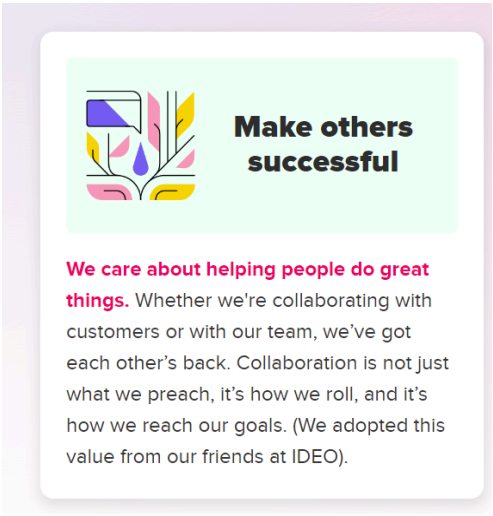
Accessibility and Design Handoff

In this module, you'll learn:

- What accessibility has to do with design handoff.
- Why accessibility is important for handoff.
- How to make sure your design is accessible before handoff.
- Best practices for design handoff and accessibility.

What's accessibility got to do with it?

Design handoff is the point where you work with developers to implement your designs. Handoff means you've created a clear and consistent UI with annotations that will **"make others successful" (one of MURAL's values!)** in implementing an accessible product.



Accessibility annotations are a crucial part of the handoff process and the foundation for ensuring your design is implemented in an inclusive way.

As a MURAL designer, you'll want to deliver your work in an easy-to-follow and meaningful format, making sure developers are clear on the accessibility needs throughout the UI.

Why is accessibility important for handoff?

The user experience for all user types (mouse, keyboard, screen reader, etc.) should be driven by design, rather than development. End-users should have the best experience no matter how they interact with a product. That experience starts with accessibility.

Why is this important?

- Designers have a more holistic view of the user experience while developers often dive deeper into smaller areas and may lack the holistic context.
- Skipping the handoff process leaves a lot of room for questions and confusion from developers about accessibility.
- When developers have to guess what needs to be done, it results in inconsistent designs and missed opportunities for accessibility that affect the end-user's experience.

Proper handoff supports better developer estimation and roadmap planning and cuts down on revisions, making the team more efficient overall.

How can I make sure my design is accessible and easy to follow before handoff?

Developers usually prefer straightforward technical explanations of design elements. Your design should explain in detail how interaction works and include specs and annotations. Be sure to complete the following tasks before handing off content to developers:

Start conversations with designers and developers early in the design process

Good handoff starts with:

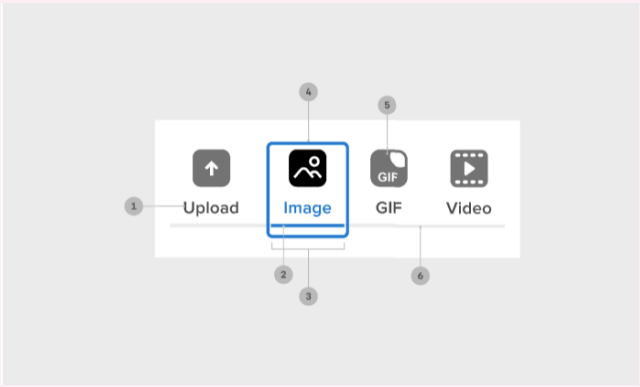
- A clear definition of accessibility requirements from the beginning (e.g., Is your team aiming for WCAG [Web Content Accessibility Guidelines] conformance Level AA or Level AAA?)
- Creating prototypes to communicate interaction intent and provide links or demos. From user flow walkthroughs to desired interactions, prototypes do much more than help everyone understand the desired result of a design.

Add annotations and more annotations!

Annotations should describe the content, state, and function of each element and how those relate to accessibility considerations:

- Include separate notes if needed. While your documentation should be thorough and avoid making assumptions, you should also avoid overwhelming the wireframe with notes.
    - Separate notes are typically created by using numbered annotations or multiple artboards that break down different information in your design. The notes may include information on landmarks, tab order, screen reader labels, keyboard navigation, and shortcuts (e.g., notes that show the overall tab order of tab stops and highlight the arrow key navigation within a specific related group.)
- Annotations should be worded in a way that is as self-explanatory as possible.

Example: The screenshot below shows an example of numbered annotations that can be used for wireframe documentation at MURAL.



Document rem unit specs

It’s helpful if designers document rem units as part of their specs for handoff. Designers use pixel units in Figma since that's what Figma supports. However, developers use `rem` (the most common) or `em` due to relative sizes that can scale based on user settings.

There are a couple of ways to do this. You can use the available tokens or components in the DS Designer Toolkit for Typography, or divide the font size by the baseline font size:

- In the Classic theme, the baseline is 10px. In Hana, the baseline is 16px. For example, a 20px font is 2 rem in Classic and 1.25 rem in Hana.

Definitions:

Em: a relative unit of measurement used to scale an element’s sizes based on the parent's size.

Rem (root em): a unit of measurement that is relative to the root (HTML) font size and used to scale an element based on the root size, no matter what the parent size is.

Add annotations for semantic elements

A semantic element helps the developer understand the meaning of its content. Semantic annotations clarify the intended purpose of elements before implementing your design. This is particularly important when a product is being redesigned since design specs aren’t always aligned with what’s in the code.

When writing annotations, be sure to:

- Call out specific components and variants of them, especially components that visually look similar in their default state (e.g., dropdown vs. select vs. menu and row and column headings for table cells).

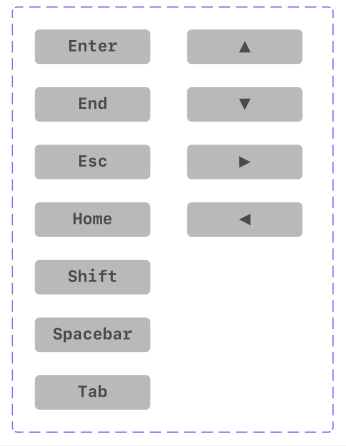
    - Use existing components whenever possible to create long-term consistency in designs and ensure components are accessible.
- Identify the role, name, state, and properties of a custom control and how it should work.
- Document headings. Headings provide structure to a webpage. Semantics should determine headings rather than visual style so that a developer will know the intended hierarchy of information.

Ensure you've documented the entire keyboard user journey

The keyboard user journey tells the story of how an end-user completes tasks using only a keyboard. Follow design system best practices to keep these annotations consistent.

Resource: Refer to the **Designing for Keyboard Accessibility** module to learn more about documentation for keyboard users.

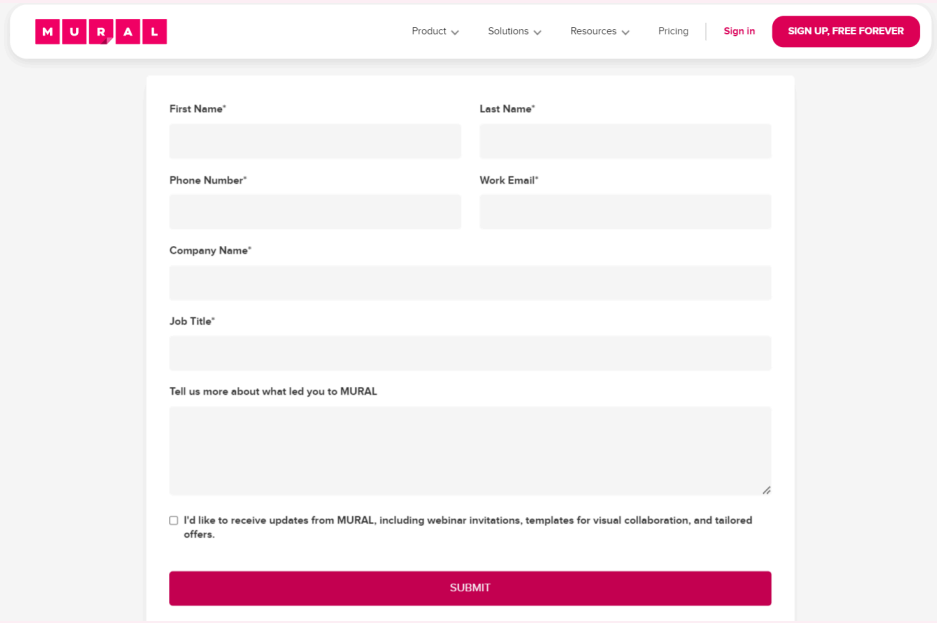
Example: The screenshot below shows how the arrow keys are sometimes used at MURAL to enable interactions.



Note the tab order of focusable elements

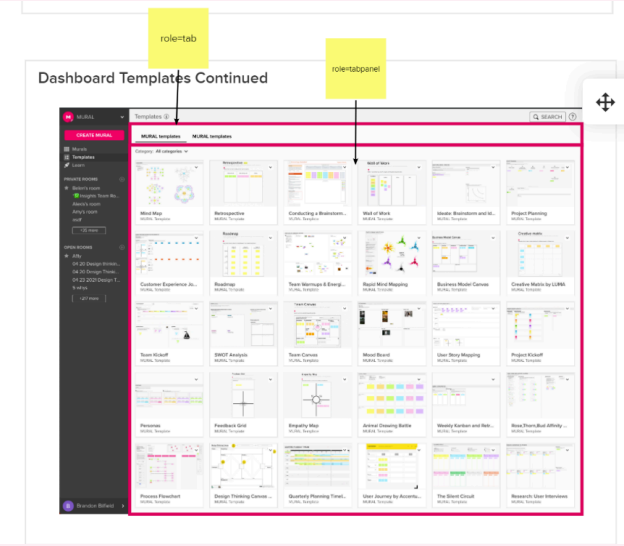
Be sure to note the tab order of elements such as links, buttons, input fields, etc., and ensure the order is logical:

Example: Ensuring there's a logical tab order in your design means verifying end users complete tasks in an expected way such as filling out form fields before tabbing to the submit button.



- Make sure the wireframe shows how interactive components or hover triggers are included in the tab order or can be operated with a keyboard.

Example: The screenshot below shows how MURAL designers notate the tab order in the wireframe and document how interactive components and hover triggers are included in that order.



- Document groups of related components, including tab stops and the keys (usually the arrows) that will enable interaction between groups.

- Notate the arrow keys between individual components and the single tab stop for groups of components.

Specify landmarks

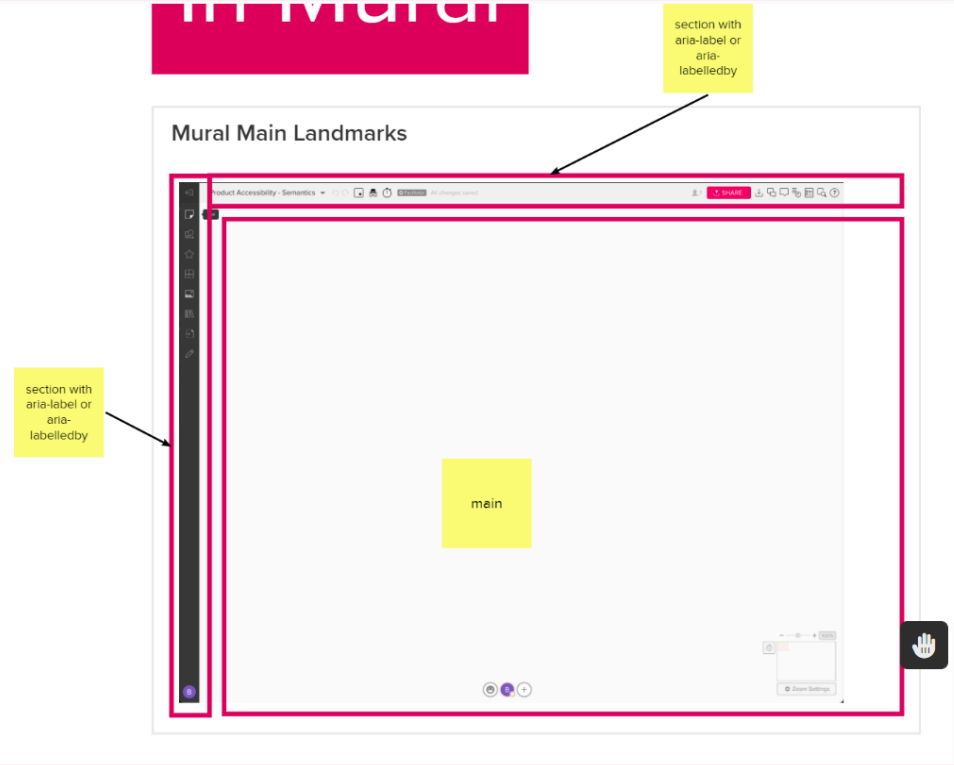
Landmarks help screen reader users obtain information quickly and efficiently about a page and how it's organized:

Definition:

Landmarks: Structural elements in HTML that define the main regions of a webpage or application. Screen readers and other assistive technologies use landmarks to navigate and determine page content (e.g., header, main, footer, and custom landmarks).

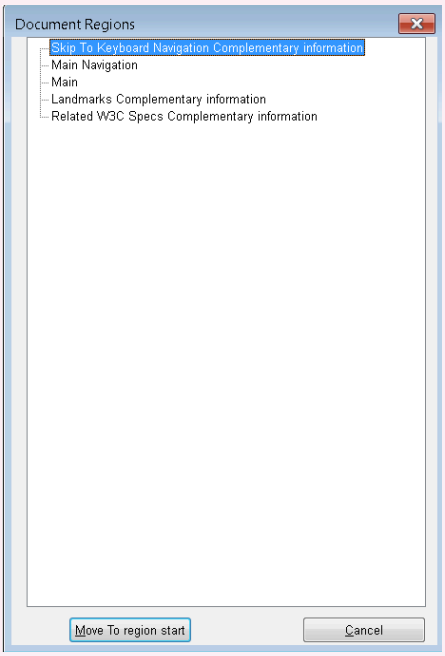
- Provide unique names for landmarks to ensure the end user understands the purpose of the landmarks.
- Note what content goes under which landmarks. Header, main, search, navigation, and footer are the most common landmarks.

Example: The screenshot below shows an example of how designers notate landmarks for MURAL.



- Keep it short. If there are a lot of landmarks, users may spend too much time hunting for sections of the page.
- Provide a semantic name for custom landmarks rather than a visual descriptor (e.g. collaboration tools vs bottom bar).

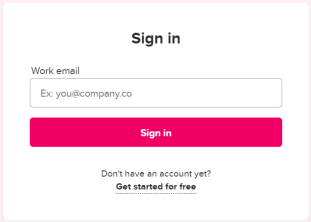
Example: Screen reader users will sometimes have JAWS or other screen readers pull up the list of landmarks and read them. The screenshot below is an example of how JAWS compiles landmarks into a list for convenience.



Provide visual labels where possible and include clear, consistent, and concise instructions and task-oriented buttons whenever user input is required.

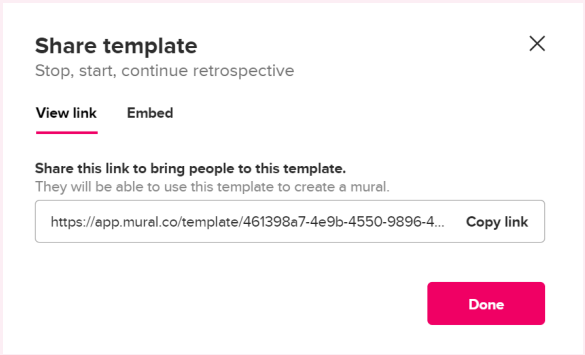
- Annotations for labels are only needed if there aren't visual labels. While it's best to provide visual labels, always check your design for any missing labels, particularly those that are hidden or invisible to the end user.

Examples: The label in the screenshot below indicates that a work email is required to sign into MURAL. The button is task-oriented, instructing the end user to sign in.



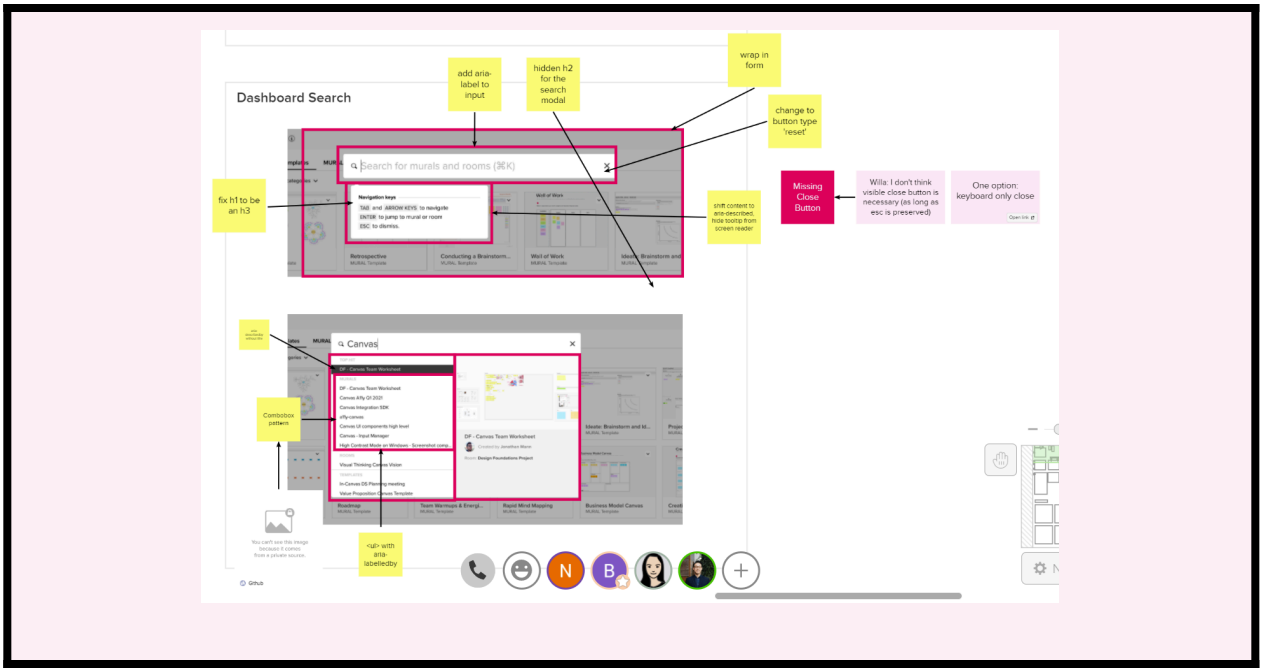
- Include an accessible name, such as an aria label, for invisible labels.

Example: The close button in the MURAL Share template modal displays the letter 'x'. The property aria-label="close" can be used to provide an accessible name to the button.



- Indicate required fields with an accessible label or legend.
- Connect each label in the wireframes.

Example: The screenshot belows show how MURAL designers document labels and accessible names consistently throughout the wireframe.



- Don’t replace labels or crucial information with placeholder text, as assistive technologies behave inconsistently.

Use accessibility verification and handoff tools.

Tools like Stark help you design accessible products and check UI elements and code to ensure they meet WCAG (Web Content Accessibility Guidelines).

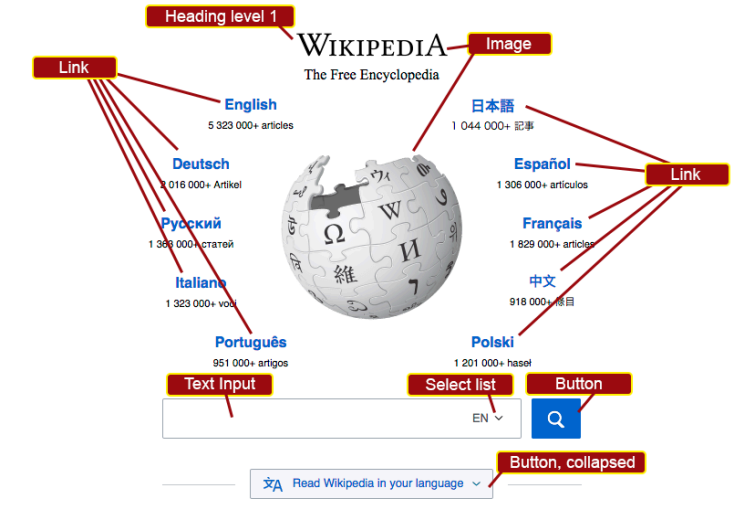
Best practices, in a nutshell

- **Add annotations and more annotations!**
- **Document rem unit specs.**
- **Document the entire keyboard user journey**.
- **Note the tab order of focusable elements such as links, buttons, input fields, etc., and ensure the order is logical.**
- **Specify landmarks so that screen reader users can obtain information quickly and efficiently about a page and how it’s organized.**
- **Make sure your design includes clear, consistent, and concise labels, instructions, and task-oriented buttons whenever user input is required.**
- **Use accessibility verification and handoff tools.**

Test your knowledge

Choose one or more answers to the following questions.

1. The red labels in the following image are used to demonstrate how a screen reader user experiences the web. The screen reader would read each red label aloud. Can you spot what's wrong with this page based on a screen reader user's experience of it?



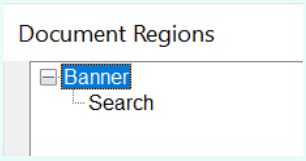
   A. There are no landmarks listed.

   B. Nothing is wrong with this page.

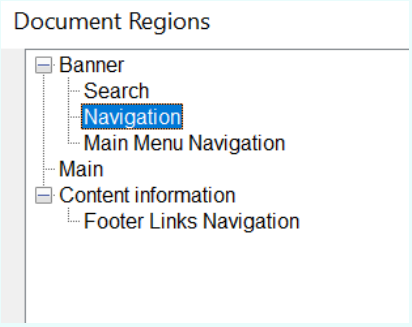
   C. The search button does not have a label.

   D. Both A and C.

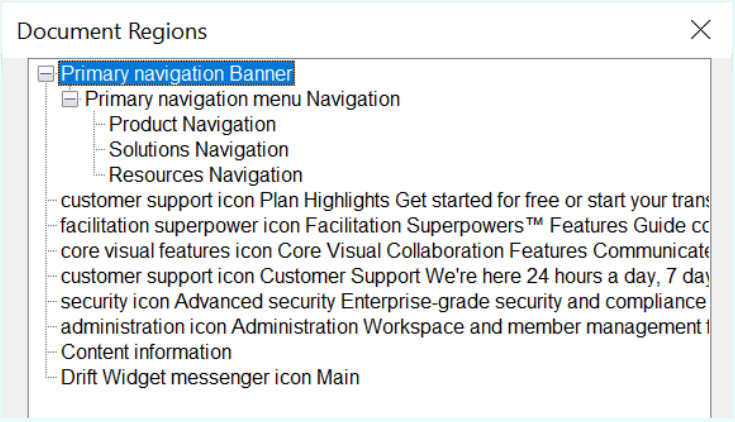
2. Using JAWS screen reader, Sophia has pulled up a list of landmarks of the various pages she's visited. Which list is most helpful for her to understand and navigate the page?



   A. Because the simpler the landmarks are, the better.



B. Because the landmarks are distinguishable from one another.



C. Because there are a lot of landmarks.

3. Eduardo is designing a survey for MURAL collaborators. He's placed instructions and labels in the form fields. What else does he need to do before handing off his design?

   A. Ensure required fields are noted.

   B. Ensure participants can just skip the form fields and submit the survey.

   C. Avoid noting landmarks - they're not needed in a survey.

   D. Nothing - he's all set!

4. In Eduardo's survey for MURAL collaborators, he's noticed that keyboard users would reach the green button to submit the form before actually moving through the form. What does he need to do before handing off his design?

   A. Nothing, Karen. They can move through it and submit it. We're cool.

   B. Make sure the developer knows this is how the design works. You're the boss of them.

C. Change the tab order to move through the form fields before going to the submit button.

D. Make sure the button says “submit.”

E. Both C and D.