



Aetna’s Digital Accessibility Center

# IMQ Collection thru FormFire (Asset #1075 )

# Spot Check Review

Version 4

## Executive Overview

### Review details

Date completed: April 22, 2020

Reviewer: Pallavi Vardhamane, Level Access Inc.

Date completed: January 23rd, 2020

Reviewer: Shruthi Gopalakrishna, Level Access Inc.

Date completed: November 20, 2019

Reviewer: Pallavi Vardhamane, Level Access Inc.

Date completed: August 27, 2019

Reviewer: Nishant Bochare, Level Access Inc.

### Project scale

Medium

### Project complexity

Medium

### Project iteration

Fourth validation review

### Project accessibility rating

6 out of 10 (Many blocker issues, Many important issue)

### Scope covered

FormFire – Web application

### Summary of findings

Blockers:

* **[Partially Fixed]** Input fields does not have valid label
* **[Partially Fixed]** The image buttons do not have valid label
* [Fixed] The Radio buttons available on the Condition selector page are labeled with one common id
* [Fixed] Elements are not keyboard accessible.
* [Fixed] Focus does not shift to the opened slide out content and does not provide any state about the open slide
* **[Partially Fixed]** There is no valid error identification
* [Fixed] Radio buttons do not have proper role

Important:

* [Fixed] On the Signup page, tooltip information does not have meaningful label is not keyboard accessible
* **[Partially Fixed]** Color contrast issue
* [Fixed] The height input fields are not grouped under same fieldset
* [Fixed] Keyboard focus is not indicated visually
* [Fixed] Heading structure on the personal information is not hierarchical
* [Fixed] No Main Landmark found on the page and multiple nav regions are available

## Introduction

The DAC has conducted a review of Formfire website by an accessibility expert. Please note that this may have been a limited review and/or a point-in-time assessment conducted to get a high-level understanding of the asset’s conformity to WCAG v2 AA accessibility standard. The review provides insights into issues and may provide suggestions for remediation. We assume that the asset team will use recommended accessibility testing methods including automated, keyboard navigation, and screen reader tools as part of their remediation efforts. After the asset team has completed remediation the DAC many conduct a follow-up validation review and provide a rating.

**Note:** a minimum score of **8** out of 10 is required to be considered highly accessible or passing. (1=lowest, 10=highest)

## Scope

Credentials:

<https://www.formfire.com/interview/Login/#/>  
Username: AetnaEE1  
PW: FormFire1

The following pages or screens were included in this audit:

* Login pages
* Home page
* Mainly all 4 Forms pages/links.
* Settings pages

## Methodology and Tools

To conduct this audit, we used a combination of automated and manual testing. Manual testing of web pages requires physically navigating pages through normal use as well as using assistive technology.

* Audible testing was conducted via the screen reader, JAWS 2020(Desktop), Firefox.

## Goals and next steps

* This document is intended as a guide to show major blockers and not a comprehensive list of findings. A full audit may be necessary.
* Estimate the work effort to fully remediate and retest the asset to be WCAG 2.0 AA compliant.
* Develop and distribute a remediation plan to work towards WCAG 2.0 AA.

## Summary

The FormFire web application is partially compliant with the WCAG 2.0 A & AA standard. The application has been given a 6 out of 10 rating as there are still some blockers and important issues that will prevent assistive technology users from accessing the application.

The remaining blocker issues include invalid or no label for form fields, inaccessible error handling and lack of required error field indication for screen reader user. These issues make it difficult to complete the form. There is an unlabeled “Clear search” buttons on the “Resource center” widget.

There is also one important issue remaining that needs to be addressed. That is the text color contrast that does not meet the required contrast ratio.

## Findings

### Form fields does not have valid label

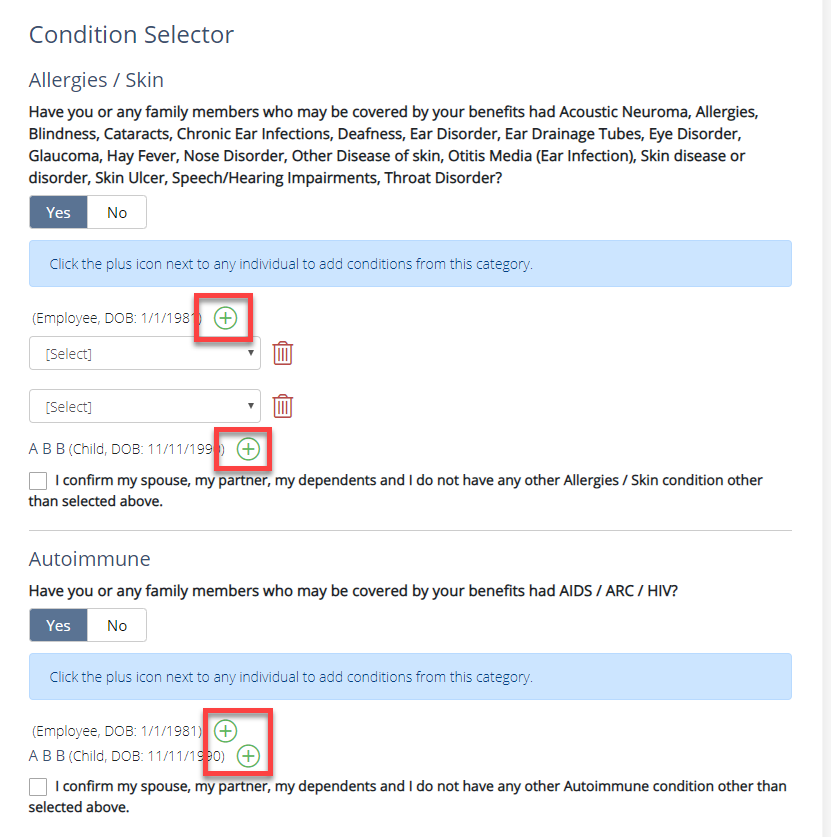
*[Severity Level: Blocker]*

**[Partially Fixed]-04/22/2020**

Note:

* ***(Fixed)*** Checkbox available on the table on “your account information” section: These checkboxes do not receive two-tab stops as tabindex=0 set on all label tags are removed.
* ***(Fixed)*** Checkbox available on the table on “your account information” section: The first check box in the table that selects all the rows in the table is labelled using aria-label attribute.
* ***(Not Fixed)*** The add condition (+) controls under the 3rd form section “Health section” has aria label attribute. However, previously these controls are implemented as links with aria-hidden attribute set on it. Now, these controls (+) are not links anymore. These add condition (+) controls are implemented as icon fonts with aria-label attribute assigned to it causing these controls are not keyboard accessible nor is rendered to screen reader users without proper role assigned to icon tags.

Ensure to implement these controls as link or button controls. Add aria-label attribute to anchor element and add aria-hidden attribute to icon image.



*Figure: Add condition controls are icon images with aria-label assigned without role=”img”.*

[Non-compliant code]

<i class="fal fa-plus-circle add" aria-label="Add Allergies / Skin Condition for A B B"></i>

[Compliant code example]

<a href="#/" aria-label="Add Allergies / Skin Condition for A B B"><i class="fal fa-plus-circle add" aria-hidden=”true”></i>

</a>

* ***(Fixed)*** The “Edit” controls under “Review and sign” page is associated using aria-describedby attribute to section headings giving additional context to screen reader users.

**[End of note]**

**[Partially Fixed]- 01/23/2020**

* Checkbox available on the table on “your account information” section: The labels are associated using for/id attribute and labels are now rendered to screen reader users. However, each checkbox receives two-tab stops which announces the control twice.

Developers must consider removing the tabindex=0 set on all label tags as all native form elements receive tab stop by default.

[Non compliant code sample]

<label tabindex="0">

<input type="checkbox" id="check-all" name="all">

<span class="overlay"></span>

</label>

<label tabindex="0">

<input aria-labelledby="IsSubscribedLabel0" class="check-email-preference checkboxPref" id="IsSubscribed0" name="FormModel.NotificationPreferences[0].IsSubscribed" type="checkbox" value="true"><input name="FormModel.NotificationPreferences[0].IsSubscribed" type="hidden" value="false" data-la-initdispnone="true">

<span class="overlay"> </span>

</label>

[Compliant code sample]

<label>

<input type="checkbox" id="check-all" name="all">

<span class="overlay"></span>

</label>

<label>

<input aria-labelledby="IsSubscribedLabel0" class="check-email-preference checkboxPref" id="IsSubscribed0" name="FormModel.NotificationPreferences[0].IsSubscribed" type="checkbox" value="true"><input name="FormModel.NotificationPreferences[0].IsSubscribed" type="hidden" value="false" data-la-initdispnone="true">

<span class="overlay"> </span>

</label>

* **[New issue]** Checkbox available on the table on “your account information” section: Also the first check box in the table that selects all the rows in the table is not labelled.

Developers must add a label to the control using aria-label property

[Non compliant code sample]

<label tabindex="0">

<input type="checkbox" id="check-all" name="all">

<span class="overlay"></span>

</label>

[Compliant code sample]

<label >

<input type="checkbox" aria-label=”**Select all Email Notification Preferences**” id="check-all" name="all">

<span class="overlay"></span>

</label>

* **[New issue]** The add condition (+) controls under the 3rd form section “Health section” has aria label attribute. However, these links are hidden using aria-hidden attribute. As a result, the labels are not being announced to the users of a screen reader when the element receives the focus.

Developers must consider removing aria-hidden attribute and make aria-label of the control more meaningful as there are multiple (+) add condition controls present on the same page.

[Non compliant code sample]

<a href="#/" class="fal fa-plus-circle add" aria-label="Add Condition" aria-hidden="true"></a>

<a href="#/" class="fal fa-plus-circle add" aria-label="Add Condition" aria-hidden="true"></a>

[Compliant code sample]

<a href="#/" class="fal fa-plus-circle add" aria-label="Add Allergies/Skin Condition for Employee"></a>

<a href="#/" class="fal fa-plus-circle add" aria-label="Add Allergies/Skin Condition for A B B child"></a>

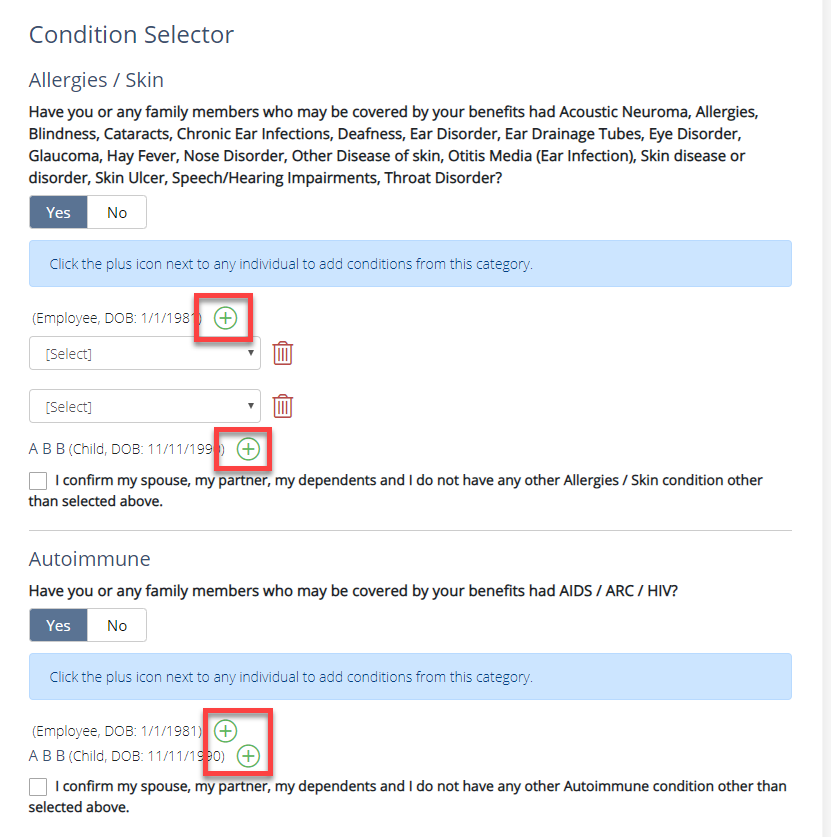


Figure: Duplicate add condition controls that are hidden using aria hidden property.

* **[New issue]** The “Edit” control under “Review and sign” page is duplicate. Each section i.e., Personal Info, Contact Info, Employer Info, Family & Household, etc., has “Edit” controls. However, users of screen readers cannot differentiate to which section each edit controls are associated to.

Developers must consider using aria-described by attribute to associate each section heading with the “Edit” controls so the users of screen readers can get the context of each control.

[Non compliant code sample]

<h3>Health Questions</h3>

<a class="btn btn-default" href="/employee/ConditionSelector?RedirectRouteName=ReviewSign.Summary.Index" tabindex="0">Edit</a>

[Compliant code sample]

<h3 id=”heading1”>Health Questions</h3>

<a class="btn btn-default" href="/employee/ConditionSelector?RedirectRouteName=ReviewSign.Summary.Index" tabindex="0" aria-describedby=” heading1”>Edit</a>

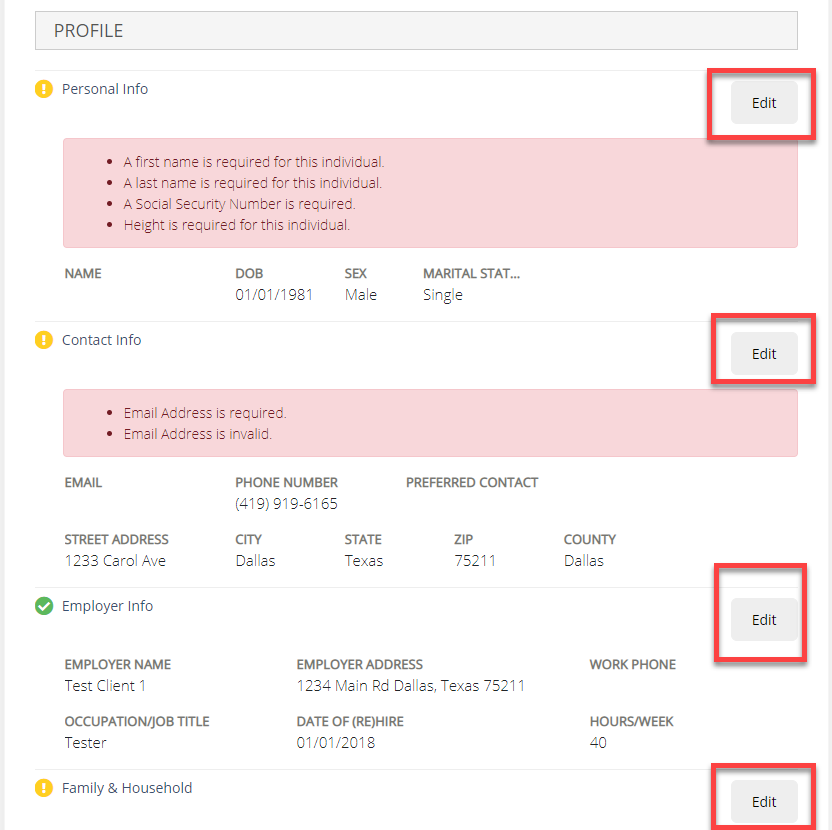


Figure: Duplicate Edit buttons in “Review and sign” page

**[End of note]**

[Partially Fixed]- 11/20/2019

Note: Now, most of the form fields across the website have valid labels. However, there are few instances with invalid labels present on the site as follows:

* (Fixed) Language button present in the footer section: Ensure to remove explicit label. Also, explicit labels shouldn’t be used with buttons as some screen readers will not announce an explicit label on buttons.

[Non-Compliant Code Example]

<button class="btn btn-language dropdown-toggle" type="button" id="lang-dropdown" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" tabindex="0">

<label for="lang-dropdown">Language</label>

<span class="caret"></span>

</button>

[Compliant Code Example]

<button class="btn btn-language dropdown-toggle" type="button" id="lang-dropdown" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" tabindex="0">

Language

<span class="caret"></span>

</button>

* (Fixed) Form fields from Profile page: All input fields are except height in inches have valid labels. This “in” field has label attribute with for value but label value is missing. If developers do not want to provide visible labels, they can add screen readers only label for this input field.
* (Partially fixed) Checkbox available on the table on “your account information” section: Though labels are associated using for/id attribute, labels are not rendering to screen reader users as a checkbox and label both are present in different <td> elements. Ensure to replace the <label> with <span> element. Ensure to provide ‘aria-labelledby’ attribute on input field referring to id of span.

[End of note]

Most form fields do not provide valid label. In the absence of valid label for form field, it is very difficult for users of assistive technology to enter or complete form information. Most of the form fields including combo boxes and check boxes have a visible label available however programmatically it is not associated with these elements.

For example, the “First Name”, “Last Name” and “Last 4 SSN” form field on the forget username page does not have valid label. Currently, a visual label is present for each input field, however the for and id does not match for the First Name, Last Name and Last 4 SSN field. The First name input field has given an id=”last-name” which affects both input fields. The for/id attribute available for SSN input field does not match.

Similarly, there are other instances for invalid labels for form fields including:

1. *(Fixed)* The “username input” field on the reset password page does not have valid label. The Username input field has given an id=”last-name” which affects both input field.(Figure 1)
2. *(Fixed)* The “Employer Code” input field available on the new user signup page does not have a valid label.
3. *(Partially Fixed)*The “language” combo box does not have valid label.
4. *(Fixed)*The “Reason for making a change” combobox on the home page does not have a valid label. On selecting different options from “Reason for making a change” combo box, the Date of Death, Loss/Gain Date, and Divorce Date input field appears which are not properly labeled. (Figure 2)
5. *(Partially Fixed)* All the following input fields on the FormFire Profile page- personal information do not valid labels:

First name, middle name, last name, Date of birth social security number, date married, height and weight.

The two comboboxes namely sex and marital status does not have a valid label. There is a mismatch with for/id on the height and weight input field. (Figure 3)

1. *(Fixed)*All the following input fields on the FormFire Profile page- Contact information does not valid labels:

Email address, phone number, city state and zip, preferred contact and country comboboxes does not have a valid label. (Figure 4)

1. *(Fixed)*All the following input fields on the add dependent slide out pane do not have valid labels:

First name, middle name, last name, Date of birth social security number, date married, height and weight.

The two comboboxes namely “sex” and “Relationship to You” does not have a valid label. There is a mismatch with for/id on the height and weight input field. Also, the input fields of dependent address are not programmatically labeled. (Figure 5)

Similar unlabeled input fields are found on all the slide out panes.

1. *(Fixed)*The “Reason for Waving” comboox on the coverage tab does not have valid label. (Figure 6)
2. *(Fixed)*The policy number, effective date and end date input fields does not have valid label.

The Medicare Number, Medicare reason, and Nature of disability input field and other input fields appear on selecting checkboxes on the add other coverage record pane and these input fields do not have valid labels.(Figure 7)

1. *(Partially Fixed)* Checkbox available on the table on “your account information” does not have valid label in association with “email me when” condition. Screen reader users need to check back and forth to validate the association of checkbox with the 2nd column text and it is overwhelming for screen reader user. (Figure 8)

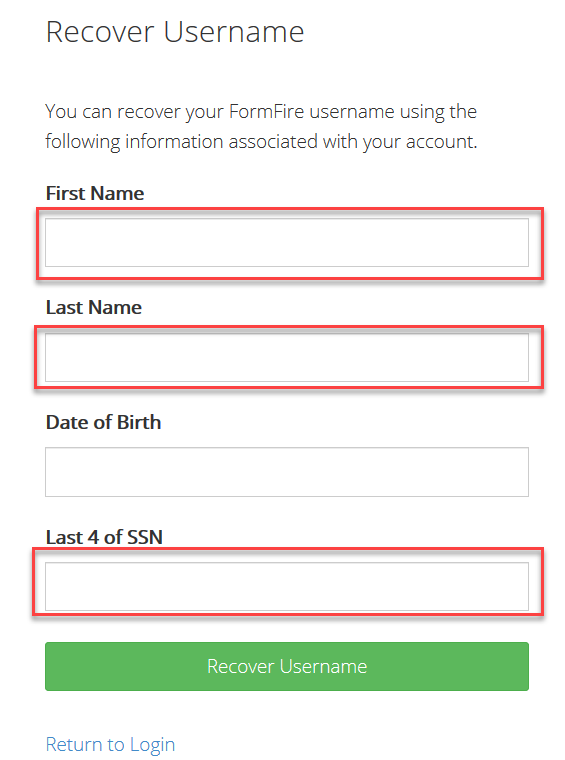


Figure 1: Input fields on the page Forgot username page does not have valid label.

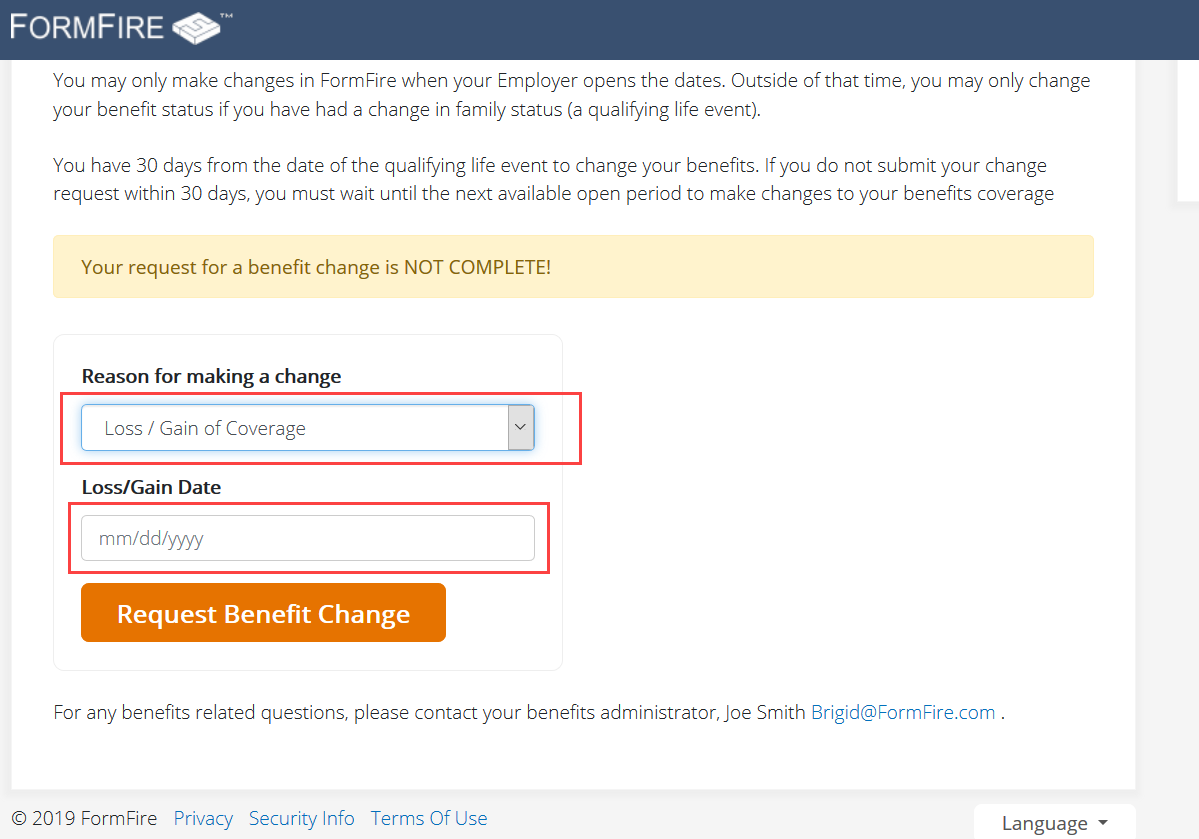


Figure : The “Reason for making a change” combobox on the home page does not have a valid label.

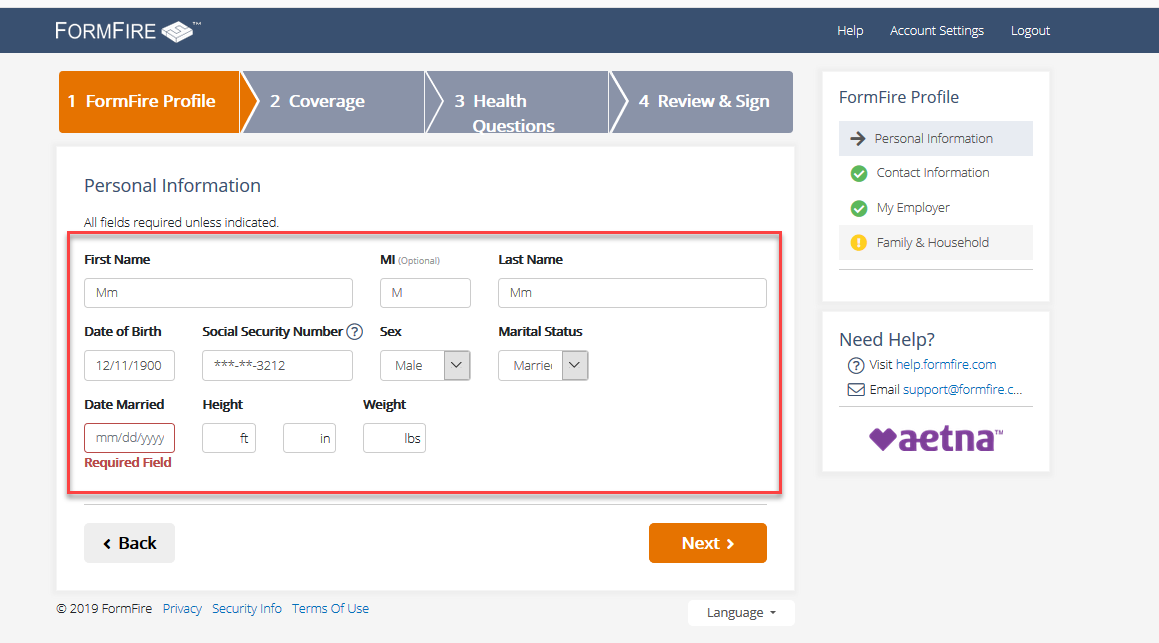


Figure : Input edit fields on the FormFire Profile page- personal information does not a valid label.

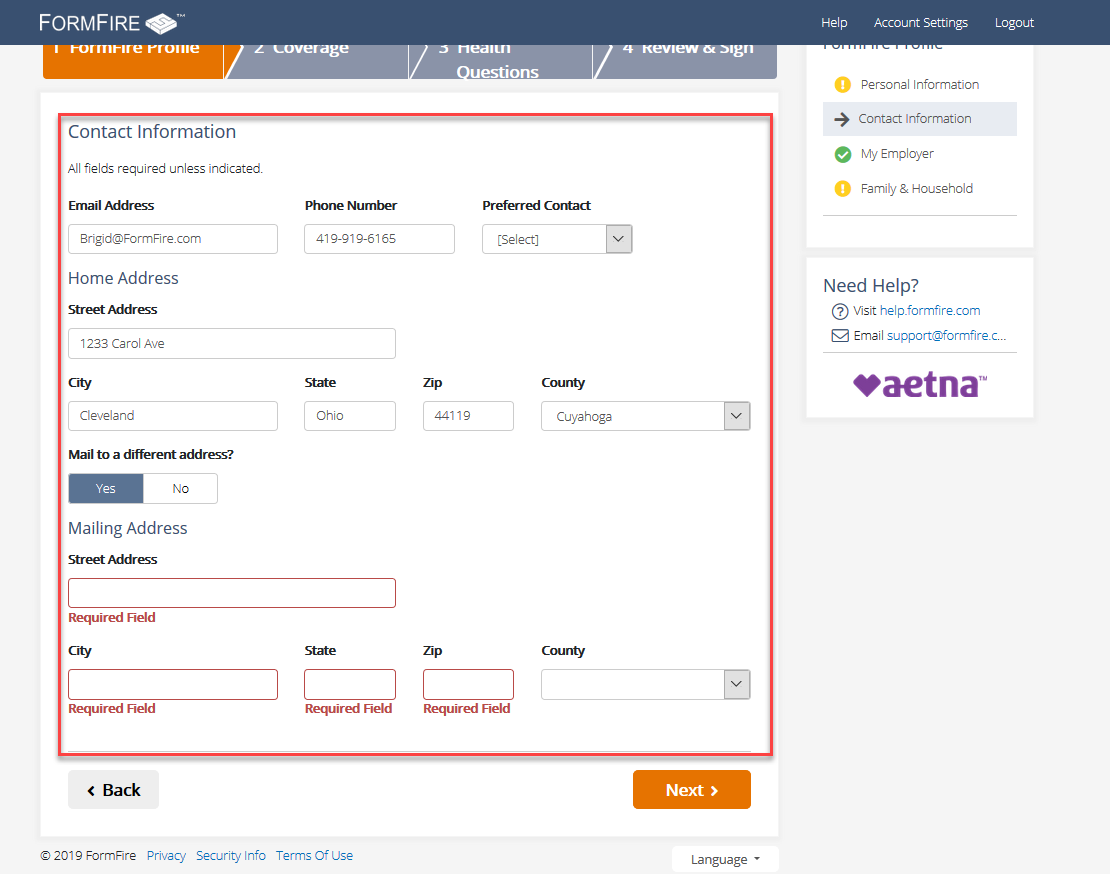


Figure : Input edit fields on the FormFire Profile page- contact information does not a valid label

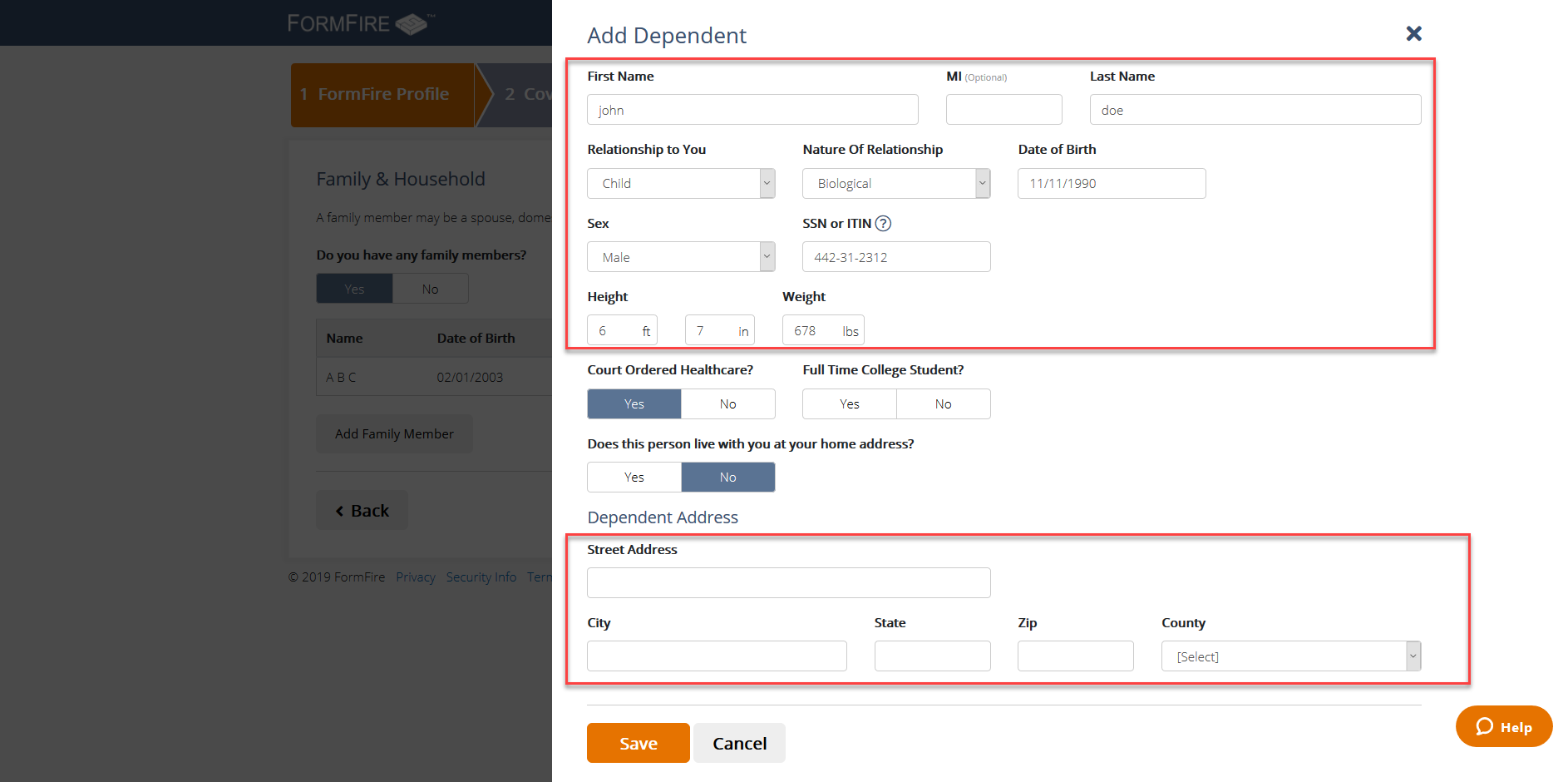


Figure : Input edit fields on the add dependent slide out pane does not a valid label

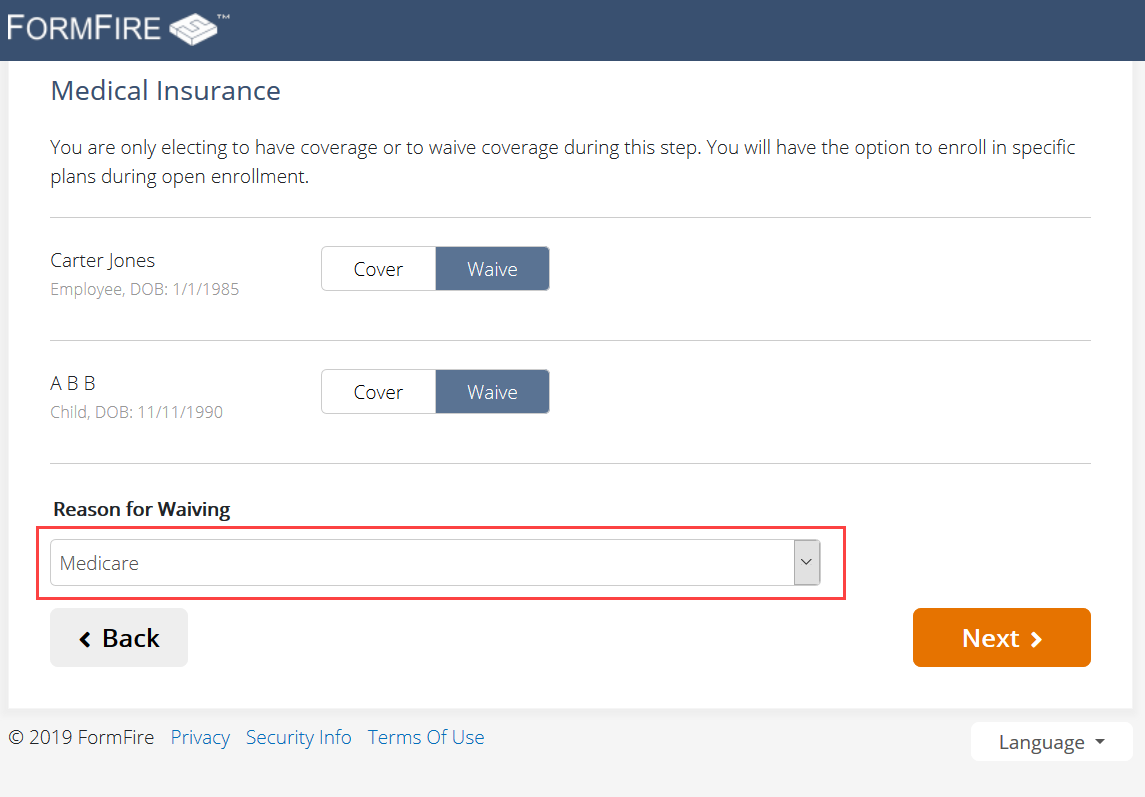


Figure : “Reason for Waiving” does not a valid label

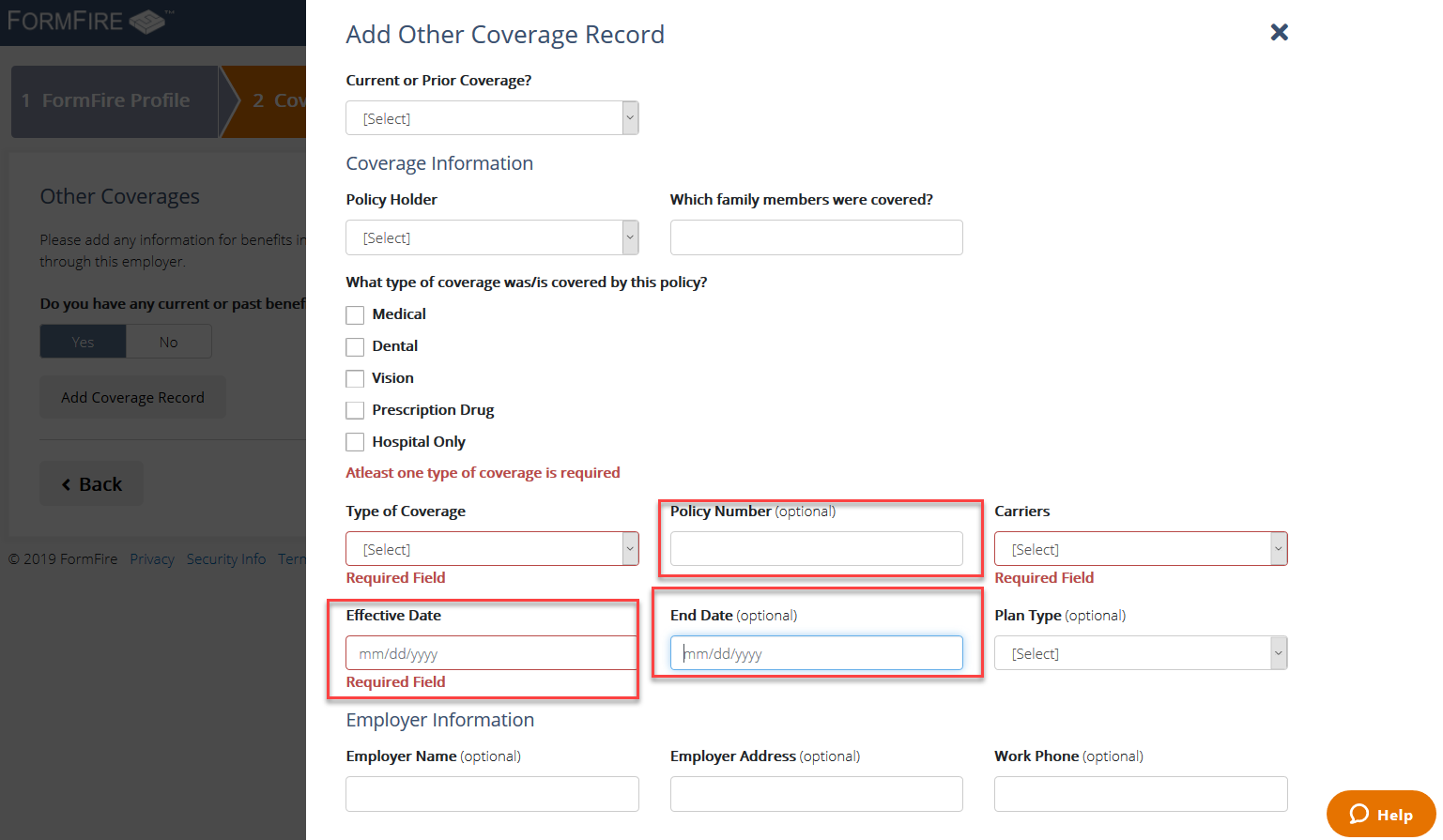


Figure : Input edit fields on the “add other coverage record” pane does not a valid label

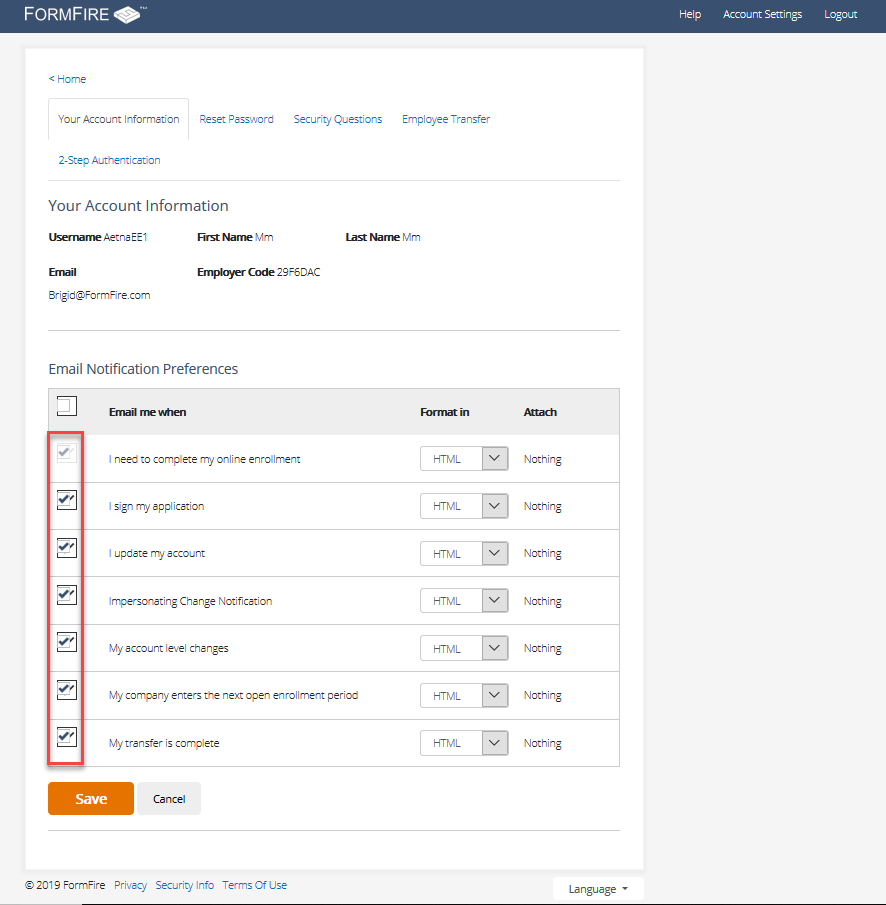


Figure : All the checkboxes does not have valid association with visual label.

[Recommendation]

Ensure form field are labeled using valid for/id association between the label and the input field. For example, for=” ” attribute available on the label tag should match with the associated input’s id=” ” tag.

*Compliant Example: (Forgot Username Page)*

<label for="first-name">First Name</label>

<input class="form-control format-name error" id="first-name" name="FormModel.FirstName" type="text" value="" data-validation="required" style="border-color: rgb(185, 74, 72);" aria-required="true" data-validation-error-msg="">

### The image buttons do not have valid label

*[Severity Level: Blocker]*

**[Partially Fixed]-04/22/2020**

Note: No change is observed from the last validation. Refer to notes below.

**[End of note]**

**[Partially Fixed]- 01/23/2020]**

This help widget component has now changed. However, in the new “Resource center” widget’, the clear search button in the search for articles edit field is not labeled. It is announced as “unlabeled 2 button” for the users of screen readers.

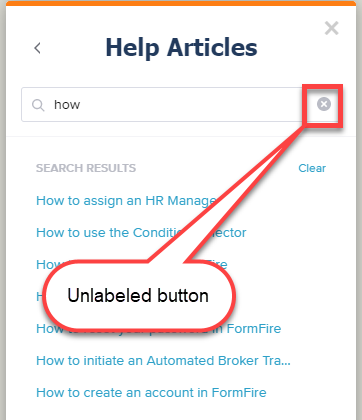


Figure: Unlabeled clear search image button

[Non compliant code sample]

<button class="\_pendo-ext-search-controller-clear\_"><svg width="14" height="14" viewBox="0 0 18 18" xmlns="http://www.w3.org/2000/svg"><path d="M9 0C4.03953488 0 0 4.03953488 0 9c0 4.9604651 4.03953488 9 9 9 4.9604651 0 9-4.0395349 9-9 0-4.96046512-4.0395349-9-9-9zm4.227907 11.7418605l-1.4860465 1.4860465-2.72093027-2.7209303L6.3 13.227907l-1.4860465-1.4860465 2.7209302-2.72093027L4.8139535 6.3 6.3 4.8139535l2.72093023 2.7209302 2.72093027-2.7209302L13.227907 6.3l-2.7209303 2.72093023 2.7209303 2.72093027z" fill="#BABCC5"></path></svg></button>

[Compliant code sample]

<button class="\_pendo-ext-search-controller-clear\_" **aria-label=”clear search”** ><svg aria-hidden=”true” width="14" height="14" viewBox="0 0 18 18" xmlns="http://www.w3.org/2000/svg"><path d="M9 0C4.03953488 0 0 4.03953488 0 9c0 4.9604651 4.03953488 9 9 9 4.9604651 0 9-4.0395349 9-9 0-4.96046512-4.0395349-9-9-9zm4.227907 11.7418605l-1.4860465 1.4860465-2.72093027-2.7209303L6.3 13.227907l-1.4860465-1.4860465 2.7209302-2.72093027L4.8139535 6.3 6.3 4.8139535l2.72093023 2.7209302 2.72093027-2.7209302L13.227907 6.3l-2.7209303 2.72093023 2.7209303 2.72093027z" fill="#BABCC5"></path></svg></button>

**[End of note]**

[Partially Fixed]-11/20/2019

Note: Most of the image buttons do have a valid label. The aria-label is used to provide an accessible label for add condition and delete condition image buttons. However, the search image in Help is not button anymore, it is just SVG image. Ensure to provide role=”button”, tabindex=”0” to div container so search button is in focus order. In addition, ensure to provide an accessible name this button by using aria-label attribute on div with a role of button and hide the SVG image using aria-hidden=”true” and focusable=”false” attributes.

[End of note]

[Issue]

There some image buttons on the website which does not have an accessible label. If the image buttons are unlabeled, screen reader user are unable to get the context of these buttons. Here are the two examples of similar instances.

1. Search button available on the chat box is unlabeled.

2. The “add condition” icon button and “delete” buttons on the condition selector page.

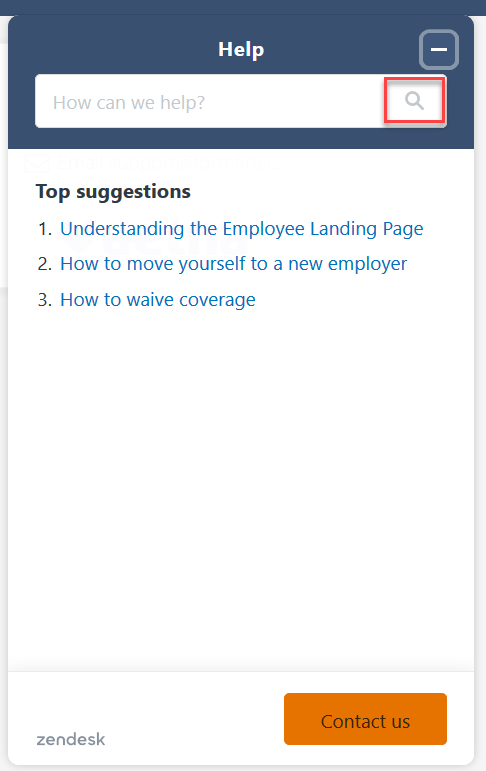


Figure 9: Unlabeled search button on the chat box

[Recommendation]

Developer should use an aria-label=”search” to provide an accessible label. Also, developers can use aria-hidden=”true” for the SVG to avoid unnecessary announcements of svg icons.

Developers can use an aria-label for add and delete button. The i element is used to express presentational information only. Instead, all presentational information should be expressed using CSS. Developers can use an anchor tag with standalone icon image and href attribute to make add and delete buttons keyboard accessible for example:

<a href="javascript:print()">

<img src="print.png" alt="Print this page">

</a>

### The Radio buttons available on the Condition selector page are labeled with one common id.

*[Severity Level: Blocker]*

**[Fixed]-11/20/2019**

[Issue]

After focusing on the “any” radio button available for conditions, the screen reader will announce only the first “Allergies/Skin” label. Only one common id is used to label all the radio buttons. Screen reader users can not complete the form because every button announces the first condition. There are duplicate values for ID attributes for every label/aria-labelledby.

[Code Reference]

<label **id=”condition-label”>**

<input id=”FormModel\_CategoryModel\_1\_\_ConditionCategoryHeading” name=”FormModel.CategoryModel[1].ConditionCategoryHeading” type=”hidden” value=”Have you, your spouse or any dependents had AIDS / ARC / HIV?” data-la-initdispnone=”true”>

Have you, your spouse or any dependents had AIDS / ARC / HIV? </label>

<label **id=”condition-label”>**

<input id=”FormModel\_CategoryModel\_0\_\_ConditionCategoryHeading” name=”FormModel.CategoryModel[0].ConditionCategoryHeading” type=”hidden” value=”Have you, your spouse or any dependents had Acoustic Neuroma, Allergies, Blindness, Cataracts, Chronic Ear Infections, Deafness,…

<div class=”btn-group btn-group-toggle category-yesno” data-toggle=”buttons” role=”radiogroup**” aria-labelledby=”condition-label”**>

<label class=”btn btn-default active” aria-checked=”true” tabindex=”0”>

…

[Recommendation]

There are duplicate values for ID attributes on the page. Each ID value must be unique per label/labelledby. This is especially important since assistive technology will use the ID attribute to correctly identify associated label sections of the page.

Developers must use a unique “id“ to for a label (question) and use the same id value for the aria-labelledby which link the question with the (Yes/No) radio buttons. Also make sure to use different id/aria-labelledby value for each set of “question-radio buttons”.

For example:

Question one

<label **id=”condition-label-01”>**

<div role=”radiogroup**” aria-labelledby=”condition-label-01”**>

Question two

<label **id=”condition-label-02”>**

<div role=”radiogroup**” aria-labelledby=”condition-label-02”**>

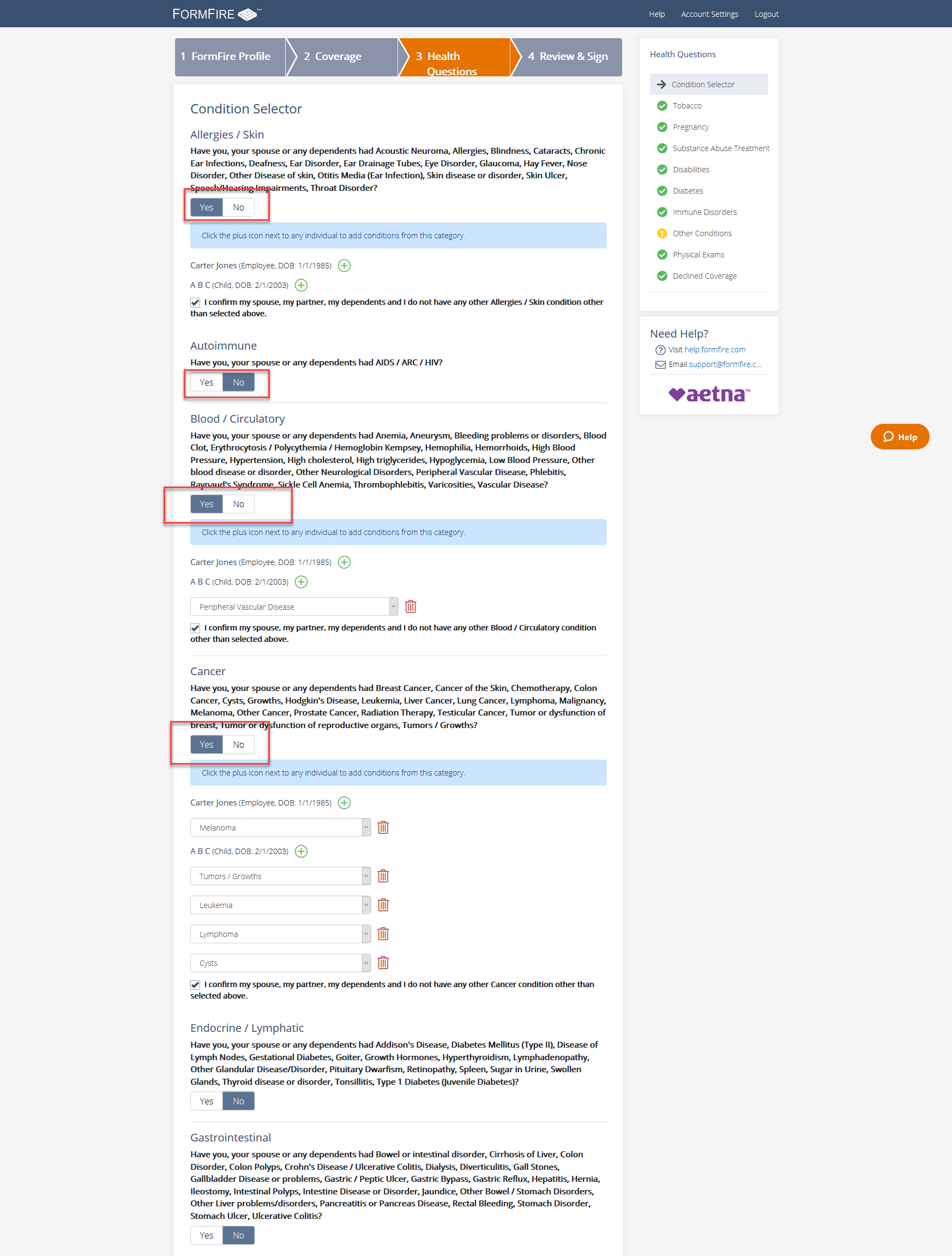


Figure 10: The Radio buttons available on the Condition selector page are labeled with one common id.

### Elements are not keyboard accessible

*[Severity Level: Blocker]*

**[Fixed]- 01/23/2020]**

The “add condition” icon button and “delete” buttons can now be activated using keyboard.

**[End of note]**

Partially Fixed- 11/20/2019

Note: Mostly all actionable elements receive focus and can be activated using keyboard except controls such as add, delete present under Condition Selector section. Although, add condition icon and delete buttons receives keyboard focus but it cannot be activated using a keyboard as only mouse event handlers are present for these buttons. Ensure to provide a keyboard event handler for these controls.

[End of note]

[Issue]

There many actionable elements which do not receive keyboard focus and cannot be activated using the keyboard when it does receive focus. Developers have applied tabindex=”-1” for these elements, which removes these buttons from natural navigation flow. Here is some other instance of this issue:

1. *(Fixed)*Buttons available on navigation region on every page do not receive keyboard focus. The buttons are Personal information, contact information, My employer, Family and household.
2. *(Fixed)*The links under “footer” and “need help” sections are not in focus order.
3. *(Fixed)*The FormFire Profile, Coverage, Health Questions, and review and sign heading links are not in focus order and not keyboard actionable.
4. *(Partially Fixed)*The “add condition” icon button and “delete” buttons do not receive keyboard focus.

[Code Reference]

(Footer and help section links)

1. <a href="http://help.formfire.com" target="\_blank" title="http://help.formfire.com" tabindex="-1">help.formfire.com</a>
2. <a href="mailto:support@formfire.com" id="email-address" title="support@formfire.com" tabindex="-1" style="margin-left: 0px;">support@formfire.com</a>
3. <a target="\_blank" href="https://www.formfire.com/privacy" tabindex="-1">Privacy</a>
4. <a target="\_blank" href="https://www.formfire.com/security" tabindex="-1">Security Info</a>

<a target="\_blank" href="https://www.formfire.com/interview/termsofuse" tabindex="-1"> Terms of Use </a>

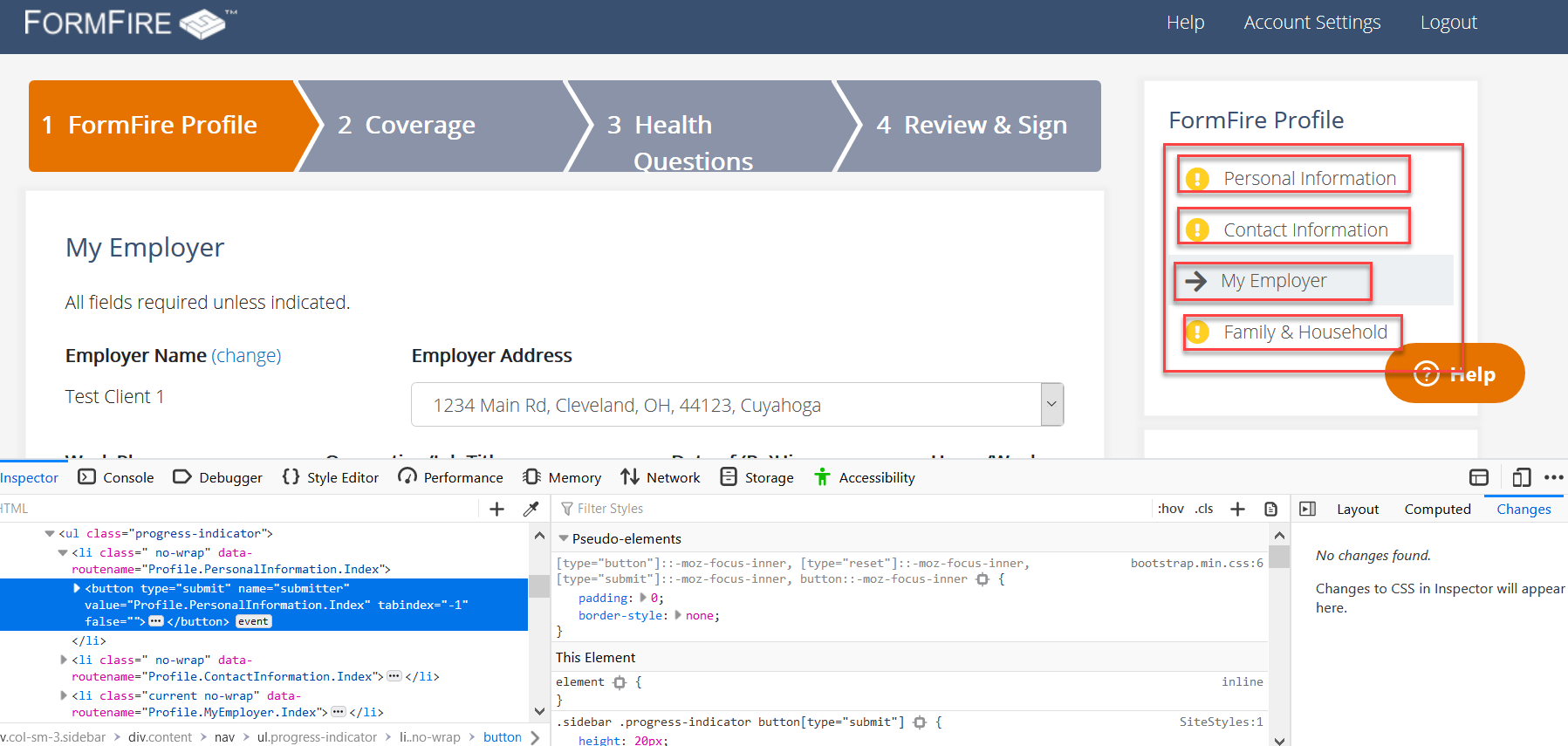


Figure 11: Buttons are not keyboard accessible

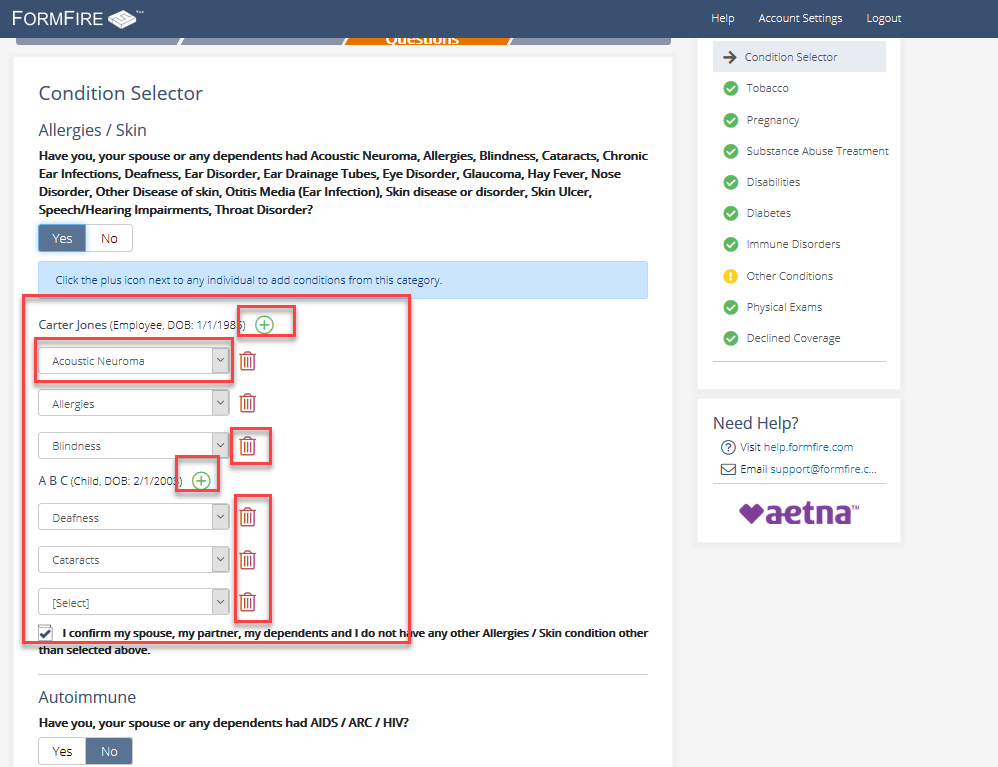


Figure 12: The add and delete buttons are not keyboard accessible.

[Recommendation]

Remove the tabindex=”-1” from the link markup to make these links available on the natural DOM order.

Developers should avoid negative tab indexes unless developers want to create any important custom controls. Native elements like button do not require any tabindex value to place the element into the natural DOM order.

Developers can use anchor tag with standalone icon image and href attribute to make “add” and “delete” buttons keyboard accessible. For example:

<a href="javascript:print()">

<img src="print.png" alt="Print this page">

</a>

### Focus does not shift to the opened slide out content and does not provide any state about the open slide

*[Severity Level: Blocker]*

**[Fixed]- 01/23/2020]**

“Add family member” control now announces its state of expanded or collapsed.

[Code snippet]

<a id="add-slideout1" data-slideout="" data-size="65%" href="/employee/FamilyAndHousehold/Add" class="btn btn-default btn-slideout" aria-expanded="false" tabindex="0">Add Family Member</a>

**[End of note]**

[Partially Fixed]- 11/20/2019

Note: Now, the focus is moving to the first actionable element on the opening “add dependent form” drawer when the “add family member” link is activated. However, it does not announce the state as expanded to provide more context to screen reader users. Ensure to add an aria-expanded attribute on the <a> element of “Add family member”.

[End of note]

[Issue]

When activating the “add family member” button, focus does not move to the first actionable element on the opened “add dependent form” drawer. In addition to this, the screen reader does not announce the state of “expanded” and the name of the drawer. Similar issues are found on every slide out activate button and pane.

Screen reader users cannot get any context about this slide out unless there is a focus change and spoken feedback to screen reader user. Keyboard only users require to press multiple tab key to reach to the open slide out.

[Recommendation]

Developers should use JavaScript focus method to move the focus to the first actionable item on the opened slide out drawer. In addition, the implementation must also allow screen readers to announce the presence of this menu. Developer should use an appropriate aria attribute, such as aria-expanded to provide more context to screen reader user.

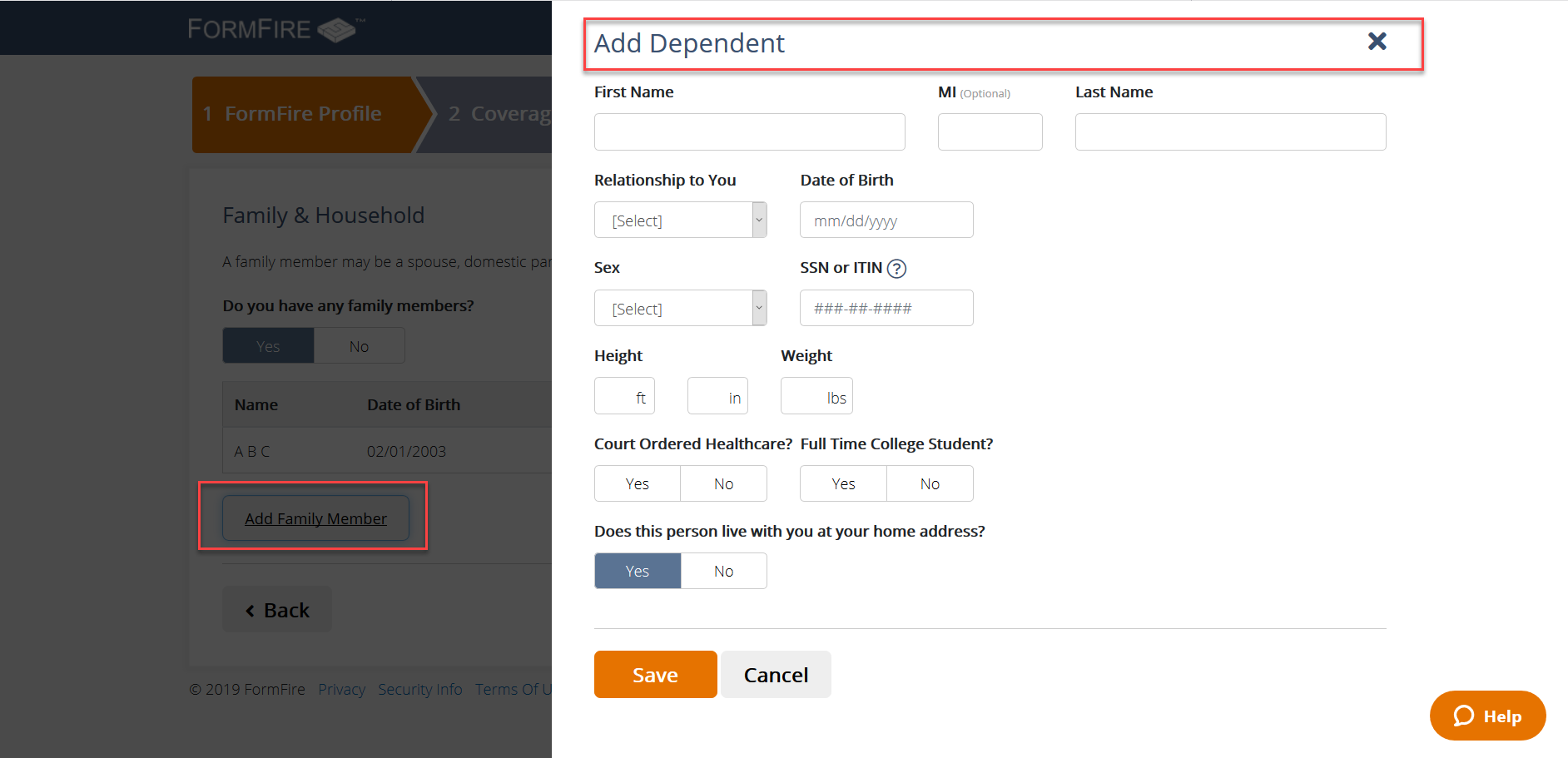


Figure 13: Focus does not shift to the opened slide out content and does not provide any state about the open slide.

### There is no valid error identification

*[Severity Level: Blocker]*

**[Partially Fixed]- 04/22/2020**

Note: No changes are observed from the last validation except aria-hidden attribute is removed from error icons. Developers must ensure to implement the following changes.

* Error summary must be available in each form section
* Error icon’s alternative equivalent must be announced to the users of screen reader users. Add role=”img” along with aria-label=”Error” to icon image. Without proper name and role information, the icon elements will not be rendered to screen reader users.

[Non-compliant code]

<input aria-required="true" class="form-control format-name error" id="first-name" aria-invalid="true" ..>

<span class="help-block form-error" aria-describedby="first-name"><i class="fal fa-exclamation-circle" alt="Error"></i>Required Field</span>

[Compliant code example]

<input **aria-describedby="first-name-err"** aria-required="true" class="form-control format-name error" id="first-name" aria-invalid="true" ..>

<span **id="first-name-err"** class="help-block form-error" …><i class="fal fa-exclamation-circle" **role=”img” aria-label="Error"**></i>**First name is required**</span>

* The inline error messages must be associated with its respective form input field. The aria-describedby attribute can be used to associate the error message with a form field.
* Ensure to remove role=”alert” from div elements of error summaries displayed for each section on “Review and sign” page. The role=”alert” on all div element causes to announce random error summary section on page load as it overlaps the other alerts on the page. This will cause screen reader users to miss out on the other alert messages. Instead, it is recommended to keep tabindex=”0” attribute on div elements of error summaries of each section and move the focus to first error summary div element container on page load using JavaScript focus()method.

**[End of note]**

**[Partially Fixed]- 01/23/2020**

The error summary which was appearing at the beginning of the form is now not available. Users can now navigate to next steps even with empty form fields. All the form errors appear at the “Review and sign” step.

For example: if the user skips filling details in the first from i.e., “Personal information” and activate “Next”. The “Contact Information” form appears without prompting for errors in “Personal information” form. In the last step I.e., “Review and Sign”, error related to each form is indicated with an edit option. However, this error implementation has following issues:

* when the users activate “Edit” control of each section, the error summary is not available in each section
* Inline error messages are not associated with its input field
* The error icon (!) has alt value set as “Error” however, it is hidden from the screen reader using “aria-hidden” to true

Developers must ensure that error notification is implemented for each form so the users of screen readers can clearly address the issues of the current form and then navigate to the next section. In this case,

* Error summary must be available in each form section
* Error icon’s alt must be announced to the users of screen reader users. This can be done by removing the “aria-hidden” from the image
* The Inline error messages must be associated with its respective form input field. The aria-describedby attribute can be used to associate the error message with a form field

[Non compliant code sample]

<div class="form-group has-error">

<label for="first-name" class="">First Name</label>

<input aria-required="true" class="form-control format-name error" id="first-name" name="FormModel.FirstName" type="text" value="" data-validation="required" current-error="<i class='fal fa-exclamation-circle' alt='Error'></i>Required Field" style="background-image: url(&quot;data:…;); background-repeat: no-repeat; background-attachment: scroll; background-size: 16px 18px; background-position: 98% 50%; cursor: auto; border-color: rgb(185, 74, 72);" aria-invalid="true">

<span class="help-block form-error" aria-describedby="first-name"><i class="fal fa-exclamation-circle" alt="Error" aria-hidden="true"></i>Required Field</span></div>

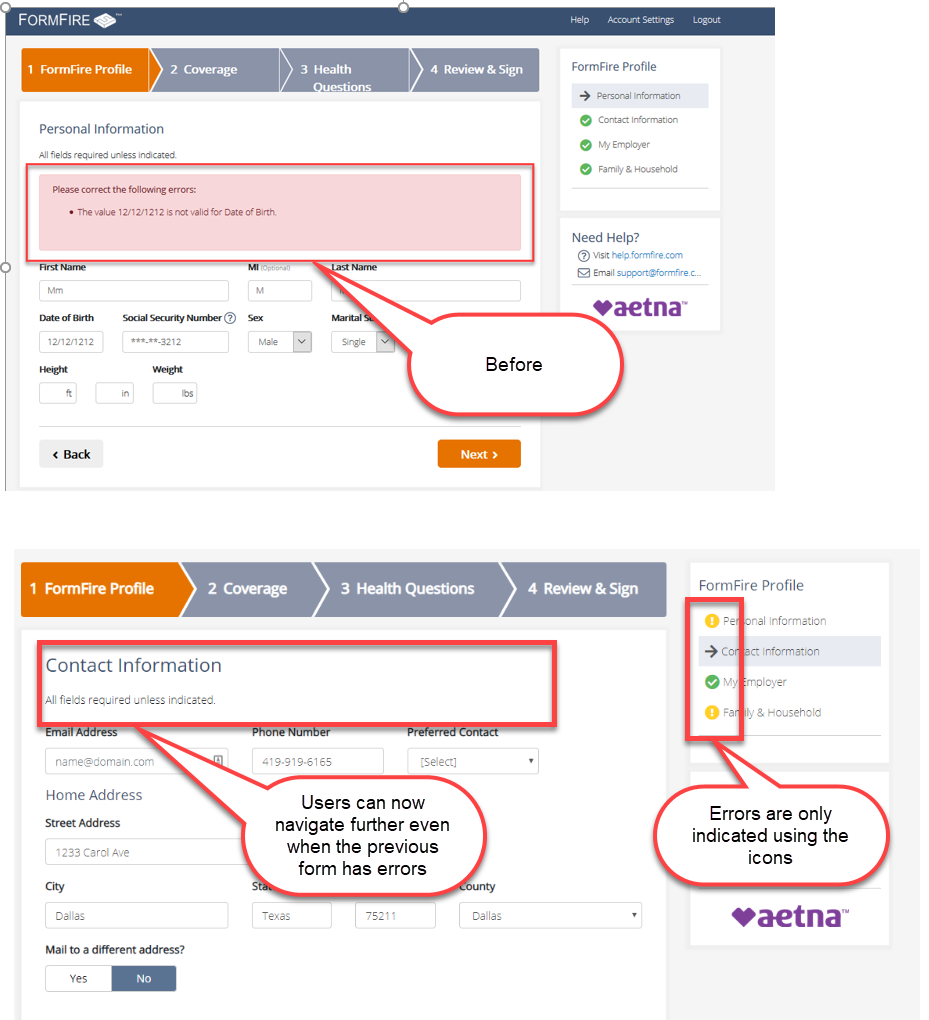


Figure: No clear indication of form error

**[End of note]**

[Partially Fixed]-11/20/2019

Note: Now, for FormFire Personal information page, when the form is submitted with the errors, errors are displayed at the top of the page and focus is moving to this error summary on invalid submission. However, these error messages are not associated with their respective form fields using aria-describedby attribute.

For “Condition selector”, “Sign up”, “Reset password” pages, inline error messages are displayed, and these inline error messages are not correctly associated to its form fields. Currently, aria-describedby attribute is applied on the error div referring to the id of input field which is not valid making it not announcing inline error messages. Ensure to provide aria-describedby attribute on the input field referring to the id of error container.

[Non-Compliant Code Example]

<input class="form-control ng-pristine ng-invalid ng-invalid-required ng-valid-pattern ng-touched" id="userName" ng-keyup="vm.hideInputValidation()" ng-model="vm.userName" name="UserName" ng-change="vm.validInfo=true" ng-pattern="/^[^@]\*$/" style="background-color: #fff !important;" required="" autofocus="" autocomplete="off" aria-required="true" aria-invalid="true">

<div ng-show="(vm.usernameRequired &amp;&amp; form.UserName.$error.required)" class="errorMessage ng-binding" aria-describedby="userName">

<i class="fal fa-exclamation-circle" alt="Error" aria-hidden="true"></i> You must enter a username

</div>

[Compliant Code Example]

<input class="form-control ng-pristine ng-invalid ng-invalid-required ng-valid-pattern ng-touched" id="userName" ng-keyup="vm.hideInputValidation()" ng-model="vm.userName" name="UserName" ng-change="vm.validInfo=true" ng-pattern="/^[^@]\*$/" style="background-color: #fff !important;" required="" autofocus="" autocomplete="off" aria-required="true" aria-invalid="true" aria-describedby="err1">

<div ng-show="(vm.usernameRequired &amp;&amp; form.UserName.$error.required)" class="errorMessage ng-binding" id="err1">

<i class="fal fa-exclamation-circle" alt="Error" aria-hidden="true"></i> You must enter a username

</div>

[End of note]

Error identification on the website for all the form fields is not valid. Inline required fields are not being announced to screen reader users. Screen reader users are unable to get the context of the any error on the page. Sole use of color with the word “require field” makes it difficult for users with limited color perception, users with low vision, and screen reader users to locate the field-specific error messages.

There is no error identification alert, focus change to the error field or message available for screen reader users on activating submit button. Screen reader users are unable to get the context of error without any spoken alert message or focus change to the error field.

Error identification is not consistent across the website, some errors shown inline and some of the errors are shown at the top of form field.

Below are few instances listed from the website where there is no error identification alert or messaging available for screen reader users:

1. On the “Condition selector” page, activating next button
2. On triggering an error on the security question page
3. On the FormFire personal information page, errors are not rendered on the top of the form.
4. On the “sign up” and “Reset Password” pages

**Note:** Onlyon entering an incorrect username and password on the login page is there an alert message “The username or password you entered is incorrect. Please call (216)367-9730 for assistance”.

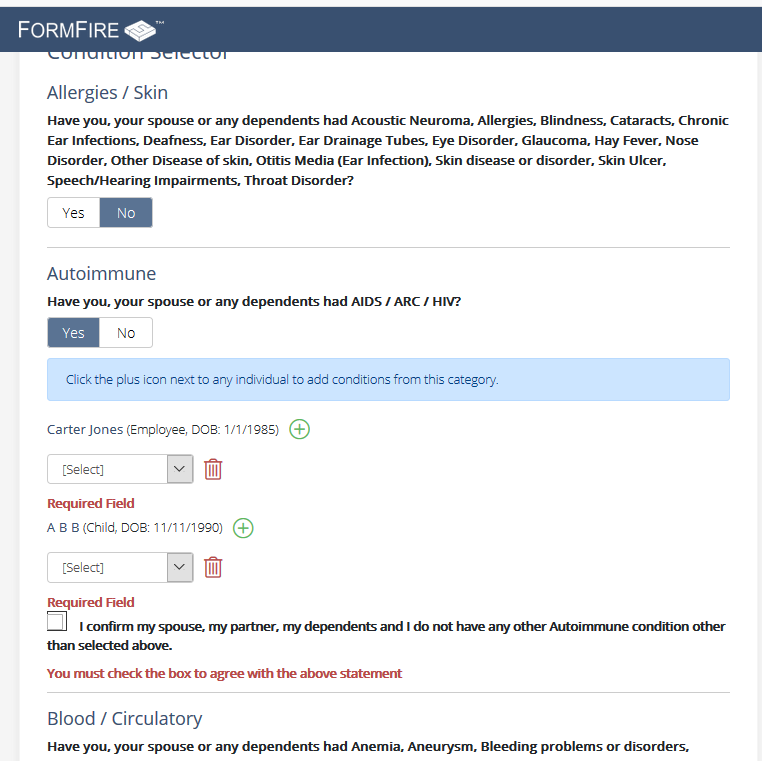


Figure 14: On the “Condition selector” there is no valid error identification for screen reader user

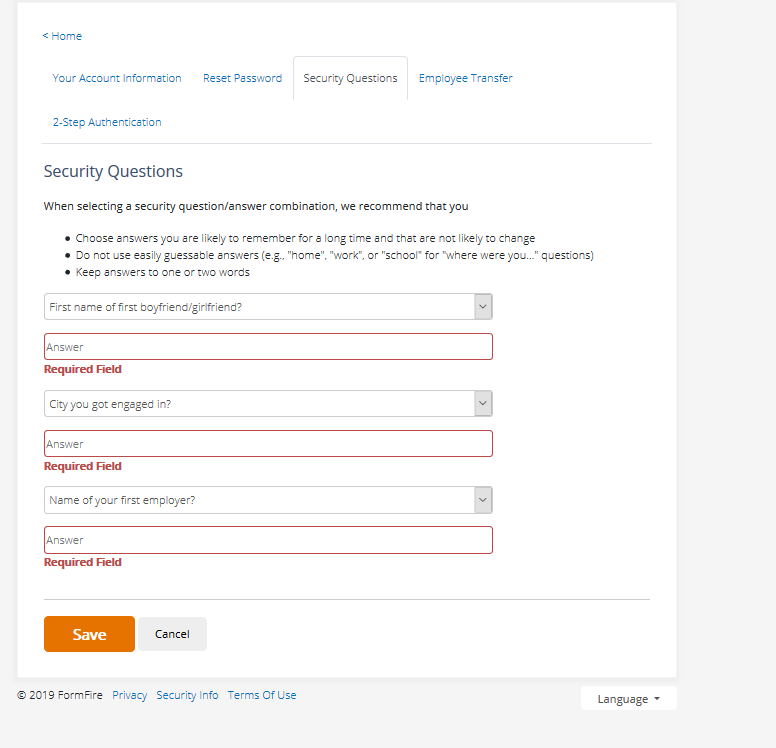


Figure 15: There is no valid error identification on security question page

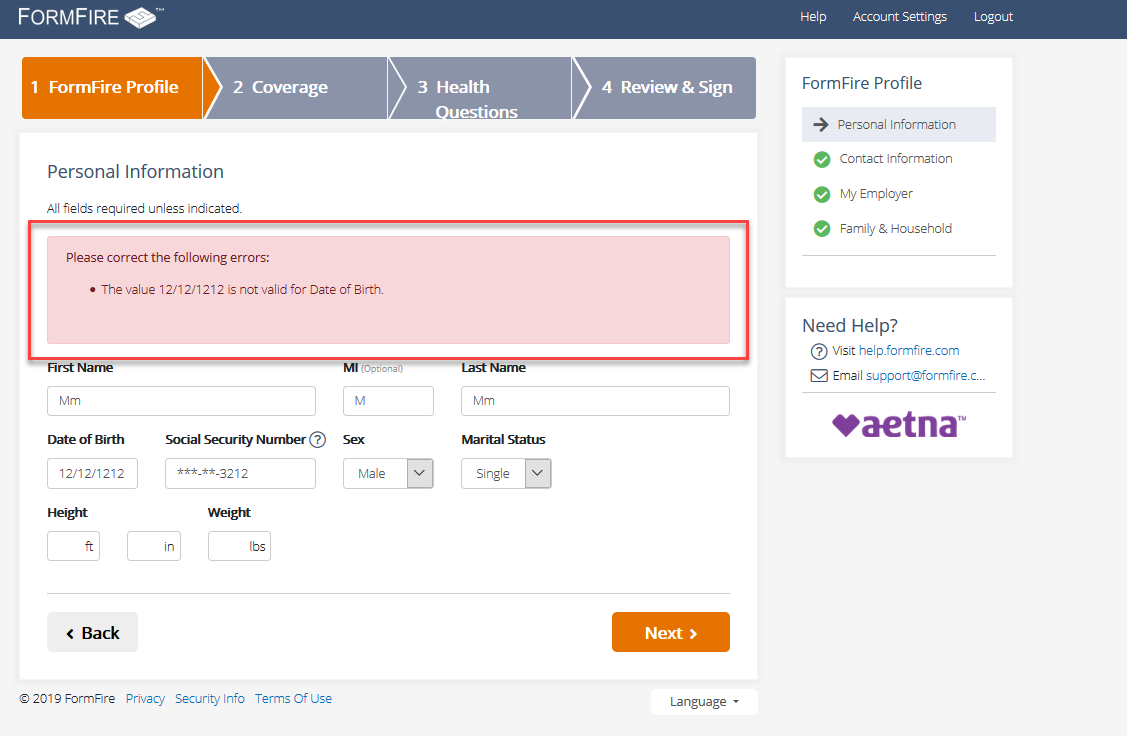


Figure 16: There is no valid error identification on activating next button from the FormFIre personal information.

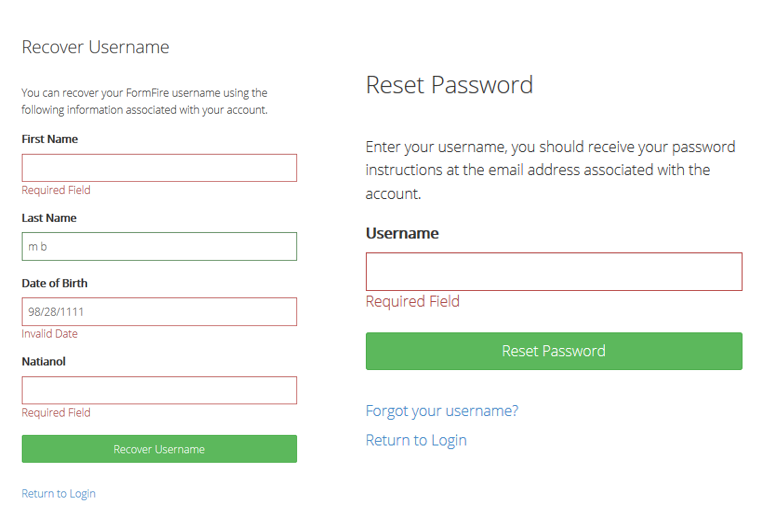


Figure 17: There is no valid error identification the “sign up” and “Reset Password” pages.

[Recommendation]

It is recommended to add text "Error" or using an inline image (error icon) with the alternative text that indicates error followed by the textual error messages. Error identification should consistent across the website, as per the user interface it is best practice to use inline error identification.

Properties such as the aria-invalid state attribute, also associating field-specific error messages with each field in error by a technique such as aria-describedby can be used to indicate to assistive technology that a field is in error.

Developers should provide a mechanism to convey the error message to screen reader users. This can be resolved using JavaScript. The keyboard focus must be set on the error summary message using JavaScript focus(). It is best practice to provide the error summary on the top of the form. Developers should use aria-required=” true” to communicate that the input is not optional.

### Radio buttons do not have proper roles

*[Severity Level: Blocker]*

**[Fixed]-11/20/2019**

[Issue]

There are custom Radio buttons used as toggle buttons on the website. These buttons do not announce the state as selected or checked. These buttons do not have any role such radio. A screen reader user is unable get the context of active yes/no button.

Yes/No buttons on the dependent page do not have valid role and state. Similarly, on the Coverages page the first “Cover and Waive” radio buttons do not have valid role and state. It does not announce the state as selected or checked.

[Recommendation]

Developer should use a role=”radio” on the <label> available for the input field.

For more on the implementation, refer to this example: <http://whatsock.com/tsg/Coding%20Arena/ARIA%20Radio%20Buttons/ARIA%20Radio%20Buttons/demo.htm>

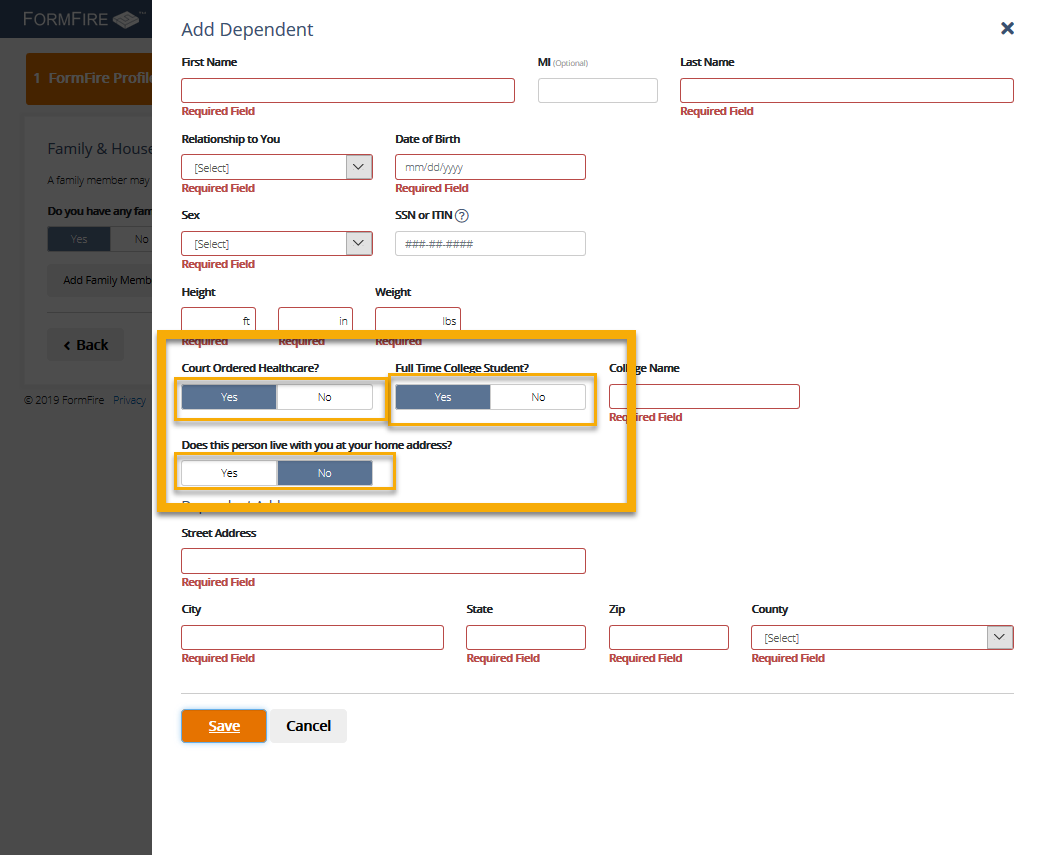


Figure 18: Yes/No button does not have proper role.

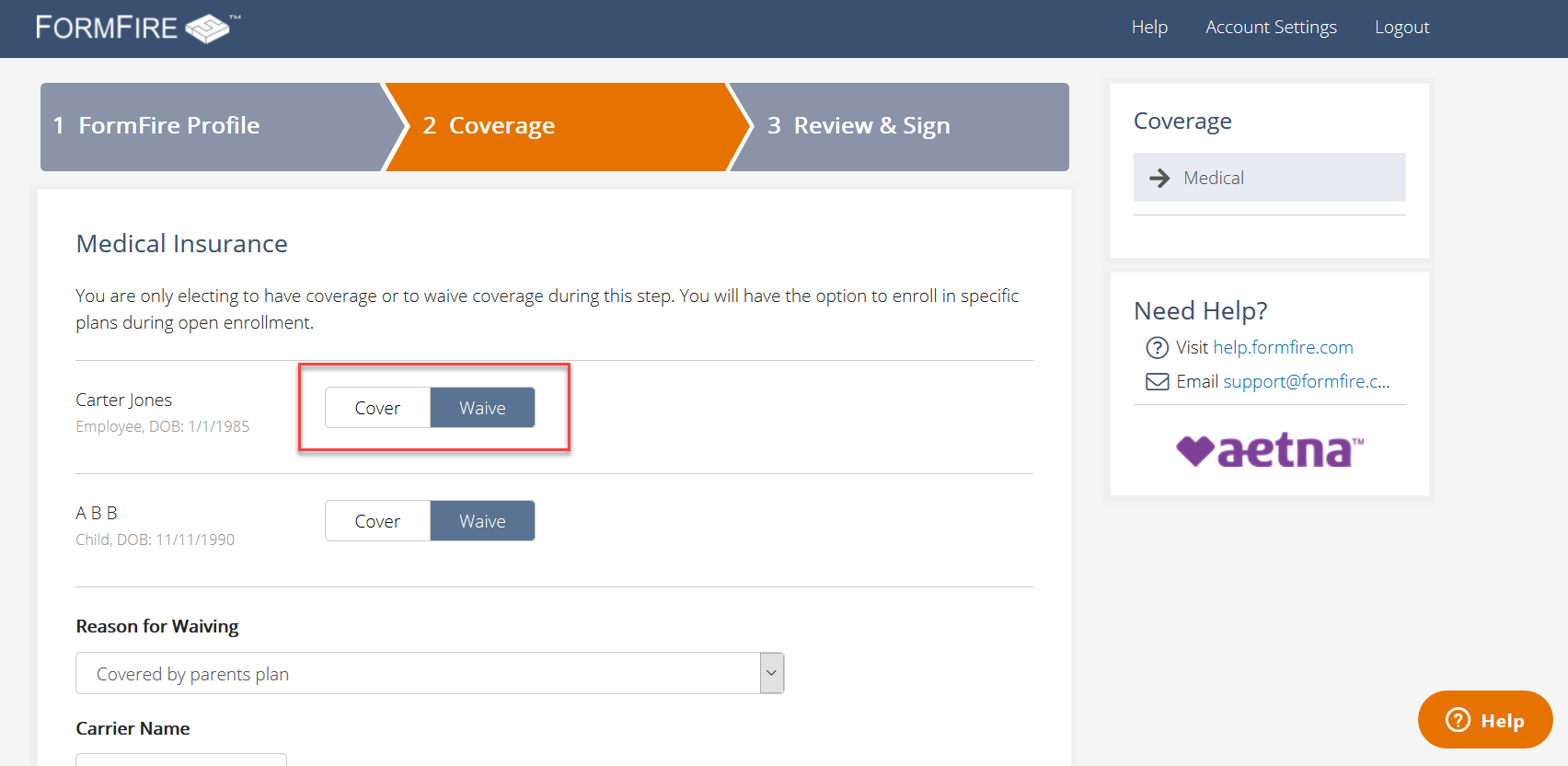


Figure 19: Cover/Waive button does not have proper role.

### On the SignUp page, tooltip information does not have a meaningful label and is not keyboard accessible

*[Severity Level: Important]*

**[Fixed- 01/23/2020]**

Tool tip is now announced to the users of screen readers.

**[End of note]**

[Partially Fixed]-11/20/2019

Note: Now, the tooltip link receives keyboard focus. The aria-label is used to provide accessible label to this link. However, the tooltip content is still not rendered to screen reader users. Ensure to provide aria-describedby attribute to link referring to the id of tooltip content div element. Also, add role=”region”, aria-label attributes to tooltip div element.

[End of note]

[Issue]

The tooltip link and information on the Sign Up page is not keyboard accessible. Only mouse event handlers are used to activate the question mark link. Also, link does not have any accessible name. Keyboard only user and screen reader users are unable to access the tooltip and associated information.

*Code Reference*

<a class="icon helpPopover " id="employerTooltip" data-toggle="popover" data-placement="top" data-original-title="" title="">

<span class="glyphicon icon-questionmark\_circle helpIcon trn" data-trn-key=""></span>

</a>

[Recommendation]

Developers should ensure that the tooltip is accessible by keyboard only users and screen reader users can read the information. To solve this issue, developers can use a keyboard event handler and an aria-label for the information tooltip.

For more on the implementation, refer to this example:

<http://whatsock.com/tsg/Coding%20Arena/Tooltips/Tooltip%20(Internal%20Content)/demo.htm>

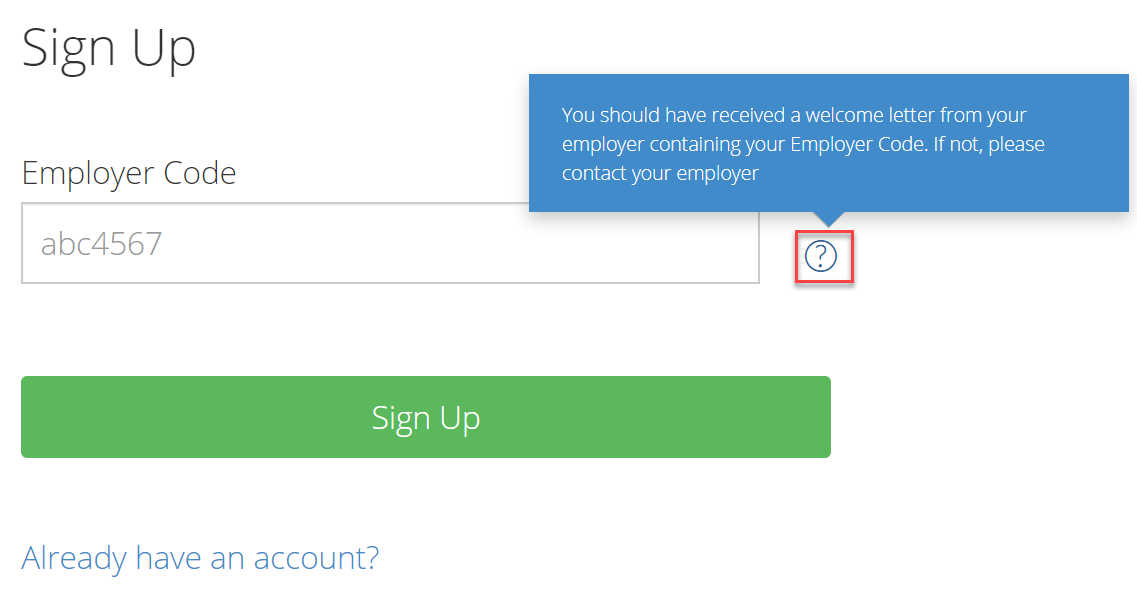


Figure 20: The “Change” link required a meaningful text on the profile page

### Color contrast issue

*[Severity Level: Important]*

**[Partially Fixed]- 01/23/2020**

Note: There are still a few instances with color contrast issues present on the website.

* Controls such as sign up, log in with the white text (#FFFFFF) against a green background (#5CB85C) have a contrast ratio of 2.5:1.
* Questions under ‘Help Articles’ modal with the blue text(#55B3D2) against white background(#FFFFFF) has color contrast ratio of 2.4:1.
* In resource center section, the grey text (#BBBDC6) against white background(#FFFFFF) has color contrast ratio of 1.9:1.

Ensure to modify foreground or background color to meet sufficient contrast ratio. Standard text less than 18 point (or less than 14 point if bold) must have a luminosity contrast ratio of 4.5:1 or more. Text 18 points or larger (14 points or larger if bold) must have a luminosity contrast ratio of 3:1 or more.

\*Please note that developers need to fix all color contrast issues across the websites and not just issues listed in the examples.

**[End of note]**

**[Partially Fixed]- 01/23/2020**

There are still many instances with color contrast issues present on the website.

For example: The blue link texts(#428BCA) against white background(#FFFFFF) has color contrast ratio of 3.6:1.

The white text(#FFFFFF) against green background (#60B350) in landing page has contrast ratio of 2.6:1.

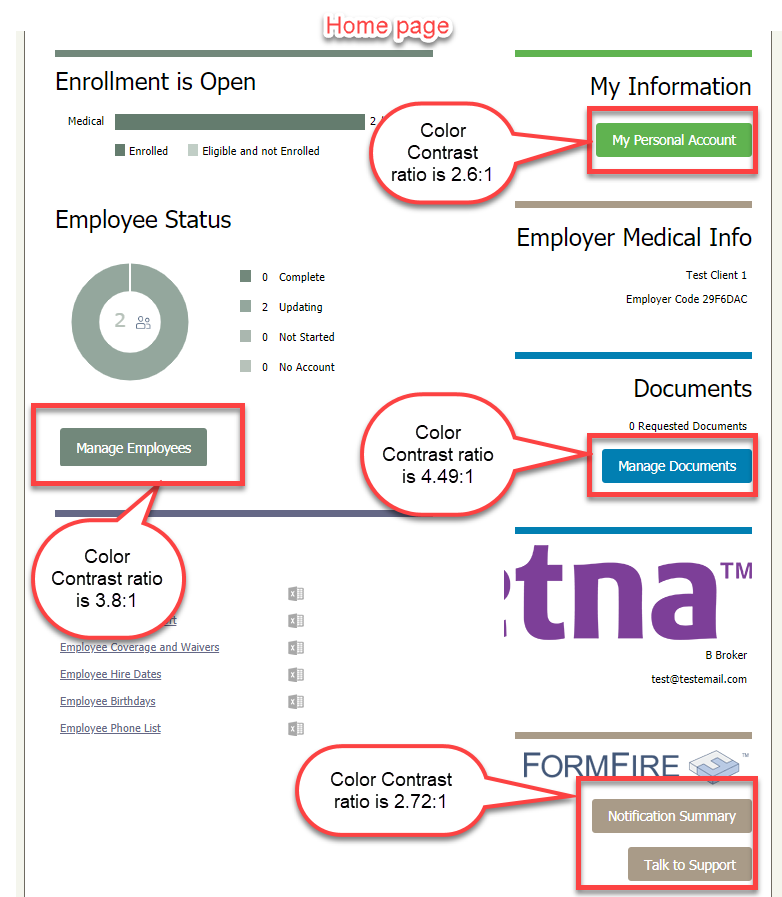
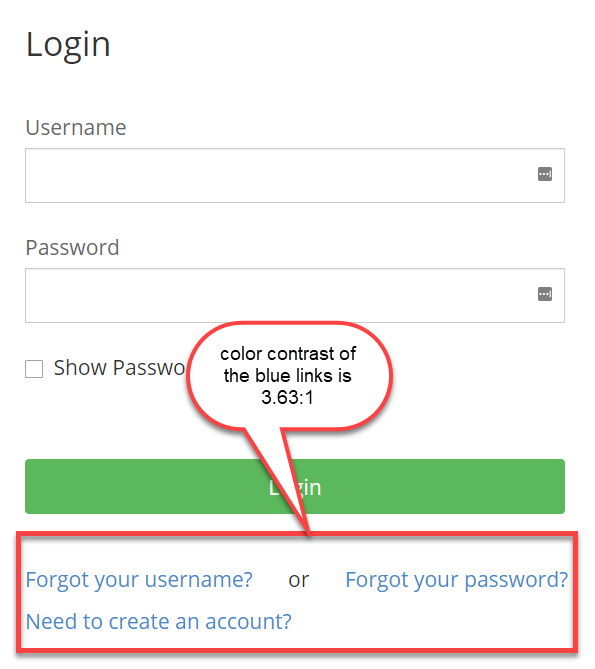


Figure: The color contrast issues in home page and login page

**[End of note]**

Not Fixed- 11/20/2019

Note: Currently, only light grey text against white background contrast is fixed. However, there are still many instances with color contrast issues present on the website.

For example,

Dark grey text against light grey background on “Employee Summary” page

Foreground:#999999

Background:#EEEEEE

The contrast ratio is: 2.5:1

[End of note]

[Issue]

There must be sufficient color contrast between foreground and background text and images of text. Overall color contrast ratio of the webpages is sufficient, however following are the few instances where foreground text color and background color does not satisfy the standard 4.5:1 color contrast ratio.

Following are the color contrast issue found the website:

1. *(Not Fixed)*White text and orange background

Foreground:#FFFFFF

Background:#E67300

The contrast ratio is: 3.1:1

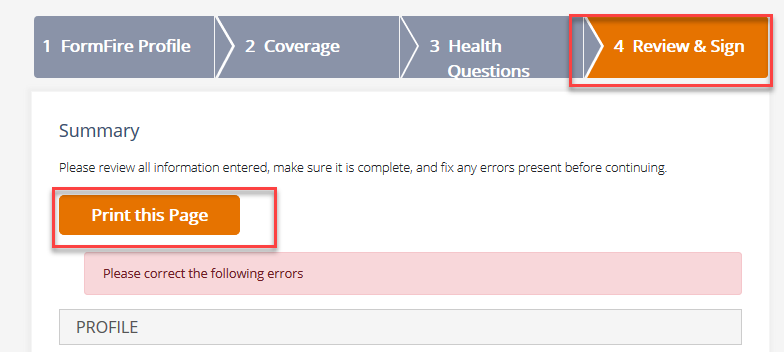


Figure 21: The color contrast issue1 White text and orange background

1. *(Not Fixed)*White text and grey background

Foreground:#FFFFFF

Background:#8A93A8

The contrast ratio is: 3.1:1

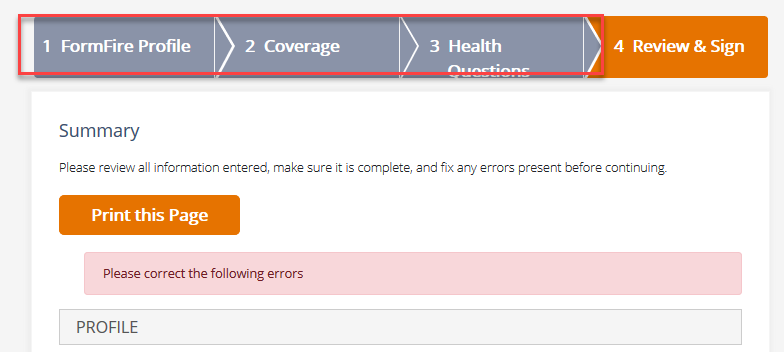


Figure 22: The color contrast issue 2 White text and grey background

1. *(Fixed)*Light grey text and white background

Foreground:#CCCCD9

Background:#FFFFFF

The contrast ratio is: 1.6:1



Figure 23: The color contrast issue

[Recommendation]

Developers must ensure that the color contrast for all text (and images of text) meets the WCAG 2.0 Level AA requirements unless specifically excluded. Designers or developers should modify the foreground color, background color, and/or font size so that sufficient contrast is attained. Standard text less than 18 point (or less than 14 point if bold) must have a luminosity contrast ratio of 4.5:1 or more. Text 18 point or larger (14 point or larger if bold) must have a luminosity contrast ratio of 3:1 or more.

### The height input fields are not grouped under same fieldset

*[Severity Level: Important]*

**[Fixed]- 11/20/2019**

[Issue]

The height has two input sections named as foot and inches; however, it is not grouped together. By grouping all the elements in one category users of assistive technologies can understand the relationship of the elements.

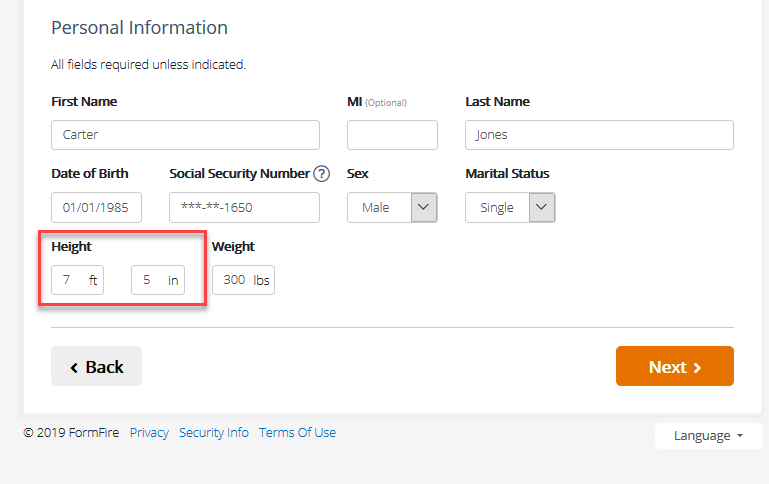


Figure 24: The height input fields are not grouped under same fieldset.

### Keyboard focus is not indicated visually

*[Severity Level: Important]*

**[Fixed]-11/20/2019**

[Issue]

The overall visual keyboard focus indication is good on the website, however some of the element are missing visual focus outline. The Keyboard only users are relying on the visual keyboard focus indication to follow their interaction with elements on the screen. Keyboard only user cannot check their path of navigation without visual focus. Here are the two instances regarding this issue.

Close button on the opened slide out pane does not have visual keyboard focus indication. Similarly, there is no visual keyboard focus indication on the show password checkbox, login, sign up, reset password and recover username buttons.

[Recommendation]

Developer should avoid outline: none which removes the default focus outline of the browser. Also, developer can user customize focus outline with enhanced focus indication.

*Compliant Example 1: (close button)*

.slideout .slideout-content .slideout-header button.slideout-close:focus {

outline: 2px solid #9acffc;

*Compliant Example 2: (other buttons)*

.btn:focus, .btn:active {

outline: 2px solid #5f94e6 !important;

[Recommendation]

Developers should ensure to group both input fields under height label. Utilize the fieldset element to group together related groups. For each fieldset element, ensure that a legend has been defined and contains an appropriate value.

### Heading structure on the personal information is not hierarchical

*[Severity Level: Important]*

**[Fixed]-11/20/2019**

[Issue]

Heading on most of the pages are not in hierarchical order. The h1 level heading is skipped on most of the pages.

Screen reader users mostly use the heading structure to map the web page and get the context of the page structure. Without proper heading structure on the page, it is difficult for screen reader users to understand the structure. Also, skipping in the heading level might be confusing for screen reader user.

[Recommendation]

Ensure to have main h1 heading and then web page should follow the heading hierarchical map from h2, h3, h4, etc. Nesting of heading should be meaningful and in sequential order.

### No Main Landmark found on the page and multiple nav regions are available

*[Severity Level: Important]*

**[Fixed]-11/20/2019**

[Issue]

There is no proper landmark structure on the webpages. There is no main region found on the webpages. There are multiple nav region on most of the page. Screen reader users use landmarks to quickly navigate through pages. If there are multiple nav region available, then it will be confusing for screen reader user to locate the correct section.

[Recommendation]

There must be a main landmark which communicates the main content of the page. Developers must ensure that duplicate nav elements contain meaningful accessible labels to assist users of screen readers to identify and distinguish the section, region or landmark. Common methods of providing accessible labels are the aria-labelledby attribute (best used when a visual label already exists in the page) or the aria-label attribute (when no visual label exists).