WCAG 2.0 AA Checklist

Checkmark =done

Circle=not done

Minus sign=N/A

# Section 1:

* Alternatives Alt text (1.1.1): All images and non-text content that conveys necessary meaning or information needs a text alternative. (Developer, Content Editor)
* Video & Audio alternatives (1.2.1): All video-only and audio-only content has a text transcript. Transcripts are clearly labeled and available near the media. (Content Editor)
* Closed captioning (1.2.2): All video with sound has closed captioning. (Content Editor)
* Audio description (1.2.3): For any video where meaningful information is not available audibly, add a transcript or an audio description describing that information. (Content Editor)
* Live captions (1.2.4): More formal, live presentations must have closed captions. (Content Editor)
* Audio description (1.2.5): An audio description is optional under 1.2.3 level A but not in 1.2.5 AA. The best route is to provide both a transcript (optional) and an audio description but the audio description is required. (Content Editor)

# Section 2:

* Presentation Website structure (1.3.1): Use proper HTML markup techniques to structure your website’s content so that meaning and information presented visually and audibly is available programmatically. (Developer, Content Editor)
* Meaningful order (1.3.2): Make sure that content is ordered in a way that preserves meaning programmatically. (Developer)
* Sensory characteristics (1.3.3): When providing detailed instructions, make it so they aren’t reliant on a single sensory ability. (Content Editor)
* Use of color (1.4.1): Do not rely on color alone to convey information. (Designer, Content Editor)
* Audio control (1.4.2): Any audio must be able to be paused, stopped, or muted. (Developer)
* Color contrast (1.4.3): There must be a color contrast ratio of at least 4.5:1 between regular text and background and 3:1 for large text. (Designer)
* Text resize (1.4.4): Text must be able to be resized up to 200% without negatively affecting the ability to read content or use functions. (Designer)
* Images of text (1.4.5): Do not use images of text unless necessary (e.g., logo). (Designer, Content Editor)

# Section 3:

* **User Control Keyboard only (2.1.1): All functionality on a website must be accessible by keyboard only (i.e., no mouse). (Developer) Tab funkar ej till länkar i td och th**
* No keyboard trap (2.1.2): Keyboard-only users must never get stuck on any part of the website; they must be able to navigate forwards and backwards. (Developer)
* Adjustable time (2.2.1): If there any time limits on a website, users have the ability to turn it off, adjust it, extend it. (Developer)
* Pause, stop, hide (2.2.2): If there is content that blinks, scrolls, moves, users must have the ability to pause, stop, or hide it. (Designer, Developer, Content Editor)
* Three flashes or below (2.3.1): Web pages do not contain anything that flashes more than three times in any one second period. (Designer, Content Editor)
* **Bypass Blocks (2.4.1): A “Skip to Content” link or other mechanism allows users to bypass repeated blocks of content (i.e., the header with menu) and go straight to the main content. Other mechanisms to bypass blocks include headings and landmarks. (Developer, Designer)**

# Section 4:

* Understandable Page titles (2.4.2): Each page of a website needs to have a unique and descriptive page title. (Content Editor)
* Focus order (2.4.3): Users must be able to navigate through focusable (interactive) elements in a logical, sequential order that preserves meaning. (Developer)
* Link anchor text (2.4.4): The purpose of each link should be clear based on its anchor text (e.g., don’t use “click here”, etc.) and/or within programmatic context. (Content Editor)
* **Multiple ways (2.4.5): There are multiple ways to access content/pages on a website (e.g., search bar, nav menus, sitemap, breadcrumbs, etc.). (Designer)**
* Descriptive headings and labels (2.4.6): Headings and labels must be clear and descriptive. They do not need to be lengthy. (Content Editor)
* Focus indicator (2.4.7): Any “user interface control” that receives focus from a keyboard user should have a focus indicator that indicates that focus visually. (Designer)
* Website language (3.1.1): Set the language for your website. (Developer)
* Language changes (3.1.2): Indicate any language changes for an entire page or within the content. (Content Editor)

# Section 5:

* Predictability No focus change (3.2.1): No context changes and nothing is activated merely because an item receives focus. (Developer, Designer)
* No input change (3.2.2): User interface changes or actions shouldn't automatically occur due to input (e.g., form doesn’t auto submit once all fields are filled out). (Developer)
* Consistent navigation (3.2.3): Keep navigation links / layout consistent throughout all pages of the website (e.g., links remain in same relative order). (Designer)
* Consistent identification (3.2.4): Components that have the same function within a website are identified consistently (but not necessarily identically). (Developer, Designer)
* Error identification (3.3.1): Make any form errors easy to identify, understand, and correct. (Developer, Designer)
* Form labels and instructions (3.3.2): Provide clear visual and programmatic labels or instructions for user input fields, ensuring users understand what information is required and how to input it. (Developer)
* **Error suggestions (3.3.3): If an input error is automatically detected, then suggestions for correcting the error should be provided. (Developer)**
* Error prevention on important forms (3.3.4): For pages that create legal commitments or financial transactions or any other important data submissions, one of the following is true: 1) submissions are reversible, 2) the user has an opportunity to correct errors, and 3) confirmation is available that allows an opportunity to review and correct before submission. (Developer)
* Parsing (4.1.1): Make sure HTML code is clean and free of errors, particularly missing bracket closes. Also, make sure all HTML elements are properly nested. (Developer)
* Name, role, value (4.1.2): For all user interface components (including forms, links, components generated by scripts), the name, role, state, and/or value should all be able to be programmatically determined and set by the user. (Developer)

## WCAG 2.1 AA Checklist Note: 2.1 AA includes all of the preceding 2.0 AA success criteria plus the success criteria below.

* Orientation (1.3.4): Web pages and content does not lock on portrait or landscape mode, unless necessary. (Developer)
* Input (1.3.5): The purpose of an input element can be determined so browsers and assistive technology can help guide and facilitate inputting information (e.g., provide autocomplete option). (Developer)
* **Reflow (1.4.10): Ensure someone can zoom in on your website without requiring scrolling or causing poor experience. (Designer)**
* Non-text contrast (1.4.11): All meaningful non-text content (e.g., buttons, form fields, icons, etc.) on your website should have a minimum 3:1 color contrast ratio to ensure they stand out. (Designer)
* **Text spacing (1.4.12): Make sure your text spacing can be adjusted without causing a poor experience. (Designer)**
* Content on hover or focus (1.4.13): Make it so any additional content (e.g., pop-ups, submenus) can be dismissed or remain visible if the user desires. (Designer, Developer)
* Keyboard shortcuts (2.1.4): If you have a keyboard shortcut, make sure a user can either 1) turn it off, 2) there’s a way to add another key in the shortcut, and/or 3) have the shortcut only active while focusing on a specific component. (Developer)
* Pointer gestures (2.5.1): Provide simple alternatives (e.g., single tap vs. swipe) to potentially complex finger motions on touch screens. (Developer)
* Pointer cancellation (2.5.2): Provide a way to cancel the trigger action when you activate a function with a “down-event” using a mouse or press /touch with your finger. (Developer)
* **Label in Name (2.5.3): Make sure any programmatic labels you make are aligned with the corresponding visual text. (Developer)**
* Motion Actuation (2.5.4): For any functions that are activated by motion, provide a simpler, alternative means of action. Also, give users the option to turn off motion activation. (Developer)
* Status Messages (4.1.3): When a status message appears, it should be coded with role or properties so that people using assistive technologies (e.g., screen readers) are alerted without losing focus. (Developer)