ACCESSIBILITY EVALUATION OF SYMPA

SEC. 1 - PROJECT NAME

SYMPA

SEC.2 - EXECUTIVE SUMMARY

Our goal in University of Illinois Urbana-Champaign Technology Services is to evaluate products to ensure they meet the University's **Minimum Digital Accessibility Standards**. Our team simply evaluates and points out the violations in a report. We do not determine whether this product can be used. If you wish to pursue the use of this product, and violations have been identified, an exception must be filed with the **Technology Accessibility Review Committee**.

The report's findings cover WCAG 2.1 A and AA guidelines. This accessibility evaluation was performed using Windows 10 as the platform Chrome Version 116.0.5845.180 (Official Build) (64-bit) and Firefox Version 102.15.0esr (Official Build) (64-bit) were tested as browsers. NVDA was used as assistive technology. The evaluation results in this report are based on an evaluation conducted on 9/25/2023. This evaluation covers Sympa (Campus mailing list) with Subscriber, Normal Owner, Privileged Owner, Listmaster, Listmaster Admin roles. These were tested with the combination of automated tools, the use of assistive technology, and manual checks.

Sympa has showstopper issues related to keyboard inaccessibility, focus order, and focus visible. There are few P1 issues related to color contrast, no labels etc. More details are in the below sections.

For any questions, please contact the evaluator **pp25@illinois.edu**. For re-evaluation requests, please see the **Disclaimer** section for further instructions on how to submit a **new request**.

SEC.3 – FUNCTIONAL DISABILITIES IMPACTED

The accessibility issues in this ICT will negatively impact individuals in the following interaction modes (Check all that apply):

☒**Without vision or with limited vision**, where a visual mode of operation is provided.

☒**Without perception of color**, where a visual mode of operation is provided.

☐**Without hearing or with limited hearing**, where an audible mode of operation is provided.

☐**Without speech**, where speech is used for input, control, or operation.

☒**With limited manipulation, reach, and strength**, where a manual mode of operation is provided.

☒**With limited language, cognitive, and learning abilities;** making the operation of the ICT more difficult for individuals with limited cognitive, language, and learning abilities.

SEC.4 - PRIORITY & RANKING

The following review is structured in a hierarchical fashion with priorities indicated as follows. Please note all findings are violations of accessibility regulations and must be fixed – this is the suggested priority in which to do so.

1. *Show Stopper (SS)* – An issue is a showstopper if it meets either of the following conditions:
    a. The issue prevents some individuals from using an application or completing a critical task.
    b. The issue makes using the application so time consuming or difficult that it prevents a functionally equivalent timeframe for access.
2. *Priority 1 (P1)* – Priority 1 issues make using an application difficult for most users; however, the difficulty is generally not severe enough to be considered as preventing equivalent access.
3. *Priority 2 (P2)* – Priority 2 issues make using an application annoying or confusing for many users.

SEC. 5 – HTML CODE VALIDATION

The purpose of code review is to make sure that the HTML follows the Document Type Declaration (DTD), which sets out the rules and grammar for the flavor of markup, thus enabling the browser to render the content accordingly. The HTML5 validator **NuValidator** was used in this evaluation. Code validation should be done to better conform with HTML standards and thus accessible coding practices. Although several of the errors uncovered by the Validator are minor, compliance means the site will operate as expected when used by assistive technology (AT) (WCAG 2.0 **4.1.1 Parsing**). Often, fixing one or two of those errors early on can resolve many of the errors flagged, since it has a waterfall effect.

Errors (P2)

1. X-UA-Compatible HTTP header must have the value **IE=edge**, was **IE=100**.
2. Element **ul** not allowed as child of element **menu** in this context. (Suppressing further errors from this subtree.).
3. No **p** element in scope but a **p** end tag seen.

Warnings

1. The **type** attribute is unnecessary for JavaScript resources.
2. The **name** attribute is obsolete. Consider putting an **id** attribute on the nearest container instead.
3. The **navigation** role is unnecessary for element **nav**.
4. The **main** role is unnecessary for element **main**.
5. The **contentinfo** role is unnecessary for element **footer**.

6. Consider using the **h1** element as a top-level heading only
(all h1 elements are treated as top-level headings by many screen readers and other tools).

SEC. 6 - AUTOMATED TOOL EVALUATION

AInspector and **Siteimprove Accessibility Checker** were used as automated accessibility evaluation tools. The issues and warnings found are noted below:

1. *P1* – All rendered content must be placed inside of container elements with appropriate ARIA landmark roles. Landmarks provide a way to organize the various types of content on a page for users of assistive technologies. The organization of content regions using landmarks is functionally like the way visual designers organize information for people who rely on a graphical rendering of the content.
   WCAG 1.3.1 Info and Relationships (Primary)
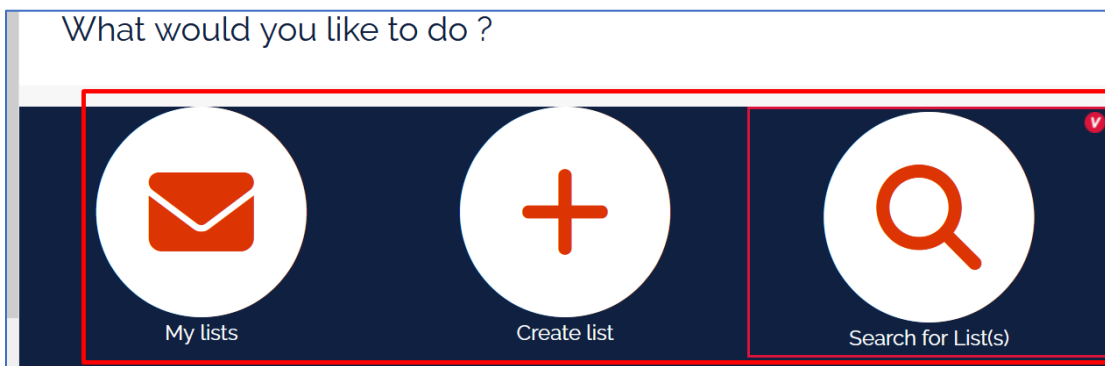
Figure 1 Elements not being inside container with proper landmark roles.

2. *P1* – Text content must exceed Color Contrast Ratio (CCR) of 3.1 for large and/or bolded text and 4.5 for any other size or style of text. The higher the color contrast of text the easier it is to read, especially for people with visual impairments. **1.4.3 Contrast (Minimum) (Primary)**
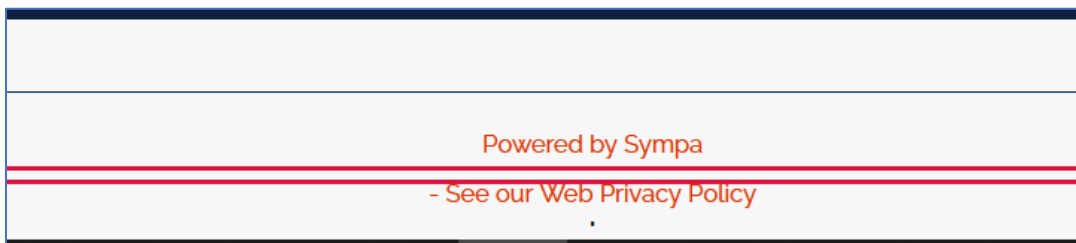

Figure 2 Color contrast issue.

3. *P1* – Landmarks must be uniquely identifiable. Multiple instances of landmarks with the same role must have unique accessible names. When there is more than one landmark of the same type on the page (e.g., multiple NAVIGATION and/or REGION landmarks), additional labeling through the use of author-defined accessible names is needed to allow users to differentiate among them. Change

3

the accessible name "" of the NAVIGATION landmark (or the other duplicates) so that it is unique on the page. **1.3.1 Info and Relationships (Primary)**
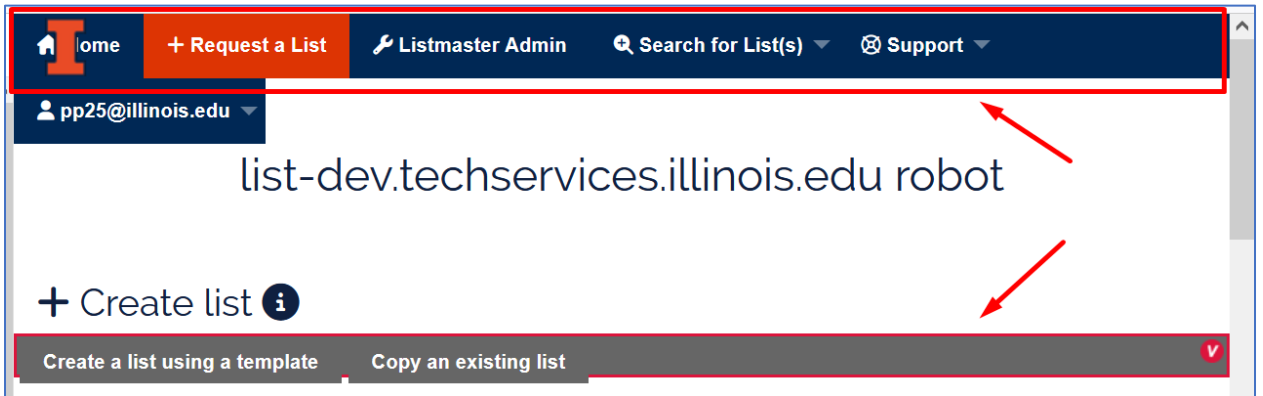

Fig 3 Multiple landmarks with no unique accessible names.

4. ***P1*** –Radio buttons must have grouping label. A related group of radio buttons must have a grouping label. Screen readers treat grouping labels differently than standard labels, typically speaking the grouping label only once when focus is first moved to one the groups controls.

    WCAG **3.3.2 Labels or Instructions (Primary)**
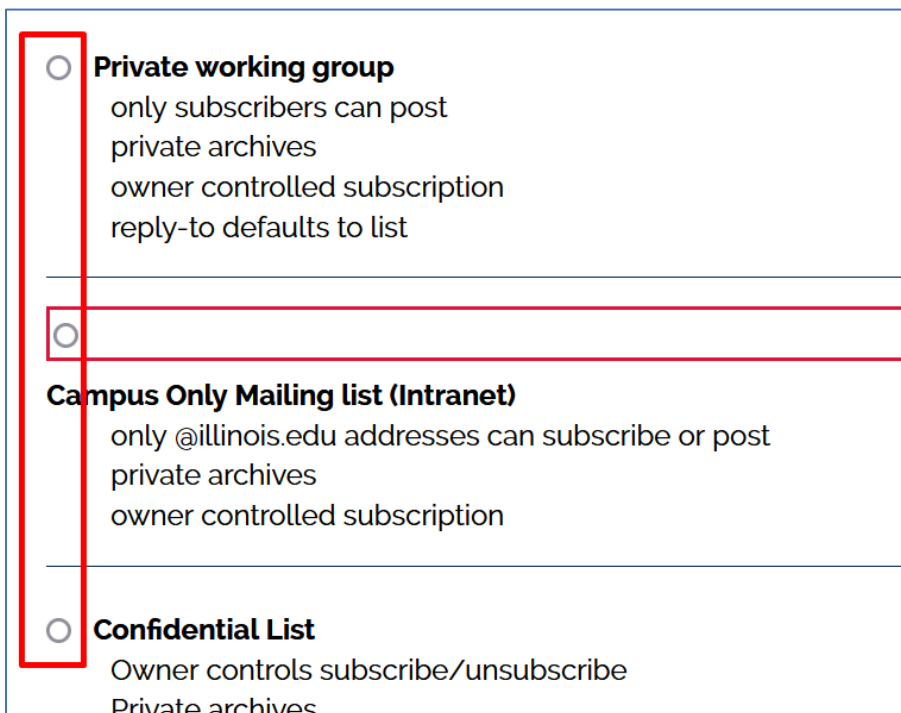
Fig 4 No grouping label on radio buttons.

5. ***P1*** –Each LABEL element using the FOR attribute must reference a form control on the page. LABEL elements are useful for accessibility only when they reference or encapsulate form controls. Change the LABEL element with the FOR attribute value 'owner' to reference a form control. Also change the LABEL

element with the FOR attribute value 'list_type' to reference a form control. WCAG Level A **3.3.2 Labels or Instructions (Primary)**

Fig 5 Label attribute with no form control issue.

6. ***P1*** –Headings must be properly nested. Headings must be properly nested within a landmark. Adjust the level of the H4 heading or other heading elements such that the headings are properly nested in the MAIN landmark.WCAG Level A **1.3.1 Info and Relationships (Primary)**

Fig 6 Inconsistent heading structure.

7. ***P1*** –Role is not allowed. Some HTML elements are restricted in the role allowed based on the semantics of the element (e.g. LI elements can only have list related ARIA roles). When a role conflicts with the implicit semantics of an HTML element the accessibility API provides confusing and/or conflicting information to

screen readers. In this case, the LABEL element does not allow the implict role of the element to be changed. Remove the HEADING role from the element. WCAG Level A **4.1.2 Name, Role, Value (Primary)**
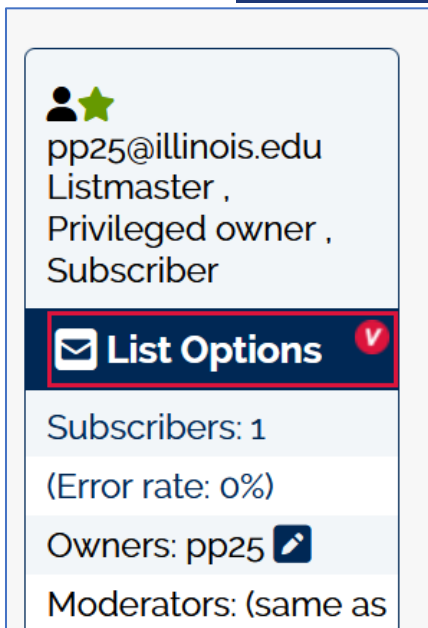

Fig 7 Inconsistent heading structure.

8. *P1* –Link text must describe the link target. The accessible name of a link must accurately describe the target or purpose of the link. When the accessible name of a link does not describe its target or purpose, users will not have the information they need to determine the usefulness of the target resources. WCAG Level A **2.4.4 Link Purpose (In Context) (Primary)**
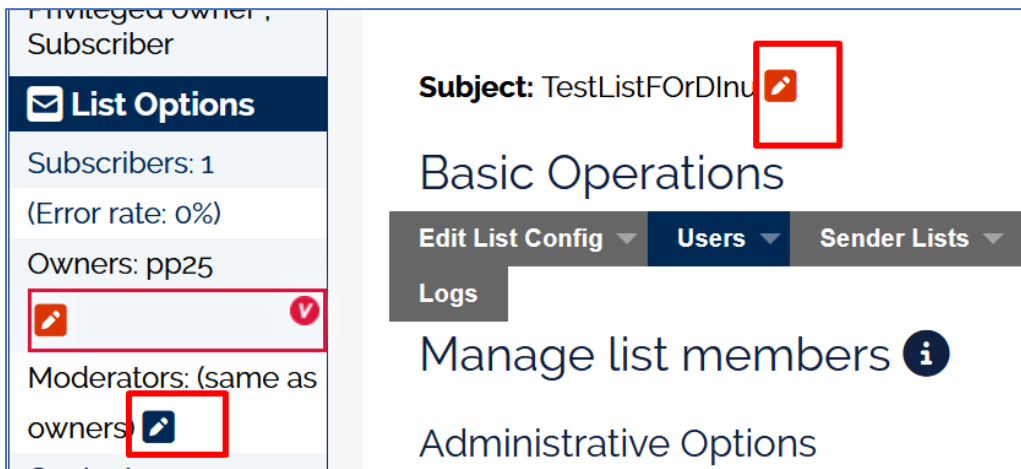
Fig 8 Link text without link target.
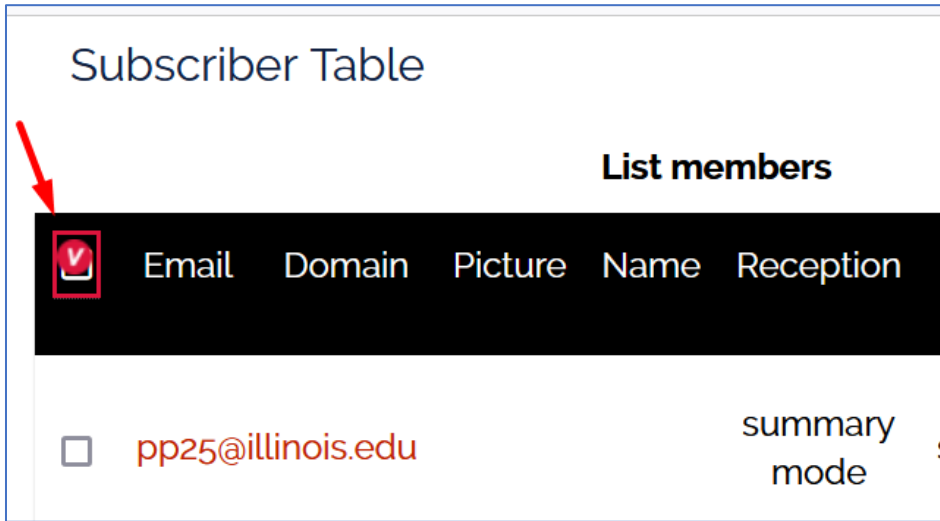
Fig 9 Link text without link target on checkboxes.

9. **P1** –Data cells in complex data tables must use HEADERS attribute to identify header cells. The data cells in complex data tables need to use the HEADERS attribute to identify the appropriate header cells, since simple row/column relationships cannot be relied upon to provide header information. WCAG Level A **1.3.1 Info and Relationships (Primary)**
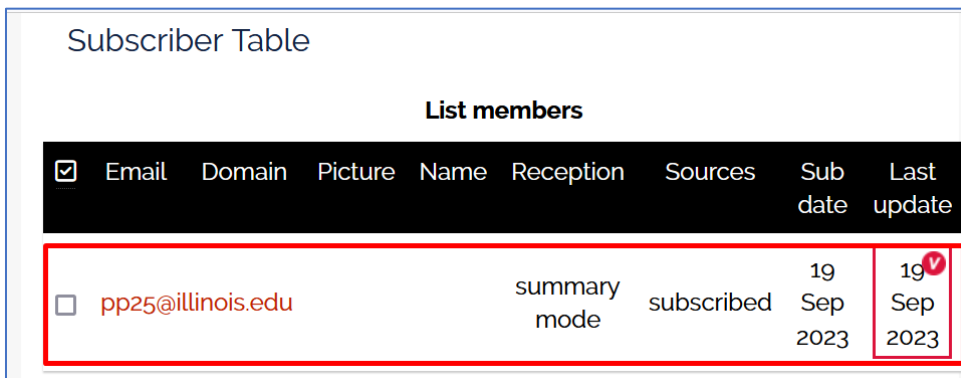

Fig 10 Data cells must have headers.

10. **P1** –Forms must use INPUT[TYPE="SUBMIT"] or other form control buttons for submitting forms. Using a submit button allows users to easily identify and control how to submit form information to a server for processing. Forms that are submitted through changes in focus or selection may result in users prematurely submitting the form or not being aware they submitted the form. WCAG Level A **3.2.2 On Input (Primary)**
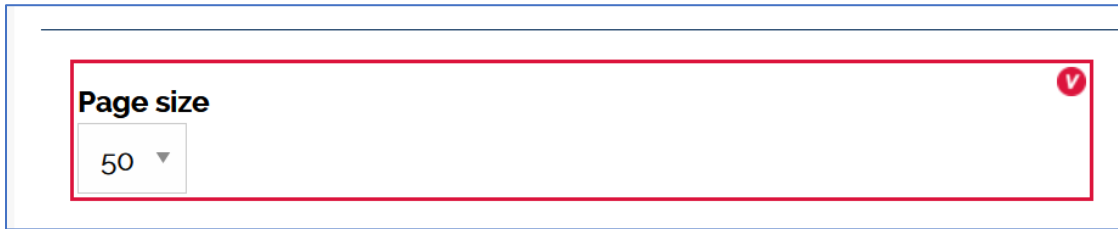
Fig 11 Forms must use control buttons.

11. **P1** –Form controls must have labels. Each INPUT, SELECT, TEXTAREA, PROGRESS, METER and OUTPUT element must have an accessible label. A label associated with a form control ensures that information about the form control is spoken by screen readers when it receives focus. Tag Names: ul[role=tree], ul[role=treeitem]. WCAG Level A **3.3.2 Labels or Instructions (Primary)**
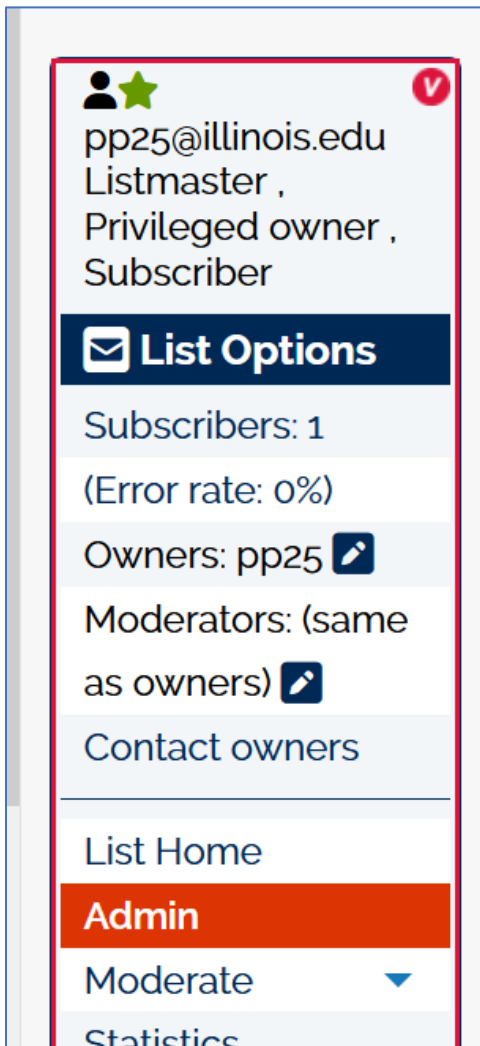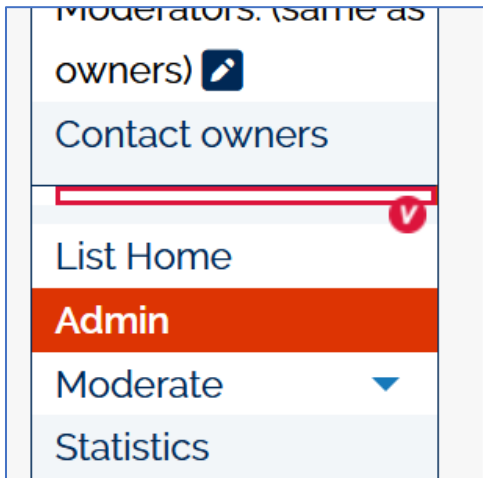

Fig 12 Forms control must have labels.

Fig 13 Forms control must have labels.

12. **P1** –Widgets must have a label. Also, the label for elements with a widget roles on a page must sufficiently describe its purpose. A label associated with a element with a widget role ensures that information about the widget is spoken by screen readers when it receives focus. Labels that are sufficiently descriptive make it possible for people to understand the purposes of elements with widget roles on the page. WCAG Level A **4.1.2 Name, Role, Value (Primary)**
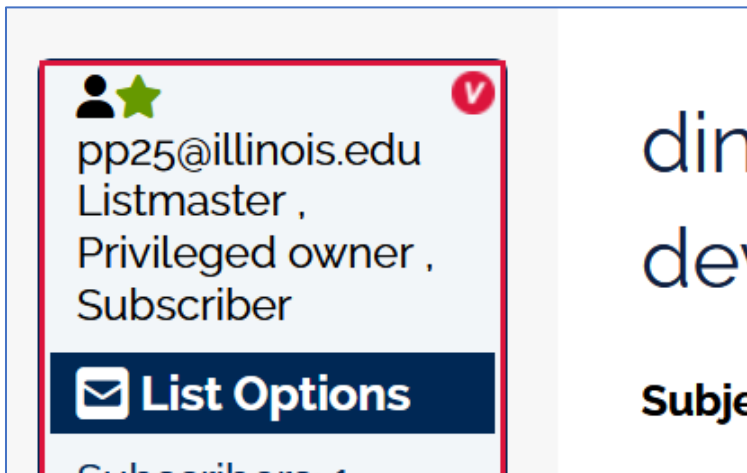
Fig 14 Widget must have labels.

13. **P1** –Role must have a required parent role using the HTML DOM structure or the ARIA-OWNS attribute. Parent/child relationships are used by assistive technologies for computing the number of items owned by a container and the position of an item (e.g. "third of five links"). The ROW role requires a parent GRID, ROWGROUP, TABLE or TREEGRID role, check your HTML DOM structure to ensure an ancestor element or an ARIA-OWNS attributes identifies a required parent role. WCAG Level A **4.1.2 Name, Role, Value (Primary)**
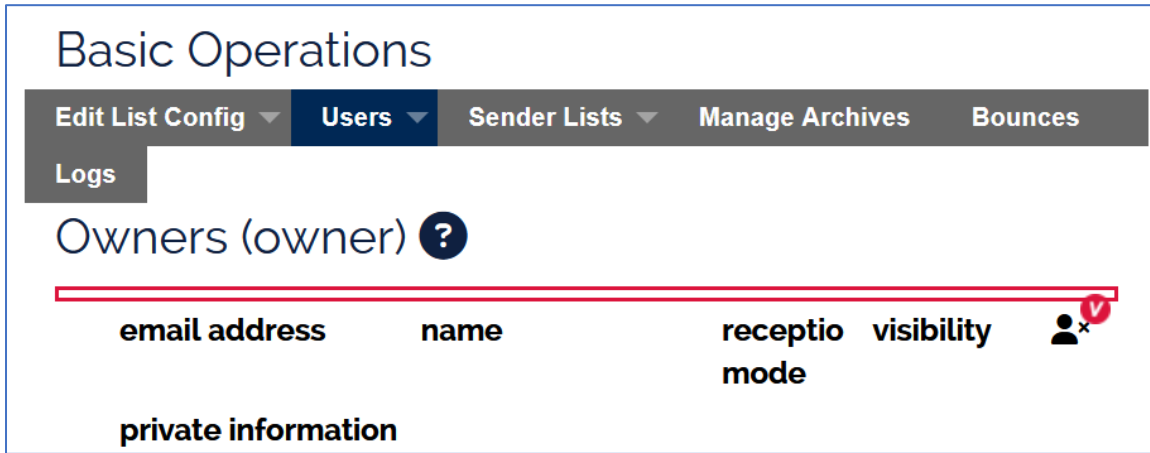
9

Fig 15 No parent role for ARIA roles.

SEC.7 - MANUAL CHECKS

The following review consists of functional testing performed by the user navigating with keyboard only or using assistive technology such as screen readers.

1. **P1** – Menus that open when receiving keyboard focus is not a great option regarding accessibility. When there are lot of submenus in each of those menus, either user must navigate to all of them to reach the main content or have an extra keyboard stroke of esc key for each menu to collapse those submenus. So, it is ideal to open the menu with the enter key or space bar. The menu in the screenshot below opens as soon as they receive keyboard focus. WCAG Level A **2.1.1 Keyboard**
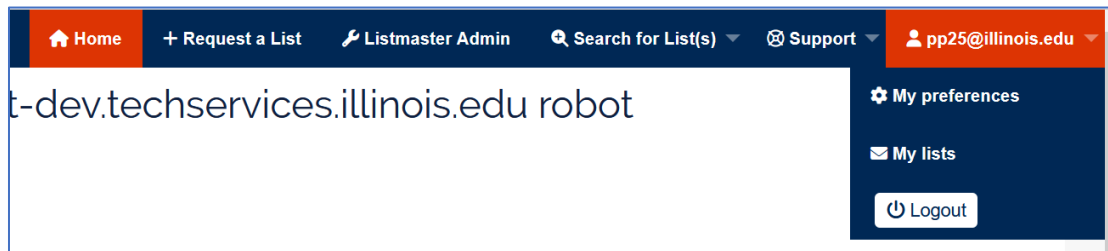

Fig 16 Menu open on keyboard focus.

2. **SS** – There is no clear visual focus indicator when accessing elements by keyboard. Users with poor vision would not understand where the current focus is. In the screenshot below, my current focus is on word prixlist, but since it doesn't have a focused box/border around the element, it is not distinguishable. WCAG Level AA **2.4.7 Focus Visible**
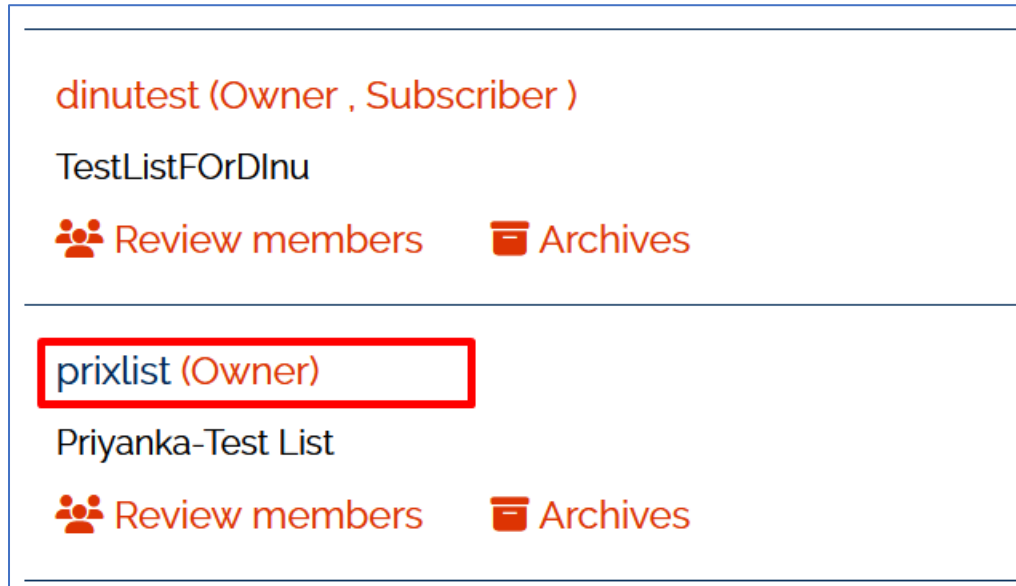
Fig 17 Focus indictor issue.

3. **SS** – When accessing the navigation menu using keyboard, once a tab is selected, the focus goes back to top bar navigation before it goes into the selected tab contents. Users expect to get into the content if they selected that tab. Focusable components need to receive focus in an order that preserves meaning and operability only when navigation sequences affect meaning and operability. WCAG Level A **2.4.3 Focus Order**
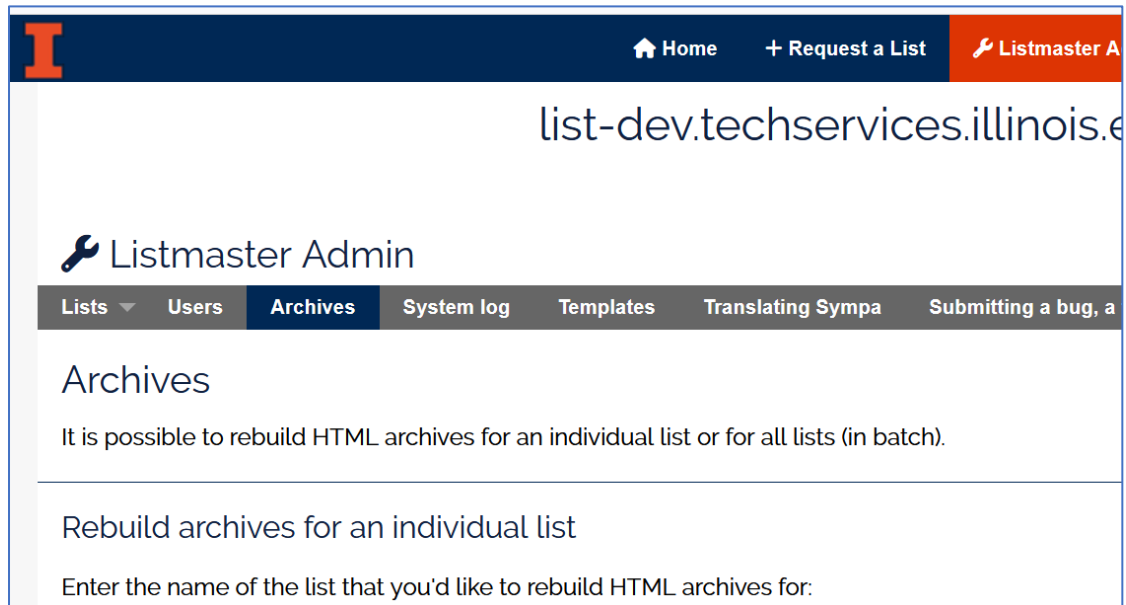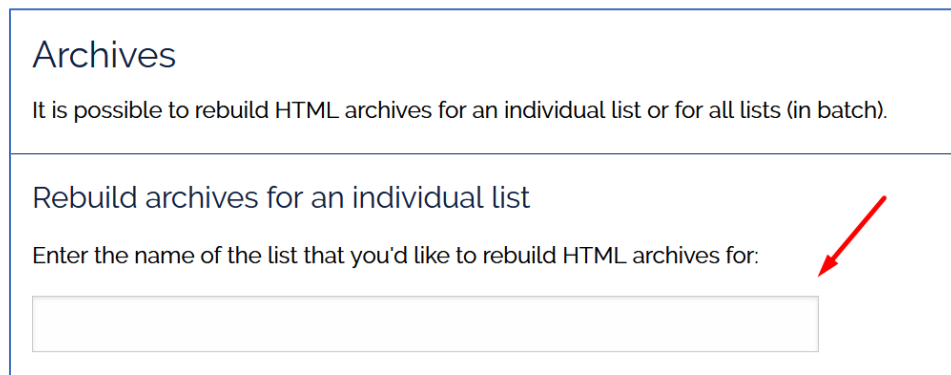

Fig 18 Focus order issue.

4. **SS** – Screen reader announces the text fields as "edit blank". Labels are required to understand the purpose of the text field. WCAG Level A **3.3.2 Labels or Instructions**

Archives

It is possible to rebuild HTML archives for an individual list or for all lists (in batch).

Rebuild archives for an individual list

Enter the name of the list that you'd like to rebuild HTML archives for:

Fig 19 No label on elements.

5. **SS** – When the screen is reduced to a mobile view, the submenus are not accessible by keyboard. Back button on Basic Operations menu is also not keyboard accessible. WCAG Level A **2.1.1 Keyboard**
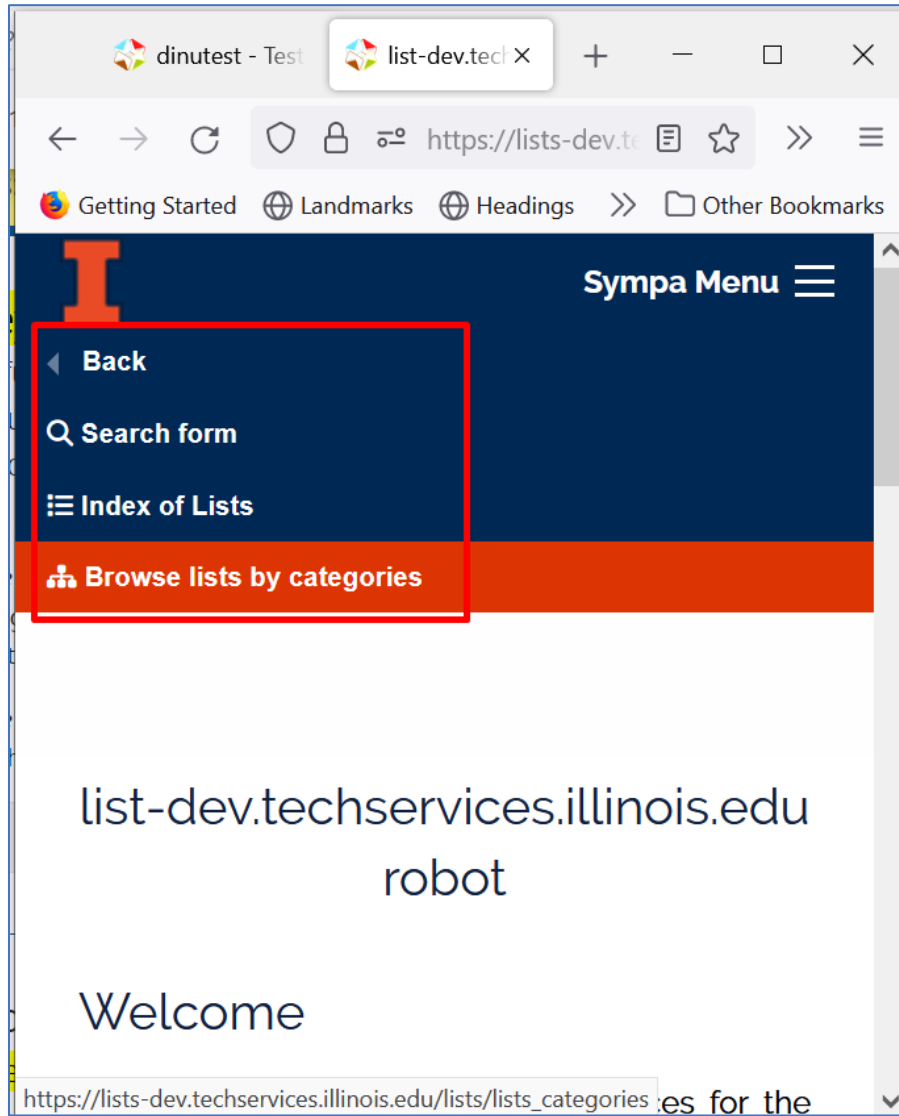
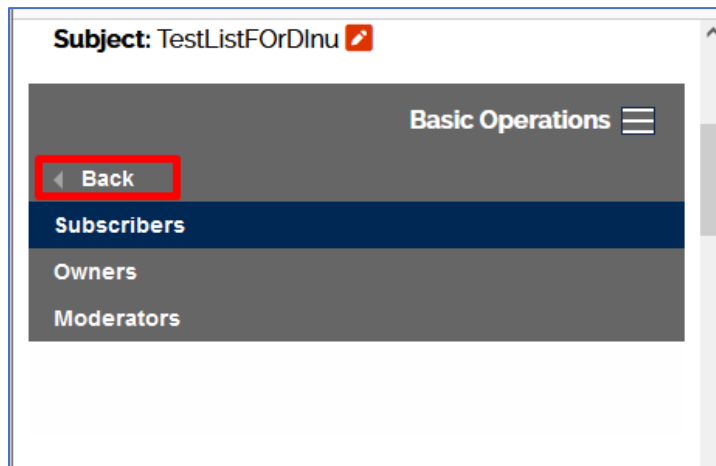*Fig 20 Keyboard accessibility issue.*

Fig 21 Keyboard accessibility issue on menu .

6. **SS** – "Moderate" option is not keyboard accessible, the submenu for that cannot be viewed when the screen size is reduced. **2.1.1 Keyboard**
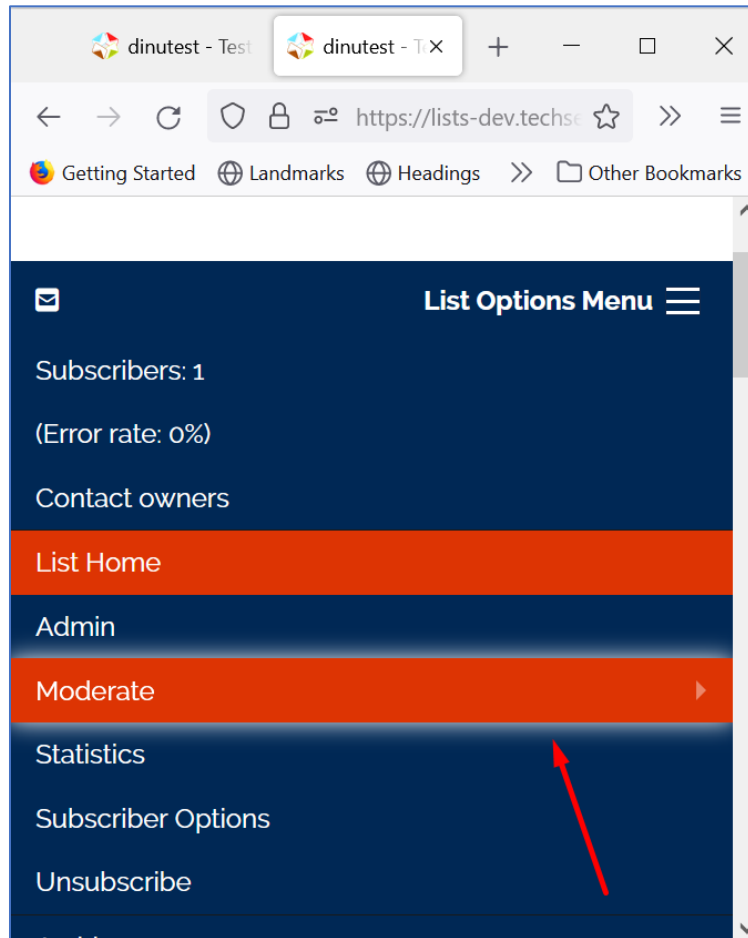

Fig 22 Keyboard accessibility issue.

7. **P1** –Content must be presented without loss of information or functionality, without requiring scrolling in two dimension. Text reflow must be within the width of the viewport. Avoiding the need to scroll in the direction of reading in order to reveal lines that are cut off by the viewport is important, because such scrolling significantly increases the effort required to read. WCAG Level AA **1.4.10 Reflow**
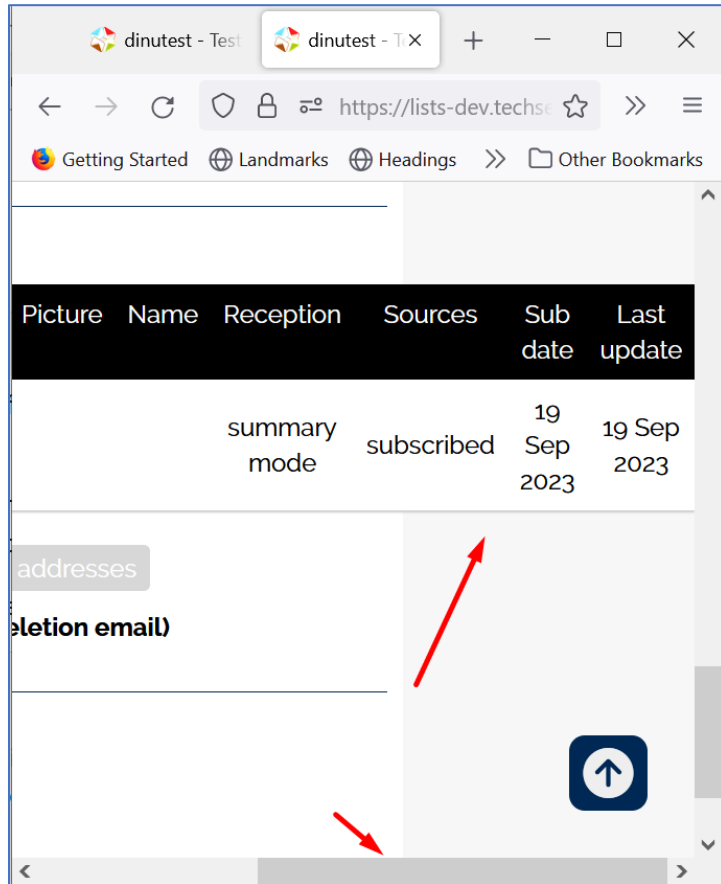
14

Fig 21 Content presentation issue on responsive website.

8. **SS** – Unable to select date from the calendar picker using keyboard. When navigating through keyboard, the calendar picker popups when hitting enter but tabbing into the picker makes the picker disappear. **2.1.1 Keyboard**
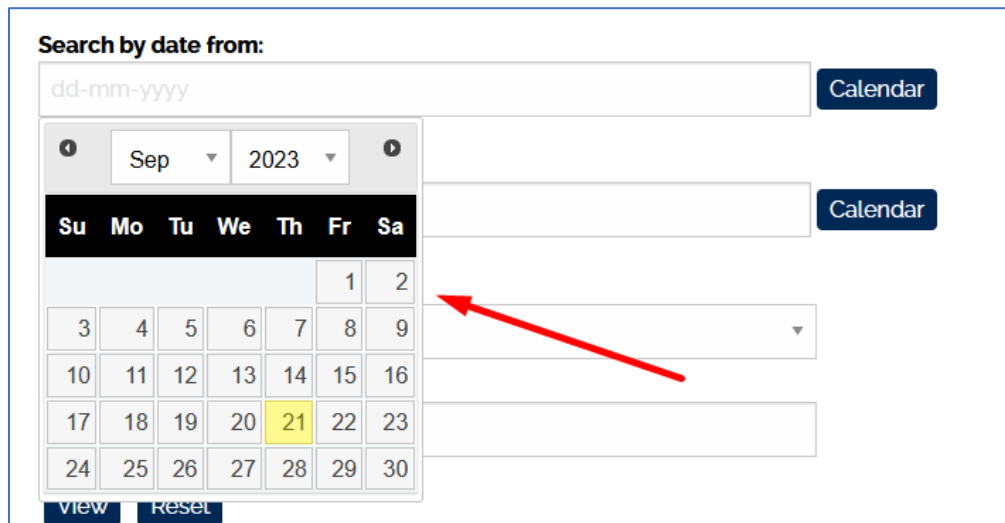

Fig 22 Keyboard accessibility issue on date picker.

9. **SS** – The items in the multiselect box are not readable fully. Items are cut off. When this is zoomed to 200%, the readable content gets lesser. WCAG Level **AA 1.4.4. Resize Text**
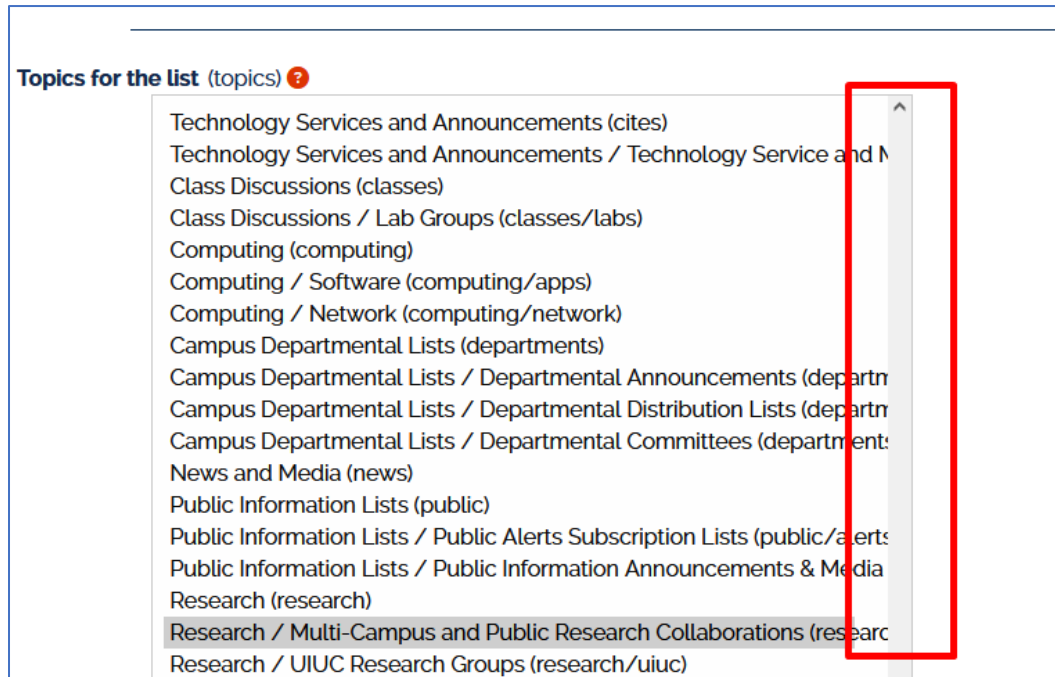

Fig 23 Text being cut off issue.

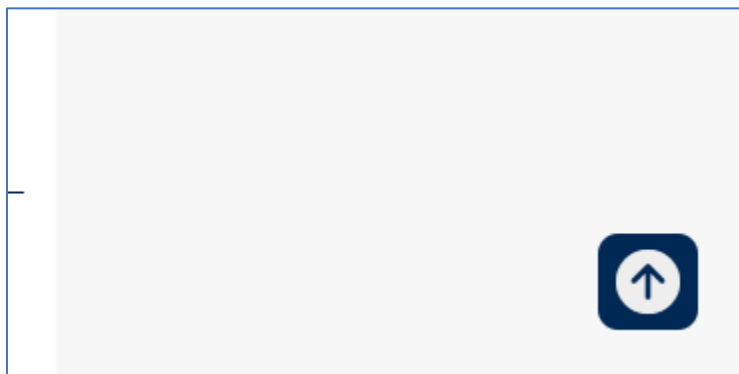10. **P1** –Back to Top button is keyboard inaccessible. **2.1.1 Keyboard**

Fig 24 Back to Top button not keyboard accessible

SEC.8 - DISCLAIMER

Any accessibility audit should not be considered exhaustive and is a process that combines automated software tools with human judgment **(Manual Checks)**. There may have been issues that were missed. Accessibility evaluation is iterative and needs to be conducted periodically, especially if changes are made to the site / software.

Therefore, it is highly recommended that a follow-up evaluation be conducted once the initial issues have been addressed. For any questions, please contact the evaluator **Evaluator email here**.

For re-evaluation requests:

- Will not move forward with re-testing until all Showstoppers and P1 issues have been resolved.
- To initiate a re-evaluation, please fill out and attach the included spreadsheet with issue resolution details and submit a **new request**. Requests that do not include the resolution details will be rejected.
- The request will be addressed in the order it was received.
- No more than three evaluations on any single product will be performed within one calendar year.