

WCAG 2.0 Compliance Audit

Jimmy John's

<http://www.jimmyjohns.com/>

John Green
Accessibility Engineer
September 30, 2014

Table of Contents

Executive Summary	3
Findings	4
All non-text content must have a text alternative that serves the equivalent purpose (WCAG Guideline 1.1.1).....	4
CAPTCHA elements must have text alternatives that describe their purpose as well as output modes for non-visual users (WCAG Guideline 1.1.1).....	5
All user interface controls must have a name that describes its purpose and that name must be able to be programmatically determined (WCAG Guidelines 1.1.1 and 4.1.2).....	5
The <label> element should only be used to associate text with form controls (WCAG Guideline 1.3.1) ..	7
Heading structure should convey relationships between content (WCAG Guideline 1.3.1)	8
The relationship between data cells and their associated row and column headers must be explicitly defined (WCAG Guideline 1.3.1).....	9
Lists of items should be marked up using the appropriate list markup tags (WCAG Guideline 1.3.1).....	9
The visual presentation of text and images of text must have a contrast ratio of at least 4.5:1 (WCAG Guideline 1.4.3).....	10
All functionality must be operable through a keyboard-only interface (WCAG Guideline 2.1.1).....	11
Blocks of content that are repeated on multiple pages should be able to be easily bypassed (WCAG Guideline 2.4.1).....	11
Triggering a control should not reset the current focus to its initial state (WCAG Guideline 2.4.3)	12
The purpose of each link should be clearly defined by the link text alone (WCAG Guideline 2.4.4)	12
Links that point to targets within the page should point to targets that exist (WCAG Guideline 2.4.4)	12
All focusable elements must have clear focus states (WCAG Guideline 2.4.7).....	13
Input errors must be communicated to the user in an identifiable manner (WCAG Guideline 4.1.1).....	14
All id attribute values on a page must be unique (WCAG Guideline 4.1.1)	14
Anchor (<a>) elements should not be used solely as targets for other anchor elements (WCAG Guideline 4.1.2).....	16
When changes are made to the state of a user control, the user should be notified of the change (WCAG Guideline 4.1.2).....	16
All listbox elements should have a value initially selected on page load (WCAG Guideline 4.1.2).....	16
Additional Recommendations	17
Characters which are used for visually decorative purposes should be hidden from assistive technologies	17
HTML5 sectioning elements should have appropriate role attributes.....	17
Custom versions of default web components should expose appropriate roles, states, and properties to accessibility APIs as well as mimic default keyboard controls.....	18

Executive Summary

A WCAG 2.0 AA (<http://www.w3.org/TR/WCAG20/>) compliance audit was conducted on four components of jimmyjohns.com:

1. Brand site – <https://www.jimmyjohns.com/>
2. Blog site – <http://www.freakyfast.com/>
3. Store site – <https://store.jimmyjohns.com/>
4. Gift Card site – <https://www.mercury-gift.com/JimmyJohns/>

In general, the inspection focused on identifying any issues that may cause users difficulty in use or navigation of the site, including those users who rely on assistive technologies to browse the web. When possible, suggestions are provided to correct any deficiencies.

Note that user-generated blog content was not specifically analyzed. However, in general, images added to blog posts were discovered to not be given alt text values that were appropriate, if they were given alt text at all.

Findings

Each issue listed below is presented at a high level with a general fix followed by specific occurrences on the site for which the issue appears and the fix should be applied.

All non-text content must have a text alternative that serves the equivalent purpose (WCAG Guideline 1.1.1)

Information that is provided in a non-text manner (usually via images) must have a text alternative so that users who are unable to view the content, either due to disability or technological barrier, still have access to the content.

For images, the alt attribute should be populated with a value that conveys the same meaning as the image (as opposed to describing the visual characteristics of said image). This value is commonly referred to as "alt text."

- Global (Desktop/Mobile) – the five social icons found near the end of the page have inappropriate alt values. The alt text should describe what the icons do when triggered, such as "share this page on Facebook."
- Global (Mobile) – both versions of the images within the "Find a JJ's" link have a null alt attribute. Screen readers will read this area as "Find a".
- Public – Homepage (Mobile) – the image of the sandwich on the mobile homepage does not have an alt attribute. As this image appears to be purely decorative, a null alt attribute would be appropriate.
- Public – Homepage (Desktop) – the second homepage theater slide ("Find A JJ's") has an inappropriate alt text value of "Slide 3".
- Public – Homepage (Desktop) – the Jimmy John's logo which serves as the Twitter avatar does not have an alt attribute.
- Public – Homepage (Desktop) – the Instagram "loading" graphic has no alt text.
- Public – Homepage (Desktop) – the left and right (arrow) buttons used within the open Instagram panel do not have alt attributes.
- Public – Find a JJ's (Desktop/Mobile) – the building graphic used before the map populates has no alt text.
- Public – Find a JJ's (Desktop/Mobile) – the loading graphic used while the map populates has a null alt attribute.
- Public – JJ Store (Desktop/Mobile) – the "JJ's Gear" heading on this page is presented as an image with an inappropriate alt text value of "Splitter Clothes Heading". If this text cannot be presented as text, it should have an alt attribute value that matches the text content (i.e. "JJ's Gear").
- Public – Sustainability (Desktop) – the tab icons for "Loyalty" and "Quality" have inappropriate alt text values of "Tab Icon loyalty" and "Tab Icon quality" – these should be more descriptive and explain their function.

- Public – Sustainability (Desktop) – the ribbon and chicken images preceding the header texts do not have alt attributes. As these images appear to be purely decorative, a null alt attribute would be appropriate.
- Blog – Landing and Post Detail (Desktop/Mobile) – both the Facebook “Share” and Pinterest “Pin It” buttons do not have text alternatives.
- Store – Landing (Desktop/Mobile) – the headings on this page (“HATS”, “CLOTHING”, “MORE”) are each presented as images that have null alt text values.
- Store – Email a Friend (Desktop/Mobile) – the product image does not have an alt attribute

CAPTCHA elements must have text alternatives that describe their purpose as well as output modes for non-visual users (WCAG Guideline 1.1.1)

Non-visual users are unable to complete traditional CAPTCHA tasks such as those currently implemented on the site. A CAPTCHA using additional, non-visual modalities (such as an audio CAPTCHA) is necessary.

Additionally, the alt attribute value should be updated to describe the purpose of the CAPTCHA or the element should be masked entirely from assistive technologies while ensuring the form can still be completed.

CAPTCHA elements are currently in use on the following page:

- Gift Cards – Check Balance (Desktop/Mobile)

All user interface controls must have a name that describes its purpose and that name must be able to be programmatically determined (WCAG Guidelines 1.1.1 and 4.1.2)

For those users who access the site via assistive technologies, all user interface controls and form elements must be clearly described as their purpose may not be clear from a non-visual perspective.

For <form> elements, this can be accomplished by providing a <label> element which describes the control and connecting the pair by using identical values for the for attribute on the <label> element and the id attribute on the control. This has the added bonus of providing another clickable or tappable region for which users can trigger to activate the control.

- Global (Desktop) – the text field in the global header (id = “searchTextField”) is not properly associated with its label text. The “Start an Order” text wrapper should be changed from a <p> to a <label> and given a for attribute value of “searchTextField”.
- Public – Feedback (Desktop/Mobile) – the three listboxes (month, day, and year) for the Date Visited group do not have individual labels associated with them. The month listbox has a machine-generated id attribute

("ctl00_ctl00_ContentPlaceholder1_companyContentPlaceholder_ddMonth"), the day listbox has an id attribute value of "ddDay", and the year listbox has an id attribute value of "ddYear".

- Public – Find a JJ's (Desktop/Mobile) – the State listbox has a machine-generated id attribute ("ctl00_ContentPlaceholder1_ucfindAJJs_drpdwnState"). This should be updated to "stateSelect" to pair it with its associated <label> text.
- Public – Find a JJ's (Desktop/Mobile) – the City listbox does not have an id attribute. This attribute should be added and set to "citySelect" to pair it with its associated <label> text.
- Public – Jobs (Desktop/Mobile) – the State listbox has a machine-generated id attribute ("ctl00_ctl00_ContentPlaceholder1_companyContentPlaceholder_ucjobs_ddStates"). This should be updated to "state" to pair it with its associated <label> text.
- Public – Jobs (Desktop/Mobile) – the Job Type listbox does not have an id attribute. This attribute should be added and set to "jobType" to pair it with its associated <label> text.
- Public – Menu (Desktop/Mobile) – the listbox which allows the user to choose a type of bread does not have an id attribute. Additionally, the "Choose Your Bread" header wrapper should be changed from an <h3> to a <label> and given a for attribute value that matches the id attribute value for the listbox.
- Public – Menu (Desktop/Mobile) – the "Cut in half" checkbox is not properly associated with its label as the checkbox is not a proper form control and is lacking an id attribute value. Additionally, the <label> element associated with it is lacking a for attribute.
- Public – Menu (Desktop/Mobile) – each instance of "XTRA", "REG", "EX", and "NO" is lacking an id attribute and is not associated with its label as the labels are marked up using <a> tags instead of <label> tags and therefore do not have for attributes.
- Blog – Landing (Desktop/Mobile) – the search input field does not have an id attribute nor is there a provided label to link with a for attribute value.
- Blog – Search Results (Desktop/Mobile) – the search input field does not have an id attribute nor is there a provided label to link with a for attribute value.
- Blog – Archive Landing (Desktop/Mobile) – the year listbox has an id attribute value of "archiveYearSelect" but no label text is provided to associate via a for attribute.
- Gift Cards – Check Balance (Desktop/Mobile) – the text field for the gift card number suffix does not have a label associated with it. The text field has an id attribute value of "BusinessId".
- Gift Cards – Reload Card (Desktop/Mobile) – the text field for specifying a non-full amount does not have a label associated with it. The text field has an id attribute value of "ReloadAmount".
- Gift Cards – Billing Info (Desktop/Mobile) – the State/Province listbox does not have a label associated with it. The listbox has an id attribute value of "State".

There are two labels that appear to be associated with this control, but neither of them have matching for attribute values:

- the “State” label has a for attribute value of “stateLabel”
- the “Province” label has a for attribute value of “provinceLabel”
- Gift Cards – Billing Info (Desktop/Mobile) – the Country listbox does not have a label associated with it. The listbox has an id attribute value of “Country” but the nearby label has a for attribute value of “BillingAddress_Country”
- Gift Cards – Billing Info (Desktop/Mobile) – the Expiration Date year listbox does not have a label associated with it. The listbox has an id attribute value of “PaymentInfo_ExpirationYear”
- Store – Register/Login (Desktop/Mobile) – the checkbox that keeps a user logged in has a machine-generated id attribute (“ctl00_PageContent_AspDNSFLoginControl_ctrlLogin_RememberMe”). The label associated with this checkbox should be given a for attribute value that matches the control.
- Store – Create Account (Desktop/Mobile) – the checkbox that duplicates the shipping address as the billing address has a machine-generated id attribute (“ctl00_PageContent_chkShppingEqBilling”). The label associated with this checkbox should be given a for attribute value that matches the control.
- Store – Item Detail (Desktop/Mobile) – the quantity input on each item detail page has a unique id attribute value but the related label element does not have a for attribute. These attribute values should match.
- Store – Item Detail (Desktop/Mobile) – the size listbox does not have a label associated with it nor is there one present on the page to associate with.
- Store – Shopping Cart/Wish List (Desktop/Mobile) – the individual quantity inputs do not have labels associated with them. A potential solution to this is adding a title attribute with a value of “Quantity” to each input.
- Store – Shipping Address/Billing Address (Desktop/Mobile) – the “Use Saved”/“Create New” toggle is actually a pair of radio buttons which do not have their labels associated with them as neither the controls nor labels have the proper associating attributes and values.
- Store – Payment Info (Desktop/Mobile) – the Expiration Date year listbox does not have a label associated with it. The listbox has a machine-generated id attribute (“ctl00_PageContent_ctrlCreditCardPanel_ddICCExpYr”).

The <label> element should only be used to associate text with form controls (WCAG Guideline 1.3.1)

The <label> element represents descriptive text for a user interface control.

- Public – Find a JJ’s (Desktop/Mobile) – the “OR” text that is between the listboxes and the Zip Code is marked up with a <label> tag but is not associated with any particular form element via a for attribute.

- Gift Cards – Buy a Card (Desktop/Mobile) – the “SHIPPING OPTIONS” header text is marked up with a <label> tag but is not associated with any particular form element via a for attribute.
- Gift Cards – Buy a Card (Desktop/Mobile) – the “SHIPPING AND HANDLING SCALE:” header is marked up with a <label> tag but is not associated with any particular form element via a for attribute.
- Gift Cards – Reload Card (Desktop/Mobile) – the “CHOOSE AMOUNT” header is marked up with a <label> tag and is associated with a non-existent form element via a for attribute value of “cardAmount.”

Heading structure should convey relationships between content (WCAG Guideline 1.3.1)

To properly communicate the structure of the document to assistive technologies, heading tags (<h1>, <h2>, <h3>, and <h4> are used on the site) should be properly nested.

- Public – Jobs (Desktop/Mobile) – multiple <h4> tags (one for each job posting) are nested under an <h2> tag with no intermediate <h3> tags
- Public – Menu (Desktop/Mobile) – multiple <h2> tags (one for each sandwich category) are not properly nested under an <h1> tag
- Public – Menu (Desktop/Mobile) – multiple <h4> tags (one for each sandwich type) are nested under <h2> tags with no intermediate <h3> tags
- Public – Privacy Policy (Desktop/Mobile) – thirteen <h3> tags are nested under an <h1> tag with no intermediate <h2> tags
- Public – Real Estate (Desktop/Mobile) – five <h3> tags are nested under an <h1> tag with no intermediate <h2> tags
- Public – Terms & Conditions (Desktop/Mobile) – twenty <h3> tags are nested under an <h1> tag with no intermediate <h2> tags
- Public – Who We Let Do This Deal (Desktop/Mobile) – the “Think you have what it takes?” heading is marked up using an <h4> tag which is nested under an <h2> tag with no intermediate <h3> tag
- Public – Why Franchise? (Desktop/Mobile) – two <h3> tags are nested under an <h1> tag with no intermediate <h2> tag
- Blog – Landing and Post Detail (Desktop/Mobile) – the “Archives” heading is marked up as an <h4> tag which is nested under an <h2> tag with no intermediate <h3> tag
- Blog – Search Results (Desktop/Mobile) – multiple <h3> tags (one for each blog title) are nested under a <h1> tag with no intermediate <h2> tags
- Gift Cards – Landing Page (Desktop/Mobile) – two <h4> tags are nested under an <h1> tag with no intermediate <h2> or <h3> tags
- Gift Cards – Buy a Card (Desktop/Mobile) – two <h3> tags are nested under an <h1> tag with no intermediate <h2> tags

- Gift Cards – Billing Info (Desktop/Mobile) – two <h3> tags are nested under an <h1> tag with no intermediate <h2> tags
- Gift Cards – Balance Details (Desktop/Mobile) – four <h4> tags are nested under an <h1> tag with no intermediate <h2> or <h3> tags
- Store – Create Account (Desktop/Mobile) – three <h3> tags are nested under an <h1> tag with no intermediate <h2> tag.
- Store – My Account (Desktop/Mobile) – three <h3> tags are nested under an <h1> tag with no intermediate <h2> tag.
- Store – Shipping & Returns (Desktop/Mobile) – two <h3> tags are nested under an <h1> tag with no intermediate <h2> tag.
- Store – Privacy & Security (Desktop/Mobile) – nine <h4> tags are nested under an <h1> tag with no intermediate <h2> or <h3> tags.
- Store – Shipping Address (Desktop/Mobile) – one <h3> tag is nested under an <h1> tag with no intermediate <h2> tag
- Store – Billing Address (Desktop/Mobile) – one <h3> tag is nested under an <h1> tag with no intermediate <h2> tag

The relationship between data cells and their associated row and column headers must be explicitly defined (WCAG Guideline 1.3.1)

For those users navigating a data table in a non-visual way, data must be properly linked with its row and column headings.

- Public – Initial Investment (Desktop/Mobile) – while the data table on this page appears to be a simple table, it is actually comprised of multiple tables to ensure proper display on mobile. Accordingly, to properly associate table headers with the table data:
 - Each <th> tag must be given a unique id attribute value
 - Each <td> tag must be given a headers attribute which is comprised of a space-separated list of the <th> tags associated with its content.

Lists of items should be marked up using the appropriate list markup tags (WCAG Guideline 1.3.1)

When encountering a series of grouped items, assistive technologies announce the list as such as well as the number of items contained within. By ensuring all lists of items are marked up as lists, users are able to quickly bypass the list.

- Public – Homepage (Desktop) – the various images which makeup the Instagram panel should ideally be marked up as an unordered list.
- Public – Jobs (Desktop/Mobile) – the job search results should ideally be marked up as an unordered list.

- Public – Menu (Desktop/Mobile) – the menu of items itself should ideally be marked up as an ordered list which contains smaller lists, some ordered and some unordered.

The visual presentation of text and images of text must have a contrast ratio of at least 4.5:1 (WCAG Guideline 1.4.3)

Enough contrast must be provided between text and its background to ensure that readers with moderately low vision (i.e. those who may have a color deficiency but do not use contrast-enhancing assistive technologies) are able to read it.

Text that is of large-scale must have a contrast ratio of at least 3:1.

- Global (Desktop) – the five footer links (Privacy Policy, Terms & Conditions, Jobs, Feedback, Owners Site) have a contrast ratio of 3.11:1 – #656565 foreground on #ffffff background.
- Public – Homepage (Desktop) – the theater control stars have a contrast ratio of 1.69:1 – #c8c7c7 foreground on #ffffff background.
- Public – Homepage (Desktop) – all blue links found in the Twitter widget (Twitter username, links and hashtags found in the tweet, and the date of the tweet) have a contrast ratio of 4.25:1 – #1083b5 foreground on #ffffff background.
- Public – Homepage (Desktop) – all gray links found in the Twitter widget (Reply link, Retweet link, Favorite link) have a contrast ratio of 2.12:1 – #b2b2b2 foreground on #ffffff background.
- Public – Find a JJ's (Desktop/Mobile) – the “OR” text that is between the listboxes and the Zip Code input has a contrast ratio of 4.29:1 – #7a7a7a foreground on #ffffff background.
- Public – Franchise pages (Desktop/Mobile) – the active page link in the left navigation has a contrast ratio of 4.29:1 – #7a7a7a foreground on #ffffff background.
- Public – Sustainability (Desktop) – the green Sustainability link in the global navigation has a contrast ratio of 3.43:1 – #539c06 foreground on #ffffff background.
- Gift Cards – Global (Desktop/Mobile) – the circular numbers that indicate next steps in a transaction have a contrast ratio of 4.29:1 – #ffffff foreground on #7a7a7a background.
- Gift Cards – Global (Desktop/Mobile) – the circular numbers that indicate previous steps in a transaction have a contrast ratio of 1.64:1 – #ffffff foreground on #cacaca background.
- Gift Cards – Blog Post (Desktop/Mobile) – the posting time at the bottom of each blog post has a contrast ratio of 3.11:1 – #929292 foreground on #ffffff background
- Store – Global (Desktop/Mobile) – the active page link in the left navigation has a contrast ratio of 4.29:1 – #7a7a7a foreground on #ffffff background.

- Store – Register/Login (Desktop/Mobile) – the “OR” text that separates the register section from the login section has a contrast ratio of 4.29:1 – #7a7a7a foreground on #ffffff background.
- Store – Product Detail (Desktop/Mobile) – the magnifying glass underneath the product image has a contrast ratio of 2.43:1 – #a6a6a6 foreground on #ffffff background
- Store – Global (Desktop/Mobile) – the circular numbers that indicate next steps in a transaction have a contrast ratio of 4.29:1 – #ffffff foreground on #7a7a7a background.
- Store – Global (Desktop/Mobile) – the circular numbers that indicate previous steps in a transaction have a contrast ratio of 1.64:1 – #ffffff foreground on #cacaca background.

All functionality must be operable through a keyboard-only interface (WCAG Guideline 2.1.1)

When content can be operated solely through a keyboard interface, it is operable by people with no vision, those people who must use alternate keyboards or devices that act as keyboard emulators, as well as users who prefer to use the keyboard to interact with page functionality.

Elements on the page are keyboard accessible when either (a) they are correctly marked up with <a> or <button> tags (as appropriate) or (b) are given a tabindex attribute value of “0” (zero).

- Public – Homepage (Desktop) – the theater control stars are not keyboard accessible – they are marked up using <a> tags but they are lacking the href attribute
- Public – Find a JJ's (Desktop/Mobile) – the City listbox is not keyboard accessible – it is assigned a tabindex attribute with a value of “-1” which makes it only accessible via scripting languages such as JavaScript.
- Public – Jobs (Desktop/Mobile) – the Job Type listbox is not keyboard accessible – it is assigned a tabindex attribute with a value of “-1” which makes it only accessible via scripting languages such as JavaScript.
- Public – Menu (Desktop/Mobile) – the contents of the JJ's Menu tab are not keyboard accessible as the category headers are not marked up using keyboard accessible tags.

Blocks of content that are repeated on multiple pages should be able to be easily bypassed (WCAG Guideline 2.4.1)

For those users who navigate through pages sequentially, direct access to the primary content of the page should be provided to bypass any repeated content.

- Global (Desktop/Mobile) – a “Skip Navigation” link should be provided near the top of the page to allow the user to quickly jump past the header section present on every page.

Triggering a control should not reset the current focus to its initial state (WCAG Guideline 2.4.3)

When navigating through a page, some elements may be removed when they are triggered. When this occurs, the user's keyboard focus should not revert back to the start of the page.

- Public – Find a JJ's (Desktop) – when paging through search results, triggering the “Next 5” link to load the last batch of results or the “Previous 5” link to load the first batch of results causes that link to disappear, resetting focus to its initial state
- Public – Menu (Desktop) – triggering either the “Customize Sandwich” or “Show Menu” button causes the other to be swapped in and focus reset to its initial state. In this instance, focus should remain on the replacement button.

The purpose of each link should be clearly defined by the link text alone (WCAG Guideline 2.4.4)

All links should have context to define their purpose and destination. This includes assigning alt text values to any linked images.

- Store – Global (Mobile) – the cart icon located in the header of the page does not have any link context – it is an anchor with no inner text.
- Store – Global (Desktop/Mobile) – the “x” button does not have any link context – it is an anchor with no inner text. These circular “x” controls are found on the following pages:
 - Shopping Cart
 - Wish List
 - Saved Addresses

Links that point to targets within the page should point to targets that exist (WCAG Guideline 2.4.4)

It is a common (though not best) practice that, when using anchors (<a> tags) as triggers for on-page functionality, the href attribute is given a value of “#”. Anchors used in this manner should not be given extended fragment href attribute values as assistive technology may still report the link target to the user, despite the target not existing.

Ideally, these anchors would be replaced with the more appropriate.

In each of the following cases, the noted href attribute value points to a target on the page that does not exist.

- Global (Desktop/Mobile) – the Delivery button has an href attribute value of “#delivery”
- Global (Desktop/Mobile) – the Pickup button has an href attribute value of “#pickup”
- Public – Homepage (Desktop) – the “CLOSE” button at the top of the Instagram panel has an href attribute value of “#instClose”
- Public – Homepage (Desktop) – the “x” (close) button within the Instagram panel modal has an href attribute value of “#fullInstCloseBtn”
- Public – Homepage (Desktop) – the “<” (previous) button within the Instagram panel modal has an href attribute value of “#fullInstLeftBtn”
- Public – Homepage (Desktop) – the “>” (next) button within the Instagram panel modal has an href attribute value of “#fullInstRightBtn”
- Public – Homepage (Mobile) – the “Top” button has an href attribute value of “#top”
- Public – Menu (Desktop) – the “Customize Sandwich” button has an href attribute of “#customizeItem”
- Public – Menu (Desktop/Mobile) – the “Show Menu” button has an href attribute of “#showMenu”
- Public – Menu (Desktop/Mobile) – the “Nutrition Menu” button has an href attribute of “#expandNutrition”
- Public – Menu (Mobile) – the “Customize Sandwich” button has an href attribute of “#customizeMobile”

All focusable elements must have clear focus states (WCAG Guideline 2.4.7)

Maintaining a clear focus state on all focusable elements allows the user to visually determine what element will receive keyboard input.

Visible focus is often controlled using the :focus state of keyboard-accessible elements as well as the outline CSS property.

- Global (Desktop) – the Delivery and Pickup buttons in the global header do not have defined focus states.
- Global (Desktop) – all red buttons (i.e. those with a class attribute value of “redBtn”) do not have a clear focus state as the CSS property outline is set to a value of “initial”, which uses the text color of the element. As the text color in these buttons is white, the focus outline is indiscernible from the background of the page.
- Public – Homepage (Desktop) – the images in the theater carousel are clickable but have had their focus states programmatically disabled by setting the CSS property outline to a value of “none”.
- Public – Find a JJ’s (Desktop) – the search results do not have a focus state.

Input errors must be communicated to the user in an identifiable manner (WCAG Guideline 4.1.1)

While input validation messages are presented visually and focus is assigned to the first invalid input in a form on submission, a user of assistive technology may not understand why focus has changed after triggering form submission.

To account for this, validation messages should be given a role attribute with a value of "alert" – this will announce the message to the user when it is present on the page.

The following pages were found to use inline form validation messaging:

- Public – Feedback (Desktop/Mobile)
- Public – Find a JJ's (Desktop/Mobile)
- Public – Franchise Application (Desktop/Mobile)
- Gift Cards – Buy a Card (Desktop/Mobile)
- Gift Cards – Billing Info (Desktop/Mobile)
- Gift Cards – Check Balance (Desktop/Mobile)
- Gift Cards – Reload Card (Desktop/Mobile)

All id attribute values on a page must be unique (WCAG Guideline 4.1.1)

Per HTML standards, no two elements on a page may have the same value for their id attribute. As assistive technologies often rely on these values, duplicates can lead to confusion, either for the technology itself or via its presentation to the user.

Duplicate id attribute values may also trigger some browsers to switch to "Quirks Mode" as the page is no longer valid HTML.

Most issues found violating this criterion are related to the "SelectBoxIt" plugin which is being used for the custom listboxes throughout the site.

- Public – Find a JJ's (Desktop/Mobile) – the original City listbox is lacking an id attribute which has resulted in five instances of an id attribute value of "" (null).
- Public – Find a JJ's (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with "0" (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Public – Jobs (Desktop/Mobile) – the original "Job Type" listbox is lacking an id attribute which has resulted in multiple instances of an id attribute value of "" (null).
- Public – Jobs (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with "0" (zero). Accordingly,

when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.

- Public – Franchise Application (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Public – Feedback (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Public – Menu (Desktop/Mobile) – the original Choose Your Bread and Vito Meat listboxes are lacking an id attribute which has resulted in ten instances of an id attribute value of “” (null).
- Public – Menu (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Blog – Landing Page (Desktop/Mobile) – each blog post has a <div> element with an id attribute value of “fb-root” for the Facebook “Like” control. When multiple blog posts appear on a single page, these id attributes are duplicated.
- Gift Cards – Buy a Card (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created. Additionally, the <label> elements in the shipping options section are also given numeric id attribute values based on the value attribute of their associated radio buttons.
- Gift Cards – Billing Info (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Store – Create Account (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Store – Shipping Address/Billing Address (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Store – Payment Info (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning with “0” (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.
- Store – Add New Address/Edit Address (Desktop/Mobile) – the SelectBoxIt plugin gives each option in a listbox a sequential numeric id attribute beginning

with "0" (zero). Accordingly, when multiple listboxes appear on the same page, multiple duplicate id attribute values are created.

Anchor (<a>) elements should not be used solely as targets for other anchor elements (WCAG Guideline 4.1.2)

It is possible to link to sections within a page by assigning an id attribute to an element and then setting up a link to that section by having an anchor with an href attribute of a "#" (hash) followed by the value of said id attribute. These targets should be setup on existing, appropriate elements.

- Public – Sustainability (Desktop/Mobile) – an anchor with an id attribute value of "tabpanelco" is present. This anchor has no clear purpose as its parent element is hidden on both desktop and mobile views.
- Public – Terms & Conditions (Desktop/Mobile) – an anchor with an id attribute value of "online_policy" is present. This attribute and value should instead be placed on its parent <h2> tag.
- Public – Terms & Conditions (Desktop/Mobile) – an anchor with an id attribute value of "online_copyright" is present. This attribute and value should instead be placed on its parent <h2> tag.
- Blog – Post Detail (Desktop/Mobile) – an anchor with an id attribute value of "comments" is present. This attribute and value should instead be placed on a parent container.

When changes are made to the state of a user control, the user should be notified of the change (WCAG Guideline 4.1.2)

For custom user controls, the current state and any changes to said states should be available to the user, both visually and via any assistive technologies.

- Public – Menu (Desktop/Mobile) – the collapsed or expanded state of each menu category is visually indicated by the downward or upward pointing caret but is not indicated to assistive technologies. To accomplish this, the category headers require aria-expanded attributes with values of "true" or "false" as appropriate.
- Gift Cards – Balance Details (Desktop/Mobile) – the collapsed or expanded state of the "Transaction Details" section is visually indicated by the downward or upward pointing caret but is not indicated to assistive technologies. To accomplish this, the control requires an aria-expanded attribute with values of "true" or "false" as appropriate.

All listbox elements should have a value initially selected on page load (WCAG Guideline 4.1.2)

This is accomplished by assigning the default or initial value a selected attribute with a value of "selected".

- Global (Desktop/Mobile) – this applies to all <select> elements used site-wide.

Additional Recommendations

While not requirements under the WCAG, the following suggestions will improve the user experience for users of assistive technologies.

Characters which are used for visually decorative purposes should be hidden from assistive technologies

As screen readers read the contents of the page, symbols created with custom web fonts may not be read aloud as expected or intended.

To combat this, the `` element wrapped around the symbol should be given an `aria-hidden` attribute with a value of `"true"`.

The list below is not comprehensive, but instead lists one occurrence of each symbol. All elements on the site which have a class attribute with a value of `"iconJJCommons"` should be checked.

- Global (Desktop) – the star before “Login”, “Delivery”, and “Pickup” in the header will be read aloud by some screen readers as “star”.
- Global (Desktop) – the leaf which follows “Sustainability” in the global navigation will be read aloud by some screen readers as “and”.
- Public – Menu (Desktop/Mobile) – the downward pointing caret at the end of each menu category will be read aloud by some screen readers as “v”.
- Public – Menu (Desktop/Mobile) – the rightward pointing caret at the end of each sandwich line in the menu will be read aloud by some screen readers as “greater” or “greater than”.
- Public – Menu (Desktop/Mobile) – the caret which appears as part of the custom SelectBoxIt dropdown button will be read aloud by some screen readers as “bar”.
- Store – Global (Desktop/Mobile) – the shopping cart which appears after the word “Cart” near the top of the page will be read aloud by some screen readers as “number”.

HTML5 sectioning elements should have appropriate role attributes

Much like older browsers require special attention to properly display the new HTML5 sectioning elements, older screen readers also require help. These elements and the appropriate role attributes are as follows:

- `<nav>` elements should have a role attribute value of `"navigation"`.

Custom versions of default web components should expose appropriate roles, states, and properties to accessibility APIs as well as mimic default keyboard controls.

When replacing default controls with custom implementations, care should be taken to ensure that the default controls' usability is not sacrificed in order to create a more pleasing visual display.

Two custom control implementations are used on the site – SelectBoxIt (for listboxes) and Kalypto (for checkboxes and radio buttons).

- SelectBoxIt – the version of SelectBoxIt used on the site is 3.5.0 with heavy customization. This plugin has support for both role and various aria-* attributes, but this support was disabled in order to support IE7. The custom version of this plugin used on the site supports some but not all default keyboard controls.
- Kalypto – the version of Kalypto used on the site is 0.2.2. This plugin does not have support for either role or the various aria-* attributes. The plugin supports some but not all default keyboard controls.

Details on appropriate role and aria-* attribute values for these controls can be found on the W3C's site:

- Listbox – <http://www.w3.org/WAI/PF/aria-practices/#Listbox>
- Checkbox – <http://www.w3.org/WAI/PF/aria-practices/#checkbox>
- Radio Button – <http://www.w3.org/WAI/PF/aria-practices/#radiobutton>

To ensure complete accessibility, both of these plugins should be updated to newer versions which utilize role and aria-* attributes (if available), replaced with plugins that do so, or removed in favor of the default controls.