controls](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-47 MEDIUM PRIORITY OPEN

Interactive elements with no accessible name

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03, 07, 08, 13

Users Affected Vision

Component Link, Button

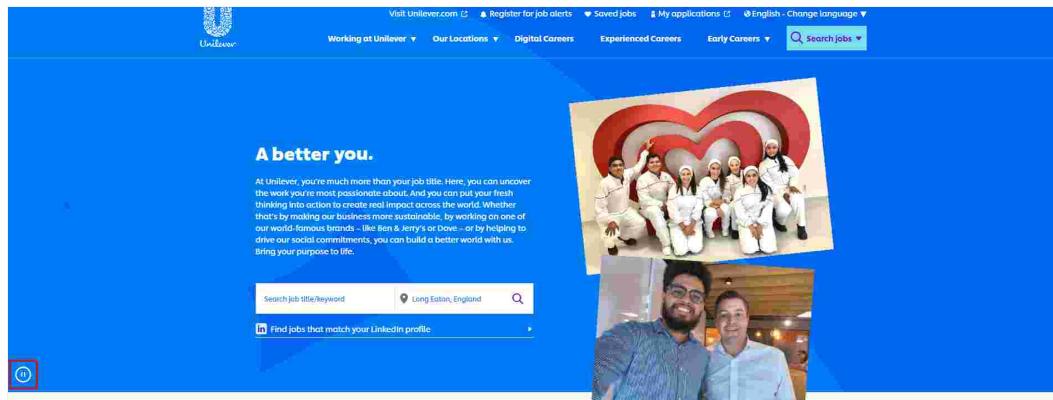
What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are interactive elements that do not have an accessible name. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the accessible name is not programmatically determined may confuse or mislead users.

Where it happened

This can be seen on page 3, on desktop, with the pause/play button provided for the hero banner video. The pause/play button does not have an accessible name.



Page 3: desktop, hero banner video pause/play button

```
<div style="position: absolute; z-index: -1; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-position: 50% 0%; background-image: none;">
  <video loop muted style="margin: auto; position: absolute; z-index: -1; top: 0%; left: 50%; transform: translate(-50%, 0%); visibility: visible; opacity: 1; width: 1899px; height: auto;"></video>
</div>
<div class="play-pause-button" paused="false"></div>
```

Page 3: DOM, hero banner video pause/play button

This can also be seen on page 7, on desktop and iOS, with the dismiss button in the “Inclusive policies” panel under the “What we offer” section. The dismiss button does not have an accessible name.

Note that only the “Inclusive policies” panel was in scope for this audit. However, this behaviour is likely to also affect the other panels.

What we offer

Here at Unilever, we believe that our brands with purpose grow, but that this is only made possible by our people. Therefore, it's just as important that people with purpose thrive within our company. We encourage our employees to live their purpose everyday and offer whole range of benefits, including:

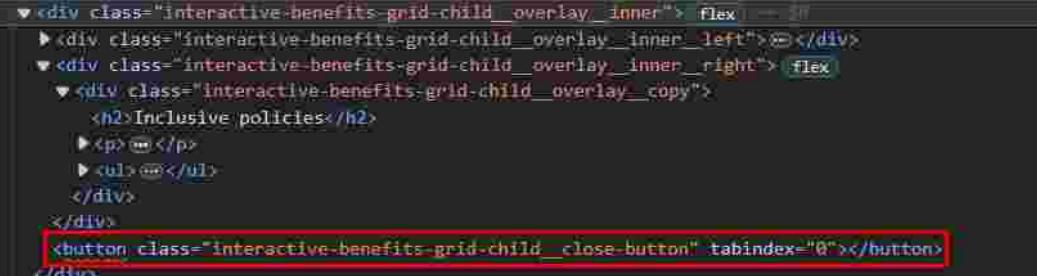


Inclusive policies

We encourage an inclusive culture, which comes to life in for instance through the option to swap public holidays, use paid paternity leave of 6 weeks, and through our transgender policy.

- Interchangeable holidays: Do you celebrate Diwali, Eid, Hanukkah, Lunar New Year or other holidays that are not reflected by the official National Public Holidays in the Netherlands? In line with our Diversity, Equity, and Inclusion priorities, we want to better represent all employees that work at Unilever in the Netherlands. This comes with the recognition that cultural or religious holidays are equally important, but individually driven. This special leave policy allows you to take these days and celebrate your holidays by choosing to exchange/swap holidays.
- Paternity leave: next to the regular 16 weeks maternity leave we offer to our female colleagues, we offer 6 weeks fully paid paternity leave after the birth of a child for the partner;

Page 7: desktop, dismiss panel button



```
<div class="interactive-benefits-grid-child__overlay_inner"> ... </div>
  <div class="interactive-benefits-grid-child__overlay_inner_left"> ... </div>
  <div class="interactive-benefits-grid-child__overlay_inner_right"> ... </div>
    <div class="interactive-benefits-grid-child__overlay_copy">
      <h2>Inclusive policies</h2>
      <p>...</p>
      <ul>...</ul>
    </div>
  </div>
  <button class="interactive-benefits-grid-child__close-button" tabindex="0"></button>
</div>
```

Page 7: DOM, dismiss panel button

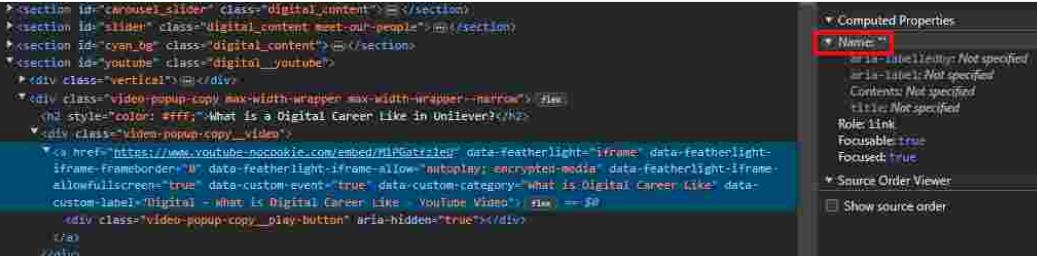
This can be seen on page 8 with the “What is a Digital Career Like in Unilever?” video link. The link does not have an accessible name.



What is a Digital Career Like in Unilever?



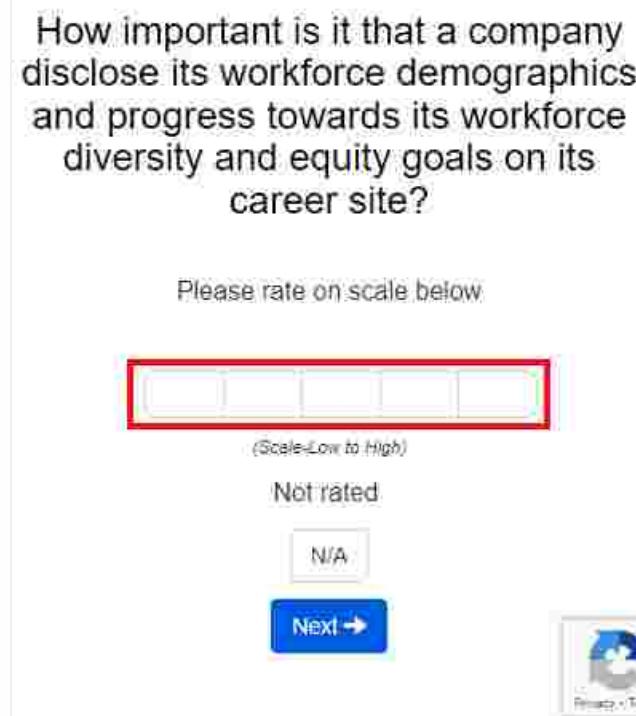
Page 8: desktop, “What is a Digital Career Like in Unilever?” video link



```
<section id="carousel_slider" class="digital_content"> ... </section>
<section id="slider" class="digital_content meet-our-people"> ... </section>
<section id="cyan_bg" class="digital_content"> ... </section>
<section id="youtune" class="digital_youtube">
  <div class="vertical"> ... </div>
  <div class="video-popup-copy max-width-wrapper max-width-wrapper-narrow"> ... </div>
    <h2 style="color: #fff;">What is a Digital Career Like in Unilever?</h2>
    <div class="video-popup-copy_video">
      <a href="https://www.youtube-nocookie.com/embed/HlPQatZleU" data-featherlight-iframe="Frameborder=0" data-featherlight-allow="autoplay encrypted-media" data-featherlight-iframe-allowfullscreen="true" data-custom-event="true" data-custom-category="What is Digital Career Like" data-custom-label="Digital - What is Digital Career Like - YouTube Video" data-ssl="1" data-title="Digital - What is Digital Career Like - YouTube Video" data-unique="1"> ... </a>
      <div class="video-popup-copy__play-button" aria-hidden="true"> ... </div>
    </div>
  </div>
</div>
```

Page 8: DOM, video link does not have an accessible name

This can be seen on page 13 on desktop and iOS with the ratings buttons under “Please rate on scale below”. The buttons do not have accessible names.



Page 13: desktop, ratings buttons

```
<div style="margin-left:15px;margin-top:40px">
  <div class="btn-group" role="group">
    <!-- ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button"
      'border-radius:first':label.firstButton,
      'border-radius:last':label.lastButton" ng-style="{}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button"
      'border-radius:first':label.firstButton,
      'border-radius:last':label.lastButton" ng-style="{}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button"
      'border-radius:first':label.firstButton,
      'border-radius:last':label.lastButton" ng-style="{}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button"
      'border-radius:first':label.firstButton,
      'border-radius:last':label.lastButton" ng-style="{}"
    <!-- end ngRepeat: label in q.hilo.labels -->
    <button ng-repeat="label in q.hilo.labels" class="btn btn-default check-button"
      'border-radius:first':label.firstButton,
      'border-radius:last':label.lastButton" ng-style="{}"
    <!-- end ngRepeat: label in q.hilo.labels -->
  </div>
```

Page 13: DOM, first ratings button does not have an accessible name

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the name is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate name. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` and provide them with an accessible name through labels or internal text. For example:

```
<label for="firstName">First Name</label>
<input id="firstName" type="text" autocomplete="given-name">
<button>Submit</button>
```

When there is no visible text and you cannot add text to the page, use **aria-label** to manually name elements. For example:

```
<button class="closeIcon" aria-label="Close Modal Dialog"></button>
<a href=".../example.pdf" download class="pdfIcon" aria-label="Download PDF report">
</a>
```

For more information, see [WAI Tutorials - Labelling Controls](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-48 MEDIUM PRIORITY OPEN

Button element has no role

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 03

Users Affected Vision

Component Button

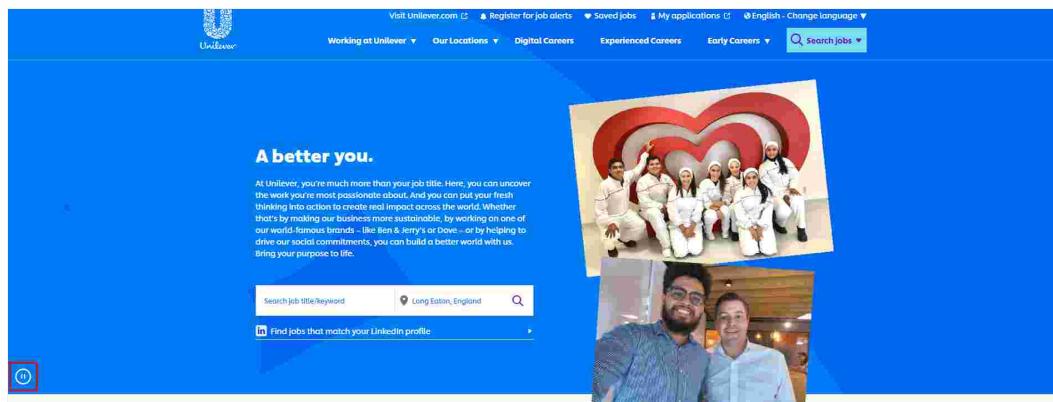
What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are buttons elements that do not have an appropriate accessible role. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the role has not been programmatically determined may confuse or mislead users.

Where it happened

This can be seen on page 3, on desktop, with the pause/play button provided for the hero banner video. The pause/play button does not have a programmatically determined role.



Page 3: desktop, hero banner video pause/play button

```
<div style="position: absolute; z-index: -1; inset: 0px; overflow: hidden; background-size: cover; background-color: transparent; background-repeat: no-repeat; background-position: 50% 0%; background-image: none;">
  <video loop muted style="margin: auto; position: absolute; z-index: -1; top: 0%; left: 50%; transform: translate(-50%, 0%); visibility: visible; opacity: 1; width: 1899px; height: auto;"></video>
</div>
<div class="play-pause-button" paused="false"></div>
```

Page 3: DOM, hero banner video pause/play button does not have a role

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the button role is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate role. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` which have an accessible role by default. To add roles onto custom components, use the `role` attribute with an appropriate value. For example:

```
<button>Submit</button>  
<span tabindex="0" role="button">Submit</span>
```

For a list of the ARIA roles available, see [ARIA Specification - Role Definitions](#) and [MDN - Using ARIA Roles](#).

For more information, see [MDN - Button element](#), [MDN Accessibility - ARIA button role](#), and [WAI ARIA Practices - Button](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-93 MEDIUM PRIORITY OPEN

Pressed element is not programmatically identified

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 13

Users Affected Vision

Component Button

What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are pressed elements that are not programmatically determined as such. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the pressed state is not programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 13 on desktop and iOS with the ratings buttons. After selecting a rating and navigating back to this page, the pressed state of the currently pressed rating button is not programmatically determined.



Page 13: desktop, first rating button is pressed

The screenshot shows a SonarQube code review interface. On the left, there is a block of AngularJS code within a `div.btn-group`. The code uses `ng-repeat` to generate multiple buttons based on the value of `q.hilo.labels`. Each button has a class of `btn btn-default check-button` and a scope of `three-five-labels`. The buttons have border-radius values of `border-radius-first:label.firstButton` and `border-radius-last:label.lastButton`, and a background color style of `background-color:label.b`.

On the right, the analysis results are displayed:

- generic**:
 - generic**:
 - group**:
 - button**

ARIA Attributes: No ARIA attributes.

Computed Properties:

 - Name:** ""
 - `aria-labelledby: Not specified`
 - `aria-label: Not specified`
 - `From label: Not specified`
 - `Contents: Not specified`
 - `title: Not specified`

Role: button

Page 13: DOM, pressed rating button does not have pressed state

Steps to reproduce

1. Observe the pressable interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the pressed state is not programmatically determined.
5. Observe that the state is not available through text.

Recommendation

Make sure that the pressed state is programmatically determined.

Use **aria-pressed** and update the **true** and **false** value appropriately. Make sure that the attribute is on the interactive elements themselves, rather than parent or child elements. For example:

```
<button aria-pressed="true">This is pressed</button>
<button aria-pressed="false">This is not pressed</button>
<span role="button" aria-pressed="true" tabindex="0">This is pressed</span>
<span role="button" aria-pressed="false" tabindex="0">This is pressed</span>
```

For more information, see [WAI-ARIA Practices - Toggle Button](#), [WAI ARIA - Disclosure](#), and [ADG - Marking elements activatable](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-107 MEDIUM PRIORITY OPEN

Card disclosure element has no role

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Mobile (iOS)

Pages Affected 11

Users Affected Vision

Component Button

What happened

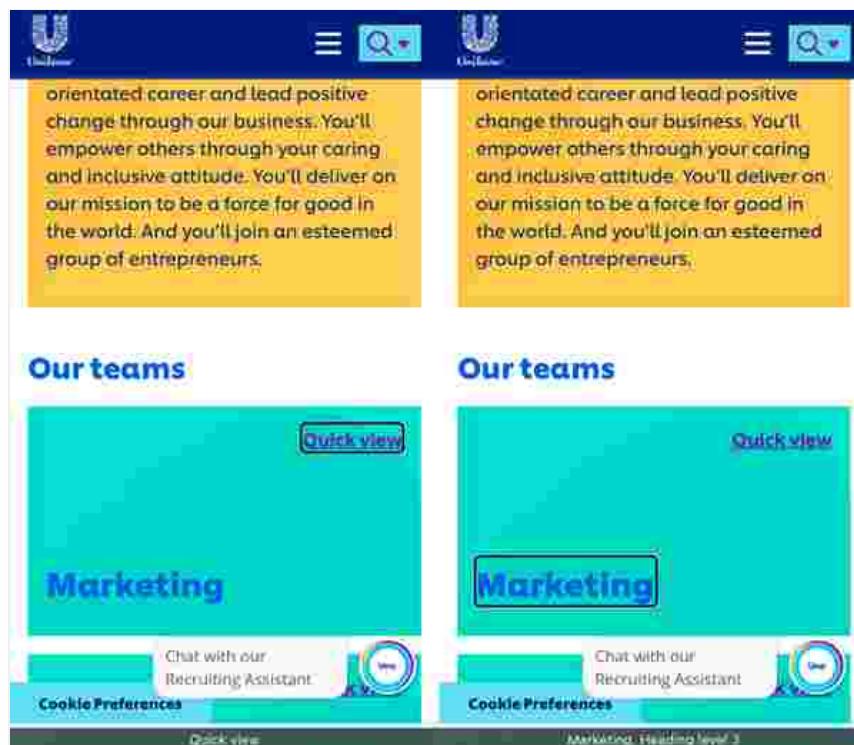
Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are interactive elements that do not have an accessible role. This may particularly disadvantage users who interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the role has not been programmatically determined may confuse or mislead users.

Where it happened

This can also be seen on page 11 on iOS with the “Quick view” and heading element, such as “Marketing”, in each card under “Our teams”. These elements can be activated with a screen reader to disclose further content. However, they do not have appropriate interactive roles, such as “Button”.

Note that, on desktop, the cards are triggered on mouse hover or when the “Find out more” or “Search jobs” link inside the disclosure receives keyboard focus.



Page 11: iOS, “Quick view” text and “Marketing” heading do not have roles

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.

4. Observe that the role is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate role. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` which have an accessible role by default.

In this case, to avoid having separate elements with the same functionality, we recommend removing the “Quick view” element and giving “Marketing” a button role, or removing the flipcard triggering functionality from the “Marketing” element and giving the “Quick view” element a button role.

To add roles onto custom components, use the `role` attribute with an appropriate value. For example:

```
<button>Submit</button>
<span tabindex="0" role="button">Submit</span>

<a href="...">Homepage</a>
<span tabindex="0" role="link">Homepage</span>
```

For a list of the ARIA roles available, see [ARIA Specification - Role Definitions](#) and [MDN - Using ARIA Roles](#).

For more information, see [WAI-ARIA Practices](#), [Inclusive Components](#), [a11y Project](#), and [Accessibility Developer Guide](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-111 MEDIUM PRIORITY OPEN

Interactive elements relies on accessible description for its accessible name

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12

Users Affected Vision

Component Button

What happened

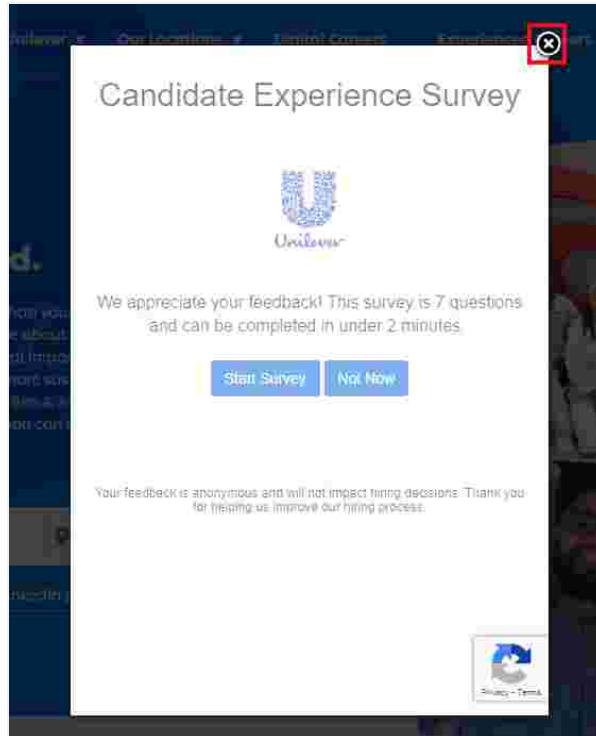
Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are interactive elements that do not have an accessible name. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the accessible name is not programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 12 on desktop and iOS with the close button for the “Candidate Feedback” survey modal. The close button has an accessible name of an empty space caused by the non-breaking space character inside the button.

Note that the text “Close” from the **title** attribute is still associated with the button as an accessible description, and was observed to be announced by assistive technology. However, this is not a robustly determined accessible name.



Page 12: desktop, close button for survey modal

The screenshot shows the AbilityNet Accessibility Inspector interface. On the left is the DOM tree with several nodes highlighted in blue, indicating they are interactive elements. On the right is a panel with tabs for 'Styles', 'Computed', 'Layout', 'Event Listener', and 'Accessibility'. The 'Accessibility' tab is active, showing the 'generic' section which includes a 'button' node under 'StaticText'. The 'Name' field is empty, and the 'Description' field contains the value 'Close'. Other properties shown include 'aria-labelledby' (Not specified), 'aria-label' (Not specified), 'aria-describedby' (Not specified), 'Role' (button), 'Invalid user entry' (false), and 'Focusable' (true). A red box highlights the 'Description' field.

Page 12: DOM, close button has empty accessible name and relies on accessible description for its name

Steps to reproduce

1. Observe the interactive elements.
2. Inspect the elements and review the DOM.
3. Review the computed accessibility properties.
4. Observe that the name is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate name. Where possible, use native elements, such as `<button>`, `<input type="...">`, ``, and `<select>` and provide them with an accessible name through labels or internal text. For example:

```
<label for="firstName">First Name</label>
<input id="firstName" type="text" autocomplete="given-name">
<button>Submit</button>
```

When there is no visible text and you cannot add text to the page, use `aria-label` to manually name elements. For example:

```
<button class="closeIcon" aria-label="Close Modal Dialog"></button>
<a href=".../example.pdf" download class="pdfIcon" aria-label="Download PDF report">
</a>
```

For more information, see [WAI Tutorials - Labelling Controls](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

ADR_20590-53 MEDIUM PRIORITY OPEN

Search results not programmatically identified

Success Criteria 4.1.3: Status Messages [2.1 AA]

Devices Desktop

Pages Affected 04

Users Affected Vision

Component Plain Text

What happened

Dynamic updates need to be programmatically determined.

There are dynamic search results that are not programmatically determined as such. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the updates are not programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 4 on desktop with the search results summary heading, such as “5 search results”. This element updates dynamically along with the search results when filters are applied. However, it is not programmatically determined as a status message.

The screenshot shows a search interface with a header 'Filter your search' and dropdown menus for 'Category', 'Location' (set to 'Remote (5)'), and 'Remote working'. A red box highlights the text '5 Search results' in blue, which is positioned above a 'Filtered by: Remote' section. Below this, there are three job listing cards: 'Associate Brand Manager-Liquid I.V. International', 'Manager, First Party Data Strategy', and 'Territory Development Manager - Las Vegas, NV'. Each card has a small heart icon and the text 'Multiple locations'.

Page 4: desktop, “5 search results” message

```
<section id="search-results" data-keywords="Marketing" data-location="United States" data-distance="100" data-latitude="39.76" data-longitude="-98.5" data-show-radius="false" data-total-results="8" data-total-job-results="5" data-total-pages="1" data-current-page="1" data-records-per-page="12" data-active-facet-id="0" data-custom-facet-name data-facet-term data-facet-type="0" data-search-results-module-name="Search Results" data-sort-criteria="0" data-sort-direction="0" data-search-type="1" data-ajax-url="/search-jobs/results" data-ajax-post-url="/search-jobs/results/post" data-server-error-message="There was an error retrieving your results. Please refresh and try again." data-keyword-type data-location-type="2" data-location-path="6252001" data-organization-ids="54195" data-selection-name="searchresults" data-refined-keywords="[]" data-postal-code data-save-jobs="true" data-results-type="0" data-use-noindex="true" data-index="1">
  <div>5 Search results</div>
  <div id="search-results-list">
    <div id="applied-filters" class="search-results-options">
      <div>
        <h2 id="applied-filters-label">Filtered by:</h2>
        <ul aria-labelledby="applied-filters-label" role="list"><li></li></ul>
      </div>
    </div>
    <ul class="global-job-list--white" role="list"><li></li></ul>
  </div>
</section>
```

Page 5: DOM, “5 search results” message is not programmatically determined as a status message

Steps to reproduce

1. Observe the dynamic search results that are visually communicated.
2. Inspect the update and review the DOM.
3. Note that the dynamic search results are not programmatically determined.

Recommendation

Make sure that status messages are programmatically determined. To achieve this, use a polite live region, such as **role="status"**, and **aria-atomic** to make sure that the entire text node is always announced.

For live regions to work, an empty container with live region semantics must be present on page load (before the region message is triggered). The live region message needs to be dynamically added to the container when triggered. For example, rather than toggling the visibility of the text using CSS. Also note, that you need to empty this container again, before dynamically adding a new message. For example:

```
<span role="status" aria-atomic="true"></span>
<span role="status" aria-atomic="true">10 results found</span>
<span role="status" aria-atomic="true"></span>
<span role="status" aria-atomic="true">No results found</span>
```

For more information, see [MDN - Using the status role](#) and [MDN - ARIA Live Regions](#).

[Return to priority table](#)

ADR_20590-79 LOW PRIORITY OPEN

Decorative icon not hidden from assistive technology

Success Criteria 1.1.1: Non-text content [2.0 A]

Devices Desktop

Pages Affected 08, 13

Users Affected Vision

Component Image

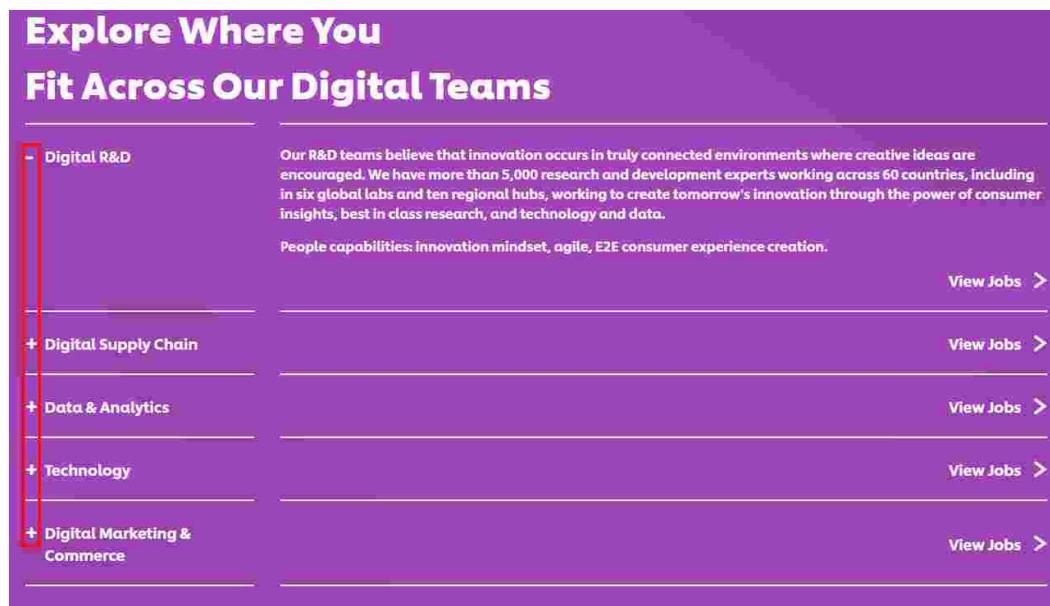
What happened

Non-text content must have appropriate alternatives. This is required for the content to be understood by all users. Images, icons and so on may otherwise be miscommunicated by assistive technology. There are decorative icons that have not been hidden from assistive technology.

We have classed the images as decorative as they do not communicate any information that is not otherwise available on the page.

Where it happened

This can be seen on page 8 on desktop with the plus and minus icons beside the accordion buttons “Digital R&D”, “Digital Supply Chain”, and so on. The icons are decorative because the state of the button is communicated by the **aria-expanded** attribute. However, the icons are exposed to assistive technology.



Page 8: desktop, plus and minus icons for accordion buttons

The screenshot shows the DOM structure for the "Digital R&D" accordion item. A red box highlights the minus icon inside the "is-focused" state of the "is-expanded" attribute. The developer tools show the CSS rule for the "contents" pseudo-element:

```
.digital-page section#accordion .accordion-item .contents::before { content: "\2212"; font-size: 1.5em; line-height: 0.5em; transition: transform 0.3s ease-in-out; left: 0; top: 5px; position: absolute; }
```

Page 8: DOM, minus icon for expanded “Digital R&D” button is exposed to assistive technology

This can also be seen on page 13 on desktop and iOS with the arrow icon in the “Next” button. This icon is decorative because it does not communicate information beyond what the button text communicates. However, it is exposed to assistive technology.

Please rate on scale below

(Scale-Low to High)

Not rated

N/A

Next →

Privacy Terms

Page 13: desktop, “Next” button with arrow icon

```
<div ng-style="{'margin-top': e.marginTop}" ng-hide="submit || introQ" class="style">
  <span ng-hide="firstQ" class="ng-hide" style></span>
  <button class="btn btn-primary mobile-button-default ng-binding" ng-style="{'background-color': survey.buttonTextColor || 'white'}" ng-click="nextQ=true; $ctrl.next(); $ctrl.$emit('nextQ');">
    Next <span class="glyphicon glyphicon-arrow-right">
      ::before == $0
    </span>
  </button>
</div>
<div ng-show="q.required && !answered" class="margin-top-15 required-answer ng-binding">
```

Page 13: DOM, arrow icon is exposed to assistive technology

Steps to reproduce

1. Observe the icons.
2. Observe that they do not visually communicate any more information than is other available.
3. Observe that they do not provide any further context or clarity than the information on the page.
4. Inspect the icons.
5. Confirm that they have not been hidden from assistive technology.

Recommendation

Decorative images that do not convey meaning or unique information need to be programmatically hidden.

Images marked up with an `` tag must have an `alt` attribute. Use an empty `alt` attribute to programmatically hide decorative images. For example:

```
Error: The email address field cannot be left blank
```

Inline SVG images can use `aria-hidden` and `focusable="false"` for the same purpose. For example:

```
<svg aria-hidden="true" focusable="false">
  ...
</svg>
```

The **aria-hidden** attribute can also be used with icon fonts and CSS generated content. For example:

```
<div>
  <span aria-hidden="true">::before</span>
  Error: The email address field cannot be left blank
</div>
<div>
  <i class="fas fa-exclamation-triangle" aria-hidden="true"></i>
  Error: The email address field cannot be left blank
</div>
```

For decorative **<svg>** elements, use the **aria-hidden="true"** and **focusable="false"** attributes. For more information, visit [CSS Tricks - Accessible SVGs](#).

For decorative icon fonts and CSS-generated content, use the **aria-hidden="true"** attribute. For more information, visit [MDN - aria-hidden, a11y Project - How to Hide Content](#), and [CloudFour - Hidden Content and Accessibility](#).

For more information, see [Web Accessibility Tutorials - Informative Images](#), and [Alt Attribute Decision Tree](#).

[Return to priority table](#)

ADR_20590-35 LOW PRIORITY OPEN

Unordered list not marked up semantically

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 04, 05

Users Affected Vision

Component List

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There are elements that are visually communicated as being interrelated, unordered list items. However, this structure has not been programmatically determined.

Where it happened

This can be seen on page 4 on desktop and iOS with the filter triggering buttons such as “Category”, “Location”, and so on under “Filter your search”. These elements are visually communicated as being interrelated, unordered list items. However, they are not marked up as such.

The screenshot shows a user interface for filtering search results. At the top, there's a header 'Filter your search'. Below it is a row of four dropdown menus labeled 'Category', 'Location', 'State', and 'City', followed by a 'Remote working' dropdown and a 'Remove filters' link. All these elements are highlighted with a red box. Below this row, the text '5 Search results' is displayed in blue.

Page 4: desktop, filter triggering buttons are visually a list

```
<h2>Filter your search</h2>
<section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="1" data-expandable-id="thyjrr7ijhf" class="expandable">
  <button class="expandable-parent" id="category-toggle">
    "Category"
    <:after>
  </button>
  <ul class="search-filter-list" aria-hidden="false" aria-expanded="true"></ul>
</section>
<section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="2" data-expandable-id="37du9qexvn" class="expandable">
  <button class="expandable-parent" id="country-toggle">
    "Location"
    <:after>
  </button>
  <ul class="search-filter-list" aria-hidden="false" aria-expanded="true"></ul>
</section>
<section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="3" data-expandable-id="1shncszujck" class="expandable"></>
<section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="4" data-expandable-id="a7u4091mz7h" class="expandable"></>
<section data-expandable="true" data-speed="0" data-enabled="true" data-filter-id="5" data-expandable-id="69np4tpxom4" style="display: none;" class="expandable"></>
```

Page 4: DOM, filter buttons are not marked up as a list

This can be seen on page 5 on desktop and iOS, with the “Culture for Growth”, “Top Notch Employee Health & Well Being Benefits”, “Every Voice Matters” and so on items that are present under the text “What We Can Offer You”. These elements are visually communicated as being interrelated, unordered list items. However, they are not marked up as such.

- Experience working with different customer channels
- Ability to operate in a fast-paced, dynamic environment, and to provide innovative approaches in a team-based structure
- Ability to deliver both verbal and written communication in a simple, compelling manner
- Ability to think strategically, both from a brand and portfolio view

What We Can Offer You

[Culture for Growth](#) | [Top Notch Employee Health & Well Being Benefits](#) | [Every Voice Matters](#)
[Global Reach](#) | [Life at Unilever](#) | [Careers with Purpose](#) | [World Class Career Development Programs](#) | [Check Out Our Space](#) | [Focus On Sustainability](#)

Pay: The pay range for this position is \$110,700 to \$166,000. Unilever takes into consideration a wide range of factors that are utilized in making compensation decisions including, but not limited to, skill sets, experience and training, licensure and certifications, qualifications and education, and other business and organizational needs.

Bonus: This position is bonus eligible.

Page 5: desktop, related links within the “What We can Offer You” text

```
* <div class="ats-description">
  > <div>= </div>
  > <div>= </div>
  > <div>
    > <div>
      > <a href="https://www.unilever.com/about/who-we-are/our-vision/" target="_blank" class="external-link" rel="noopener">
        > <span>Culture for Growth</span>
      </a>
      > <span class="visually-hidden">opens in a new window</span>
    </div>
    > <b>= </b>
    > <a href="https://drive.google.com/file/d/0Bzuvnqct1-texvxezbhaptw/view" target="_blank" class="external-link" rel="noopener">
      > <span>Top Notch Employee Health & Well Being Benefits</span>
    </a>
    > <span class="visually-hidden">opens in a new window</span>
  </div>
</div>
```

Page 5: DOM, related links not marked up semantically as unordered list items

Steps to reproduce

1. Observe the unordered list items.
2. Inspect the elements in the DOM.
3. Observe that the list items have not been marked up with **ul** and **li**.

Recommendation

Mark up the unordered list using **** and ****. Make sure that each set of list items is contained in a single **** and that the list items themselves are contained within individual **** tags. Only use separate **** tags when the list items are contained logically and visually in separate, distinct lists. For example:

```
<ul>
  <li>Pears</li>
  <li>Apples</li>
  ...
</ul>
```

Nested lists can be implemented in a similar fashion. For example:

```
<ul>
  <li>Pears</li>
  <li>Apples
    <ul>
      <li>Granny Smith</li>
      <li>Golden Delicious</li>
    </ul>
  </li>
  <li>Oranges</li>
  ...
</ul>
```

For more information, see [WAI Tutorials - Unordered Lists](#).

[Return to priority table](#)

ADR_20590-70 **LOW PRIORITY** **OPEN**

Block quote not marked up semantically

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 08

Users Affected Vision

Component Block Quote

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determinable. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

That are elements that are visually communicated as being block quotes. However, this structure has not been programmatically determined.

Where it happened

This can be seen on page 8 on desktop and iOS with the block quotations in the “Meet Our People” carousel. These are not marked up as block quotations.



Page 8: desktop, block quotation for Yuveer Ramchandani" in "Meet Our People" carousel

```
<div class="two-two-grid__child two-two-grid__child--topcontent">
  <div class="two-two-grid__child-copy">
    <h2>
      "Meet Our"
    <br>
      "People"
    </h2>
    <p>"At Unilever everyone is encouraged to be the best that they can be. I get to adapt, experiment and tinker without the fear of failure. I love the variety of new innovations and challenges I get to explore every day"'</p>
  </div>
</div>
```

Page 8: DOM, block quotation for Yuveer Ramchandani is not marked up

Steps to reproduce

1. Observe unordered the visual quote or citation.
2. Inspect the elements in the DOM.
3. Observe that the quotes have not been marked up.

Recommendation

Use the `<blockquote>` to contain the element. Use the `<cite>` for the source. For example:

```
<blockquote>
  <p>An unexamined life is not worth living.</p>
  <cite>Socrates</cite>
</blockquote>
```

In this example, there are internal `<p>` tags and the `<cite>` is nested in the `<blockquote>`. Both are allowed in HTML5, but this precise structure is not required. For example, the `<cite>` can be outside the `<blockquote>`.

For more information, see [WAI Tutorials - Quotes](#), [HTML Specification - Block Quote](#), and [HTML Doctor - Block Quote and Cite Reloaded](#).

[Return to priority table](#)

ADR_20590-90 LOW PRIORITY OPEN

Ordered list not marked up semantically

Success Criteria 1.3.1: Info and Relationships [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 11

Users Affected Vision

Component List

What happened

If a relationship between elements is expressed with visual cues, it also needs to be programmatically determined. This enables assistive technology to communicate the relationships to users. Content may otherwise seem unstructured or disorganized.

There are elements that are visually communicated as being interrelated, ordered list items. However, this structure has not been programmatically determined.

Where it happened

This can be seen on page 11, on desktop and iOS, with the ordered list of steps in the “Application process” section. The steps “Online application”, “Interview with Talent Advisor” and so on, are visually presented as interrelated items, numbered from 1 to 4. However, the ordered list is not programmatically determined.

Note that, in this case, there is no alternative provided to communicate the ordered list nature of the items, such as a semantic ordered list. As well, the visual numbers are not exposed to assistive technologies.



Page 11: desktop, “Application process” section

```
<section class="application-process max-width-wrapper max-width-wrapper--narrow"> ...</section>
<h2 class="application-process__header">Application process</h2>
<div class="application-process__grid">[flex]
  <div class="application-process__step one">
    <button class="application-process__number-button one" data-custom-event="true" data-custom-category="Custom Event" data-custom-label="Application process">[flex]
      <span class="application-process__number-button__copy">Online application</span>[flex]
      ::after [flex]
    </button>
    <div class="application-process__hidden-content">@1</div>
  </div>
  <div class="application-process__line one"></div>
</div>
<div class="application-process__step two">
  <button class="application-process__number-button two" data-custom-event="true" data-custom-category="Custom Event" data-custom-label="Application process">[flex]
    <span class="application-process__number-button__copy">Interview with Talent Advisor</span>[flex]
    ::after [flex]
  </button>
  <div class="application-process__hidden-content">@2</div>
</div>
<div class="application-process__line two"></div>
<div class="application-process__step three">@3</div>
<div class="application-process__line three"></div>
<div class="application-process__step four">@4</div>
</div>
</section>
```

Page 11: DOM, “Application process” section

Steps to reproduce

1. Observe the ordered list items.

2. Inspect the elements in the DOM.
3. Observe that the list items have not been marked up with **ol** and **li**.
4. Observe that there is no alternative provided for the ordered list semantics.

Recommendation

Mark up the ordered list using `` and ``. Make sure that each set of list items is contained in a single `` and that the list items themselves are contained within individual `` tags. Only use separate `` tags when the list items are contained logically and visually in separate, distinct lists. For example:

```
<ol>
  <li>Pears</li>
  <li>Apples</li>
  ...
</ol>
```

For more information, see [WAI Tutorials - Ordered Lists](#).

[Return to priority table](#)

ADR_20590-98 **LOW PRIORITY** **OPEN**

Colour relied upon to communicate information (job results)

Success Criteria 1.4.1: Use of Color [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 04

Users Affected Vision

Component Button

What happened

Understanding content and operating interactive elements must not rely on colour. There is information that is visually communicated exclusively through colour. This may particularly disadvantage low vision and colour blind users who may be excluded by the visual reliance on colour.

This has been raised as a low priority issue because the information communicated by the colour is likely to be unclear to all users.

Where it happened

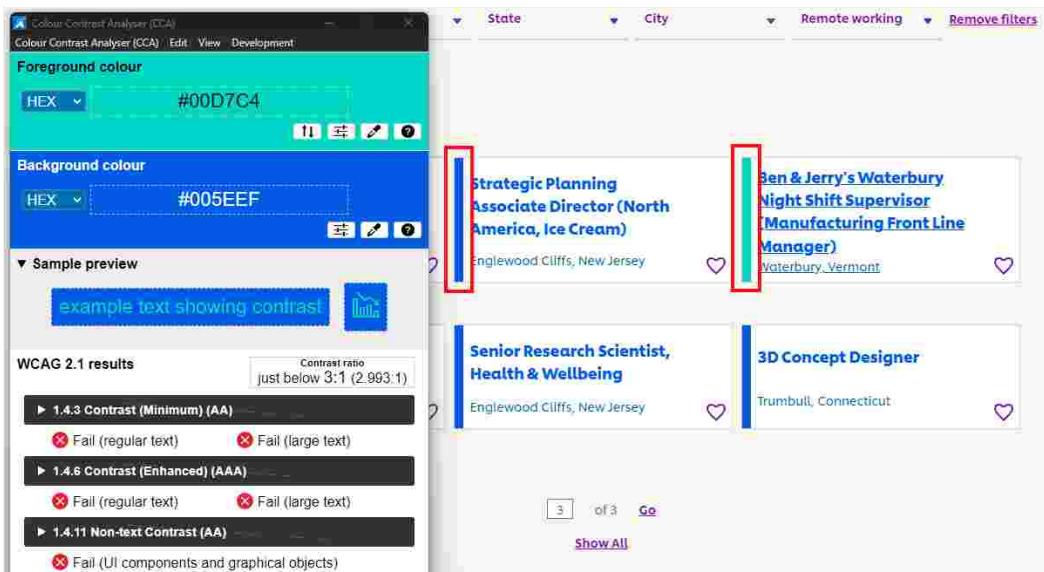
This can be seen on page 4 on desktop and iOS with the dark blue and teal bars.

Jamie Robertson of Unilever confirmed in an email on 13/06/24 that the colours convey information: “Teal - Blue collar work jobs”, “Blue - Experienced professional jobs”, “Violet - Early Career jobs”.

Because the teal bars (#00D7C4) have a slightly less than 3:1 contrast ratio with the dark blue bars (#005EEF), colour is relied on to convey this information. However, the meaning of the colours is not specified on the page and is likely to be unclear for all users.

Note that this colour-coding was only observed on page 4. However, similar components across the site that use the same card format may also be affected.

Note that the violet colour was not observed in the jobs available on this page during testing.



Page 4: desktop, Colour Contrast Analyser, colour is relied on to convey information with the job categories

Steps to reproduce

1. Access the information.
2. Observe that colour is used to communicate the information.

3. Observe that there is not an alternative way to visually access the information.

Recommendation

Make sure that information is not exclusively visually communicated through colour. Colour can be used to communicate information. However, colour must not be the only visual means of conveying information.

As well, hidden information and ARIA cannot be the only alternative to colour. It is important that information communicated through colour is programmatically determined, such that it is exposed to assistive technology. However, this success criterion requires that there be visual cues that are available to all users.

The simplest solution is providing the information that the colour is communicating through text. For example:

```
<ul>
  <li class="GreenUp">Tesla (up 2%)</li>
  <li class="RedDown">Apple (down 1%)</li>
</ul>
```

Otherwise, graphs can use patterns, textures, and tables to communicate information. For examples, see [Highcharts - Line Chart Accessibility Demo](#) and [GOV - Magna Charta Example](#). States, such as selected, activate, and invalid, can use additional visual cues, such as changed font weight, text decoration, icons, or text. For example:

```
<button class="boldText" role="tab" aria-selected="true"> ... </button>
<button class="standardText" role="tab" aria-selected="false"> ... </button>
```

For more information, see [PSU - Charts & Accessibility](#), [Penn State - Colour Coding](#), and [Princeton - Colour Alone](#).

[Return to priority table](#)

ADR_20590-100 **LOW PRIORITY** **OPEN**

Informative non-text content fails contrast (job results)

Success Criteria 1.4.11: Non-text Contrast [2.1 AA]

Devices Desktop, Mobile (iOS)

Pages Affected 04

Users Affected Vision

Component Button

What happened

When non-text content communicates information that is not available through text, the non-text content must pass colour contrast requirements with adjacent colours. For example, when using icons to communicate states, priority, and so on, the icon must pass the colour contrast requirements.

The required contrast ratio is 3:1. There is informative non-text content that does not meet the contrast ratio. This may particularly disadvantage low vision and colour blind users who may be excluded by the low contrast. Users might struggle to perceive the component that fails the colour contrast requirements.

This has been raised as a low priority issue because the information conveyed by the element is likely to be unclear for all users.

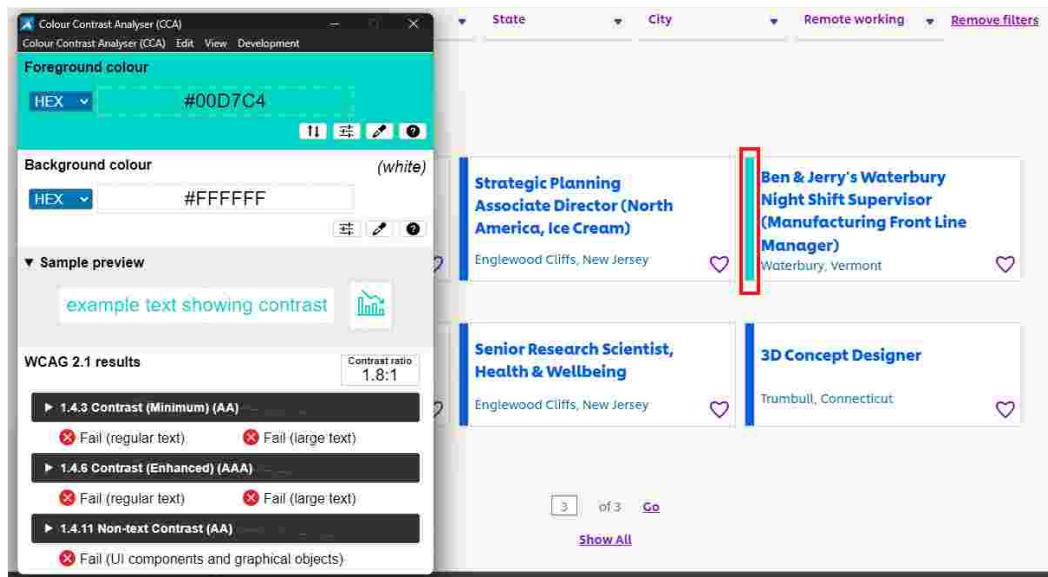
Where it happened

This can be seen on page 4 on desktop and iOS with the teal bars.

Jamie Robertson of Unilever confirmed in an email on 13/06/24 that the bars convey information: "Teal - Blue collar work jobs", "Blue - Experienced professional jobs", "Violet - Early Career jobs".

The teal bars (#00D7C4) have a 1.8:1 contrast ratio with the white internal button background (#FFFFFF). The contrast ratio with the grey button border and the off-white page background are lower than this.

Note that the violet colour was not observed in the jobs available on this page during testing.



Page 4: desktop, Colour Contrast Analyser, teal bar does not have sufficient contrast with white background

Steps to reproduce

1. Get the colour values of the informative non-text content.
2. Insert these values into a colour contrast analyser.

3. Observe that the analyser reports that the contrast ratio is not sufficient.
4. Observe that there is not an alternative way to access the same information.

Recommendation

When using informative non-text content, make sure that the colours pass 3:1 colour contrast requirements.

Adjust the foreground and background colours accordingly to meet the contrast ratio requirements. If you cannot adjust the colours, an alternative is to use native semantics and default browser stylings, which this success criterion considers to be out of scope. For examples, see [Native Form Elements](#).

The colour contrast ratio between the adjacent colours can be tested with tools such as [TPG - Colour Contrast Analyser](#) and [Tanaguru - Contrast Finder](#). WCAG AA compliant colour palettes can be generated with tools such as [Colorsafe](#) or [Accessible Color Matrix](#).

For more information, see [GOV - Colour Contrast Why Does It Matter?](#) and [WAI - Colours with Good Contrast](#).

[Return to priority table](#)

ADR_20590-50 LOW PRIORITY OPEN

Inappropriate page titles

Success Criteria 2.4.2: Page Titled [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 07

Users Affected Vision

Component Not Applicable

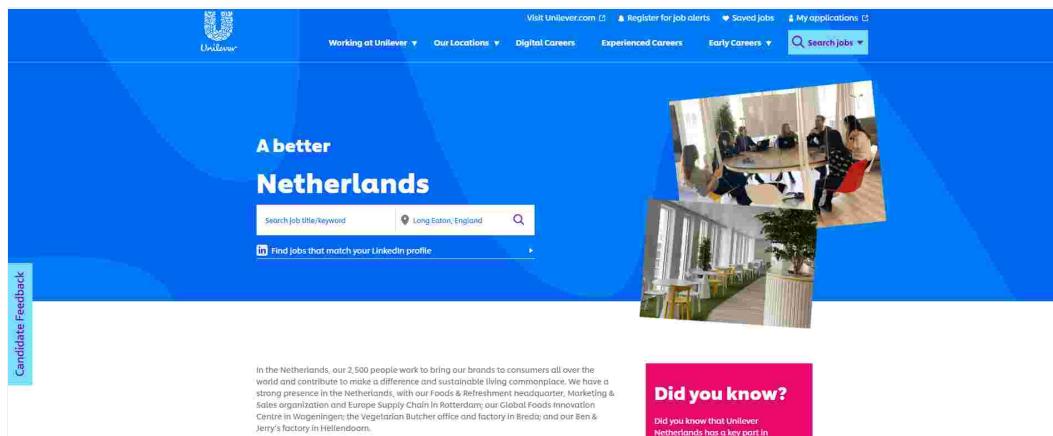
What happened

Webpages must have accurate and descriptive page titles.

There are inappropriate page titles. They do not describe the topic or purpose of the page. Page titles orientate users, communicating the nature of the page. They are also used by assistive technology, such as screen readers that announce them on page load. Due to the inappropriate page titles, users may struggle to orientate themselves.

Where it happened

This can be seen on page 7, on desktop and iOS. The page is titled “Unilever Netherlands”. This does not describe accurately the purpose of the page, which is around career and job opportunities.



Page 7: desktop, screenshot of the page

```
<meta name="theme-color" content="#005ee9">
<title>Unilever Netherlands</title>
<link rel="stylesheet" href="//tbcn.talentbrew.com/company/34155/css/31939-Content-pages.css">
```

Page 7: DOM, page title

Steps to reproduce

1. Observe the topic and purpose of the page.
2. View the source of the page.
3. Observe **<title>** element contains information that does not describe the topic or purpose of the page

Recommendation

Make sure that webpages have accurate and descriptive page titles. It is not required that every page title be unique or exhaustively descriptive. However, the page titles must accurately describe the topic or purpose of the page.

Where possible, use a consistent format, such as page name, page category, and site name. For example:

```
<!-- Homepage -->
<head>
  <title>Homepage - ACME Corp</title>
</head>

<!-- Online banking logon page -->
<head>
  <title>Logon - Online Banking - ACME Corp</title>
</head>

<!-- First step of the registration pages -->
<head>
  <title>Step 1: Personal details - Registration - ACME Corp</title>
</head>
```

Whatever format you use, use it consistently throughout. With single-page applications, if you meaningfully change the topic and purpose of the page, the page title needs to be dynamically updated, as well. This can be done with JavaScript using the `document.title` property.

For more information, see [Indiana University - Page Titles](#), [Washington University - Informative Page Title](#), and [MDN - Document Title element](#).

[Return to priority table](#)

ADR_20590-112 LOW PRIORITY OPEN

iFrame has no title

Success Criteria 4.1.2: Name, Role, Value [2.0 A]

Devices Desktop, Mobile (iOS)

Pages Affected 12

Users Affected Vision

Component Not Applicable

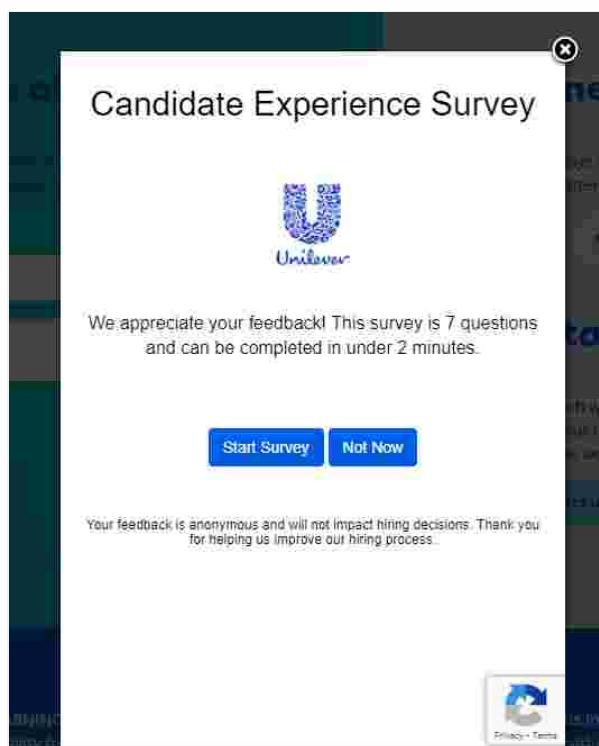
What happened

Interactive elements need a programmatically determined name, role, and value. These properties must also be accurate and appropriate.

There are iFrame elements that do not have an appropriate accessible name. This may particularly disadvantage users that interface with the accessibility properties of elements, such as screen reader and voice recognition users. That the name has not been programmatically determined may confuse or mislead users.

Where it happened

This can be found on page 12 on desktop and iOS with the iFrame for the “Candidate Experience Survey”. The element does not have a **title** attribute with an appropriate value.



Page 12: desktop, “Candidate Experience Survey”

A screenshot of the DOM inspector showing the iFrame element. The element is identified by the ID "survaley-survey-iframe" and has a "name" attribute set to "h1718707110480". The "src" attribute points to a URL starting with "https://app.surveyle.com/survey/index.html?". The "title" attribute is explicitly set to "Not specified". The "aria-label" and "aria-labelledby" attributes are also listed as "Not specified". The "role" attribute is set to "Iframe".

Page 12: DOM, “Candidate Experience Survey” iFrame does not have a title

Steps to reproduce

1. Inspect the iFrame and review the DOM.
2. Review the computed accessibility properties.
3. Observe that the iFrame title is not programmatically determined.

Recommendation

Make sure that interactive elements have an appropriate name. iFrame elements need a **title** attribute with an appropriate value. For example:

```
<iframe src="..." title="Informative and Accurate iFrame Name">  
...  
</iframe>
```

For more information, see [18F Accessibility Guide - iFrames](#), [Penn States Accessibility - Frames and iFrames](#), and [WCAG - H64 Using the title attribute of the frame and iframe elements](#). As well, for guidance on reviewing accessibility properties, see [Chrome - Accessibility properties](#), [Edge - Accessibility Pane](#), and [Firefox - Accessibility Inspector](#).

[Return to priority table](#)

Additional Issues by Priority

These are issues do not fail the conformance standard. They do not require resolving for the project to be conformant with the standard.

Note that the **additional** and **language** issues listed in the table below are not included in the final count.

High priority (2 issues)

Can't use: some users are excluded from using part of the product.

Specific user group(s) are excluded from using part of the product. For example, widgets cannot be opened with keyboard or critical visual information is not communicated by screen readers.

Issue	Success criteria Users affected	Status
ADR_20590-25 Footer - Significant elements overlapped at default text size and zoom levels	Additional Issue  	OPEN
ADR_20590-38 Significant elements overlapped at default text size and zoom levels	Additional Issue 	OPEN

Medium priority (0 issues)

Causing problems: some users will experience problems using the product.

Specific user group(s) will experience problems but they are not prevented from using the product.

Nonetheless, they may find accessing and operating content frustrating, confusing, or otherwise challenging. For example, expanding and collapsing elements do not communicate this behaviour to screen reader users.

No issues at this priority level.

Low priority (0 issues)

Minor and cosmetic issues: something is wrong, but it won't affect users too much.

Specific user group(s) will experience minor problems. They are not excluded from the product and won't face major obstacles. Nonetheless, these issues will make interaction bothersome and the product seem unpolished. For example, a small number of cosmetic images with redundant alt text.

Note that our priorities are context-sensitive. For example, if a page is swamped with occurrences of the same low priority issue, it may be bumped to medium priority.

No issues at this priority level.

Breakdown of Additional issues

ADR_20590-25 **HIGH PRIORITY** **OPEN**

Footer - Significant elements overlapped at default text size and zoom levels

Success Criteria Additional Issue

Devices Mobile (iOS)

Pages Affected 02

Users Affected Motor, Cognitive

Component Link, Button

What happened

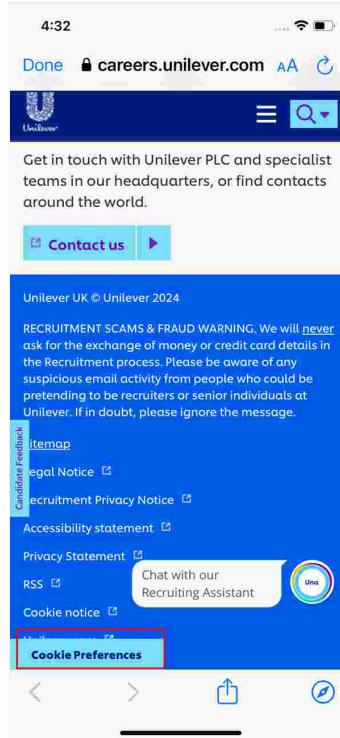
Users must be able to access content and functionality when the page is viewed with default text size and zoom settings. There should be no content or functionality that is partially off-screen, obscured by other content, or is otherwise not visually perceivable. There are significant elements that are not available at default text size and zoom settings.

This has been raised as an Additional Issue because the issue occurs when the page is viewed with default text size and zoom settings. WCAG requires that content and functionality remain available when text size and zoom levels are increased. It does not cover default text size and zoom levels.

This has been raised as a high-priority issue because the missing elements are significant.

Where it happened

This can be seen on page 2 on iOS, with the “Unilever.com” link present within footer section. At default page size of 100%, this link is overlapped by the “Cookie Preferences” button.



Page 2: iOS, “Unilever.com” link is overlapped by “Cookie preferences” button

Steps to reproduce

1. Navigate onto the page with the default text size setting and zoom settings.
2. Navigate through the content without changing these settings.

3. Observe that the element is overlapped.

Recommendation

Make sure that significant content and functionality can be accessed at default text size and zoom levels. In such cases, make sure that content is not off-screen, obscured by other content, or is otherwise not visually perceivable.

Decorative content can be lost. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. In this example, users can access the same functionality.

This can be achieved by using modern, flexible layout techniques and relative units, such as `em`, `ex`, `rem`, `%`, `vw`.

For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), [CUBE CSS](#), [MDN - Relative Units](#), and [CSS Tricks - font-size](#).

[Return to additional issue priority table](#)

ADR_20590-38 **HIGH PRIORITY** **OPEN**

Significant elements overlapped at default text size and zoom levels

Success Criteria Additional Issue

Devices Desktop, Mobile (iOS)

Pages Affected 05

Users Affected Cognitive

Component Link

What happened

Users must be able to access content and functionality when the page is viewed with default text size settings. There should be no content or functionality that is partially off-screen, obscured by other content, or is otherwise not visually perceivable. There are significant elements that are not available in the default text size settings.

This has been raised as an Additional Issue because the issue occurs when the page is viewed with default text size settings. WCAG requires that content and functionality remain available when text size and zoom levels are increased. It does not cover default text size and zoom levels.

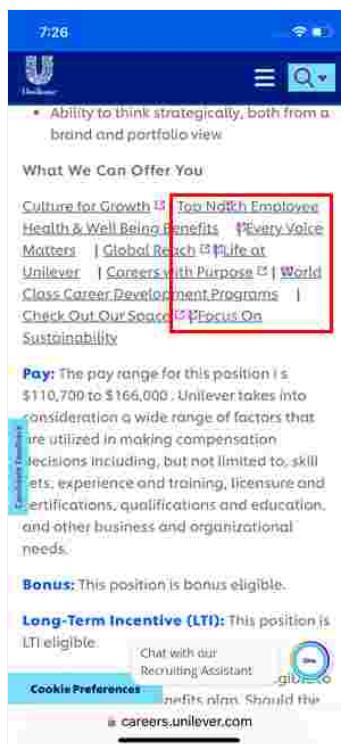
This has been raised as a high-priority issue because the missing elements are significant.

Where it happened

This can be seen on page 5 on desktop and iOS, with links such as “Top Notch Employee Health & Well Being Benefits”, “Every Voice Matters”, “Careers with Purpose”, “World Class Career Development Programs” and so on present below “What We Can Offer you” section. At default page size of 100%, these links are overlapped as they are placed closely adjacent to each other.

The screenshot shows a webpage with a dark blue header containing navigation links: "Working at Unilever ▾", "Our Locations ▾", "Digital Careers", and "Experienced". Below the header, there is a section titled "What We Can Offer You". Under this section, several links are listed in a horizontal row. Some of these links are highlighted with a red border, specifically "Top Notch Employee Health & Well Being Benefits", "Every Voice Matters", "Careers with Purpose", "World Class Career Development Programs", and "Focus On Sustainability". The text next to these links is partially cut off. Below this row, there is a paragraph about pay, followed by sections for bonus and long-term incentive (LTI), and finally a benefits section. The entire screenshot is framed by a light blue border.

Page 5: desktop, links overlapped on default page size



Page 5: iOS, links overlapped on default page size

Steps to reproduce

1. Navigate onto the page with default text size.
2. Navigate through the content without changing these settings.
3. Observe that the elements are overlapped.

Recommendation

Make sure that significant content and functionality can be accessed at default text size. In such cases, make sure that content is not partially off-screen, obscured by other content, or is otherwise not visually perceptible.

Decorative content can be lost. Similarly, the design of the page can change. For example, a mega menu can be converted into a burger menu. In this example, users can access the same functionality.

This can be achieved by using modern, flexible layout techniques and relative units, such as `em`, `ex`, `rem`, `%`, `vw`.

For more information, see [CSS Tricks - Guide to Flexbox](#), [MDN - Basic Concepts of Flexbox](#), [CUBE CSS](#), [MDN - Relative Units](#), and [CSS Tricks - font-size](#).

[Return to additional issue priority table](#)

Links and references

These resources are complementary to the recommendations given in individual issues above. They can be useful for debugging, understanding how to retest issues yourselves, and develop new features in an accessible fashion from the start.

Understanding WCAG

- [WCAG Quick Reference](#)
- [Web Content Accessibility Guidelines — for People Who Haven't Read Them](#) on 24Ways
- [The full WCAG specification](#)

Accessibility resources and tutorials

- [W3C Web Accessibility Tutorials](#) for structure, images, tables and forms
- [The A11Y project](#): A community-driven effort to make web accessibility easier
- [A11Y Style Guide](#) for developers
- [Accessibility Developer Guide](#)
- Checking colour contrast: [Contrast-Ratio.com](#) (online), [Colorable](#) (online), [Accessible Color Matrix](#) (online), [Colour Contrast Analyser](#) (Windows), [Contrast](#) (macOS), [Stark](#) (Sketch plugin)

ARIA resources

- [WAI-ARIA Authoring Practices](#), your first point of call for checklists and examples of accessible components
- [Inclusive Components](#), in-depth resources for complex patterns
- [ARIA Browser support matrix](#) on CanIUse

For native app development

- [Apple Developer documentation for Accessibility](#)
- [Android accessibility overview](#)