Web Accessibility Report

Prepared for S603 – Workshop on Web Accessibility

February 28, 2013

Name of evaluator: Alan Rhoda

Web site evaluated: Epicurious.com

Three representative pages on the site were evaluated for web accessibility in accordance with the W3C's Web Content Accessibility Guidelines 2.0 (WCAG 2.0):

- Home page (http://www.epicurious.com/)
- Search results page (http://www.epicurious.com/tools/searchresults?search=cookies&x=0&y=0)
- Recipe page (http://www.epicurious.com/recipes/food/views/Glittering-Lemon-Sandwich-Cookies-362630)

In the evaluation below, these pages are referred to as **Home**, **Search**, and **Recipe**, respectively.

Evaluation tools used:

Many evaluation tools used were used to test for different accessibility issues. Tools used are listed here by category. All of these tools are freely available. Hyperlinks point to where the tool may be downloaded and/or accessed. General-purpose, multi-function accessibility tools are listed at the top. More specialized tools are grouped by the primary accessibility issue that they check.

General web development:

- Firebug – a Firefox extension (https://addons.mozilla.org/en-us/firefox/addon/firebug/)
- W3C markup validation service (http://validator.w3.org/)
- HeadingsMap (https://addons.mozilla.org/en-US/firefox/addon/headingsmap/)

Multi-function accessibility evaluation:

- AChecker (http://achecker.ca/checker/index.php)
- WAVE Toolbar (http://wave.webaim.org/toolbar)
- Juicy Studio Accessibility Toolbar (https://addons.mozilla.org/en-US/firefox/addon/juicy-studio-accessibility-too/)
- Accessibility Evaluation Toolbar (https://addons.mozilla.org/en-us/firefox/addon/accessibility-evaluation-toolb/)
- Web Accessibility Toolbar for IE (http://paciellogroup.com/resources/wat/ie)

Screen reader accessibility:

- Fangs (https://addons.mozilla.org/en-US/firefox/addon/fangs-screen-reader-emulator/)
- ChromeVox (http://www.chromevox.com/index.html)

Visual accessibility:

- Colorblind Design (https://addons.mozilla.org/en-us/firefox/addon/colorblind-design/)
- Colorblind Web Page Filter (http://colorfilter.wickline.org/)

Mobile accessibility:

- Cowemo Mobile Emulator (http://www.mobilephoneemulator.com/)
- Mobile Phone Screen Emulator (http://web.redevelopments.co.uk/mobile-phone-emulator/mobile-phone-emulator.asp)
- mobiReady (http://ready.mobi/launch.jsp?locale=en_EN)
- Screenfly (http://quirktools.com/screenfly/)

Evaluation Results:

The following evaluation follows the sequence of the WCAG 2.0 guidelines. There are twelve guidelines subdivided into four accessibility principles according to which content and functionality on websites should be:

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

For each problem identified, a suggested solution is proposed. Severity of problems is indicated in accordance with the following rubric (from the UIUC center for IT Accessibility):

- Priority 1 – Issues that prevent some individuals from using an application. These are the most important to address.
- Priority 2 – Issues that make using an application difficult and/or disorienting.
- Priority 3 – Issues that are annoying or enhancements that could be made to the application.

Proposed solutions are also graded in terms of whether they are *easy*, *moderate*, or *difficult* to implement.

1. Perceivability

Guideline 1.1 – Provide text alternatives for any non-text content (i.e., images).

- Rationale: Supplying text alternatives for images that describe the information contents of the images renders that content accessible to users who are visually impaired or whose equipment does not support graphics. Unlike images, text alternatives can be programmatically presented in other forms (e.g., large print, braille, or synthesized speech) according to user needs.
- According to AChecker, 23 images on **Home**, 19 images on **Search**, and 24 images on **Recipe** are missing an ALT attribute. Some of these images appear to be purely decorative, but many of them are informative (e.g., logos, pictures of cookbooks, etc.).

  Problem: Images without descriptive ALT attributes are inaccessible to the blind and anyone who must rely on a screen reader.

  Priority: 1

  Solution: (Easy) Add an ALT attribute containing a brief description of the content to all informative images (e.g., logos, pictures of cookbooks, etc.).

Guideline 1.2 – Provide alternatives for time-based audio–visual media.

- Rationale: It is important to render time-based audio–visual content accessible in multiple modalities so that users who are unable to access the content in one modality (e.g., blind and/or deaf users) can still access it in other way.
- This guideline appears not to be applicable to either **Home** or **Search**. No time-based media was found on those pages.
- On **Recipe**, however, there is a "video" tab containing video content. This content is neither closed-captioned, nor accompanied by a transcript.

  Problem: Video without a text alternative via closed-captioning or transcript is inaccessible to the deaf.

  Priority: 1

  Solution: (Moderate) Add closed-captioning and/or link to a transcript file that lists times so that the text can be synced up with the video.

Guideline 1.3 – Adaptable: Content should be presentable in different ways without losing information or structure.

- Rationale: The purpose of this guideline is to ensure that page information and structure can be programmatically determined and thus rendered by assistive technologies for a wide array of users. To test this, I checked the heading structure and other semantic labeling in the HTML code and suppressed styles to ensure that the structure of the information was not style dependent.
- According to the Fangs screen reader emulator, the heading structure of **Home** is problematic. There is no H1 heading, heading levels are skipped (with H4 instead of H3 coming after H2), there are a couple empty headings, and in general the heading hierarchy doesn't make any clear sense. In contrast, the heading structure of both **Search** and **Recipe** is pretty clear, with the exception that at one point the heading structure skips from H1 to H3.

  Problem: A clear and orderly heading structure helps screen readers and other assistive technologies present page information to users in a meaningful order. Lack of a clear and sensible hierarchy can be very disorienting for disabled users.

  Priority: 2

  Solution: (Easy/Moderate) Repair the heading structure, making sure there is a unique H1 tag and a sensible headings hierarchy, with no skipped levels.

- When styles are disabled using the WAVE toolbar, **Search** fares quite well. We are very quickly told what web site we're on and that we're on a "search results" page. We are then presented with a hierarchically organized list of filtering options, followed by the actual search results. **Home** and **Recipe**, however, are not so well-structured. On both we have to wade through a page or more of sign in and social media links before we are even told what website we're on. After that comes the page title and then about another page of sign in stuff and miscellaneous adds until we come to the main page content.

  Problem: On the **Home** and **Recipe** pages styling is being used to control much of the information structure, but this means that much of that structure is programmatically

inaccessible, and thus the pages are likely to be confusing for those using assistance technologies.

Priority: 2

Solution: (Moderate) Restructure the HTML code for the pages so that, for example, the site name, page title, and other top-level information appear right at the top.

- Additional suggestions (Priority 3):
  - (Easy) Where appropriate, use semantically informative HTML tags like <header>, <footer>, and <nav>, instead of the generic .
  - (Moderate) Use WAI-ARIA roles and properties to provide additional semantic information to assistive technologies.

Guideline 1.4 – Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

- Rationale: A sizable percentage of users have sub-normal visual and/or hearing acuity. This includes not only persons who are blind and deaf, but those who are colorblind or who just have poor eyesight and/or hearing.
- According to Juicy Studio, while many display elements on each page easily passed the W3C's AA and even AAA standards for color brightness and contrast, there were many brightness and color failures:
  - **Home** had 101 luminosity contrast ratio failures, 112 brightness difference failures, and 144 color difference failures.
  - **Search** had 100 luminosity contrast ratio failures, 107 brightness difference failures, and 135 color difference failures.
  - **Recipe** had 103 luminosity contrast ratio failures, 105 brightness difference failures, and 140 color difference failures.
- A qualitative test using Colorblind Design to view each page as it would be seen by someone with protanopia (red–green colorblindness; no red cones), deuteranopia (red–green colorblindness; no green cones), tritanopia (blue–yellow colorblindness; no blue cones), or achromatopsia (total color blindness) yielded the following:
  - On each of the pages, small and/or light-grey text was generally very hard to read.
  - Visited links were very hard to read under the protanopia filter.
  - Small and orange to reddish text on a light background was very hard to read under the deuteranopia filter, as was small and light text on an orange to reddish background.
  - Small and light-grey text and and visited links were hard to read under the tritanopia filter.
  - Under the achromatopia filter, light text on a white background was very hard to read (due to low contrast). In addition, the main navigation menu on **Search** was also very hard to read.
- Given that small text was an issue, I checked whether manually enlarging text in the browser would help. It did, but only to a point. More than two enlargements caused text in the main navigation menu to wrap and sub-menu items to truncate in ways that made them difficult to use. Additional enlargement caused further wrapping elsewhere that obscured items and impaired page functionality.

- Visual perceptibility summary:

  Problem: Summarizing the above tests, major perceivability problems on **all three** pages stem mainly from small text and low contrast.

  Priority: 1

  Solution: (Easy/Moderate) Increase the font sizes of very small text and bold and/or darken lighter text to increase contrast. This will make the page overall more perceptible and alleviate the need for manually enlarging the text.

2. Operability

Guideline 2.1 – Keyboard accessibility: Make all functionality available from a keyboard.

- Rationale: Not all users are able to use a mouse. This includes not only screen reader users, but also users with various types of motor disabilities.
- After extensive testing of the web pages using only a keyboard, it was found that:
  - On the positive side, there appear to be no *keyboard traps* (i.e., places that users can't TAB out of) and *nearly* all page functionality is keyboard accessible.
  - On the negative side, however, *not all* functionality is keyboard accessible:
    - On **Search**, I could not find a way to access the "refine this search by …" box using the keyboard.
    - On **Home**, I could not find a way to access the Flash slideshow near the top-middle of the page. If one could, it would likely become a keyboard trap. But still, this is keyboard-inaccessible content.
    - On **Recipe**, I could not find a way to toggle the checkboxes in the "newsletter sign up" box on the right-hand side. Focus never goes to the checkboxes themselves, but only to the nearby link text. Pressing the spacebar jumps one down the page. It doesn't toggle the checkbox.

  Problem: All three pages have keyboard inaccessible areas.

  Priority: 1

  Solution: (Moderate) Keyboard inaccessible areas on each page should be addressed. I don't know what is preventing focus from the elements on the **Search** and **Recipe** pages, but this should be fixable (perhaps by using one of the suggestions under solution "b" above). There may be no easy way to do this with the Flash slideshow on **Home**, but a link to an alternative presentation of the content could be provided.

Guideline 2.2 – Provide users enough time to read and use content.

- Rationale: Not all users are able to process information and operate controls as quickly as others. They may need more generous time allowances.
- There is no apparent time-limited functionality on any of these pages. Thus, the pages are fully accessibility with respect to this guideline.

Guideline 2.3 – Do not design content in a way that is known to cause seizures.

- Rationale: Web pages containing content that flashes more than three times per second can induce seizures in some people.

- There is no flashing content on any of these pages. Thus, the pages are fully accessibility with respect to this guideline.

Guideline 2.4 – Navigability: Provide ways to help users navigate, find content, and determine where they are.

- Rationale: Users need to be able to determine where they are in a web page (i.e., where the focus is) and how to navigate, whether by mouse, keyboard, or other device, to where they may want to go. Most important is having an easily accessible navigation menu with readily understandable links and making it possible for users to jump between major sections of a web page without necessarily having to wade sequentially through everything in between.
- On the positive side, pages are aptly titled and, while there is no "breadcrumbs" trail showing where one is in the site hierarchy, in the main menu the active section of the site (e.g., "Recipes & Menus") appears in a different color.
- Several usage problems were encountered when trying to navigate the pages with only a keyboard.
  - There is no initial focus. Focus jumps around quite a bit and it's often not clear where the focus is.
  - It takes very long time to tab to the main navigation menu. 128 tabs on **Home**.
  - The dropdown menu items on the main navigation menu cannot be accessed directly with a keyboard. On **Home**, however, the main navigation menus are reproduced (for the most part) down the left-hand side beginning about a third of the way down the page, but the menus are not reproduced on either **Search** or **Recipe**. While one can press ENTER in the main navigation menu to go to another page containing all the sub-menu items, this is inconvenient because one can't easily browse other sub-menu options without tabbing (100+ times) through all the content to get back to the main menu.
  - There is no "skip navigation" link, i.e., a way to skip directly to the main content of the page. This is not much of an issue on **Home**, as that page is just a portal to the rest of the site, but it is an issue on both **Search** and **Recipe**, where a user (esp., one using a screen reader) would probably want to be able to skip directly to the search results / recipe details.
- Some other navigation issues were encountered as well.
  - While color shading of the active main navigation tab is nice, it's not at all helpful to screen reader users. That's why something like a "breadcrumbs" trail would be nice.
  - Apart from using a mouse (or a lot of tabbing), there's no easy way to move between different content regions of a page.
- Navigation summary:

  Problem: Lots of areas in which navigation accessibility could be improved.

  Priority: 2–3

  Solutions:

  a. (Priority 2, Easy) A "skip navigation" or "skip to main content" link should be added at the top of at least the **Search** and **Recipe** pages. (**Home** is more of a portal page. It doesn't have any clear main content section.)

b. (Priority 2, Easy) Styling can be used to place an observable border around the element in focus.
c. (Priority 3, Moderate/Difficult) Steps should be taken to control the tab order so that one can get to important areas, like the main navigation menu, quickly. Judicious use of the tabindex attribute and/or WAI-ARIA roles and properties may help here. It may also help facilitate easier navigation between different content regions on a page.
d. (Priority 3, Easy) Place a "breadcrumbs" navigation trail above the main content area.

3. Understandability

Guideline 3.1 – Make text content readable and understandable.

- Rationale: In order for an assistive technology to present text in a proper format for users it needs to be able to programmatically determine the language. Also, language (e.g., technical jargon) that is unclear to many users is a barrier to website usage and should therefore either be avoided or explained.
- A perusal of the HTML code with Firebug reveals that the language, in this case "en-US", can be programmatically determined.
- Most of the language on these pages is not particularly technical and should pose few barriers to most adult English speakers. Nevertheless, some of the filtering options on **Search** and directions on pages like **Recipe** may contain terms (like "hors d'oevres") or units (English vs. metric) that culinary novices may be unfamiliar with. No glossary of terms is provided that I can see, but a helpful unit conversion tool is provided on **Recipe**.

  Problem: Specialized cooking terminology may be unclear to many users.

  Priority: 3

  Solution: (Moderate) Create a glossary of possibly unfamiliar cooking terminology and link terms to the glossary. Or perhaps put SPAN tags around terms with descriptive ALT text so that it will display a tooltip when focused.

- When a list of links was inspected in Fangs, a few of the links (e.g., "go") were very uninformative as to their purpose.

  Problem: Because screen readers cannot capture the spatial context of links very well, it should not be presumed that the purpose of ambiguous link text can be inferred from context. Wherever possible, link text should convey a clear sense of its purpose.

  Priority: 2

  Solution: (Easy) Review link text and revise where unclear (e.g., "go to search results" rather than "go").

Guideline 3.2 – Make Web pages appear and operate in predictable ways.

- Rationale: Consistency of layout and operation makes it easier for users to navigate a website and find the information they need with an economy of effort.

- One positive is that the structure and position of the main navigation menu and search bar are consistent from page to page.
- On the other hand, the video content on **Recipe** has *no relation* to the dish that the recipe pertains to.

  Problem: This is confusing because the content is on a "video" tab that stands in parallel with a series of other tabs ("recipe", "ratings", "photo", etc.) that all pertain to the recipe in question.

  Priority: 2

  Solution: Remove irrelevant videos from the "video" tab. If desired, this content could be placed on another page or placed in a separate "videos" area of the same page. The point is to keep distinct content distinct so as not to confuse.

Guideline 3.3 – Input assistance: Help users avoid and correct mistakes.

- Rationale: Form controls should make it clear to users what sort of input is expected or required, helpfully inform users when input is incorrect so that it can be corrected, and inform users about potential liabilities and allow them a chance to either confirm or back out before incurring those liabilities.
- AChecker detected that many form controls on **all three** pages were lacking associated labels.
- There are seven forms on **Home**: (1) a user sign-in form, (2) a recipe search form in the middle of the page about half-way down, (3) a global search form in the header, (4) a email newsletter sign-up form, (5) a magazine subscription drop-down, (6) a drop-down with links to "sister sites", and (7) a poll.
  - None of the forms has any textual indication of which, if any, fields are required. This is mostly an issue for form (1), where both username and password should be required. All of the other forms consist either of only one input field or just a single set of option buttons (7).
  - Forms (1) and (4) require a valid email address. Both provided appropriate error messages when an invalid address was entered.
  - None of the complex forms—(1), (2), (3), (4), and (7)—have their fields semantically grouped with either fieldset or legend tags.
- With the exception of the user sign-in form, **Search** has the same forms as **Home**, as well as two additional recipe search forms and a drop-down control for sorting search results. Similar issues apply.
- With the exception of the user sign-in form, **Recipe** has the same forms as **Home**, as well as buttons for rating and reviewing recipes. Similar issues apply.
- Summary of form issues:

  Problem: Many forms and form controls on **all three** pages are lacking appropriate semantic information. In order for the purpose of a form, the role its various controls, and whether a field is required or not to be programmatically determined so as to appropriately inform users of assistive technologies, this information need to be supplied in the HTML.

  Priority: 2

Solution: (Easy) Supply legends for forms, fieldsets for complex forms, and required="required" attributes when an input is required. Add a label for each form control and associate it with that control's ID attribute. Label text should describe the purpose or function of the control.

4. Robustness

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

- Rationale: For maximal compatibility with current and future user agents, web pages must properly parse and the names, roles, and states of all functional components must be programmatically determinable. Additional robustness can be provided by ensuring that access to content is not dependent upon client-side technologies like JavaScript and Flash and that all content (or as much as feasible) is accessible from other viewing devices, like mobile "smart" phones.
- Parsing was checked using the W3C's Markup Validation Service. The service found 260 parsing errors and 329 warnings on **Home**, 928 errors and 1670 warnings on **Search**, and 344 errors and 352 warnings on **Recipe**. The vast majority of these alerts resulted from the use of ampersand ('&') characters, where the & entity reference should have been used. But there were a few other problems as well, including unrecognized tags and attributes and elements with missing or unfinished end tags.

  Problem: Parsing failures could cause a page to not display or function properly in devices that are not as tolerant as many modern web browsers.

  Priority: 1

  Solution: (Easy) Go through the W3C's validation report and fix the errors. At the very least, the web pages should fully conform to the specifications of the stated document types, in this case XHTML 1.0 Transitional.

- Juicy Studio was used to check for WAI-ARIA roles and properties. Apart from a couple third-party widgets provided by Facebook, none were found on any of the pages.

  Problem: WAI-ARIA stands for the Web Accessibility Initiative's Accessible Rich Internet Applications standard. WAI-ARIA roles and properties provide more detailed semantic information than do standard HTML tags and attributes about the roles, states, and properties of a web page's interactive components. Assistive technologies, especially screen readers, can utilize this markup to help screen reader users more effectively navigate and utilize the information on a page.

  Priority: 3

  Solution: (Moderate) Incorporate WAI-ARIA roles and properties into the site's web pages.

- Dependence of site functionality upon client-side technologies was checked by using the Accessibility Evaluation Toolbar to disable scripting and to identify event handlers. Event handlers are attributes like "onclick", "onfocus", "onmouseover", etc. that tell browsers and scripting languages like JavaScript to look for a certain kind of user

behavior (e.g., moving the mouse over a button), at which point a specified function is performed (e.g., rolling out a drop-down menu).

- The Accessibility Evaluation Toolbar's "List of Events" feature identified 12 elements with event handlers on **Home**, 40 on **Search**, and 34 on **Recipe**. In most cases, the event handlers are either device-independent (like "onfocus" and "onblur") rather than device-specific (like "onmouseout" and "onmouseover") or occur in conjunction with device-specific event handlers. In some cases, however, specifically eight times on **Search** and three times on **Recipe**, a tag has device-specific event handlers applied, but no device-independent alternatives.

  Problem: Use of device-specific event handlers without a device-independent alternative means that some functionality is available only by mouse (and/or touchpad) users. This is problematic for screen reader and keyboard-only users and users who for one reason or another either do not have or (because of motor limitations) cannot use a mouse (or touchpad).

  Priority: 1

  Solution: (Easy) Make sure that for every device-specific event handler there is a device-independent alternative with equivalent functionality.
- In **all three** pages, when JavaScript is disabled, the drop-down navigation menus at the top of each page cease to drop down when the main menu item receives focus. Also disabled are the search boxes and form error checking (i.e., the email field). In addition, on **Home**, the Flash slideshow near the top-middle of the page and the item toggle buttons in the "the epicurious shop" box on the right hand side are also disabled.

  Problem: This is problematic because not all browsing devices are JavaScript enabled, thereby preventing some users from having full access to the site.

  Priority: 1

  Solution: (Moderate) Where possible, alternatives to JavaScript should be implemented for this functionality (e.g., server-side error-checking or search box processing, CSS-driven drop-down menus, etc.).

- Finally, mobile device accessibility was checked using a Samsung SCH-1200 (Android) phone. Up until about a week ago, Epicurious maintained a separate mobile mirror site (http://m.epicurious.com), that site is now apparently defunct and has been replaced by a mobify.com site. I was unable to access the access the mobile site in a desktop browser or in any mobile phone emulator. Cowemo, Screenfly, and the Mobile Phone Screen Emulator did not display the Epicurious site as it appears on an Android phone. mobiReady does not attempt to emulate mobile sites, but when the user agent is set to a mobile OS (in this case Android), it generates an accessibility report highlighting possible issues.
  - Visual inspection on an Android phone reveals an attractive and apparently usable page. The page is greatly simplified as compared with the non-mobile site and nicely resized to fit on a mobile screen.
  - When I tried searching for "cookies", however, my search text did not display in the search bar as I was typing it. Moreover, trying to perform the search repeatedly crashed my Android browser. Without further testing I cannot be sure that these are

problems with the site and not either idiosyncratic to my phone or results of a poor Internet connection.

Problem: There appear to be systematic usability problems with the mobile search interface.

Priority: 1

Solution: (Moderate) Given that Epicurious has recently overhauled their mobile interface, it is possible that the new system is still relatively buggy. In particular, search functionality on a variety of mobile systems (esp. Android and i-OS) needs to carefully tested. If the documented problems are confirmed, then this needs to be fixed.

- When evaluated in mobiReady, the site received an overall score of 3 (Fair), with two failures and two warnings. The former had to do with (1) failure to use the XHTML Mobile Profile (XHTML-MP), and (2) six instances of invalid markup. The latter had to with (1) incorrect or missing MIME types and (2) failure to use access keys. (Note: When I tried to replicate these results in mobiReady, again specifying the user agent as "Android" mobiReady generated a very different evaluation. The much larger file size reported, however, makes me think it was evaluating the *non*-mobile site.)

  Problem: The failure should be addressed as these involve parsing issues that could impair or limit cross-platform functionality.

  Priority: 2

  Solution: (Easy) Change the document type to XHTML-MP and address the invalid markup.

Reflection Questions:

Q. 1: Describe how web accessibility affects use of the Internet for everyone, including users with disabilities. Include any relevant topics discussed during class (Universal Design, assistive technology, legal issues, WCAG 2.0, mobile, etc.).

While fully blind, deaf, and/or severely impaired persons may be a small minority of the population, a very sizable percentage of the population has at least some degree of disability that may hinder website interaction, whether poor vision, colorblindness, bad hearing, dyslexia, etc. Moveover, most accessibility solutions (e.g., ALT attributes on images; skip-navigation links; normal sized fonts with high color contrast, etc.) are easy to implement and result in web pages that are easier to use for *everyone*, not just the severely disabled.

Q. 2: What are your recommendations to a web developer seeking to create an accessible web site?

1. Become aware of the range and prevalence of various types of disabilities.
2. Become familiar with WCAG 2.0 and Section 508 accessibility recommendation. Many of them are very easy to implement.
3. Realize that in many contexts (e.g., institutions receiving money from the federal government), accessibility is legally mandated.
4. There are many great accessibility tools and plug-ins (e.g., Juicy Studio, Colorblind Design, etc.) which can greatly simplify the job of checking for accessibility issues. Use them!