TEXAS TECH UNIVERSITY
HEALTH SCIENCES CENTER™
EL PASO

TTUHSC El Paso

Accessibility Guidelines
Levels of Conformance

Version 1.2

2/8/2024

Prepared for:

Texas Tech University

Health Sciences Center

El Paso

Document Revision History

Document Version	Revised Date	Modified By	Section, Page(s) and Text Revised	Comments
1.0	05/27/2020	Mario Alvarado	Entire document	Initial version revised by Adrian Cazares
1.1	5/28/2020	Adrian Cazares	Scope	Policy information included
1.2	2/8/2024	Mario Alvarado	Entire document	Updating WCAG version

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide basic guidelines to make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. Following these guidelines will also often make your Web content more usable to users in general.

1.2 Scope

This document is intended to be a complement of the TTUHSC El Paso programming standards documentation according to our 56.40.06 - Applications Development, Life Cycle, and Coding Standards policy, it also provides guidance on the architecture, formatting, commenting, naming, and programming style of C# source code and is applicable to component libraries, web sites, web services, and client – server applications, therefore accessibility assessment will be validated whenever there is a new release for in-house applications. Accessibility compliance will be validated for third party applications as well in the case of new acquisitions or upgrades.

2. Web Content Accessibility Guidelines (WCAG) 2.2

2.1 Introduction

Web Content Accessibility Guidelines (WCAG) is developed through the W3C process in cooperation with individuals and organizations around the world, with a goal of providing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments internationally [1].

The WCAG documents explain how to make web content more accessible to people with disabilities. Web “content” generally refers to the information in a web page or web application, including:

- Natural information such as text, images, and sounds.
- Code or markup that defines structure, presentation, etc.

2.2 Layers of Guidance Definitions

2.2.1 Principles

1. Perceivable - Information and user interface components must be presentable to users in ways they can perceive.
2. Operable - User interface components and navigation must be operable.
3. Understandable - Information and the operation of user interface must be understandable.
4. Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies. [1]

2.2.2 Guidelines

Under the principles are 13 guidelines that provide basic goals that authors/developers should work toward in order to make content more accessible to users with different disabilities. The guidelines are

not testable, but provide the framework and overall objectives to help authors understand the success criteria and better implement the techniques. [1]

1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
1.2 Provide alternatives for time-based media.
1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
1.4 Make it easier for users to see and hear content including separating foreground from background.
2.1 Make all functionality available from a keyboard.
2.2 Provide users enough time to read and use content.
2.3 Do not design content in a way that is known to cause seizures.
2.4 Provide ways to help users navigate, find content, and determine where they are.
2.5 Make it easier for users to operate functionality through various inputs beyond keyboard.
3.1 Make text content readable and understandable.
3.2 Make Web pages appear and operate in predictable ways.
3.3 Help users avoid and correct mistakes.
4.1 Maximize compatibility with current and future user agents, including assistive technologies.

2.2.3 Success Criteria

For each guideline, testable success criteria are provided to allow WCAG 2.2 to be used where requirements and conformance testing are necessary such as in design specification, purchasing, regulation, and contractual agreements. In order to meet the needs of different groups and different situations, three levels of conformance are defined: A (lowest and minimum for some entities), AA (TTUHSCEP goal), and AAA (highest). [1]

2.2.4 Sufficient and Advisory Techniques

For each of the guidelines and success criteria in the WCAG 2.2 document itself, the working group has also documented a wide variety of techniques. The techniques are informative and fall into two categories: those that are sufficient for meeting the success criteria and those that are advisory. The advisory techniques go beyond what is required by the individual success criteria and allow authors to better address the guidelines. Some advisory techniques address accessibility barriers that are not covered by the testable success criteria. Where common failures are known, these are also documented. [1]

3. Levels of Conformance

3.1 Levels

The success criteria arranged in 3 levels:

- **A** - Minimum level –This level affects the broadest group with the most benefits and is essential.
- **AA** - More accessible (TTUHSCEP goal) – Even with minimum support some barriers will still exist for some people. Not meeting criteria at this level may have an impact on certain groups of users and address the criteria may impact the look of a page to a greater extent or affect site logic.
- **AAA** – Even more accessible – Some Level AAA criteria cannot be applied everywhere, so level AAA is generally not required. That being said, even meeting level AAA does not make web pages accessible to everyone. [2]

3.2 WCAG 2.2 Guidelines by Success Criteria

3.2.1 Level A

Guideline	**Summary** [3]	**Justification** [4] [5]	**Solution** [4] [5]
1.1.1 – Non-text Content	Provide text alternatives for non-text content	These can be read by the user or voiced by assistive technology.	Add a text alternative to your images
			If the content is video or audio, add a short description of the topic
			If a control or input field is non-text, add a name
1.2.1 – Audio-only and Video-only (Pre-recorded)	Provide an alternative to video-only and audio-only content	By providing the same information in a different format, these users can access the content by other means, such as text or assistive technology.	Provide a text transcript that conveys the same information as audio-only media;
			Provide a text transcript that conveys the same information as video-only media;
			Provide an audio-track that conveys the same information as video-only media.
1.2.2 – Captions (Pre-recorded)	Provide captions for videos with audio	Presenting this in caption means these users can fully enjoy the content.	Add captions to all videos with sound.
			Caption all spoken word.
			Identify speakers.
			Caption non-speech information (such as sounds).

1.2.3 – Audio Description or Media Alternative (Pre-recorded)	Video with audio has a second alternative	While captions can provide some assistance, adding an audio description track or text transcript helps more users enjoy your content.	Provide a full text transcript of the video; Provide a version of the video with audio description.
1.3.1 – Info and Relationships	Logical structure	Assistive technology often relies on correct formatting and logical structures to work. When a user experiences the site through a screen reader, other assistive technology or without CSS they should still understand the content.	Break up content with subheadings for new sections Mark headings with HTML header tags Use lists, tables and other formats where needed Use the correct HTML for all structural elements Use valid HTML everywhere Use clear labels and alternative text on forms
1.3.2 – Meaningful Sequence	Present content in a meaningful order	Users who rely on assistive technology (such as a screen reader) to interpret content, require content to be presented in a meaningful order. If this is presented out of sequence, users may become disorientated and will not understand the content.	Present all content in a meaningful order Separate navigation menus from the content Use paragraphs in order Nest headings from H1 downwards to show their relative importance Choose whether a list needs numbering or not Use valid HTML
1.3.3 – Sensory Characteristics	Use more than one sense for instructions	Saying things like 'Use the search bar on the right' isn't helpful to a user who doesn't understand what right is. The main solution is to use more than one indicator for instructions.	Use more than one sense for instructions. Avoid instructions that rely on sound.
1.4.1 – Use of Colour	Don't use presentation that relies solely on colour	Users with visual impairment need help when you use color on your website. As many as 1 in 12 men have some degree of color blindness.	Instructions must not rely on color alone. Other information (like charts and graphs) must not rely on color alone.
1.4.2 – Audio Control	Don't play audio automatically	Automatically playing sounds doesn't help your users. Some of your users will have problems focusing and unexpected sounds can distract them further.	Don't have any audio that plays automatically. Have all the audio you want, just let your users choose when they want to hear it.

2.1.1 – Keyboard	Accessible by keyboard only	Customers with motor impairment, including many elderly customers, need your help to navigate your website.	A clean HTML and CSS website will often have keyboard accessibility without further work
2.1.2 – No Keyboard Trap	Don't trap keyboard users	You must make sure that keyboard-only users can't get stuck anywhere on your site. All parts of your website should be reachable by keyboard alone	Test your website to make sure you can navigate away from, as well as to, all parts of your website by keyboard only.
			Make sure all navigation is controllable by either the 'Tab' or arrow keys, which is a standard many people are familiar with.
2.1.4 – Character Key Shortcuts	Allow users to turn off or remap single-key character shortcuts.	Keyboard shortcuts can help some users, but cause difficulty for those using speech input and some users with motor impairments. They can also cause issues on mobile screens as the functional area is reduced on a mobile keyboard.	Don't use single-key shortcuts
			If you really want to: *give users a way to turn off the shortcut; *allow users to remap the shortcut to use non-character keys; or *ensure the shortcut only works when an element has focus.
2.2.1 – Timing Adjustable	Time limits have user controls	If any of your content is time-controlled, you risk losing users who need more time to read and understand the information on your website.	If your website uses a time limit:
			Give your users an option to turn off the time limit before it begins (for example, a landing page before the time-controlled page can display a message that shows your customers what to do); or
			Give your users the option to adjust the time limit before it begins, over a range of at least ten times the default setting (you can do this with a landing page too); or
			Give your users the option to extend the period at least twenty seconds before it expires. This must be a simple action like clicking a button and must be available to use at least ten times.

			If your website has moving or animated text, users must be able to pause the movement. If your website has a feature that updates automatically (for example, with the latest football scores), you must allow your users to delay the frequency of the updates by at least ten times the default setting.
2.2.2 – Pause, Stop, Hide	Provide user controls for moving content	Content that moves or auto-updates can be a barrier to anyone who has trouble reading stationary text quickly as well as anyone who has trouble tracking moving objects. It can also cause problems for screen readers.	Moving, blinking or scrolling content must have an option to pause, stop or hide it; and Auto-updating content must have the same options; or An option to control frequency
2.3.1 – Three Flashes or Below	No content flashes more than three times per second	Individuals who have photosensitive seizure disorders can have a seizure triggered by content that flashes at certain frequencies for more than a few flashes.	A Web site has video of muzzle flash of machine gun fire, but limits the size of the flashing image to a small portion of the screen below the flash threshold size. A movie with a scene involving very bright lightning flashes is edited so that the lightning only flashes three times in any one second period.
2.4.1 – Bypass Blocks	Provide a 'Skip to Content' link	it may be difficult for people with some disabilities to reach the main content of a Web page quickly and easily	Add a 'Skip to Content' link to all pages on your website. It's best if you make the link visible.
2.4.2 – Page Titled	Use helpful and clear page titles	When titles appear in site maps or lists of search results, users can more quickly identify the content they need.	Give each page on your website a unique and descriptive title.
2.4.3 – Focus Order	Logical order	Some users can't use a mouse and will 'tab' through your website –	A well-made HTML website will often comply with this guideline.

		they must access elements in an order that makes sense.	Unplug your mouse and verify that you can use the 'Tab' key to navigate to every part of your website and use every function, including search boxes and forms – make sure each page has a sensible focus order.
2.4.4 – Link Purpose (In Context)	Every link's purpose is clear from its context	Users with assistive technology, like a screen reader, often hear all the links on a page to help them find where they want to go.	The purpose of the link is clear from the link text (for example, 'My blog'); or The purpose of the link is clear from the surrounding content, meaning the same sentence, paragraph or cell in a table (for example, 'Visit my blog'); or If the link is an image, the alt text of the image makes the link purpose clear (for example, 'Luke McGrath – Visit my blog'); and Links with the same destination have the same description (but links don't share a description if they point to different places).
2.4.7 – Focus Visible	Keyboard focus is visible when used.	Where there are multiple elements on a webpage, it helps users to highlight which element has keyboard focus. This helps users who rely on a keyboard to navigate as it shows them which element the keyboard will interact with. Users with attention or short-term memory limitations will also benefit from a visual cue to where focus is located.	When an element has keyboard focus, show a visual indication.
2.5.1 – Pointer Gestures	Multi-point and path-based gestures can be operated with a single pointer.	Provide alternative means of interaction for individuals who may have difficulties performing complex or multi-point gestures.	Where you have a function that requires a multi-point or path-based gesture, provide a way for a user to operate the same function with a single pointer.

2.5.2 – Pointer Cancellation	Functions don't complete on the down-click of a pointer.	It ensures that users have the ability to cancel or undo accidental triggers by moving their finger or pointer away from the function.	Ensure that actions are only taken when a pointer is clicked and released within the boundary of the target. Abort actions where the pointer is released outside the boundary of the target.
2.5.3 – Label in Name	Where a component has a text label, the name of the component also contains the text displayed.	People who operate with voice interaction use the visible labels in their commands.	Ensure that the text label and programmatic name of components match.
2.5.4 – Motion Actuation	Functions operated by motion can also be operated through an interface and responding to motion can be disabled.	Some people cannot hold or move a device steadily.	Ensure users can enable and disable gesture and movement-based controls. Provide a standard interface (such as a button) in addition to motion and gesture controls.
3.1.1 – Language of Page	Page has a language assigned	Setting a language is important because the way that screen readers pronounce words depends on the HTML language assigned to your website.	Ensure that each page of your website has a language assigned to it.
3.2.1 – On Focus	Elements do not change when they receive focus	Helps people with visual disabilities, cognitive limitations, and motor impairments by reducing the chance that a change of context will occur unexpectedly.	Ensure no element changes purely by receiving focus. Avoid both behavioral and visual modifications. Ensure that: Links don't open automatically on focus; and Forms don't submit without the user taking action (such as clicking the 'Submit' button); and There are no automatic pop-ups; and Focus never jumps to another element automatically; and

			No other action that occurs on focus alone causes the page to change
3.2.2 – On Input	Elements do not change when they receive input	When a form receives input from, it must not automatically skip to another field or auto-submit. This can disorientate users.	Forms must not auto-submit when all fields are filled – this prevents your users from checking and editing what they have written.
			Focus (the field where the user will input next) must not automatically jump to the next field in a form once a field is complete.
			Using a control (like selecting yes or no) must not automatically perform the action (for example, selecting to subscribe to a newsletter in a check box must not automatically subscribe your user, they should be able to click a submit button to confirm their decision).
3.2.6 – Consistent Help	Help options are presented in the same order.	People who need help can find it more easily if it's in the same place.	Where help is offered to users on multiple pages of a website, it is done so in a consistent order.
3.3.1 – Error Identification	Clearly identify input errors	In the case of an unsuccessful form submission, re-displaying the form and indicating the fields in error is insufficient for some users to perceive that an error has occurred. Screen reader users, for example, will not know there was an error until they encounter one of the indicators.	Identify and explain to the user any mistakes that you can detect automatically.
			Add error explanation close to the error, showing what is wrong and how to fix it.
3.3.2 – Labels or Instructions	Label elements and give instructions	Users with cognitive disabilities benefits from labels or instructions to form fields and controls which require user input.	Always provide visible labels to every form fields and controls
			Provide instructions where the form fields require specific data or format
			Ensure the labels identify the fields clearly

			Do programmatically associate the labels with their respective fields
			Provide group level labels and associate them with the group of form fields where the user input is required in more than one field like phone number or credit card number; also ensure to provide individual labels through title attribute in such scenarios.
3.3.7 – Redundant Entry	Auto-fill or provide information that's required more than once in the same process.	Some people with cognitive disabilities have difficulty remembering what they entered before.	If a process requires information that a user has previously provided to be entered again in the same process: auto-fill the information; or make the information available to select.
4.1.1 – Parsing	No major code errors	Users who rely on assistive technology will benefit from a well-made website as the technology often relies on HTML parsing.	Ensure HTML elements have complete start (< >) and end (</ >) tags where needed.
			Nest all HTML elements correctly (for example, list objects within an ordered or unordered list).
			Use unique Ids.
			Check that HTML elements don't contain duplicate attributes.
4.1.2 – Name, Role, Value	Build all elements for accessibility	Enables compatibility with assistive technology, such as screen readers, screen magnifiers, and speech recognition software, used by people with disabilities.	Use HTML specifications for any script you author for your website.
			If you use a plugin or other element authored by a third party, make sure it uses valid HTML markup.

3.2.2 Level AA

Guideline	**Summary** [3]	**Justification** [4] [5]	**Solution** [4] [5]
1.2.4 – Captions (Live)	Add captions to live videos.	Enable people who are deaf or hard of hearing to watch real-time presentations.	Add captions to live video.
1.2.5 – Audio Description (Pre-recorded)	Provide audio descriptions for pre-recorded videos.	Users with visual impairments or cognitive limitations may rely on audio description to enjoy videos. Adding an audio description soundtrack to videos means these users get all information from the content.	Provide an audio described version of a video's soundtrack, selectable by the user; or Provide an alternative version of your video with audio description.
1.3.4 – Orientation	Your website adapts to portrait and landscape views.	Some users have a preferred alignment (portrait or landscape) or physical requirements and need content to adapt to their preference. Others have visual impairments and may find one way round easier to use.	All content retains meaning and function in either position.
1.3.5 – Identify Input Purpose	The purpose of input fields must be programmatically determinable.	All users, but particularly those with cognitive impairments, benefit from programmatically determinable input fields.	Specify the intention of each input field with "input type="type" Use specific autocomplete values to allow the user's browser to prefill fields where it already has the data
1.4.3 – Contrast (Minimum)	Contrast ratio between text and background is at least 4.5:1.	Some users with visual impairments need a stronger contrast than others to understand your content, so using the right colours is essential.	Make sure the contrast ratio between your text and background is at least 4.5:1.
1.4.4 – Resize Text	Text can be resized to 200% without loss of content or function.	Some users with visual impairments need to change the size text to understand it fully.	Users can resize the text content in their web browser up to 200% without loss of meaning or function.

1.4.5 – Images of Text	Don't use images of text.	Text allows for this kind of personalization, but images of text almost always don't.	Don't use an image of text when you can use plain text. Display quotes as text rather than images. Use CSS to style headings as text. Use CSS to style navigation menus as text.
1.4.10 – Reflow	Content retains meaning and function without scrolling in two dimensions.	When users enlarge content up to 400% of the default size	Ensure vertical content doesn't require a horizontal scroll at a width of 320 CSS pixels, and ensure horizontal content doesn't require a vertical scroll at a height of 256 CSS pixels
1.4.11 – Non-Text Contrast	The contrast between user interface components, graphics and adjacent colours is at least 3:1.	Some users with visual impairments need a stronger contrast than others to fully distinguish and use components such as input fields, buttons and controls	Ensure user controls have a contrast of at least 3:1 to the colour around them; Where controls change colour on focus or use, ensure the colours used have a contrast of at least 3:1; and; Ensure all graphics (for example icons, graphs and charts) have a contrast of at least 3:1 to the colour around them.
1.4.12 – Text Spacing	Content and function retain meaning when users change elements of text spacing.	Users with visual or cognitive impairments may wish to amend the default spacing around text content to make it easier for them to read and understand.	Content and function remain intact when a user changes: Line height to at least 1.5 times the font size; Paragraph spacing to at least 2 times the font size; Letter spacing to at least 0.12 times the font size; and Word spacing to at least 0.16 times the font size.

1.4.13 – Content on Hover or Focus	When hover or focus triggers content to appear, it is dismissible, hoverable and persistent.	Additional content triggered by keyboard focus or mouse hover can cause accessibility issues for users with visual or cognitive impairments	Where keyboard focus or mouse hover triggers additional content to appear, the content must be: Dismissible by the user without moving keyboard focus or mouse hover (for example by pressing the 'escape' key or closing on click); Hoverable by the mouse pointer so the pointer can be moved over the content; and Persistent until the user changes keyboard focus or mouse hover, dismisses the content or the content is no longer valid.
2.4.5 – Multiple Ways	Offer at least two ways to find pages on your website.	Some users will find certain methods easier than others, so it's important to offer a range of options.	Provide multiple ways for users to find your website's pages by: Adding a sitemap page which links to every page on your website; and Including a search function on every page (by adding it to the header); and Providing a clear and consistent main navigation menu.
2.4.6 – Headings and Labels	Headings and labels describe topic or purpose.	Users with slow reading ability or short-term memory issues benefit from headings for sections of content to make it clear what the section contains. People who use screen readers may also use headings to navigate to sections.	Use descriptive headings and subheadings in content where appropriate (a change in topic or purpose) Use descriptive labels on controls and input fields

2.4.11 – Focus Appearance (Minimum)	Focus indicators are clearly distinguishable when active.	Focus indicators help users see which element on a page currently has focus, they are especially useful for people with low-vision, memory or mobility impairments.	When a component receives keyboard focus: The focus indicator has a contrast ratio of at least 3:1 between focused and unfocused states; The focus indicator has a contrast ratio of at least 3:1 against adjacent colours; The element within focus isn't hidden; and The focus indicator area: Has an outline that's at least 1 CSS pixel thick; or Has a shape that's at least the area of a 4 CSS pixel-thick line;
2.5.7 – Dragging Movements	Functionality that uses dragging movements can be achieved with a single pointer without dragging.	Some users with mobility impairments may have difficulty using a dragging action precisely, either by mouse pointer or touch.	Where a control uses dragging, provide an alternative.
2.5.8 – Target Size (Minimum)	The target size for pointer inputs is at least 24 by 24 CSS pixels.	Users with mobility impairments may have difficulty using elements if their target area is small.	Ensure that the target areas for all pointer inputs are at least 24 by 24 CSS pixels.
3.1.2 – Language of Parts	Each part of a webpage has a default human language assigned.	Helps browsers render the content correctly and assistive technology, such as screen readers, to interpret content accurately.	Set the default language of each webpage using the "lang" HTML attribute; and Add a further "lang" attribute to content that is not in the main language.
3.2.3 – Consistent Navigation	Position menus and standard controls consistently.	Users who rely on spatial navigation, due to impaired sight, or need extra help understanding your website benefit most.	Keep navigation menus in the same location on all pages; and Present the options in navigation menus in the same order on all pages; and Keep all other standard controls (for example, your search box) in the same location on all pages.

3.2.4 – Consistent Identification	Identify components with the same function consistently.	Users with screen readers and screen magnifiers benefit even further from consistent identification.	Any icons used are consistent (for example, 'Print page' or Twitter link); and Elements with the same function are labelled and named consistently; or Elements with the same function have a consistent text alternative.
3.2.7 – Visible Controls	Give users a way to identify controls without mouse hover or keyboard focus.	Users with cognitive impairments, visual disabilities, mobility or motor issues may have difficulty using components if they are hidden until hovered over by a mouse pointer or focused on through keyboard navigation.	Ensure that the information needed to identify controls and components is available when the controls are needed. Note: This doesn't mean that controls must be visible all the time.
3.3.3 – Error Suggestion	Suggest corrections when users make mistakes.	Going beyond identifying errors to suggesting corrections is particularly helpful to users with cognitive or visual limitations.	Ensure that: You identify input errors and suggest corrections where possible. When the error is missing a required field, communicate this to the user with a text suggestion. If the error is in the format of the input, the suggestion shows the correct format (for example, 'The date must be in the form DD/MM/YYYY'). If the error is because the input needed to be from a limited list of values, provide these values and explain them.
3.3.4- Error Prevention (Legal, Financial, Data)	Check, confirm and allow reversal of pages that cause important commitments.	Some disabilities and impairments can make users more likely to make errors.	If a process results in: a legal commitment; a financial commitment; modification or deletion of stored data; or submission of test responses

3.3.8 – Accessible Authentication	Don't authenticate users through memory, transcription or cognitive tests without alternatives.	Some users will be unable to recall a password or series of gestures to access their accounts and require help or alternative means to authenticate.	If you are authenticating a user, avoid: asking for a memorized password; and requiring them to type in certain characters; and making the solve any kind of puzzle, calculation or test.
4.1.3 – Status Messages	Alert users to changes in content that aren't given focus.	Users with visual impairments and low vision can benefit from status messages to inform them of changes, results or processes that aren't clear from a change of context.	A 'status message' is a special term used for these guidelines, meaning something that provides information to the user on the: results of an action; waiting state of an application; progress of a process; or existence of errors.

3.2.3 Level AAA

Guideline	**Summary** [3]
1.2.6 – Sign Language (Pre-recorded)	Provide sign language translations for pre-recorded videos.
1.2.7 – Extended Audio description (Pre-recorded)	Provide extended audio descriptions for pre-recorded videos.
1.2.8 – Media Alternative (Pre-recorded)	Provide text alternatives for pre-recorded videos.
1.2.9 – Audio Only (Live)	Provide alternatives for live audio.
1.3.6 – Identify Purpose	The purpose of all components must be programmatically determinable.
1.4.6 – Contrast (Enhanced)	Contrast ratio between text and background is at least 7:1.
1.4.7 – Low or No Background Audio	Audio-only content is clear with no or minimal background noise.
1.4.8 – Visual Presentation	Offer users a range of presentation options for blocks of text.
1.4.9 – Images of Text (No Exception)	Don't use images of text.
2.1.3 – Keyboard (No Exception)	All functionality is accessible by keyboard with no exceptions.
2.2.3 – No Timing	No time limits on your website.
2.2.4 – Interruptions	Users can postpone or suppress non-emergency interruptions.
2.2.5 – Re-authenticating	Save user data when re-authenticating.
2.2.6 – Timeouts	Warn users about timeouts that cause data loss.
2.3.2 – Three Flashes	No content flashes more than three times per second.
2.3.3 – Animation from Interactions	Users can disable motion animation.
2.4.8 – Location	Let users know where they are on your website.
2.4.9 – Link Purpose (Link Only)	Every link's purpose is clear from its text.
2.4.10 – Section Headings	Organize content with headings.
2.4.12 – Focus Appearance (Enhanced)	Focus indicators are more clearly distinguishable when active.
2.5.5 – Target Size	The target size for pointer inputs is at least 44 by 44 CSS pixels.
2.5.6 – Concurrent Input Mechanisms	No restrictions on modes of input.
3.1.3 – Unusual Words	Define any unusual words or phrases.
3.1.4 – Abbreviations	Define any abbreviations.
3.1.5 – Reading Level	Users with nine years of schooling can read your content.
3.1.6 – Pronunciation	Define words where meaning is ambiguous without pronunciation.
3.2.5 – Change on Request	Elements do not change without a request.
3.3.5 – Help	Provide help to users.
3.3.6 – Error Prevention (All)	Check, confirm and allow reversal of pages that require users to submit information.
3.3.9 – Accessible Authentication (Enhanced)	Don't authenticate users through memory, transcription or cognitive tests.

4. Web Accessibility Initiative – Accessible Rich Internet Applications (WAI-ARIA)

The W3C Web Accessibility Initiative (WAI) develops standards and support materials to help the understanding and accessibility implementations [1]. WAI-ARIA is a technical specification that provides a framework to improve the accessibility and interoperability of web content and applications. The incorporation of WAI-ARIA is a way for an author to provide proper semantics for custom widgets to make these widgets accessible, usable, and interoperable with assistive technologies. Authors/developers need to associate elements in the document to a WAI-ARIA role and the appropriate states and properties (aria-* attributes) during its life-cycle.

ARIA attributes are predefined in the spec and are divided into two categories, roles and states & properties. Both can be added directly in the markup or via JavaScript to progressively enhance markup as necessary. The properties and states should be updated as needed based on user interactions. [3]

An ARIA role is added via a role="<ROLE TYPE>" attribute and does not change for an element once set. There are six categories of ARIA roles [3]:

- Landmark
- Document Structure
- Widget
- Window
- Live Regions
- Abstract

ARIA states and properties are often used to support ARIA roles that exist on a page. Properties often describe relationships with other elements and for the most part, do not change once they're set. States are more dynamic and are typically updated with JavaScript as a user interacts with a page. It's common to refer to states and properties collectively as just ARIA attributes. Screen readers are notified when attributes change and can announce these changes to users after an interaction takes place. [3]

References

[1] World Wide Web Consortium, "Understanding WCAG 2.0," W3C, 05 October 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/#wcag-2-layers-of-guidance. [Accessed 2020].

[2] J. Avila, "Level Access," 2020. [Online]. Available: https://www.levelaccess.com/wcag-a-and-wcag-aa-conformance/. [Accessed 2020].

[3] The A11Y Project, "Getting started with ARIA," 2014. [Online]. Available: https://a11yproject.com/posts/getting-started-aria/. [Accessed 2020].

[4] L. McGrath, "Web accessibility for developers," WUHCAG, 2015. [Online]. Available: https://www.wuhcag.com. [Accessed 2020].

[5] R. Satish Peri, "Understanding WCAG Success Criterion," 2020. [Online]. Available: https://www.digitala11y.com. [Accessed 2020].

[6] IBM, "IBM Accessibility Checklist," IBM, 2020. [Online]. Available: https://www.ibm.com/able/guidelines/ci162/accessibility_checklist.html. [Accessed 2020].