

Homework 8 — Group 7

Screen Reader

Chris Pyles	Using the screen reader was difficult because it tried to read <i>everything</i> on the page, including the ads, which made it difficult to figure out what was the main content of the page, which I was interested in, versus what was extraneous information that distracted from what I was trying to do. I tried the screen reader on https://www.imore.com/how-enable-voiceover-mac . Because of what was mentioned above, it was very difficult to navigate the page because of all the ads. Our project site, https://damp-brushlands-79654.herokuapp.com/ , was difficult because it just read the links out but without reading the text, just “Skip to content.”
Raymond Meng	The screen reader is a bit hard to use. Not everything on a web page is informative. It reads everything out loud indiscriminately. We tried it on https://www.imore.com/how-enable-voiceover-mac . The experience was bad for it is hard to use. Our project site, https://damp-brushlands-79654.herokuapp.com/ , was hard, for it read the links out without context.
Peyrin Kao	The screen reader made reading webpages much more tedious, because it read pages aloud at a much slower pace than reading text visually. I learned that when designing frontends, we should keep in mind the length of the text on any particular webpage. I tried the screen reader on https://www.imore.com/how-enable-voiceover-mac , and it was very difficult to navigate because the text was read in arbitrary order, and a lot of unnecessary/advertising text was read aloud. I then tried the screen reader on https://damp-brushlands-79654.herokuapp.com/ , our own project homepage, and it was slightly easier because our interface is more visual and less complicated than the imore.com interface. However, it was still difficult to navigate compared to using traditional mouse-and-keyboard.
Geping Qu	The screen reader we used to read was https://www.imore.com/how-enable-voiceover-mac . It was not easy to use because it read everything on the page and read it very slowly. It could be very difficult to figure out what was the main content of the page and what is appealing and what is not. It was difficult to navigate because there are adds all over. Our project website is https://damp-brushlands-79654.herokuapp.com/ . It was also difficult because it reads out irrelevant content, and it read the links without reading the text on it.
Hanson	We tried the screen reader https://www.imore.com/how-enable-voiceover-mac . It made navigation difficult due to the reader giving a lot of information in a short period of time. However, to the visually impaired, the user needs the information since they are not able to see it. Our project website is https://damp-brushlands-79654.herokuapp.com/ . Our site was not designed with accessibility in mind so the only thing read was the link.

Austin Maina	Using the screen reader was quite difficult especially with the automated voice which made it difficult to navigate through what was actually on the screen. The link we used was https://www.imore.com/how-enable-voiceover-mac .
--------------	---

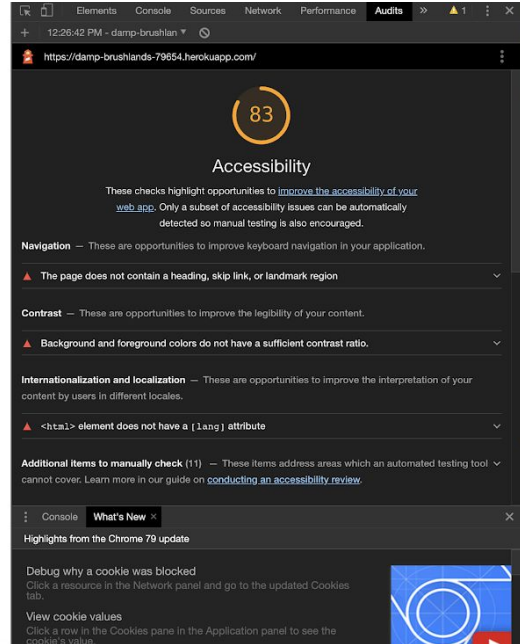
Keyboard Experience

Chris Pyles	Navigating with the keyboard was pretty easy as I use them a lot to navigate the pages I visit on a daily basis. I visited https://github.com/ . On our project site was also easy because our homepage doesn't have lots of buttons. It was more difficult on the user list page, and not useful on the schedule creation form because that interface is an HTML table not a visible form.
Raymond Meng	We tried using Facebook with keyboard shortcuts, because we are already using it on a daily basis. We also tried it on our homepage. It is hard to use it on our homepage, because when we built the HTML table we expect mouse interactions.
Peyrin Kao	I tried using Facebook with keyboard shortcuts. It was intuitive and similar to my regular browsing because I already frequently use keyboard shortcuts when browsing the Internet. I also tried using our project page https://damp-brushlands-79654.herokuapp.com . Again, because our homepage doesn't have too many buttons, and the browser makes tabbing through buttons pretty intuitive. The schedule creation form (HTML table) was built with mouse interaction in mind, so it was much more difficult to use using the keyboard only.
Geping Qu	Using keyboard shortcuts was easy to navigate. I use the keyboard shortcuts everyday when visited often visit website. I tried using Facebook with keyboard shortcuts. We tried it on our homepage, but it's hard to use because we build the interface using HTML table, so we expect mouse interaction, not so much about keyboard interactions.
Hanson	We navigated Facebook using keyboard controls which was easy since I use keyboard controls listed in the info link already. Our project home site was easily navigable using keyboard controls. However, our schedule form uses click events, so it is not usable with keyboard only.
Austin Maina	It was quite easy to navigate with the keyboard given that I am already used to using it on my laptop and hence it was very intuitive. The link I used was

Application Audit

Lighthouse Route: "/" (root path)

The tests reported that we could improve keyboard navigation, that we didn't have a sufficient contrast ratio between the back- and foreground of our app, and that the `html` tag needs a `lang` attribute.



Fast Pass

Route: "/users"

The fast pass reported that there was not enough contrast, that the `html` tag didn't have a `lang` attribute, and that some form elements needed labels.

Accessibility Insights for Web
FastPass

Target page: [Cal NERDS Schedule](#)

1 Automated checks
2 Tab stops

Automated checks

Automated checks can detect some common accessibility problems such as missing or invalid properties. But most accessibility problems can only be discovered through manual testing. The best way to evaluate web accessibility compliance is to complete an [assessment](#).

Show failures ☒ On [Export result](#)

17 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

<input checked="" type="checkbox"/>	>	Path	Snippet	How to fix
<input checked="" type="checkbox"/>	>	color-contrast: Elements must have sufficient color contrast (10) WCAG 1.4.3		
<input checked="" type="checkbox"/>	>	html-has-lang: <html> element must have a lang attribute (1) WCAG 3.1.1		
<input checked="" type="checkbox"/>	>	label: Form elements must have labels (6) WCAG 1.3.1, WCAG 3.3.2		

Failure details

Select a single failure instance from a group in the table above to see more details here.

Accessibility Insights Assessment

Route: "/users/1/schedules/new"












I was unsurprised by the results of the assessment, because I knew the ones that we would fail already from the other accessibility tests. All of the main ones appear on multiple pages (e.g. color contrast, html lang attribute, etc.). These can be fixed by altering the stylesheets and editing the application layout ERB file.

Automated checks

Getting Started

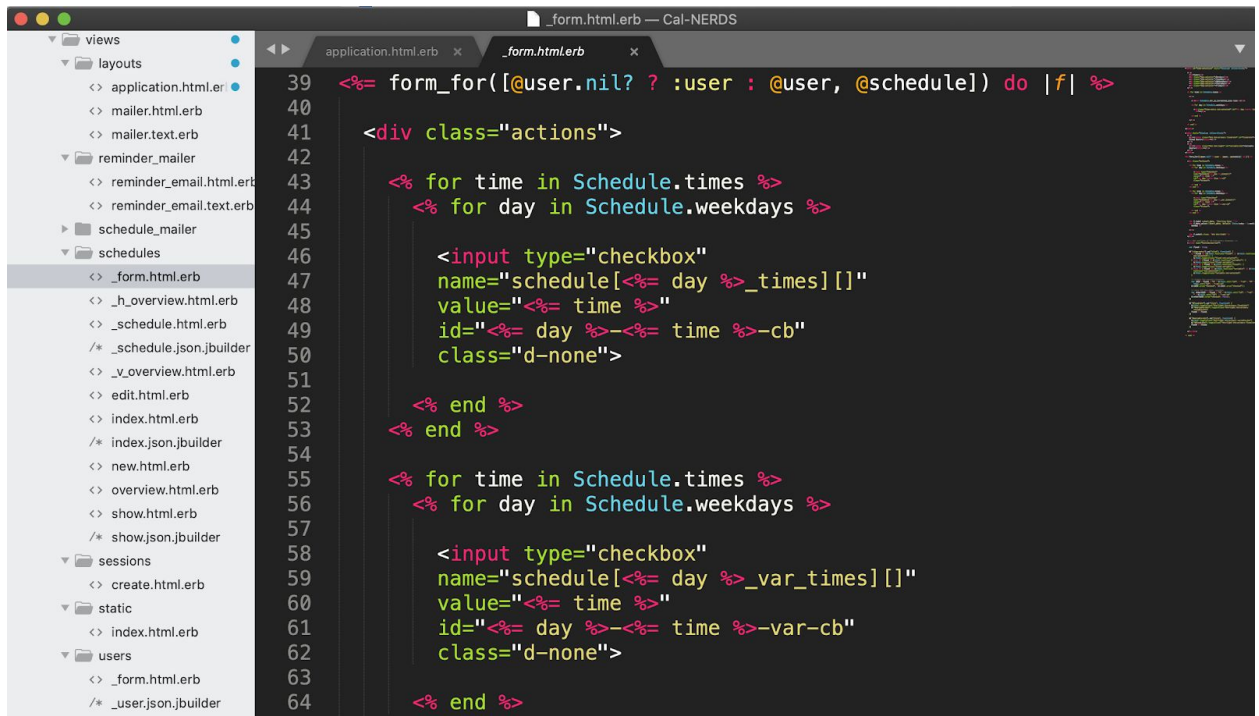
Accessibility Insights for Web automated accessibility checks can detect some of the most common accessibility issues, depending on the complexity of the site or the application.

Requirements

color-contrast. Elements must have sufficient color contrast. 	accesskeys Visual helper <input type="checkbox"/> Off No matching/failing instances were found  How to test accesskey attribute value must be unique. See more info here. Instances No matching instances
html-has-lang. <html> element must have a lang attribute. 	
label. Form elements must have labels. 	
accesskeys. accesskey attribute value must be unique. 	
area-alt. Active <area> elements must have alternate text. 	
aria-allowed-attr. Elements must only use allowed ARIA attributes. 	
aria-hidden-body. aria-hidden='true' must not be present on the document body. 	
aria-hidden-focus. ARIA hidden element must not contain focusable elements. 	
aria-input-field-name. ARIA input fields have an accessible name. 	
aria-required-attr. Required ARIA attributes must be provided. 	

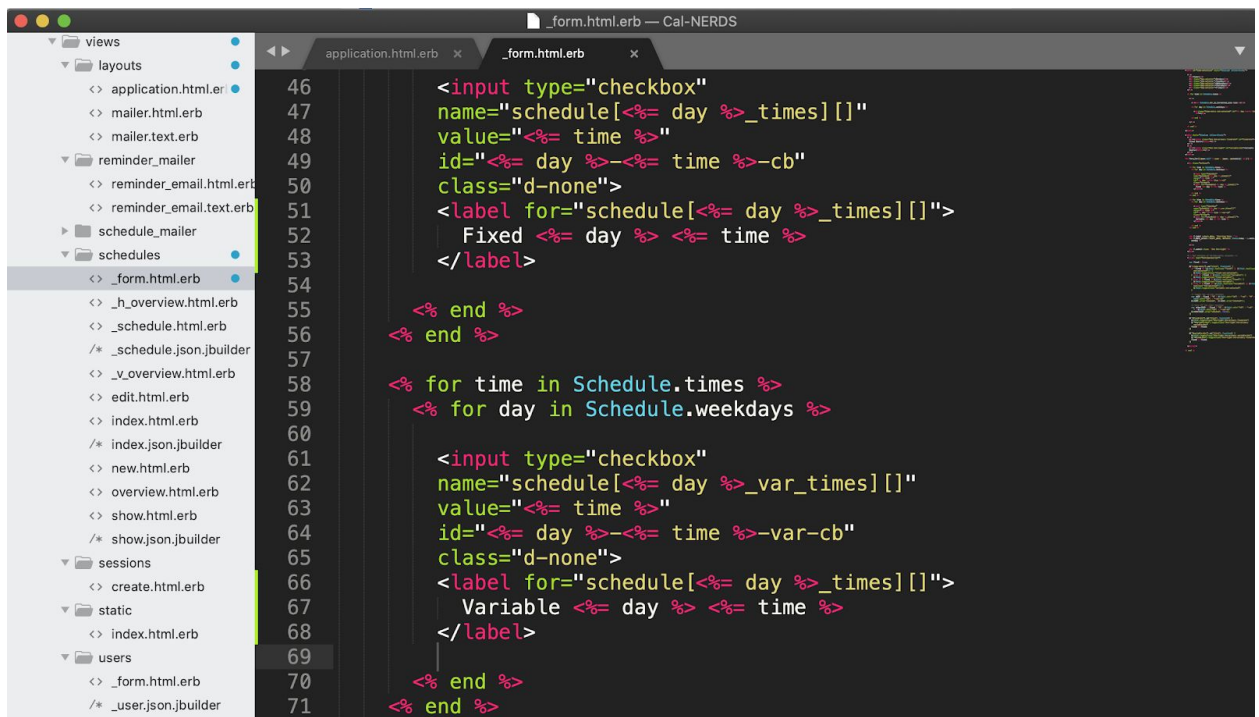
Fixes

1. we fixed the html lang attribute issue before:



```
39 <%= form_for([@user.nil? ? :user : @user, @schedule]) do |f| %>
40
41 <div class="actions">
42
43   <% for time in Schedule.times %>
44     <% for day in Schedule.weekdays %>
45
46       <input type="checkbox"
47         name="schedule[<%= day %>_times][]"
48         value="<%= time %>"
49         id="<%= day %>-<%= time %>-cb"
50         class="d-none">
51
52     <% end %>
53   <% end %>
54
55   <% for time in Schedule.times %>
56     <% for day in Schedule.weekdays %>
57
58       <input type="checkbox"
59         name="schedule[<%= day %>_var_times][]"
60         value="<%= time %>"
61         id="<%= day %>-<%= time %>-var-cb"
62         class="d-none">
63
64     <% end %>
65   <% end %>
```

after:



```
46
47   <input type="checkbox"
48     name="schedule[<%= day %>_times][]"
49     value="<%= time %>"
50     id="<%= day %>-<%= time %>-cb"
51     class="d-none">
52   <label for="schedule[<%= day %>_times][]">
53     Fixed <%= day %> <%= time %>
54   </label>
55
56   <% end %>
57 <% end %>
58
59   <% for time in Schedule.times %>
60     <% for day in Schedule.weekdays %>
61
62       <input type="checkbox"
63         name="schedule[<%= day %>_var_times][]"
64         value="<%= time %>"
65         id="<%= day %>-<%= time %>-var-cb"
66         class="d-none">
67       <label for="schedule[<%= day %>_var_times][]">
68         Variable <%= day %> <%= time %>
69       </label>
70
71     <% end %>
72   <% end %>
```