

Criteria	WCAG Version	WCAG Level	Recommendation
1.1.1 Non-text Content	2	A	All img tags must have alt attributes.
1.1.1 Non-text Content	2	A	If short alt text is sufficient to describe an image, provide the short text via the image's alt attribute.
1.1.1 Non-text Content	2	A	If a short text alternative is not sufficient to describe an image (such as in a chart, graph, or diagram), provide short text via the image's alt attribute, and include a long description in nearby text.
1.1.1 Non-text Content	2	A	If an image or icon is used as a button or link, the image has a text alternative sufficient to describe the purpose of the button or link.
1.1.1 Non-text Content	2	A	Images that are decorative, used for formatting, or contain content already conveyed in text have a null alt attribute or are implemented as CSS background images.
1.1.1 Non-text Content	2	A	Frames and iframes have descriptive titles.
1.1.1 Non-text Content	2	A	Minimize the number of adjacent links to the same destination by combining adjacent images and text into a single link, rather than creating a separate link for each element.
1.2.1 Audio-only and Video-only (Prerecorded)	2	A	For pre-recorded audio (without video), provide a descriptive transcript that includes dialogue and all other meaningful sound.
1.2.1 Audio-only and Video-only (Prerecorded)	2	A	For pre-recorded video (without audio), provided a text alternative or audio descriptions that provide the same information presented
1.2.2 Captions (Prerecorded)	2	A	Provide captions for prerecorded audio content in non-live synchronized media.
1.2.3 Audio Description or Media Alternative (Prerecorded)	2	A	For non-live video, provide a descriptive transcript or an audio description.
1.2.4 Captions (Live)	2	AA	Provide captions for live audio and video.
1.2.5 Audio Description (Prerecorded)	2	AA	Videos should include "radio style" narration so that content makes sense if someone is consuming just the audio track. Include any text elements in the narration.
1.3.1 Info and Relationships	2	A	Use semantic markup to designate headings, lists, figures, emphasized text, etc.
1.3.1 Info and Relationships	2	A	Organize pages using properly nested HTML headings.
1.3.1 Info and Relationships	2	A	Use ARIA landmarks and labels to identify regions of a page.
1.3.1 Info and Relationships	2	A	Reserve tables for tabular data, use table headers appropriately, and use table captions.
1.3.1 Info and Relationships	2	A	When the appearance of text conveys meaning, also use appropriate semantic markup.
1.3.1 Info and Relationships	2	A	Avoid emulating links and buttons. Use the a and button tags appropriately. Avoid using a tags for buttons.
1.3.1 Info and Relationships	2	A	Avoid using div, span, etc. tags for links or buttons.
1.3.1 Info and Relationships	2	A	Avoid using whitespace characters for layout purposes.
1.3.2 Meaningful Sequence	2	A	Ensure that the source order presents content meaningfully. When the page is viewed without styles, all content on the page should still appear in a meaningful and logical order.
1.3.3 Sensory Characteristics	2.1	A	Do not identify content based on its color, size, shape, position, sound, or other sensory characteristics.
1.3.3 Sensory Characteristics	2.1	A	Do not convey information solely through icons or symbols.
1.3.4 Orientation	2.1	AA	All content and functionality should be available regardless of whether a mobile device is oriented vertically or horizontally, unless the orientation of the device is absolutely essential.
1.3.5 Identify Input Purpose	2.1	AA	If a form field asks for information about the user and if there is an appropriate HTML autocomplete attribute, include that autocomplete attribute.
1.4.1 Use of Color	2	A	Links should always be easily identifiable through non-color means, including both default and hover states.
1.4.1 Use of Color	2	A	The easiest and most conventional way to signify links is underlining.
1.4.1 Use of Color	2	A	Required fields and fields with errors must include some non-color way to identify them.

1.4.1 Use of Color	2	A	When the color of words, backgrounds, or other content is used to convey information, also include the information in text.
1.4.2 Audio Control	2	A	Do not have audio that plays automatically on the page. When providing audio, also provide an easy way to disable the audio and adjust the volume.
1.4.3 Contrast (Minimum)	2	AA	Text (including images of text) have a contrast ratio of at least 4.5:1. For text and images of that is at least 24px and normal weight or 19px and bold, use a contrast ratio that is at least 3:1.
1.4.4 Resize text	2	AA	Ensure that there is no loss of content or functionality when text resizes.
1.4.4 Resize text	2	AA	Define texts and text containers in relative units (percents, ems, rems) rather than in pixels.
1.4.5 Images of Text	2	AA	Avoid images of text, except in cases such as logos.
1.4.10 Reflow	2.1	AA	Provide responsive stylesheets such that content can be displayed at 320px wide without horizontal scrolling. (Content which must be displayed in two dimensions, such as maps and data tables, may have horizontal scrolling.)
1.4.11 Non-text Contrast	2.1	AA	Color contrast for graphics and interactive UI components must be at least 3:1 so that different parts can be distinguished.
1.4.11 Non-text Contrast	2.1	AA	When providing custom states for elements (e.g. hover, active, focus), color contrast for those states should be at least 3:1.
1.4.12 Text Spacing	2.1	AA	Avoid using pixels for defining the height and spacing (e.g. height, line height, etc) of text boxes.
1.4.13 Content on Hover or Focus	2.1	AA	For tooltips, follow corresponding ARIA authoring practice.
1.4.13 Content on Hover or Focus	2.1	AA	For content that appears on hover and focus: the content should be dismissible with the escape key; the content itself can be hovered over; and the content remains available unless it is dismissed, it is no longer relevant, or the user removes hover and focus.
1.4.13 Content on Hover or Focus	2.1	AA	To the extent possible, content that appears on hover or focus should not obscure other content, unless it presents a form input error. or can be dismissed with the escape key.
2.1.1 Keyboard	2	A	Avoid implementing access keys. When access keys and other keyboard shortcuts are implemented, they must not interfere with existing browser and screen reader provided shortcuts.
2.1.1 Keyboard	2	A	All functionality should be available to a keyboard without requiring specific timing of keystrokes, unless the functionality cannot be provided by a keyboard alone.
2.1.1 Keyboard	2	A	Avoid relying exclusively on pointer-driven events, such as onmouseover, to provide functionality when scripting. Generally, such functionality will also require scripting for keyboard operability.
2.1.1 Keyboard	2	A	In general, avoid using scripts to remove focus from an element until the user moves focus manually.
2.1.2 No Keyboard Trap	2	A	Ensure keyboard focus is never trapped on an element without an obvious way to move focus out of the element. Make sure the user can move focus to and from all focusable elements using a keyboard only.
2.1.4 Character Key Shortcuts	2.1	A	If a keyboard shortcut uses only letter (including upper- and lower-case letters), punctuation, number, or symbol characters, then the user must be able to disable the shortcut, remap the shortcut, or limit the shortcut to only when a particular interactive element has focus.
2.1.4 Character Key Shortcuts	2.1	A	If a keyboard shortcut uses only letter (including upper- and lower-case letters), punctuation, number, or symbol characters, then the user must be able to disable the shortcut, remap the shortcut, or limit the shortcut to only when a particular interactive element has focus.
2.2.1 Timing Adjustable	2	A	Do not require time limits to complete tasks unless absolutely necessary. If a time limit is necessary, the time limit should be at least 20 hours, or it can be extended, adjusted, or disabled.

2.2.2 Pause, Stop, Hide	2	A	Items on the page should not automatically move, blink, scroll, or update, including carousels. If content does automatically move, blink, scroll, or update, provide a way to pause, stop, or hide the moving, blinking, scrolling, or updating.
2.3.1 Three Flashes or Below Threshold	2	A	Do not provide any content that flashes more than three times in any 1-second period.
2.4.1 Bypass Blocks	2	A	Provide a link to skip to the main content as the first focusable link on the page.
2.4.2 Page Titled	2	A	Make sure each web page has a title tag that is descriptive, informative, and unique.
2.4.3 Focus Order	2	A	Create a logical tab order through links, form controls, and interactive objects.
2.4.3 Focus Order	2	A	When inserting content into the DOM, insert the content immediately after the triggering element, or use scripting to manage focus in an intuitive way. When triggering dialogs and menus, make sure those elements follow their trigger in the focus order in an intuitive way. When content is dismissed or removed, place focus back on the trigger.
2.4.3 Focus Order	2	A	Avoid using tab index values greater than 0.
2.4.4 Link Purpose (In Context)	2	A	The purpose of each link can be determined from the link text alone, or from the link text and the containing paragraph, list item, or table cell, or the link text and the title attribute.
2.4.4 Link Purpose (In Context)	2	A	If the visible text alone is not sufficient to convey meaning, use advanced techniques to provide additional meaning, such as ARIA attributes, screen reader only text, or the title attribute.
2.4.5 Multiple Ways	2	AA	Each website should include at least two of the following: a list of related pages; table of contents; site map; search; or list of all pages.
2.4.6 Headings and Labels	2	AA	Ensure that on each page, headings, landmark labels, and form labels are unique unless the structure provides adequate differentiation between them.
2.4.7 Focus Visible	2	AA	Provide keyboard focus styles that are highly visible, and make sure that a visible element has focus at all times when using a keyboard. Do not rely on browser default focus styles.
2.5.1 Pointer Gestures	2.1	A	Do not require multipoint or path-based gestures (e.g. pinching, swiping, dragging) for functionality unless the gesture is essential to the functionality.
2.5.2 Pointer Cancellation	2.1	A	Avoid triggering functionality on down-events, such as onmousedown. Use events such as onclick instead.
2.5.2 Pointer Cancellation	2.1	A	If a function is triggered on an up-event (e.g. onmouseup), provide a way to abort or undo the function.
2.5.3 Label in Name	2.1	A	The accessible name for a UI element must contain any visual label for the element. Accessible names for UI elements should match visual labels as closely as possible.
2.5.4 Motion Actuation	2.1	A	Avoid activating functionality through motion (e.g. shaking a phone). If motion triggers functionality, provide a way to disable the motion trigger, and provide an alternative way to activate the functionality.
3.1.1 Language of Page	2	A	Provide a lang attribute on the page's html element.
3.1.1 Language of Page	2	A	When a visual label is present for an interactive element (e.g. link or form control), the accessible name of the element should contain the visual label.
3.1.2 Language of Parts	2	AA	If a portion of the page is in a different language, use the lang attribute on that part.
3.2.1 On Focus	2	A	When the focus change, the page should not cause a change in page content, spawn a new browser window, submit a form, cause further change in focus, or cause any other change that disorients the user.
3.2.2 On Input	2	A	When a user inputs information or interacts with a control, the page should not cause a change in page content, spawn a new browser window, submit a form, cause further change in focus, or cause any other change that disorients the user. If an input causes such a change, the user must be informed ahead of time.
3.2.3 Consistent Navigation	2	AA	When components are repeated across web page, they should appear in the same relative order with regard to other repeated components on each web page where they appear.

3.2.3 Consistent Navigation	2	AA	When a navigation menu is presented on multiple pages, the links should appear in the same order on each page.
3.2.4 Consistent Identification	2	AA	When components have the same functionality across several web pages, the components are labeled consistently on each page. Programmatically indicate required fields using the required or aria-required attributes. Also, visually indicate required fields in the form's instructions or form labels. Do not indicate required fields for CSS alone.
3.3.1 Error Identification	2	A	
3.3.1 Error Identification	2	A	Make errors easy to discover, identify, and correct.
3.3.1 Error Identification	2	A	Identify errors using aria-invalid.
3.3.2 Labels or Instructions	2	A	Use semantic, descriptive labels for inputs. Visually position labels in a consistent way that makes associating labels with form controls easy. Do not rely on placeholder text in lieu of an HTML label.
3.3.2 Labels or Instructions	2	A	Provide text instructions at the beginning of a form or set of fields that describes the necessary input.
3.3.2 Labels or Instructions	2	A	When providing inline help text, use aria-describedby to associate the help text with the input.
3.3.3 Error Suggestion	2	AA	If an input error is detected and if suggestions for correction are known, provide suggestions for fixing the submission. Provide easy ways to confirm, correct, or reverse a user action where a mistake would cause a serious real-world consequence (e.g. submitting financial data, entering into a legal agreement, submitting test data, or making a transaction).
3.3.4 Error Prevention (Legal, Financial, Data)	2	AA	
4.1.1 Parsing	2	A	Validate all page HTML, and avoid significant validation / parsing errors.
4.1.2 Name, Role, Value	2	A	Avoid creating custom widgets when HTML elements already exist. For example, use a and button tags appropriately.
4.1.2 Name, Role, Value	2	A	When creating a custom interactive widget, consult the ARIA Authoring Practices Document. Use ARIA labels, descriptions, roles, states, and properties to expose information about the component.
4.1.2 Name, Role, Value	2	A	Use ARIA to enhance accessibility only when HTML is not sufficient. Use caution when providing ARIA roles, states, and properties.
4.1.3 Status Messages	2.1	AA	When conveying a status message, use ARIA live regions or ARIA alerts to announce the message to screen reader users.

Conformance	Notes
Supports	
Supports with exceptions	doesn't apply to thumb images used for template design, in those cases we use ARIA to describe the purpose and function of button
Does not support	we do not use images as button
Supports	
Supports	
N/A	
Supports	
N/A	
Supports	
Supports with exceptions	widget is injected code on page, we do not use the normal structure of a blank HTML page
Supports	
N/A	
Supports	
Supports	
Supports	
Does not support	because the nature of the widget and some design/architecture constraints (pop up on bottom right corner) this is not achievable fully
Supports with exceptions	
Supports with exceptions	when we do use images for button, those buttons use ARIA attributes to convey function and purpose
Supports	
Supports with exceptions	
Supports	
Supports	

Supports

N/A

Supports

Supports

Does not support

Supports

N/A

Supports

Supports

Does not support

Supports with exceptions

Supports

Supports

Supports

Supports with exceptions

Supports

Supports

Supports

N/A

N/A

Supports

Supports
Supports
Does not support
N/A
Supports

Supports
Supports with exceptions

Supports

Supports

N/A

Supports

Supports

Supports
Supports with exceptions
Supports

Supports

Supports
N/A

Supports
N/A

N/A

Supports

Supports

N/A

Supports

Supports

Supports

Does not support

Supports

Supports

Supports

Does not support

N/A

N/A

Supports with exceptions

Supports

Supports

Supports