STRATEGIC PLAN FOR WEB ACCESSIBILITY

1.0 ORGANIZATIONAL STATEMENT

The California Association of Health and Education Linked Professions, a Joint Powers Authority (CAHELP JPA), values diverse experiences and perspectives and strives to fully include everyone who engages with the organization. Therefore, CAHELP is committed to ensuring that individuals with disabilities have an opportunity equal to that of nondisabled peers accessing CAHELP programs, benefits, and services, including those delivered through information technology (IT). The CAHELP Strategic Plan for Web Accessibility, hereinafter referred to as "SPWA" establishes a foundation for equality of opportunity and provides guidance to ensure equal access to IT the CAHELP purchases, creates, and uses, such as websites, software, hardware, and media in accordance with applicable state and federal laws including, but not limited to, Sections 504 and 508 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act as amended (ADA).

The SPWA shall apply to all new, updated, and existing online web content and functionality. All CAHELP web content shall meet WCAG 2.0 Level AA conformance and shall be updated according to changes in WCAG standards, as best practice, and in accordance to federal and state law. WCAG 2.1 was published on June 5, 2018 by the Worldwide Web Consortium (W3C) and establishes 17 new success criteria that have been included in this policy. By conforming to WCAG 2.1, CAHELP also conforms to WCAG 2.0.

2.0 DEFINITIONS

Accessible: Refers to the concept that individuals with disabilities are able to access and use a product or system, including with the help of assistive technologies. For example, an "accessible" web site may be designed so that the text can be enlarged by the user, rather than having a fixed font size, or may be designed so that it can be interpreted and "read out loud" by screen reader software used by blind or low-vision individuals.

Accessible Information Technology: Information technology that has been designed, developed, or procured to be usable by, and therefore accessible to individuals with disabilities, including those who use assistive technologies.

Assistive Technologies: Adaptive, rehabilitative devices that promote greater independence for individuals with disabilities by changing how these individuals interact with technology. Examples include special input devices (e.g., head or foot mouse, speech recognition), screen reading software, and screen magnifiers.

Usability: Refers to how easily, effectively, and efficiently users can use a product or system to achieve their goals, and how satisfied they are with the experience.

3.0 REGULATORY REQUIREMENTS (SECTIONS 504/508; TITLE II ADA, CALIFORNIA UNRUH CIVIL RIGHTS ACT)

Accessibility awareness is an important aspect of the CAHELP's underlying legal obligation to ensure that individuals with disabilities have equal access to programs, services, and information within the same timeframe as nondisabled peers. No individual shall be excluded from participation in, deny the benefits of, or otherwise be subjected to discrimination from any of the CAHELP programs, services, and activities, including those delivered through information technology. The regulatory requirements in Sections 504 and 508 of the Rehabilitation Act of 1973, and Title II of the Americans with Disabilities Act (ADA), as amended in 1990, provide the basis for equal access and governs the overall responsibility of CAHELP content developers and approvers, webmasters, procurement officials, and all others responsible for content management, to ensure that online content and functionality are equally accessible to all.

Section 504 and Title II of the ADA are implicit and require public agencies to make web pages accessible. The ADA prohibits discrimination against individuals with disabilities by any state or local government and any of its department, agencies, or other instrumentalities. Section 504 prevents intentional or unintentional discrimination based on an individual's disability and applies to employers and organizations that receive federal financial assistance. Section 508 is limited to federal agencies but is extremely influential because its compliance standards require federal agencies to provide software and website accessibility to individuals with disabilities. The California Unruh Civil Rights Act (UCRA) is a California statute providing that all persons within the jurisdiction of this state are free and equal, and no matter what their disability are entitled to full and equal accommodations, advantages, facilities, privileges, or services in all business establishments of every kind whatsoever. The UCRA specifies that "[a] violation of the right of any individual under the Americans with Disabilities Act of 1990…shall also constitute a violation of this section."

> Title II Americans with Disabilities Act (ADA). *"…Protect qualified individuals with disabilities from discrimination on the basis of disability in the services, programs, or activities of all State and local governments. It additionally extends the prohibition of discrimination on the basis of disability established by section 504 of the Rehabilitation Act of 1973, as amended, to all activities of State and local governments, including those that do not receive Federal financial assistance. By law, the Department of Justice's Title II regulation adopts the general prohibitions of discrimination established under section 504 and incorporates specific prohibitions of discrimination from the ADA.*
>
> Section 504, Title 29 of the United States Code § 794. *"No otherwise qualified individual with a disability in the United States…shall, solely by reason of her or his disability, be excluded from participation in, be denied the benefits of, or be subjected to discrimination under any program or activity receiving Federal financial assistance."*

Section 508, Title 29 of the United States Code § 1194.1. *"…Section 508 also requires that individuals with disabilities, who are members of the public seeking information or services from a Federal agency, have access to and use of information and data that is comparable to that provided to the public who are not individuals with disabilities, unless an undue burden would be imposed on the agency."*

California Unruh Civil Rights Act (UCRA), Cal Civ. Code § 51. (a) This section shall be known, and may be cited, as the Unruh Civil Rights Act.

(b) All persons within the jurisdiction of this state are free and equal, and no matter what their sex, race, color, religion, ancestry, national origin, disability, medical condition, genetic information, marital status, sexual orientation, citizenship, primary language, or immigration status are entitled to the full and equal accommodations, advantages, facilities, privileges, or services in all business establishments of every kind whatsoever…"

Refer to Appendix D for WCAG 2 Checklist produced by Web Accessibility in Mind (WebAIM).

3.1 Legal Guidance:

- Department of Justice (DOJ) Guidance (June 2003)
- ADA/504 "generally require" equal access unless fundamental alteration or undue burden
- OCR Dear Colleague Letter (June 2010)
  Colleges and universities must make book readers and other educational technologies equally accessible
- OCR FAQs (May 11)
- Follow-up from June 2010 Dear Colleague letter – legal requirements articulated in letter apply to elementary and secondary schools
- DOJ Notice of Proposed Rulemaking (May 2016)
- Proposed rulemaking for state and local governments with regard to web accessibility

NOTE: Effective January 18, 2017, the U.S. Access Board published a final rule updating accessibility requirements for information and communication technology (ICT) covered by Section 508 of the Rehabilitation Act of 1973. Major changes in the revised Section 508 Standards include the incorporation of the web standards in WCAG 2.0 developed by the W3C and clarifies applicability to websites, electronic documents, and software. The final rule also requires all public-facing official agency business content, as well as specific categories of non-public-facing content that is official agency business, to be accessible, and that software and operating systems must interoperate with assistive technology.

4.0 COMPLIANCE/RESPONSIBILITIES

Under this strategic plan, CAHELP personnel shall:

- Adhere to the CAHELP strategic plan for web accessibility;
- Develop, purchase and/or acquire, to the extent feasible, hardware and software products that are accessible to individuals with disabilities; and
- Promote awareness of this strategic plan to all members of the CAHELP community, particularly those in roles that are responsible for creating, selecting, or maintaining electronic content and applications.

4.1 Implementation of the Policy

CAHELP management in collaboration with the CAHELP JPA Virtual Compliance Supervisor is responsible for facilitating and ensuring implementation of this strategic plan for web accessibility with fidelity.

The CAHELP JPA Virtual Compliance Supervisor in collaboration with the designated Accessibility Compliance Team (ACT) is responsible for issuing and updating any requirements, standards or guidelines that support this strategic plan and shall facilitate regular communication among organizational departments to address consistent implementation of this strategic plan throughout CAHELP.

4.2 Revisions to the Strategic Plan

The Chief Executive Officer (CEO) of CAHELP is the approver of the strategic plan for web accessibility and has the authority to approve revisions upon recommendation by the CAHELP JPA Virtual Compliance Supervisor and ACT.

The CAHELP JPA Virtual Compliance Supervisor in collaboration with the ACT has the authority to initiate revisions to the strategic plan and is responsible for regular reviews and updates.

All revisions substantive in nature to the strategic plan will be presented for approval to the CEO and subsequently presented to the CAHELP Governance Council for review and approval.

4.3 Oversight and Responsibilities

The CAHELP JPA Virtual Compliance Supervisor is responsible for online web accessibility and functionality and is a member of the ACT. He/she is responsible for establishing systems of audit, accountability, corrective action of accessibility of all online content and functionality on an ongoing basis. He/she and the ACT shall work towards ensuring equal access and opportunity to organizational programs and services for all individuals, including those delivered online. The ACT shall be comprised of the following:

- Chief Operations Officer, CAHELP
- JPA Virtual Compliance Supervisor, CAHELP
- Representative from IT Department, as needed
- Representative from Web Programmer/Host, as needed

- Representative from Desert/Mountain Children's Center
- Representative from Desert/Mountain Special Education Local Plan Area
- JPA Virtual Compliance Program Technician, CAHELP

Membership of the ACT shall be at the discretion and determination of the CEO, CAHELP.

4.3.1 Responsibilities of Accessibility Compliance Team (ACT)

The ACT responsibilities shall include, but not be limited to, all of the following:

- Report accessibility issues and recommended solutions;
- Review and recommend changes and/or modifications to the strategic plan for web accessibility;
- Evaluate effectiveness of accessibility training(s) and provide recommendations for modifications to improve training and to ensure organizational compliance;
- Participate in audit of website, web developer meetings (contract renewal, web redesign, etc.); evaluate needs of compliance team; and
- Attend regularly scheduled team meetings, appropriate accessibility workshops, trainings, etc.

4.3.2 Responsibilities of the CAHELP JPA Virtual Compliance Supervisor

The CAHELP JPA Virtual Compliance Supervisor shall work in collaboration with the ACT in support of the organization's accessibility requirements and shall:

- Create workflow and approval process for online content;
- Develop, coordinate, implement, and facilitate one-to-one and/or annual training regarding online content accessibility and functionality for content developers and approvers, and other staff as needed;
- Develop, review, revise, and implement strategic plan for web accessibility;
- Provide recommendations for implementation, or modification to establish compliance;
- Contract for services (i.e., auditor, web developer, training, etc.);
- Develop long range plan for addressing problems, taking into account identified priorities, with all proposed remedies to be completed within a reasonable timeframe;
- Set up systems of accountability and verify claims of accessibility by vendors, open sources;

- Set up a system of testing and accountability to maintain the accessibility of all online content and functionality on an ongoing basis;
- Maintain appropriate records;
- Develop, implement, and manage CAHELP strategic plan for web accessibility;
- Develop long range content management and accessibility plans and priorities for CAHELP in conjunction with accessibility requirements under federal and state law, and best practices;
- Analyze, plan, and coordinate the needs for training and educational development in designing and creating accessible materials;
- Develop, implement, and maintain a process for public input and reporting on inaccessible virtual content;
- Investigate, research, analyze, and respond to inquiries and complaints of accessibility-related issues regarding the functionality of the website and virtual content;
- Perform regular accessibility audit of CAHELP website, applications, and external platforms hosting CAHELP content;
- Perform accessibility evaluations for website and applications under consideration for purchase and/or use;
- Manage, monitor, and evaluate budget and expenditure-related activities;
- Supervise, evaluate, and train personnel assigned to the JPA Virtual Compliance team; and
- Attend regularly scheduled team meetings, appropriate accessibility workshops, trainings, etc.

4.3.3 Responsibilities of CAHELP JPA Virtual Compliance Technician

The CAHELP JPA Virtual Compliance Technician performs specialized technical work in assisting, training, and advising CAHELP employees regarding virtual regulations, compliance, policies, and procedures. The CAHELP JPA Virtual Compliance Technician:

- Advises on virtual compliance issues, regulations and procedures;
- Resolves complex virtual compliance issues;
- Reviews and makes recommendations for the procurement of software programs to ensure virtual accessibility;
- Performs monthly audits of virtual compliance;
- Attends and conducts accessibility and compliance training workshops;

- Prepares and disseminates virtual compliance reports, materials, forms, correspondence, and other written information; and
- Interprets and applies laws, regulations and procedures affecting virtual compliance.

4.3.4 Responsibilities of Designated Content Developers and Approvers, Webmaster, and Procurement Officials

The CAHELP JPA Virtual Compliance Supervisor in collaboration with the ACT shall provide and/or procure appropriate training necessary to ensure that individuals as defined below are knowledgeable and appropriately trained to create and/or develop accessible online content, maintain functionality, and procure appropriate IT software, hardware, and media.

- *CAHELP JPA Virtual Compliance Technician:* Primary support technician to the Virtual Compliance team responsible for assisting in resolving compliance issues, auditing of website and functionality, and providing accessibility support services and training to CAHELP staff;
- *Content Developers:* Individuals responsible for uploading, modifying, maintaining, and updating content on web pages;
- *Content Approvers:* Individuals responsible for review of online content and ensuring content meets principles of accessibility and WCAG guidelines;
- *Procurement Officials:* Individuals responsible for the research and procurement of IT equipment; and
- *Webmaster:* Individual(s) responsible for the overall accountability and compliance of online content and functionality.

An accessibility checklist (Appendix B) based on WCAG 2.0 Level AA is available to assist content developers and approvers, web designers, and purchasing agents in creating and procuring accessible IT. This checklist can also be used by procurement officials as a reference for vendors and contractors providing products and services to CAHELP. Many of the items in the checklist apply to web pages and web-based applications as well as electronic documents in Microsoft Word, Adobe PDF, and other formats, and other products and services that are not specifically web-based.

Refer to Appendix B for a checklist for implementing HTML-related principles and techniques for seeking WCAG 2.0 conformance produced by Web Accessibility in Mind (WebAIM).

4.3.5 Workflow for Creating/Publishing Online Content

To ensure efficiency, accountability, and implementation, designated content developers and approvers shall upload content to the CAHELP website and/or web pages in the following manner:

4.3.5.1 *Content Developers shall:*

- Receive and review proposed online content;
- Log in to CAHELP website;
- Enable "design mode" feature to edit or add content to a page;
- Create and/or develop content per accessibility checklist (i.e., headings, subheadings, text, images, video, etc.);
- Save content (Note: Web system will automatically forward an e-mail notification to the content auditor to review saved content);
- Review returned content and complete revisions as needed; and
- Publish and maintain approved online content.

4.3.5.2 *Content Approvers shall:*

- Log in to CAHELP website;
- Receive and review all e-mail notifications of pending online content for review;
- Review proposed online content;
- Approve or reject propose online content based on accessibility checklist and accessibility standards; and
- Return content to content developer for modifications.

Content developers and approvers are responsible for ensuring accurate and up-to-date information are published on the website.

Questions regarding content development and management, and accessibility requirements shall be submitted to accessibility@cahelp.org. Staff may also complete and submit a helpdesk ticket to the IT support desk. Requests for assistance shall be completed without unreasonable delay.

5.0 ACCESSIBILITY STANDARDS

The following is a set of accessibility standards provided by the W3C Web Accessibility Initiative (WAI) that are commonly recognized by governments and organizations:

- *Web Content Accessibility Guidelines (WCAG) 2.0* (applicable to all web content and applications, including on mobile, television, and other delivery channels);
- *Authoring Tool Accessibility Guidelines (ATAG) 2.0* (applicable for websites that provide users the opportunity to generate content, such as adding comments, posting to forums, or uploading image or videos; also relevant if an organization provides tools, such as content management systems (CMS), for staff or customers to manage websites and content); and
- *User Agent Accessibility Guidelines (UAAG) 2.0* (applicable when additional plug-ins, such as media players, are provided to deliver content or when custom controls are developed to provide nonstandard functionality. UAAG may also be relevant where mobile applications deliver web content as part of the application, and to the procurement process if your organization provides browsers for staff).

Given the CAHELP's commitment to providing accessible opportunities and environments, it looks to the W3C WCAG 2.0 Level AA and Web Accessibility Initiative Accessible Rich Internet Applications (WAI-ARIA) 1.0 as a target for meeting these commitments. The WCAG 2.0 includes success criterion (WCAG guidelines) organized under four general principles, which provide the foundation of web accessibility. The latest version, WCAG 2.1, extends WCAG 2.0 by adding 17 new success criteria. The W3C recommends that sites adopt WCAG 2.1 as the new conformance target to provide improved accessibility and to anticipate future policy changes. The following four principles have been adopted by CAHELP.

5.1 Principles of Accessibility (P.O.U.R.)

- **Perceivable:** Information and user interface components must be presented to users in ways they can perceive;
- **Operable:** User interface components and navigation must be operable;
- **Understandable:** Information and the operation of user interface must be understandable; and
- **Robust:** Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

CAHELP online content shall be Perceivable, Operable, Understandable, and Robust. Content developers and approvers, webmasters, procurement officials, and all others responsible for developing, loading, maintaining, or auditing web content and functionality shall implement the accessibility standards to ensure compliance with the CAHELP's underlying legal obligation to ensure individuals with disabilities are not excluded from participation in, denied the benefits of, or otherwise subjected to discrimination in any of the CAHELP's programs, services, and activities delivered online.

5.2 WCAG Guidelines

Under the four principles of accessibility there are 12 WCAG guidelines that provide the framework and overall objectives to help content developers and approvers, webmasters, procurement officials, and all others responsible for

developing, loading, maintaining, or auditing web content and functionality, understand the success criteria and better implement the techniques to meet accessibility standards. In its adoption of the four principles of accessibility, the CAHELP ensures that online content and functionality shall be developed in accordance to the 12 WCAG guidelines in each principle of accessibility.

5.2.1 Perceivable

- *Guideline 1.1. Text Alternatives:* 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;
- *Guideline 1.2. Time-based Media:* Provide alternatives for time-based media;
- *Guideline 1.3 – Adaptable:* Create content that can be presented in different ways (i.e., simpler layout) without losing information or structure; and
- *Guideline 1.4 – Distinguishable:* Make it easier for users to see and hear content including separating foreground from background.

5.2.2 Operable

- *Guideline 2.1 – Keyboard Accessible:* Make all functionality available from a keyboard;
- *Guideline 2.2 – Enough Time:* Provide users with enough time to read and use content;
- *Guideline 2.3 – Seizures and Physical Reactions:* Do not design content in a way that is known to cause seizures or physical reactions;
- *Guideline 2.4 – Navigable:* Provide ways to help users navigate, find content, and determine where they are; and
- *Input Modalities:* Make it easier for users to operate functionality through various inputs beyond keyboard.

5.2.3 Understandable

- *Guideline 3.1 – Readable:* Make text content readable and understandable;
- *Guideline 3.2 – Predictable:* Make web pages appear and operate in predictable ways; and
- *Guideline 3.3 – Input Assistance:* Help users avoid and correct mistakes.

5.2.4 Robust

- *Guideline 4.1 – Compatible:* Maximize compatibility with current and future user agents, including assistive technologies.

5.3 Levels of Conformance (Priority Levels)

W3C WAI guidelines provide three levels of conformance: Levels A, AA, and AAA:

1. *Level A:* Establishes a baseline level of conformance, and covers a basic set of core accessibility issues (such as alternate text on images and captions and videos);
2. *Level AA:* Includes additional success criteria such as providing a visible focus indicator for keyboard users, and ensuring sufficient color contrast; or
3. *Level AAA:* The highest level of conformance. Conforming to WCAG 2.0 at Level AAA would mean all success criteria have been met.

Level AA is the designated benchmark for measuring accessibility of CAHELP online content and functionality. Conformance to Level AA requires that CAHELP meet all Levels A and AA success criterion. Levels of conformance are based on impact on individuals with disabilities, feasibility, and other factors. Each of the success criteria under each principle of accessibility is identified with a conformance level. CAHELP shall ensure that all public-facing official agency business content, as well as specific categories of non-public-facing content that is official agency business, conform to all Level AA success criterion.

Example of conformance Level AA required:

Principle: UNDERSTANDABLE

Guideline 3.2 - Predictable:
Make web pages appear and operate in predictable ways.

Success Criteria	Recommendation
3.2.3 Consistent Navigation *Level AA*	*Navigation links that are repeated on web pages do not change order when navigating through the site.*

With the new added success criteria under WCAG 2.1, the following lists requirements for conformance to WCAG 2.1, as well as information about how to make conformance claims, which are optional. This information also describes what it means to be accessibility supported, since only accessibility-supported ways of using technologies can be relied upon for conformance.

5.3.1 Interpreting Normative Requirements: The main content of WCAG 2.1 is normative and defines requirements that impact conformance claims. Introductory material, appendices, sections marked as "non-normative" diagrams, examples, and notes are informative (non-normative). Non-normative material provides advisory information to help interpret the guidelines but does not create requirements that impact a conformance claim.

5.3.2 Conformance Requirements: In order for a web page to conform to WCAG 2.1, all of the following conformance requirements must be satisfied:

5.3.2.1 *One of the following levels of conformance is met in full:*

- For Level A conformance (the minimum level of conformance), the web page satisfies all the Level A success criteria, or a conforming alternate version is provided.
- For Level AA conformance, the web page satisfies all the Level A and Level AA success criteria, or a Level AA conforming alternate version is provided.
- For Level AAA conformance, the web page satisfies all the Level AA and Level AAA success criteria, or a Level AAA conforming alternate version is provided.

5.3.3 Conformance (and conformance levels) is for full web page(s) only and cannot be achieved if part of a web page is excluded.

NOTE: For the purpose of determining conformance, alternatives to part of a page's content are considered part of the page when the alternatives can be obtained directly from the page, e.g., a long description or an alternative presentation of a video. Authors of web pages that cannot conform due to content outside of the author's control may consider a Statement of Partial Conformance. A full page includes each variation of the page that is automatically presented by the page for various screen sizes (e.g., variations in a responsive web page). Each of these variations needs to conform (or needs to have a conforming alternate version) in order for the entire page to conform.

5.3.4 Complete Processes

When a web page is one of a series of web pages presenting a process (i.e., a sequence of steps that need to be completed in order to accomplish an activity), all web pages in the process conforms at the specified level or better. (Conformance is not possible at a particular level if any page in the process does not conform at that level or better).

5.3.5 Only Accessibility-Supported Ways of Using Technologies

Only accessibility-supported ways of using technologies are relied upon to satisfy the success criteria. Any information or functionality that is provided in a way that is not accessibility supported is also available in a way that is accessibility supported.

5.3.6 Non-Interference

If technologies are used in a way that is not accessibility supported, or if they are used in a non-conforming way, then they do not block the ability of users to access the rest of the page. In addition, the web page as a whole continues to meet the conformance requirements under each of the following conditions:

1. When any technology that is not relied upon is turned on in a user agent;
2. When any technology that is not relied upon is turned off in a user agent; and
3. When any technology that is not relied upon is not supported by a user agent.

In addition, the following success criteria apply to all content on the page, including content that is not otherwise relied upon to meet conformance, because failure to meet them could interfere with any use of the page:

- 1.4.2 – Audio Control;
- 2.1.2 – No Keyboard Trap;
- 2.3.1 – Three Flashes or Below Threshold; and
- 2.2.2 – Pause, Stop, Hide.

5.3.7 Conformance Claims (Optional)

Conformance is defined only for web pages. However, a conformance claim may be made to cover one page, a series of pages, or multiple related web pages.

5.3.7.1 *Required Components of a Conformance Claim*

Conformance claims are not required. Authors can conform to WCAG 2.1 without making a claim. However, if a conformance claim is made, then the conformance claim must include the following information:

1. Date of claim;

2. Guidelines title, version and URI "Web Content Accessibility Guidelines 2.1";

3. Conformance level satisfied (Level A, AA, AAA);

4. A concise description of the web pages, such as a list of URIs for which the claim is made, including whether subdomains are included in the claim; and

5. A list of web content technologies relied upon.

5.3.7.2 *Optional Components of a Conformance Claim*

In addition to the required components of a conformance claim, consider providing additional information to assist users. Recommended additional information includes:

- A list of success criteria beyond the level of conformance claimed that have been met. This information should be provided in a form that users can use, preferably machine-readable metadata.

- A list of the specific technologies that are "used but not relied upon."

- A list of user agents, including assistive technologies that were used to test the content.

- A list of specific accessibility characteristics of the content, provided in machine-readable metadata.

- Information about any additional steps taken that go beyond the success criteria to enhance accessibility.

- A machine-readable metadata version of the list of specific technologies that are relied upon.

- A machine-readable metadata version of the conformance claim.

Refer to Appendix B for WCAG 2.0 Checklist produced by Web Accessibility in Mind (WebAIM) for list of success criteria at Level A and Level AA.

5.3.8 Statement of Partial Conformance - Third Party Content

Sometimes, web pages are created that will later have additional content added to them. For example, an email program, a blog, an article that allows users to add comments, or applications supporting user-

contributed content. Another example would be a page, such as a portal or news site, composed of content aggregated from multiple contributors, or sites that automatically insert content from other sources over time, such as when advertisements are inserted dynamically.

In these cases, it is not possible to know at the time of original posting what the uncontrolled content of the pages will be. It is important to note that the uncontrolled content can affect the accessibility of controlled content as well. Two options are available:

1. A determination of conformance can be made based on best knowledge. If a page of this type is monitored and repaired (non-conforming content is removed or brought into conformance) within two business days, then a determination or claim of conformance can be made since, except for errors in externally contributed content which are corrected or removed when encountered, the page conforms. No conformance claim can be made if it is not possible to monitor or correct non-conforming content; OR

2. A "statement of partial conformance" may be made that the page does not conform but could conform if certain parts were removed. The form of that statement would be, "This page does not conform, but would conform to WCAG 2.0 at Level X if the following parts from uncontrolled sources were removed." In addition, the following would also be true of uncontrolled content that is described in the statement of partial conformance:

    a. It is not content that is under the author's control.

    b. It is described in a way that users can identify (e.g., they cannot be described as "all parts that we do not control" unless they are clearly marked as such).

A "statement of partial conformance due to language" may be made when the page does not conform but would conform if accessibility support existed for (all of) the language(s) used on the page. The form of that statement would be, "This page does not conform, but would conform to WCAG 2.0 at level X if accessibility support existed for the following language(s)."

5.3.9 Authoring Tool Accessibility Guidelines (ATAG) 2.0

Authoring Tools Accessibility Guidelines (ATAG) 2.0 provides guidelines for designing web content authoring tools that are both more accessible to authors with disabilities, and designed to enable, support, and promote the production of more accessible web content by all authors. Authors are individuals who use authoring tools to create or

modify content. Authors include roles such as content authors, designers, programmers, publishers, testers, etc. ATAG is primarily for developers of authoring tools. An authoring tool is any web-based or non-web-based application(s) that can be used by authors (alone or collaboratively) to create or modify web content for use by other authors or end users.

Examples of software that are generally considered authoring tools under ATAG 2.0:

- What-you-see-is-what-you-get (WYSIWYG) HTML editor;
- Software for directly editing source code; software for converting to web technologies (e.g., "Save as HTML" features in office document applications);
- Integrated development environments (e.g., for web application development);
- Software that generates web content on the basis of templates, scripts, command-line input or "wizard" type processes;
- Software for rapidly updating portions of web pages (e.g., blogging, wikis, online forums);
- Software for generating/managing entire websites (e.g., content management systems, courseware tools, content aggregators);
- Email clients that send messages using web content technologies;
- Multimedia authoring tools; and
- Software for creating mobile web applications.

CAHELP shall consider authoring tools that web developers, designers, writers use to produce CAHELP web content (i.e., static web pages, dynamic web applications, etc.) based on their accessibility conformance claims and ATAG 2.0 accessibility standards.

Refer to the following for additional information:

- ATAG http://www.w3.org/TR/ATAG/
- WCAG http://www.w3.org/TR/WCAG/
- WAI-ARIA http://www.w3.org/TR/wai-aria/

5.3.10 User Agent Accessibility Guidelines (UAAG)

User Agent Accessibility Guidelines (UAAG) 2.0 is part of a series of accessibility guidelines. The core target audience of UAAG are the developers of the authoring tools, but policy makers and procurement decision makers within CAHELP can equally use UAAG criteria to determine whether the user agent technologies are accessible, or UAAG can be given to other developers to use to enhance the accessibility features of the tools. User agents are defined as any software that

retrieves, renders and facilitates end user interaction with web content. UAAG 2.0 identifies the following user agent architectures:

- *Platform-based user agent, native user agent.* User agents that run on non-web platforms (operating systems and cross-OS platforms, such as Java) and perform content retrieval, rendering and end-user interaction facilitation themselves (e.g., Firefox, Internet Explorer, Chrome, Opera, Windows Media Player, QuickTime Pro, RealPlayer);
- *Embedded user agent, plug-in.* User agents that "plug-in" to other agents or applications (e.g., media player plug-in for a web browser, web view component). Embedded user agents can establish direct connections with the platform (e.g., communication via platform accessibility services);
- *Web-based user agent.* User agents that have user interfaces that are implemented using web content technologies and are accessed by users via a user agent. Web-based user agents transform content into web content technologies that the host user agent can render (e.g., web-based e-Pub reader, web-based video player).

UAAG provides guidance in designing user agents that make the web more accessible to individuals with disabilities. The goal of UAAG 2.0 is to ensure that all users, including users with disabilities, have equal control over the environment they use to access the web. A user agent that follows UAAG 2.0 will improve accessibility through its own user interface and its ability to communicate with other technologies, including assistive technologies (software that some individuals with disabilities use to meet their requirements). All users, not just users with disabilities, will benefit from user agents that follow UAAG 2.0.

Like WCAG, UAAG offers three layers of guidance: (1) principles, (2) guidelines; and (3) testable success criteria. Five principles provide a foundation for accessible user agents. Three of the five principles are parallel to WCAG 2.0, and two are specific to user agents. For each principle, there is a set of guidelines for making user agents more accessible to users with disabilities. These guidelines provide the framework to help individuals who use authoring tools to create or modify content, content authors, designers, programmers, publishers, testers, etc., understand the objectives for success criteria so they can better implement them. Under each guideline is also a set of testable success criteria that can be used wherever conformance testing is necessary, including design application, purchasing, regulation, and contractual agreements. Each success criterion is assigned a level of conformance, which are designed to meet the needs of different groups and different situations. The recommended conformance for UAAG is

AA. Much of the value of the UAAG stems from the harmonious integration of the WCAG 2.0 and the ATAG 2.0.

CAHELP will recommend that developers of authoring tools, policy makers, and procurement officials ensure that user agents utilized to support CAHELP web content and web applications meet the W3C recommended UAAG 2.0 version Level AA conformance.

Refer to the following for additional information:

- UAAG http://www.w3.org/TR/UAAG/
- WCAG http://www.w3.org/TR/WCAG/
- WAI-ARIA http://www.w3.org/TR/wai-aria/

5.3.11 Accessibility Evaluation Tools (Testing Sites and Applications)

Evaluating the extent to which the CAHELP conforms to WCAG 2.0 Level AA is a process involving several steps. The activities carried out within these steps are influenced by many aspects such as the type of website (e.g., static, dynamic, responsive, mobile, etc.); its size; complexity; technologies used to create the website (e.g., HTML, WAI-ARIA, PDF, etc.); how much knowledge the auditors have about the process used to design and develop the website; and the main purpose for the audit (e.g., to issue an accessibility statement, to plan a redesign process, to perform research, etc.).

To ensure CAHELP meets established benchmarks for accessibility, it shall implement an audit of online content and functionality as specified herein to ensure compliance with W3C WCAG 2.0 Level AA and WAI-ARIA 1.0. Auditors shall utilize the Techniques for WCAG 2.0 documented by W3C/WAI (url: https://www.w3.org/TR/WCAG20-TECHS/), and may also refer to the W3C Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0 to assist in providing a comprehensive evaluation of online content and functionality. The WCAG-EM highlights considerations for auditors to apply during the evaluation process but does not replace the need for quality assurance measures that are implemented throughout the design, development, and maintenance of the website and web applications to ensure their accessibility conformance. WCAG-EM does not in any way add to or change the requirements defined by the normative WCAG 2.0 standards and can be used in conjunction with techniques for meeting WCAG 2.0 success criteria. Go to WCAG-EM 1.0 to access this information (url: https://www.w3.org/TR/WCAG-EM/).

Outside of the WCAG-EM, there are also a number of website evaluation tools available online to assist content developers and approvers, webmasters, procurement officials, and all others responsible

for developing, loading, maintaining, or auditing web content and functionality, in determining whether or not the website meets accessibility standards. However, because these tools are limited in being able to uncover the majority of accessibility issues, the CAHELP shall procure the services of an external auditor in addition to conducting accessibility testing online, and internal auditing.

The CAHELP shall employ the following accessibility evaluation methods to audit all online content and functionality.

1. *Accessibility Audit:* An external accessibility auditor shall review the website, highlighting any accessibility issue(s) and provide recommendations to the CAHELP JPA Virtual Compliance Supervisor. The auditor shall utilize assistive software used by web users with disabilities (e.g., screen reader) to effectively carry out the audit, along with the free Web Accessibility Toolbar (WAT) developed by The Paciello Group. WAT aids manual examination of web pages for a variety of aspects of accessibility. Go to WAT to download a copy (url: https://developer.paciellogroup.com/resources/wat/).

   The auditor can be a hired external accessibility consultancy, or an in-house member who is knowledgeable of the W3C accessibility guidelines who is appropriately trained in web accessibility.

2. *Accessibility Testing:* The CAHELP JPA Virtual Compliance Technician, as designated by the CAHELP JPA Virtual Compliance Supervisor, shall coordinate testing with real users with disabilities to complete common tasks on the website while a designated moderator notes all problems the user experiences. Regular usability testing will uncover more usability issues as users with disabilities may require additional time to complete tasks.

3. *Automated Accessibility Testing:* Both internal and external auditor may utilize automated programs to evaluate the website against accessibility guidelines.

   For a list of online accessibility testing resources, see Appendix C (e.g., Useablenet, Web Accessibility Versatile Evaluator (WAVE), AChecker, SiteImprove, etc.).

The external auditor shall carry out the accessibility audit. After the findings from an accessibility audit has been implemented, the CAHELP shall initiate accessibility testing, as needed. The CAHELP JPA Virtual Compliance Supervisor shall further coordinate testing sessions with the assistance of county-operated programs and/or

inviting a group of users living with visual, auditory, physical, and/or cognitive disabilities, to participate.

5.3.12 Qualifications of Accessibility Auditor

The external auditor shall have the requisite experience and knowledge to carry out an appropriate audit and to develop a proposed Corrective Action Plan. The external auditor shall meet the approved qualifications of an auditor as specified by the Office of Civil Rights (OCR) and shall:

- Audit all content and functionality of the CAHELP website to identify any online content or functionality that is inaccessible to individuals with disabilities, including online content and functionality developed by, maintained by, or offered through a third-party vendor or an open source;
- Use W3C WCAG 2.0 Level AA and WAI-ARIA 1.0 as the benchmarks for measuring accessibility, unless the CAHELP receives prior permission to use a different standard as a benchmark; and
- Develop a proposed Corrective Action Plan.

During the accessibility audit, the CAHELP may also seek input from members of the public with disabilities, including parents, students, employees, and others associated with the CAHELP, and other persons knowledgeable about website accessibility, regarding the accessibility of its online content and functionality.

The Virtual Compliance Team shall have overall responsibility for establishing systems of audit, accountability, corrective action of accessibility of all online content, and functionality on an ongoing basis (Section 4.0 Oversight and Responsibility).

Refer to Appendix C for list of Accessible Testing resources (e.g., Useablenet, Web Accessibility Versatile Evaluator (WAVE), AChecker, SiteImprove, etc.)

6.0 PROCEDURES

See Appendix A: Getting Started with Accessibility.

7.0 IT ACCESSIBILITY CHECKLIST

The following is a checklist for content developers and approvers, web designers and developers, and purchasing agents to consider when developing and/or procuring accessible information technology that the CAHELP purchases, creates, and uses, such as websites, software, hardware, and media. Many of the items in this checklist apply to web pages and web-based applications as well as electronic documents in Microsoft Word,

Adobe PDF, and other formats, and other products and services that are not specifically web-based.

7.1 Make content and controls Perceivable by all users

- Do images have alternative text?
- Does video have captions and does audio have a transcript?
- Does the web page or document include headings, lists, ARIA landmarks, and other semantic elements to communicate document structure?
- Is the tab order and read order logical and intuitive?
- Do form fields within web pages and documents have appropriately coded labels and prompts?
- Have you avoided using visual characteristics to communicate information (e.g., “click the circle on the right” or “required fields are in red”)?
- Does the interface have sufficient contrast between text color and background color?
- Does the content scale well when text is enlarged up to 200 percent?

7.2 Make content and controls Operable by all users

- Can all menus, links, buttons, and other controls be operated by keyboard, to make them accessible to users who are unable to use a mouse?
- Does the web page include a visible focus indicator so all users, especially those using a keyboard, can easily track their current position?
- Do features that scroll or update automatically (e.g., slideshows, carousels) have prominent accessible controls that enable users to pause or advance these features on their own?
- Do pages that have time limits include mechanisms for adjusting those limits for users who need more time?
- Have you avoided using content that flashes or flickers?
- Does the web page or document have a title that describes its topic or purpose?
- Are mechanisms in place that allow users to bypass blocks of content (e.g., “skip to main content” link on a web page or bookmarks in a PDF)?
- Does the website include two or more ways of finding content, such as a navigation menu, search feature, or site map?
- Is link text meaningful, independent of context?

7.3 Make content and user interfaces Understandable to all users

- Has the language of the web page or document (or individual parts of a multilingual document) been defined?
- Have you avoided links, controls, or form fields that automatically trigger a change in context?
- Does the website include consistent navigation?
- Do online forms provide helpful, accessible error and verification messages?

7.4 Make content Robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies

- Is the web page coded using valid HTML?
- Do rich, dynamic, web interfaces, such as modal windows, drop-down menus, slideshows, and carousels, include ARIA markup?

8.0 TRAINING

CAHELP shall provide and/or procure website accessibility training for all appropriate personnel, including, but not limited to content developers and approvers, webmasters, procurement officials, and all others responsible for developing, loading, maintaining, or auditing web content and functionality. Training shall continue on a schedule designed to maintain website accessibility consistent with, or superior to, that which is required under federal law.

9.0 RELATED INFORMATION

9.1 Resources and Support for IT Accessibility

- Accessible Technology at the CAHELP
- IT Accessibility Checklist
- Access Technology Center
- World Wide Web Consortium (W3C) Web Content Accessibility Guidelines 2.0

9.2 Legal and Policy Requirements

- Section 504 of the Rehabilitation Act of 1973 (http://www2.ed.gov/about/offices/list/ocr/504faq.html)
- Americans with Disabilities Act as amended (https://www.ada.gov/2010_regs.htm)
- California Unruh Civil Rights Act (UCRA), Cal Civ. Code *§ 51.*
- Department of Justice (DOJ) Guidance (June 2003)
  - ADA/504 “generally require” equal access unless fundamental alteration or undue burden
- OCR Dear Colleague Letter (June 2010)
  - Colleges and universities must make book readers and other educational technologies equally accessible
- OCR FAQs (May 11)
  - Follow-up from June 2010 Dear Colleague letter – legal requirements articulated in letter apply to elementary and secondary schools
- DOJ Notice of Proposed Rulemaking (May 2016)
  - Proposed rulemaking for state and local governments with regard to web accessibility

10.0 REVISION HISTORY

Version Number	Revised	Governance Approval
1.0	10/27/16	4/7/2017
2.0	08/24/18	9/7/2018
3.0	01/16/20	02/07/20

APPENDIX A

A. GETTING STARTED WITH ACCESSIBILITY

To ensure accessibility standards are met, *content developers and approvers* must have an understanding of web accessibility, online content, and functionality, and an understanding of the terminology provided in Section 2.0 of this document. In designing web accessibility, *content developers and approvers* should consider these user characteristics in designing web accessibility:

A.1. Characteristics for Consideration

(1) Unable to see. Individuals who are blind use either audible output (products called screen readers that read web content using synthesized speech) or tactile output (a refreshable Braille device).

(2) Has dyslexia. Individuals with learning disabilities such as dyslexia may also use audible output, along with software that highlights words or phrases as they are read aloud using synthesized speech.

(3) Has low vision. Individuals with low vision may use screen magnification software that allows them to zoom in all or a portion of the visual screen. Many others with less-than-perfect eyesight may enlarge the font on websites using standard browser functions, such as Ctrl + in Windows browsers or Command + in Mac browsers.

(4) Has a physical disability. Individuals with physical disabilities that effect their use of hands may be unable to use a mouse, and instead may rely exclusively on keyboard or use assistive technologies such as speech recognition, head pointers, mouth sticks, or eye-gaze tracking systems.

(5) Unable to hear. Individuals who are deaf or hard of hearing are unable to access audio content, so video needs to be captioned and audio needs to be transcribed.

(6) Using a mobile device. Individuals who are accessing the web using a compact mobile device such as a phone, face accessibility barriers, just like individuals with disabilities do. They're using a small screen and may need to zoom in or increase the font size, and they are likely to be using a touch interface rather than a mouse. Also, Apple's iPhone and iPad do not support Adobe Flash.

(7) Limited bandwidth. Individuals may be on slow internet connections if they are located in a rural area or lack the financial resources to access high-speed internet. These users benefit from pages that load quickly (use graphics sparingly) and transcripts for video.

(8) Limited time. Very busy individuals may have too little time to watch an entire video or audio recording but can quickly access its content if a transcript is available.

Accessible technology works for all of these users, and countless others not mentioned.

A.2. Essential Components of Web Accessibility

Web accessibility depends on several different components of web development and interactions working together and how improvements in specific components could substantially improve web accessibility. These components include:

- Content (information in a web page or web application, including (1) natural information such as text, images, and sounds, or (2) code or markup that defines structure, presentation etc.);
- Web browsers, media players, and other user agents;
- Assistive technology, in some cases, screen readers, alternative keyboards, switches, scanning software, etc.;
- User's knowledge, experiences, and in some cases, adaptive strategies using the web;
- Developers, designers, coders, authors, etc., including developers with disabilities and users who contribute content;
- Authoring tools – software that creates web sites; and
- Evaluation tools – web accessibility evaluation tools, HTML validators, Cascading Style Sheets (CSS) validators, etc.

Authoring tools and evaluation tools are used by web developers to create web content. Individuals ("users") use web browsers, media players, assistive technologies, or other means to get and interact with content. It's important to note that there are significant interdependencies between the components. Components must work together in order for the web to be accessible. When accessibility features are effectively implemented in one component, the other components are more likely to implement them.

A.2.1. Examples

- When web browsers, media players, assistive technologies, and other user agents support an accessibility feature, users are more likely to demand it and developers are more likely to implement it in their content;

- When developers implement an accessibility feature in their content, they are more likely to demand that their authoring tool make it easy to implement;

- When authoring tools make a feature easy to implement, developers are more likely to implement it in their content; or

- When an accessibility feature is implemented in most content, developers and users are more likely to demand that user agents support it.

If an accessibility feature is not implemented in one component, there is little motivation for the other components to implement it when it does not result in an accessible user experience. If one component has poor accessibility support, sometimes other components can compensate through "work-arounds" that require much more effort and are not good for accessibility overall.

A.3. Guidelines for Different Components:

The different components were briefly covered in Section 5.0 – Accessibility Standards: WCAG, ATAG, and UAAG. *Content Developers and Approvers*, web developers, and other individuals involved in the creation and maintenance of online content and functionality may refer to the following W3C WAI accessibility guidelines for additional information on the different components:

- Authoring Tool Accessibility Guidelines (ATAG) addresses authoring tools (url: https://www.w3.org/WAI/intro/atag.php)

- Web Content Accessibility Guidelines (WCAG) addresses web content, and is used by developers, authoring tools, and accessibility evaluation tools (url: https://www.w3.org/WAI/intro/wcag.php)

- User Agent Accessibility Guidelines (UAAG) addresses web browsers and media players, including some aspects of assistive technologies (url: https://www.w3.org/WAI/intro/uaag.php)

B. HOW TO MAKE TECHNOLOGY ACCESSIBLE

The following information will provide *content developers and approvers and webmasters* how-to-pages with step-by-step guides for making particular types of content accessible. For additional information about accessibility of particular technologies, please refer to the pages that are most relevant for the technologies to be used. *Webmasters* and *content developers and approvers* shall be familiar with:

APPENDIX A

(1) Creating Accessible Documents

(2) Developing Accessible Websites

(3) Creating Accessible Videos

(4) Procuring Accessible IT

(5) Managing Projects for Accessibility

Content developers and approvers, and webmasters shall consider accessibility throughout the design and creation process of online content. The following are tips for creating accessible content and conducting simple accessibility tests:

- **Useable without a mouse:** Ensure all links, buttons, menus, and controls in web pages and applications can be used without a mouse, but instead can be navigated using only the keyboard. Whether an interface is functional using a keyboard alone is often a reliable indicator of overall accessibility;

- **Document structure:** Create web pages, Word documents, and PDF files that have good structure, including the use of headings, sub-headings, and lists that make these documents easier for users to understand and navigate;

- **Accessible images:** Include alternative text for graphics and avoid images of text. Individuals who cannot see an image rely on alternate text to access its content; and

- **Test with accessibility checker tools:** As stated in subsection 5.3.11, CAHELP will employ accessibility testing using online accessibility checkers. *Webmasters* may use accessibility checkers and/or web browser plug-ins to identify common accessibility problems and report them to the CAHELP JPA Virtual Compliance Supervisor and/or the Accessibility Compliance Team (ACT). A list of online accessibility checkers is available in *Appendix B* to assist with accessibility efforts.

Accessibility issues shall be reported to the CAHELP JPA Virtual Compliance Supervisor and/or the ACT for accountability. Issues that exceed the parameters and scope of responsibility of the CAHELP JPA Virtual Compliance Supervisor and ACT shall be referred to an accessibility expert for review and recommendation for corrective action.

B.1. Creating Accessible Documents

The core steps needed for accessibility are the same regardless of whether the document is developed in HTML (web), Microsoft Word, Adobe PDF, or another document format. The following are the required basic steps to assist *content developers and approvers* in creating accessible documents:

- Use headings;

- Use lists;
- Add alternate text to images;
- Use tables wisely; and
- Understand how to export from one format to another.

B.1.1. Headings

Identify headings and subheadings using the built-in heading features of the authoring tool. Headings (e.g., h1, h2, h3, etc.) form an outline of the page content and enable screen reader users to understand how the page is organized, and to quickly navigate to content of interest. Screen readers have features that enable users to jump quickly between headings with a single key stroke.

B.1.2. Use Lists

Use the list controls provided in the document authoring software. Content that is organized as a list should be created using the list controls. Authoring software provides one or more controls for adding unordered lists (with bullets) and ordered lists (with numbers). When lists are explicitly created as lists, this helps screen readers to understand how the content is organized. When screen reader users enter a list, their screen reader informs them that they are on a list and may also inform them of how many items are in the list, which can be very helpful information when deciding whether to continue reading.

B.1.3. Add Alternate Text for Images

Users who are unable to see images depend on content developers to supplement their images with alternate text, which is often abbreviated "alt text." The purpose of alt text is to communicate the content of an image to individuals who cannot see the image. The alt text should be succinct, just enough text to communicate the idea without burdening the user with unnecessary detail. When screen readers encounter an image with alt text, they typically announce the image then read the alt text.

Authoring tools provide a means of adding alt text to images, usually in dialog that appears when an image is added, or later within an image properties dialog.

If images are purely decorative and contain no informative content, they do not require a description. However, they may still require specific markup, so screen readers know to skip them. Also, images that require a lengthier

description, such as charts and graphs, may require additional steps beyond adding alt text.

B.1.4. Use Tables Wisely

Tables should not be used to control content layout. Tables in documents are useful for communicating relationships between data, especially where those relationships can be best expressed in a matrix of rows and columns. Authoring tools have other means of doing this, including organizing content into columns.

If the data is best presented in a table, try to keep the table simple. If the table is complex, consider whether it could be divided into multiple simpler tables with a heading above each.

A key to making data tables accessible to screen reader users is to clearly identify column and row headers. Also, if there are nested in columns and rows with multiple headers for each cell, screen readers need to be explicitly informed as to which headers relate to which cells.

B.1.5. When Exporting to PDF, Understand How to Preserve Accessibility

In order for an Adobe PDF document to be accessible, it must be a "tagged" PDF, with an underlying tagged structure that includes all of the features already described herein. There are right ways and wrong ways to export documents to PDF. Some authoring tools do not support tagged PDF at all, while others provide multiple ways of exporting to PDF, some that produce tagged PDF and some that do not. The CAHELP utilizes Adobe Acrobat DC which provides accessible tags.

B.1.6. Creating High Quality Scanned Documents

When documents are in electronic form, they are easier to distribute and can be more accessible than print documents. However, in order to be fully accessible, certain steps must be followed to be sure a scanned document is of high quality. Even if a document is not needed for an individual with a disability, a poor scan often negatively impacts the end user's experience.

B.1.7. Developing an Accessible Website

In order to assure that the CAHELP website and web applications are accessible to and usable by everyone, web designers and developers must follow accessibility guidelines. The following topics address issues that are especially common on the website:

Features of an Accessible Website:

- Good structure in web pages and documents
- Good use of HTML headings;
- Accessible with keyboard;
- Accessible images;
- Accessible menus;
- Accessible forms;
- Accessible tables;
- Effective use of color;
- Meaningful link text;
- ARIA landmark roles;
- ARIA for web applications; and
- Avoiding reliance on visual characteristics.

B.1.8. Structure in Web Pages and Documents

In order to understand a document, everyone depends on understanding its structure. Screen reader users need to understand this structure and are dependent on *content developers* clearly identifying the headings, paragraphs, lists, tables, banners, menus, and other features as exactly what they are. In the world of web design this is called semantics, building a page using web elements that define the role of the object. For example, when adding a top-level heading to a web page, *content developers* shall use the built-in h1 feature that the authoring software provides. Simply making the text big and bold may look like a heading but it really is not a heading.

B.1.9. HTML headings

As discussed in Section 5.0, the core steps needed for accessibility are the same whether the document is developed in HTML (web), Microsoft Word, Adobe PDF, or another document format. The use of HTML headings is essential in developing an accessible website.

HTML headings service two purposes for non-sighted users:

- They provide an outline of the page, so users can understand how the page is structured, and how all the sections relate to one another; and

- They provide a target so users can jump from heading to heading with a single keystroke, e.g., the letter "H" in some screen readers.

Content developers shall utilize built-in heading feature in authoring tools.

B.1.10. Accessible with Keyboard

Because many users are physically unable to use a mouse and might be navigating through a web page using a keyboard alone, conducting a simple accessibility test using the keyboard will help determine whether users can (1) access all features, (2) operate all controls, and (3) easily tell where they are on the web page. *Content developers* test this feature by using the tab key to navigate between features, and other keys of doing so would seem to make sense (e.g., enter or space to "click" the element that currently has focus), arrow keys to move within a widget such as a menu or slider, and escape to close a pop-up window.

B.1.11. Testing HTML Web Pages

Content developers should navigate through the web page using a keyboard alone. Using the tab key, *content developers* should be able to access all links and controls in a predictable order based on their visual position on the page. The success of this test can also be affected by whether there is sufficient visual indication of focus.

- WCAG 2.0 Success Criterion 1.3.2 Meaningful Sequence (Level A)

- WCAG 2.0 Success Criterion 2.4.3 Focus Order (Level A)

If users are unable to tell where they are on a web page when navigating with keyboard, *content developers and approvers*, and webmasters can typically fix this with some very simple cascading style sheets (CSS). *Content developers and approvers* should consult the webmaster and/or developer of authoring tools.

Movement through a web page or application should follow a logical order. It should mirror the visual order of navigation and controls on the page. Users who are navigating by keyboard (e.g., using the tab key) expect to move sequentially from left to right and top to bottom through the focusable elements on the page.

When creating web pages, be sure the order of items in the source code matches the visual order.

B.1.12. Accessible Images

If web pages include images, the content of those images is, by default, inaccessible to individuals who are unable to see the images. Whether and how to address this issue depends on the purpose of the image within the context of the web page.

- **Simple Informative Images.** If images are designed to communicate information to the user, they must be described. Images that convey simple information must be described with alternative text, or "alt text." Alt text is a short description of the content of the image, added in such a way that is typically invisible to individuals who can see the image but is exposed to individuals who are using assistive technologies such as screen readers or Braille displays. Browsers also display alt text visibly if an image fails to load. Such simple images include logos, buttons, and photographs. The description should describe the content and functionality of the image as concisely as possible to provide access to the content of the image without burdening the user with superfluous details.

- **Adding Alt Text in Word Processing Programs or Rich Text Editors.** Word processing applications such as Microsoft Word and Google Docs; as well as online rich text editors such as those used for adding content to Canvas, WordPress, or Drupal; all include support for alt text on images. When adding an image to a web page or document, simply look for a tab or field labeled "alt text" or equivalent and enter a short description into the field. If you are not prompted for alt text when adding the image, right click on the image after it has been added and select "Image Properties" or equivalent, then look around in the image properties dialog for an "Alt text" prompt.

- **Complex Informative Images.** Complex images, such as graphs, charts, or diagrams, may contain too much information to be effectively described using alt text. Instead, these images must be described with a long description. Long description is a more detailed description that provides equivalent access to the information of the image. The question *content developers* should ask is: Given the current context, what information is this image intended to communicate? That same information must be provided to individuals who are unable to see the image. A long description can include any structure necessary to communicate the content of the image, including heading list and data tables.

- **Adding Long Description in HTML.** In HTML, long description can be added either on a separate web page or on the same page in a with id attribute. The latter can be hidden from sighted users, although *content developers* should consider whether it might be of value to some sighted users too, particularly individuals who have difficulty understanding visually symbolic content such as charts and graphs. Once the long description is in place, add a longdesc attribute to the <img> element, pointing to the URL of the long description.

- For assistance on providing accessible images and what constitutes alt text verses longdesc, consult the webmaster and/or developer of authoring tools.

- **Decorative Images.** If images are used solely for decorative purposes and does not convey meaning, they should be added to the page using CSS, not with the HTML <img> element. If for some reason an image needs to be added using HTML, the <img> element must have an empty alt attribute (alt=""). This is a standard technique for communicating to screen readers that the image should be ignored. The following are a few methods that *content developers* can tell screen readers to ignore the decorative image:

  - Avoid using the HTML <img> element for decorative images; instead present the image as a background image using cascading style sheets (CSS)
  - If using the HTML <img> element, add an empty alt attribute (alt="")
  - If using the HTML <img> element, add the following attribute: role="presentation"

References:

- HTML5: Techniques for providing useful text alternatives
- National Center for Accessible Media (NCAM) guidelines for describing complex images: Effective Practices for Description of Science Content within Digital Talking Books
- National Center on Accessible Media (NCAM): Effective Practices for Describing STEM Images
- WCAG 2.0. Success Criterion 1.1.1 Non-text Content (Level A)

B.1.13. Accessible Menus

Website navigation menus often include dropdown or fly-out menus, where submenus are hidden by default and appear visibly when mouse users hover

over or click a top-level menu item. These types of menus can present major accessibility challenges for many groups of users unless they are coded properly.

For assistance and information on creating accessible menus, consult the webmaster and/or developer of authoring tools. The webmaster and/or developer shall explore this problem in depth and provide recommendations to the CAHELP JPA Virtual Compliance Supervisor and/or the ACT.

B.1.14. Accessible Forms

To create an accessible Online Form, *content developers* shall ensure that all form fields have accurate labels or prompts so screen reader users know what each field is asking for. Forms typically have labels or prompts that are obvious to sighted users, but their association with particular form fields is made based on visual cues, such as relative position and proximity to the field. Since screen reader users do not have access to these same visual cues, labels and prompts must be explicitly associated with form fields within the HTML (web).

The following should be used by *Content Developers* or form developers:

B.1.14.1. Use Label Element

The prompt "Last name" precedes the input field, but its relationship to the field is not explicitly defined. Therefore, some screen readers will simply announce this as an "edit" field but will not prompt the user to enter "Last name" into that field. Other screen readers will guess at the label, and in the example provided below, the user will probably guess accurately. However, as forms grow in complexity, screen readers that guess at labels are more likely to guess incorrectly, which means users are more likely to complete the form incorrectly. *Content developers* or form developers shall properly label form elements.

EXAMPLE OF INCORRECT FIELD:

```
<div>
Last name:
<input type="text" name="last_name" id="last_name">
</div>
```

CORRECT LABEL:

```
<div>
```

```
<label for="last_name">Last name:</label>
<input type="text" name="last_name" id="last_name">
</div>
```

B.1.14.2. Use <fieldset> and <legend> Elements

For groups of related fields such as radio buttons and checkboxes, each form field must have a label as described in the previous section. However, that prompt alone can be meaningless if the user does not know the question. *Content developers* or form developers shall address this problem by grouping these elements together using a <fieldset> element then use a <legend> element to markup the question.

EXAMPLE:

```
<fieldset>
<legend>What is your favorite color?</legend>
<div>
<input type="radio" name="color" value="Red" id="color_red">
<label for="color_red">Red</label>
</div>
<div>
    <input type="radio" name="color" value="green" id="color_green">
</div>
<div>
    <input type="radio" name="color" value="blue" id="color_blue">
    <label for="color_blue">Blue</label>
</dvi>
</fieldset>
```

For additional assistance regarding appropriate use of labels, field sets, and legend elements, consult the webmaster and/or developer of authoring tools.

B.1.14.3. Making PDF Forms Accessible

Interactive forms in Adobe PDF have many of the same issues as those described in developing online forms (HTML). Labels and prompts must all be created in a way that explicitly associates them with their corresponding form fields. It is also important to note that PDF form fields have a tendency to be out of order, so *content developers* or form developers must be sure to test the tab order of the PDF form, to be sure that users will move through the form in a logical sequence when jumping between fields using the keyboard.

Testing PDF Documents. In Adobe Acrobat, go to View > Tools > Accessibility, and select "Touch Up Reading Order." This feature provides a visual indication of the approximate order in which content will appear if automatically re-purposed for display on a small screen.

To test an interactive PDF form, open the form in any desktop PDF reader and move through the form fields by pressing the tab key. Fields will be highlighted as they receive focus. If fields are not arranged in the expected sequence, this can be fixed in Adobe Acrobat. Go to View > Tools > Forms > Edit. All form fields will be listed in tab order in a sidebar panel. Simply drag fields to their correct position in the tab order.

References:

- WCAG 2.0 Success Criterion 1.3.1 Info and Relationships (Level A)
- WCAG 2.0 Success Criterion 1.3.2 Meaningful Sequence (Level A)
- WCAG 2.0 Success Criterion 2.4.3 Focus Order (Level A)

B.1.14.4. Avoiding CAPTCHA (Completely Automated Public Turing Test)

CAPTCHA is a type of form field that is sometimes used to determine whether a user is human, in an effort to prevent computers from automatically submitting online forms. Often CAPTCHAs assume the form of distorted characters.

CAPTCHA is inaccessible to many groups of users, including individuals who are blind or dyslexic. If audio CAPTCHA is

provided as an alternative for these users, that still is not a solution for individuals who are deaf-blind. Also, CAPTCHAs are burdensome for everyone, and increase the likelihood that individuals will fail to submit the form or complete the task. *Content developers* should consider other creative alternative solutions that do not burden the user.

B.1.15. Accessible Tables

Data tables should not be used to force content into visible columns. Multi-column layouts can now be attained using CSS to handle layout and positioning. Data tables are useful for presenting data in rows and columns. A few specific HTML tags are required in order to ensure that data tables are accessible to screen readers. Without these tags, users who are unable to see the table can find it very difficult or impossible to understand the relationship between table headers and the cells within their scope.

Content developers should determine whether the table will be simple or complex and apply the specific tags as noted below.

B.1.15.1. Simple Table

A simple table has a single header at the top of each column, and optionally a single header in the first column of each row. It has no nested columns or rows. To make a simple table accessible, apply the following techniques:

- Markup all column headers or row headers as table headers using the <th> element.
- Define the scope of each <th> using the scope attribute (the value of scope can be either “col” or “row”)

B.1.15.2. Complex Table

A complex table is any table that is not a simple table, as defined in the preceding section. There might be nested rows or columns, or headers might be located in places other than the first row or column. These sorts of tables can be very challenging for screen reader users to understand. To ensure their accessibility, apply the following techniques:

- Markup all column headers or row headers as table headers using the <th> element
- Add a unique id attribute to each <th> element

- For every table data cell (<th>), add a headers attribute that lists the ids of all headers that apply to that particular cell. If more than one header applies to a cell, separate ids with a space

For additional assistance and guidance regarding the use and development of accessible tables, consult the webmaster and/or developer of authoring tool.

B.1.16. Effective Use of Color

There are two accessibility issues related to choice of color:

B.1.16.1. Avoid Using Color to Communicate Information

Because some users are unable to perceive color differences or may not perceive color the same way others do, it is important to avoid using color alone to communicate information. For example, if link text is blue, *content developers* should also enable underline feature so users who are unable to perceive color differences can distinguish links from surrounding text.

B.1.16.2. Choose Colors with Ample Contrast

Because some users have difficulty perceiving text if there is too little contrast between foreground and background, *content developers* must use color combinations that meet clearly defined contrast ratios per W3C WCAG 2.0. CAHELP applies Level AA for contrast success criteria. In order to meet Level AA, *content developers* must ensure that text or images of text must have a contrast ratio of at least 4.5:1 (or 3:1 for large text). In order to meet the guidelines at the stricter Level AAA, the contrast ratio must be at least 7:1 (or 4.5:1 for large text).

Several free tools have been developed that make it easy to check color combinations for WCAG 2.0 compliance. *Content developers* may utilize the following resources to determine Level AA compliance for color contrast:

- Colour Contrast Analyser by the Paciello Group (for Windows or Mac) (url: *https://www.paciellogroup.com/resources/contrastanalyser/*)

- WebAIM Color Contrast Checker (url: https://webaim.org/resources/contrastchecker/)

B.1.17. Meaningful Link Text

Screen reader users navigate websites using a variety of techniques. One of those is to pull up a list of links (a feature on most screen readers) and navigate through that list. Given this, link text should be able to stand alone independently of its context. For example, links like "click here" and "more" are meaningless out of context. Also, speech recognition users can click links with a voice commence like "click" followed by the link text. Therefore, *content developers* should keep link text short and easy to say.

For both of these reasons long URLs should be avoided as link text (short URLs like cahelp.org) are okay since they are easy to say and stand-alone independently of context.

B.1.18. ARIA Landmark Roles (Accessible Rich Internet Applications)

ARIA is a new W3C specification that consists of markup that can be added to HTML in order to clearly communicate the roles, states, and properties of user interface elements. User interface includes both the "user agent user interface," i.e., the controls (e.g., menus, buttons, prompts, etc.) and mechanisms (e.g., selection and focus) provided by the user agent that are not created by content; and the "content user interface," i.e., the enabled elements that are part of content, such as form elements, links, applets, etc. This information helps screen readers and other assistive technologies to better understand the elements on a web page, and to provide a user interface that enables their users to effectively interact with those elements.

One of the easiest ARIA features to implement, and one that provides significant immediate benefits to screen reader users, is landmark roles. There are eight of these roles, each representing a block of content that occurs commonly on web pages. To use them, webmasters and/or developers of authoring tools simply add a relevant role attribute to an appropriate container within the HTML. Then, screen reader users can quickly jump to that section of the page. The eight ARIA landmark roles are:

- Role="banner"
- Role"navigation" (e.g., a menu)
- Role="main" (the main content of the page)
- Role="complementary" (e.g., a sidebar)

- Role=”contentinfo” (meta data about the page, e.g., a copyright statement)
- Role=”search”
- Role=”form”
- Role=”application” (a web application with its own keyboard interface)

If a role is used more than once on a page, the aria-label attribute should also be used in order to distinguish between the two regions. For example, a web page might have the following two navigation regions:

-
-

When role=”application” is used, there is an exception that the application has its own model for navigating and operating all controls by keyboard, and help text is easily available so users can learn the keystrokes. When assistive technologies encounter content that is marked up with role=”application”, they stop listening for users’ keystrokes and hand off all functionality to the application. This can be problematic as it defies users’ expectations. Keys that normally perform certain functions when using their assistive technology suddenly stop providing that functionality.

Therefore, webmasters and/or developers of authoring tools should use role=”application” only when an application has been carefully developed with accessibility in mind, and steps have been taken to inform users of what to expect.

For additional clarification and guidance on Aria landmark roles, consult the webmaster and/or developer of authoring tool.

B.1.19. ARIA for Web Application

Like ARIA for Landmark Roles, ARIA for web applications is W3C specification that consists of markup that can be added to HTML in order to clearly communicate the roles, states, and properties of user interface elements. This information helps screen readers and other assistive technologies to better understand the elements on a web page, and to provide a user interface that enables their users to effectively interact with those elements.

For example, imagine a web page where a user is able to click a button to trigger some action on the page. When the user clicks the button, a message appears at the top of the page informing the user of their success or failure. Using HTML alone, screen reader users would have no idea that this message has appeared, and even if they suspected it had appeared, they might not be able to easily find it. With ARIA, webmasters and/or developers of authoring tools could simply add role="alert" to the container where the message will appear. Then, when the content of that container changes, screen readers will interrupt the user by announcing the message content. The user's focus will remain in their original location so they can resume their work.

Webmasters and/or developers of authoring tools creating dynamic, rich, interactive user interface elements for web pages must include ARIA markup or there is very little possibility of their being accessible.

Testing ARIA:

- Use the W3C Markup Validation Service to check HTML against current web standards. This tool includes checks for valid use of ARIA markup.

- Test website or web application with multiple browser/screen reader combinations. Support for ARIA is a moving target, and even if the code is valid, there might be problems in the way its rendered with assistive technologies. There is no substitute for testing, especially if the website has rich, interactive content.

For additional assistance and guidance, consult the webmaster and/or developer of authoring tool. For help with testing with assistive technologies, please contact accessibility@cahelp.org.

References:

- WCAG 2.0 Success Criterion 4.1.2 Name, Role, Value (Level A)

B.1.20. Avoiding Reliance on Visual Characteristics

Content that flashes or flickers can trigger seizures in susceptible individuals. Therefore, flashing or flickering content should be avoided.

The best technique for addressing this issue is to avoid using content that flashes or flickers. Not only can it cause seizures, but it is likely to be annoying or distracting for users in general. If *content developers* must use content that flashes or flickers, test the content using methods described below to be sure the content flashes or flickers at a safe level.

Testing:

The W3C WCAG 2.0 includes specific technical requirements for determining whether content flashes or flickers at an unsafe level. In general, if content flashes more than three times per second, it is unsafe. However, the W3C provides a more precise technical formula for calculating general flash and red flash thresholds. The Trace Center at the University of Wisconsin has developed a Photosensitive Epilepsy Analysis Tools (PEAT) for measuring whether web or computer applications are likely to cause seizures.

References:

- WCAG 2.0 Success Criterion 2.3.1 Three Flashes or Below Threshold (Level A)

B.1.21. Creating Accessible Videos

Videos and audio content can help make web pages and course curriculum provided by the CAHELP Professional Learning more engaging. However, they can also erect barriers unless delivered with accessibility in mind. Videos should be produced and delivered in ways that ensure that all members of the audience can access their content. An accessible video includes captions, a transcript, audio description, and is delivered in an accessible media player. When delivering video content, the following accessibility issues must be considered by *content developers and approvers*, and other designated staff producing or delivering video:

- **Some people are unable to hear audio.** Audio content such as audio-recorded lectures or podcasts must be accompanied by a transcript, and videos must be provided with closed captions.

- **Some people are unable to see video.** Video must be carefully scripted or edited in a way that ensures all important content is accessible through the audio track. If this is not the case, any important information that is presented visually must be described in a separate narration track using a technique called audio description.

- **Some people are unable to operate a mouse.** Multimedia content should be delivered in a player that can be operated with keyboard alone, has controls that are properly labeled so that they are announced properly to screen reader users, and can be operated effectively by speech input users.

B.1.21.1. Captions

Captions are text versions of the audio content, synchronized with the video. They are essential for ensuring a video is accessible to

members of the public who are deaf or hard of hearing. Captions also help non-native English speakers to understand the video, make it possible to search for content within the video, help with the spelling of technical terms spoken in the video, and make it possible to generate an interactive transcript where users can click anywhere in the transcript to watch the video where the text is spoken.

There are two general approaches to captioning video that *content developers and approvers*, and other appropriate staff producing or delivering video can consider:

- **Outsource.** Companies such as Automatic Sync Technologies, 3PlayMedia, cielo24, and many other captioning service providers will caption videos for a fee. Consult CAHELP JPA Virtual Compliance Team prior to contacting these companies for additional information.
- **Do it Yourself.** There are free tools available online that make it possible and easy to caption video. See captioning your own video for free (*See Appendix D*).

The end product generated by the above two options is a caption file. Most caption files are plain text files with time codes indicating the start and stop times. However, there are various types of caption files with slight variations in their syntax. Once a caption file has been created, the final step is to add this file to the video. How *content developers and approvers* accomplish this depends on where the video is hosted. For specific instructions, select one of the following options:

- Adding captions to YouTube videos (link to…
- Adding captions to videos on web pages (link to…
- Adding captions to videos in Panopto (link to…
- Adding captions to videos in Canvas (link to…
- Adding captions to videos in MediaAMP (link to…

References:

- WCAG 2.0 Success Criterion 1.2.1 Audio=only and Video-only (Prerecorded) (Level A)
- WCAG 2.0 Success Criterion 1.2.2 Cations (Prerecorded) (Level A)
- WCAG 2.0 Success Criterion 1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A)

- WCAG 2.0 Success Criterion 1.4.2 Audio Control (Level AA)
- WCAG 2.0 Success Criterion 1.2.4 Captions (Live) (Level AA)
- WCAG 2.0 Success Criterion 1.2.5 Audio Description (Prerecorded) (Level AA)

B.1.21.2. Audio Description

Audio description is a separate narrative audio track that describes important visual content, making it accessible to individuals who are unable to see the video. Individuals who are blind can understand much of a video's content by listening to its audio. However, if a video includes content that is only presented visually (e.g., on-screen text or key actions that are not obvious from the audio), this visual information must be described in order to be accessible to individuals who are unable to see it.

Like captions, there are two general approaches to producing audio description for video that *content developers and approvers*, and other appropriate staff producing or delivering audio shall consider:

- **Outsource.** The American Council of the Blind has compiled a comprehensive list of commercial services for producing audio description. If the video contains a lot of visual information, this may be the best option since describing visual content effectively requires specialized skills. Typically, service providers will produce a new video that has the descriptive narration mixed in with the program audio. *Content developers and approvers*, and other appropriate staff producing or delivering audio can then provide a video in two formats: one with audio description and one without.
- **Do it Yourself.** For videos that have very little visual information, the same free online tools that are used for creating closed caption tracks can be used for creating description tracks. Description tracks are essentially the same as caption tracks—short blocks of text with timestamps that synchronize the text with the video—but their function is different. They are intended to be read aloud by screen readers, rather than voiced by a human narrator. Playing video with text-based audio description requires a media player that supports this feature, such as Able Player, the

open source media player developed at the University of Washington.

B.1.21.3. Live Captioning and Description.

If live events are simulcast over the web, live captioning is needed in order to provide access to the audio content for audience members who are deaf or hard of hearing. Similarly, live description may be needed if key visual content is not otherwise verbalized, such as in a dramatic production. At the CAHELP, these services are coordinated through the Professional Learning team with the assistance of *content developers* and *approvers*, and the CAHELP JPA Virtual team and/or the ACT.

B.1.21.4. Transcript

A transcript is a text version of the media content. A transcript should capture all the spoken audio, plus on-screen text and descriptions of key visual information that wouldn't otherwise be accessible without seeing the video. Transcripts make video content accessible to everyone, including individuals who are unable to view the video due to accessibility problems or technical limitations. They are also helpful for individuals who want to quickly scan or search a video's content but do not have the time to watch the entire video.

If *content authors* have captioned the video, a transcript is available as one of the optional output formats produced by the closed captioning process. This is true of both the free online tools and the commercial service providers. To make the transcript available simply link to it from the web page, wherever it is linked to or display the associated video.

Content developers and approvers, webmasters, procurement officials, and all others responsible for developing, loading, maintaining, or auditing web content and functionality, may consider using Able Player, the accessible open source media player developed at the University of Washington, which generates an interactive transcript automatically using the caption and/or description tracks.

B.1.21.5. Choosing an Accessible Media Player

When choosing how to deliver video, it is important that *content developers and approvers*, webmasters, procurement officials, and all others responsible for developing, loading, maintaining, or auditing web content and functionality, consider options that are fully accessible. Whether selecting a media player plugin or module for the CAHELP website or selecting a service to host videos, the following questions should be answered about the available options:

- Does the media player support close captions?
- Does the media player support audio description in a way that enables users to toggle the narration on and off?
- Can the media player's buttons and controls be operated without a mouse?
- Are the media player's buttons and controls properly labeled so they can be operated by a blind person using a screen reader?
- Is the media player fully functional, including all of its accessibility features, across platforms and in all major browsers?

Able Player, the accessible open source media player developed at the University of Washington satisfies all of the above criteria. It is a free, open-source media player developed with accessibility in mind. For additional information on Able Player, go to Able Player on Github (url: *https://ableplayer.github.io/ableplayer/*).

B.1.22. Procuring Accessible IT

The CAHELP strives to ensure that IT products developed at, purchased by, or used at the CAHELP are accessible to all individuals. To reach this aspirational goal, the ACT shall be responsible for making decisions about which products to procure and must consider accessibility as one of the criteria for acquisition. This is especially critical for enterprise-level systems and other technologies that affect a large number of students, teachers, and/or staff. The following three steps provide an example of how accessibility can be considered in the procurement process.

For additional information and guidance on procurement of products accessible to all, consult IT services or the ACT with any of these steps.

B.1.22.1. Ask vendors to provide information about the accessibility of their products.

The following is an example of accessibility language that could be used in requests for proposals (RFPs):

Mandatory Scored Requirement:

- Bidder must describe how their IT products or services are accessible to users in accordance with CAHELP guidelines;
- CAHELP refers to the WCAG 2.0 developed by W3C Level AA for guidance in meeting its IT accessibility commitments.

If there are issues that prevent a bidder's IT product or service from meeting these requirements, the bidder must describe efforts underway to address these issues, including anticipated timelines for completion.

B.1.22.2. Validate information provided by bidders and evaluate the product for accessibility

Consult the ACT for assistance. Vendors should provide detailed information about the accessibility of their product or services. One common method is by providing a Voluntary Product Accessibility Template (VPAT). This is a standard form developed to assist federal agencies in fulfilling their Section 508 requirements. VPATs can sometimes be informative, but they have limitations since they are self-reports completed by the vendors. Some vendors do not have adequate technical expertise to accurately assess their products' accessibility. Others skillfully complete their VPATs in ways that trivialize the significance of accessibility shortcomings. Therefore, VPAT claims should be independently verified and not accepted at face value. A VPAT could provide a good starting point, but ultimately vendors, particularly those whose products are selected as finalists, should be engaged in a thorough discussion about accessibility of their products.

Few IT products are fully accessible. However, vendors should at a minimum be willing to make a commitment to address their accessibility problems. Without this commitment, using the product may place the CAHELP at risk for discriminating against some of its users and/or employees.

The CAHELP procured and/or contracted web host shall provide detailed information about the accessibility of their web product or services and may provide a Voluntary Product Accessibility Template (VPAT) for consideration.

B.1.23. Include Accessibility Assurances in Contracts with Vendors

If ultimately the best product for meeting a particular need is one that fails to fully meet accessibility requirements, vendors should be asked to make a commitment to improving accessibility over a specified timeline, perhaps working with the CAHELP JPA Virtual Compliance Team and the ACT.

After procurement officials discuss accessibility issues with a vendor, the procurement contract should include language that specifically documents the agreement between vendor and procurer as to how satisfactory progress on accessibility will be measured. The vendor might provide a roadmap as an addendum to the contract with a prioritized list of accessibility issues and a timeline for addressing each issue. Contract extensions might be contingent upon satisfactory progress toward resolving the issues identified in the roadmap.

Even if the product is currently accessible, the contract should include language that assures continued accessibility as the product is updated. This is especially important for products that are developed on an ongoing rapid release cycle.

B.1.24. Managing Projects for Accessibility

It shall be the responsibility of the CAHELP JPA Virtual Compliance Supervisor to ensure that all projects related to accessibility be prioritized. All areas of the CAHELP website will be reviewed annually using the processes described at WCAG 2.0. Reviews are the responsibility of the CAHELP JPA Virtual Compliance Supervisor in collaboration with the ACT. Accessibility checks will be incorporated into the publishing workflow for all new content.