Peter Generao
Anthony Shao
Jeff Park
Alex Mutwiri
King Arthur Alagao
Sabrina Suhair

Homework 8: Accessibility Testing

What was your experience after trying out a screen reader? What did you learn? Try one site, *not your project site*, with a screen reader. What site was it? How easy or hard was it to navigate? Now, try *your own project site*, with a screen reader. What site was it? How easy or hard was it to navigate?

Jeff Park: I felt like I'm not getting the equal amount of information as compared to my ordinary experience. However, I realized with careful considerations made by the developers in the process would provide enough information for me to use the service. I tried cnn.com. It was very hard at first with huge amount of information but not too bad as I get used to. Our site was hard to navigate as we were not considerate enough for accessibility.

Sabrina Suhair: The screen reader can be quite tricky and frustrating to use, especially for a first time user like myself. I tried accessing my gmail and browsing my email using the screen reader. I found it quite tiresome to have to keep clicking the commands to navigate through the page (ctrl + option + right arrow). I learned how to use the screen reader tool to go through my inbox and actually see the contents of an email. It took about 10 minutes for me to figure out what commands to use to view the emails in my inbox (opening the email I actually wanted to view was quite tricky but I found that you need to click the enter key rather than ctrl + option + space to view a specific email). For my team's own project site, I tried navigating through the My Projects page and the Project Information page (apps/:app_id). I also managed to send an App Edit Request using the screen reader. The main issue that I observed was that the checkboxes to filter the projects in the My Projects page did not mention what the checkboxes represent, the only thing that is voiced is "checked checkbox"/ "unchecked checkbox". Overall, I think the tool is very useful (especially for people with visual impairment) and not terribly difficult to use once you get the hang of it.

King Arthur Alagao: Using the voice-over was frustrating at first. It took me some time to learn the right keys to use when navigating a website. I learned that navigating websites with the help of voice-over is very useful, but at the same time, has a steep learning curve. I tried screen reader on the actual github page corresponding to this homework: https://github.com/saasbook/hw-accessibility-audits. It was pretty hard to navigate at first. I got stuck on the github banner for about 5 minutes. I had to use my touchpad just to get out of it. I also tried the screen reader on my team's application: https://esaas-demo.herokuapp.com/. Navigating was easier, because by the time that I accessed it I already knew what keys to press.

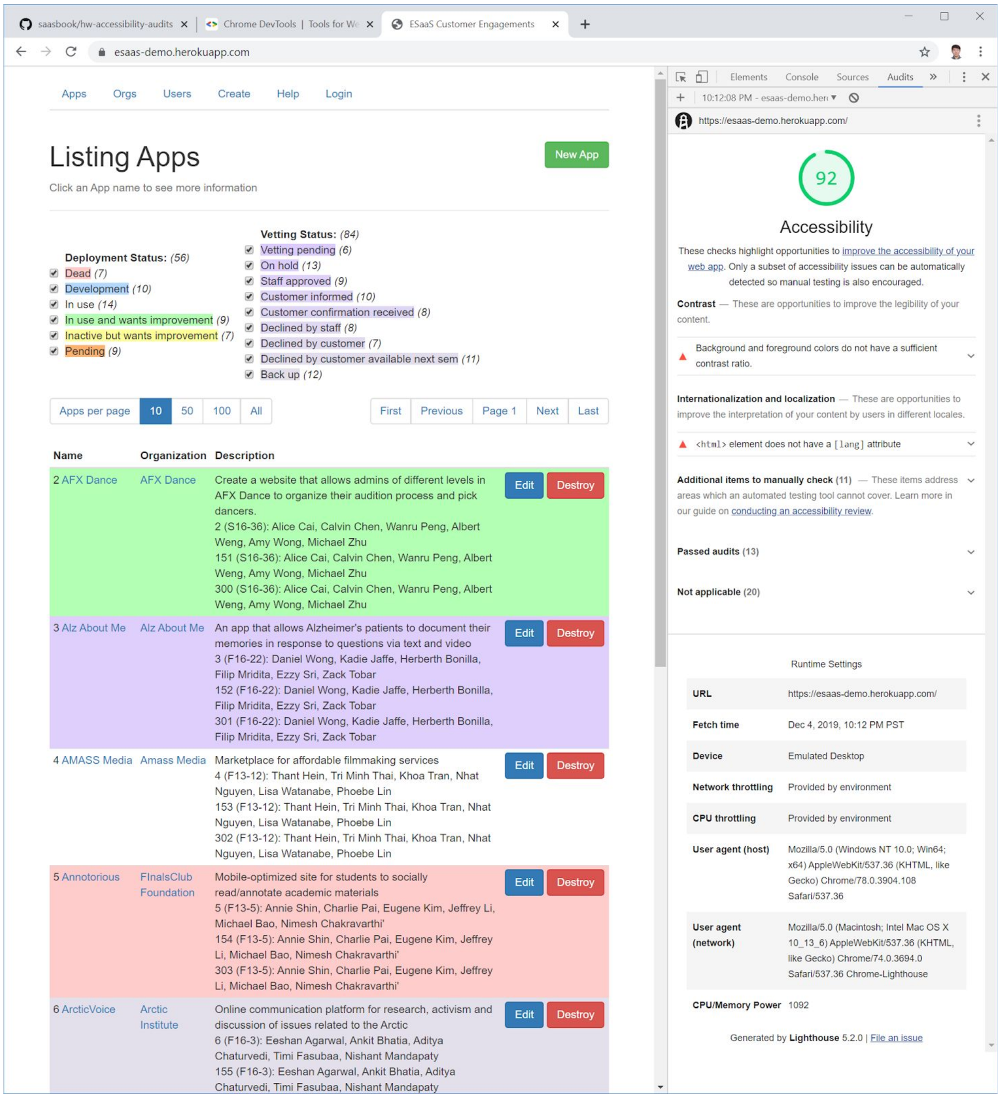
Anthony Shao: Using the screen reader to navigate through web pages was really difficult at first. I tried using the voice-over on bCourses, and as I was navigating through the site, I had to google search commands for how to move from one place to another and access different elements in each view. For our app, I used voice-over on myprojects index page. The first issue I found is the checkboxes for app status. When selecting a specific status, the description of the status aren't voiced. Also, in the description column of the app table, although the descriptions are voiced, because there is an empty line between app's description and student team information, the voice over will only read the text but not tell what it's reading is the team information.

Peter Generao: Using the screen reader to browse through the web pages was pretty hard to use at first; however, there are a lot of different commands that a user may not know about when they are first using the tool. Like King said, there is a pretty steep learning curve towards using these kinds of tools, and I have gotten stuck on pages more than once. I tried using Wikipedia at first, and it was kind of hard for me to just get it to do what I wanted it to do. I noticed that the ability to select certain parts of a page and highlighting them was very useful, and prevented me from getting lost (even without the voice-over portion). On ESaaS Engagements (i.e. our project site), it was also difficult to navigate; however, it was definitely easier since I had practice on Wikipedia. The search bar at the top also did not say anything for the navigation bar since it did not have a label, but we fixed that after adding our accessibility changes.

Alex Mutwiri: I did not know about the existence of such a program in my MacOS so it was a pleasant surprise to learn about how to use a screen reader to navigate a page. The screen reader even reads notifications that show up in the notification panel like slack messages. I learnt how to use a screen reader to navigate a page. This new skill will help me learn how to test websites for a11y against screen readers. I tried the screen reader VoiceOver on BCourses website. I was surprised that he website was seemingly optimized for access through a screen reader. However, some items on the page seemed to have been nested and the screen reader would end up saying something weird like "Close inside close" for a <span> close </span> nested in an anchor with title attribute = "close". However, I was surprised that the screen reader was able to detect that the notification bubble with 5 meant there were 5 unread notifications. It was not at all hard to navigate the site. I was able to pick-up the controls for the VoiceOver quickly and managed to navigate through a couple of links on Bcourses. Our site was harder to navigate before fixing it but we made changes to make it more accessible.

What did the accessibility tests from Audits tab in Chrome report? * Include the route in your report along with screen shots. What did the tests from Accessibility Insights "Fast Pass" report?

/ (root) : Chrome Audits - 92 / MS Insights - contrast is not sufficient, and there exists tab stops


saasbook/hw-accessibility-audits
Chrome DevTools | Tools for We
ESaaS Customer Engagements
esaas-demo.herokuapp.com
Elements
Console
Sources
Audits
10:12:08 PM - esaas-demo.her
https://esaas-demo.herokuapp.com/

Apps Orgs Users Create Help Login

Listing Apps

New App

Click an App name to see more information

Deployment Status: *(56)*

- ☑ Dead *(7)*
- ☑ Development *(10)*
- ☑ In use *(14)*
- ☑ In use and wants improvement *(9)*
- ☑ Inactive but wants improvement *(7)*
- ☑ Pending *(9)*

Vetting Status: *(84)*

- ☑ Vetting pending *(6)*
- ☑ On hold *(13)*
- ☑ Staff approved *(9)*
- ☑ Customer informed *(10)*
- ☑ Customer confirmation received *(8)*
- ☑ Declined by staff *(8)*
- ☑ Declined by customer *(7)*
- ☑ Declined by customer available next sem *(11)*
- ☑ Back up *(12)*

Apps per page | 10 | 50 | 100 | All

First | Previous | Page 1 | Next | Last

Name	Organization	Description	
2 AFX Dance	AFX Dance	Create a website that allows admins of different levels in AFX Dance to organize their audition process and pick dancers. 2 (S16-36): Alice Cai, Calvin Chen, Wanru Peng, Albert Weng, Amy Wong, Michael Zhu 151 (S16-36): Alice Cai, Calvin Chen, Wanru Peng, Albert Weng, Amy Wong, Michael Zhu 300 (S16-36): Alice Cai, Calvin Chen, Wanru Peng, Albert Weng, Amy Wong, Michael Zhu	Edit Destroy
3 Alz About Me	Alz About Me	An app that allows Alzheimer's patients to document their memories in response to questions via text and video 3 (F16-22): Daniel Wong, Kadie Jaffe, Herberth Bonilla, Filip Mridita, Ezzy Sri, Zack Tobar 152 (F16-22): Daniel Wong, Kadie Jaffe, Herberth Bonilla, Filip Mridita, Ezzy Sri, Zack Tobar 301 (F16-22): Daniel Wong, Kadie Jaffe, Herberth Bonilla, Filip Mridita, Ezzy Sri, Zack Tobar	Edit Destroy
4 AMASS Media	Amass Media	Marketplace for affordable filmmaking services 4 (F13-12): Thant Hein, Tri Minh Thai, Khoa Tran, Nhat Nguyen, Lisa Watanabe, Phoebe Lin 153 (F13-12): Thant Hein, Tri Minh Thai, Khoa Tran, Nhat Nguyen, Lisa Watanabe, Phoebe Lin 302 (F13-12): Thant Hein, Tri Minh Thai, Khoa Tran, Nhat Nguyen, Lisa Watanabe, Phoebe Lin	Edit Destroy
5 Annotorious	FInalsClub Foundation	Mobile-optimized site for students to socially read/annotate academic materials 5 (F13-5): Annie Shin, Charlie Pai, Eugene Kim, Jeffrey Li, Michael Bao, Nimesh Chakravarthi' 154 (F13-5): Annie Shin, Charlie Pai, Eugene Kim, Jeffrey Li, Michael Bao, Nimesh Chakravarthi' 303 (F13-5): Annie Shin, Charlie Pai, Eugene Kim, Jeffrey Li, Michael Bao, Nimesh Chakravarthi'	Edit Destroy
6 ArcticVoice	Arctic Institute	Online communication platform for research, activism and discussion of issues related to the Arctic 6 (F16-3): Eeshan Agarwal, Ankit Bhatia, Aditya Chaturvedi, Timi Fasubaa, Nishant Mandapaty 155 (F16-3): Eeshan Agarwal, Ankit Bhatia, Aditya Chaturvedi, Timi Fasubaa, Nishant Mandapaty	Edit Destroy

92

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

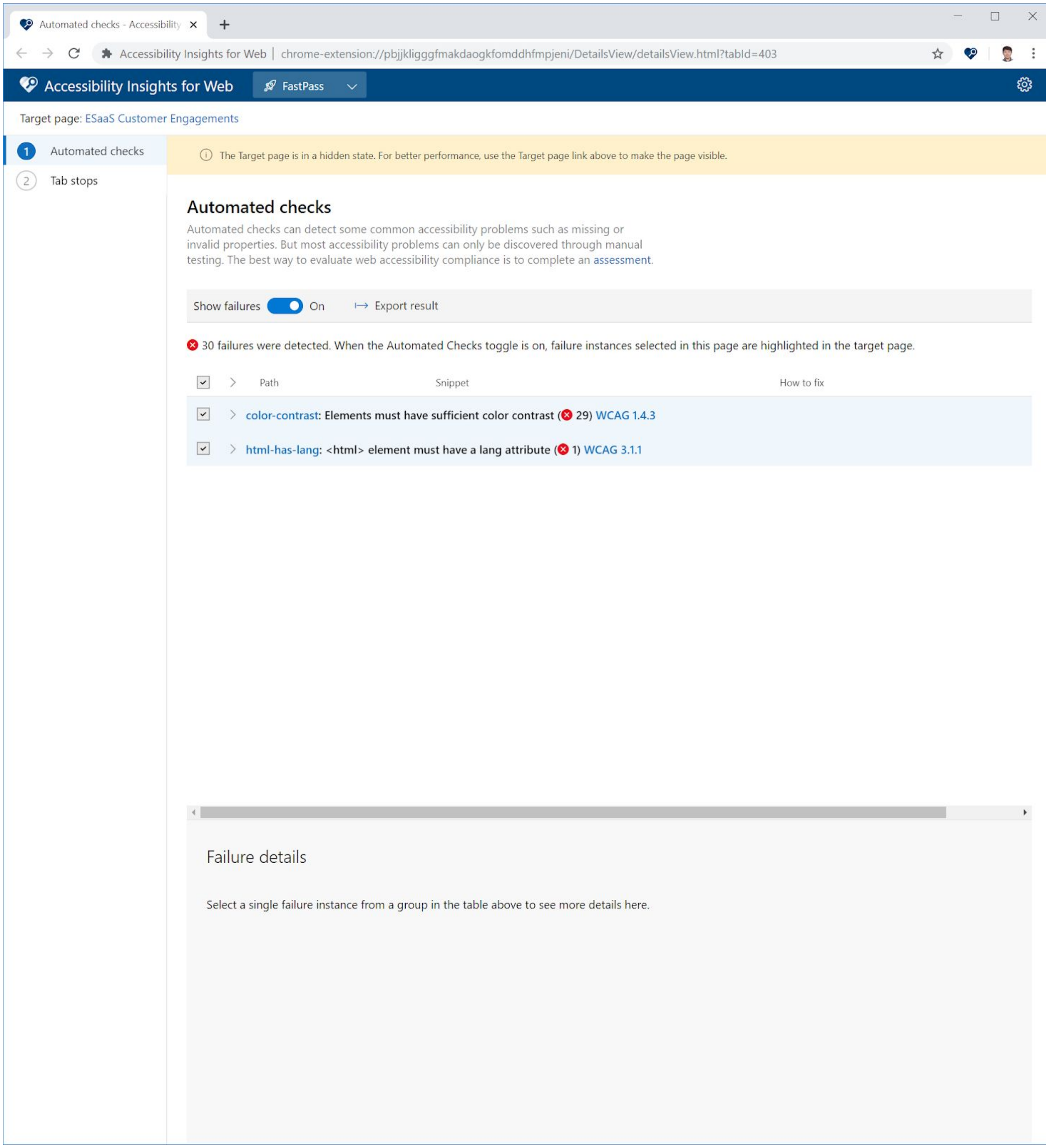
Passed audits (13)

Not applicable (20)

Runtime Settings

URL	https://esaas-demo.herokuapp.com/
Fetch time	Dec 4, 2019, 10:12 PM PST
Device	Emulated Desktop
Network throttling	Provided by environment
CPU throttling	Provided by environment
User agent (host)	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36
User agent (network)	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3694.0 Safari/537.36 Chrome-Lighthouse
CPU/Memory Power	1092

Generated by **Lighthouse** 5.2.0 | File an issue


Automated checks - Accessibility
Accessibility Insights for Web | chrome-extension://pbjjkligggfmakdaogkfomddhfmpjeni/DetailsView/detailsView.html?tabId=403
Accessibility Insights for Web
FastPass
1 Automated checks
2 Tab stops

Target page: ESaaS Customer Engagements

ⓘ The Target page is in a hidden state. For better performance, use the Target page link above to make the page visible.

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

⊗ 30 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

☑ > Path Snippet How to fix

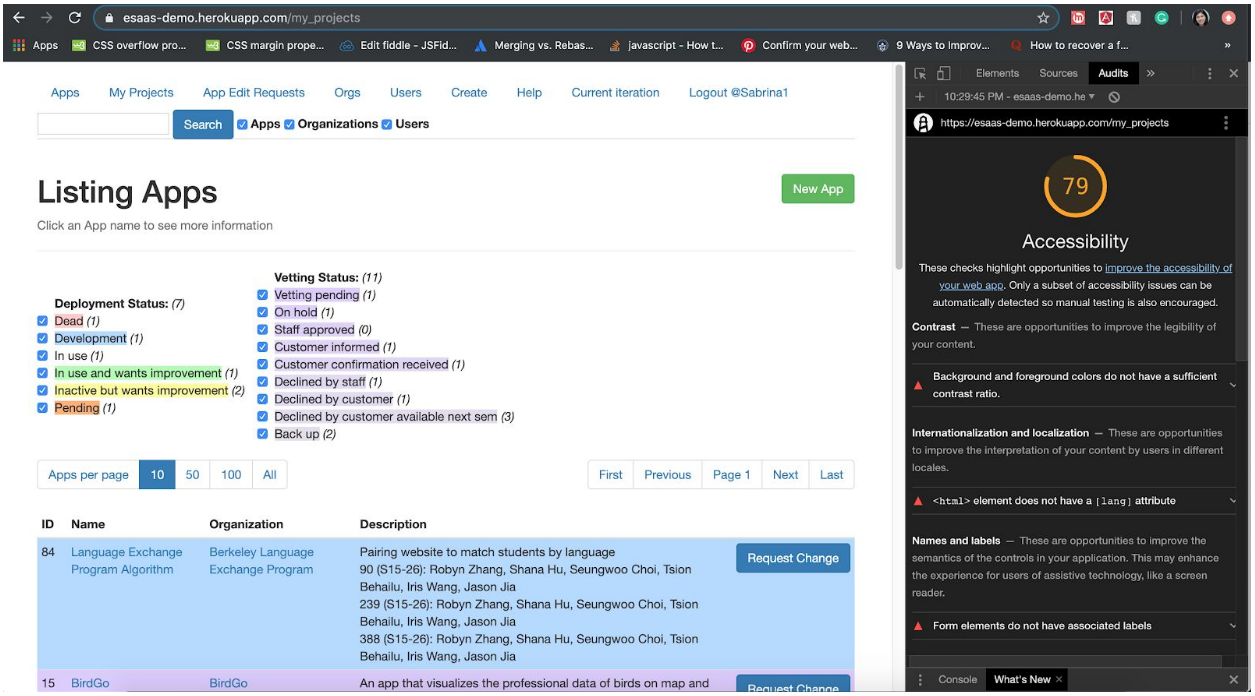
☑ > color-contrast: Elements must have sufficient color contrast (⊗ 29) WCAG 1.4.3

☑ > html-has-lang: <html> element must have a lang attribute (⊗ 1) WCAG 3.1.1

Failure details

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

/ my_projects : Chrome Audits - Accessibility score: 79 /Microsoft Accessibility Insight - contrast is not sufficient, and form elements (search box) do not have associated labels



Listing Apps

Click an App name to see more information

Deployment Status: *(7)*

- ☑ Dead *(1)*
- ☑ Development *(1)*
- ☑ In use *(1)*
- ☑ In use and wants improvement *(1)*
- ☑ Inactive but wants improvement *(2)*
- ☑ Pending *(1)*

Vetting Status: *(11)*

- ☑ Vetting pending *(1)*
- ☑ On hold *(1)*
- ☑ Staff approved *(0)*
- ☑ Customer informed *(1)*
- ☑ Customer confirmation received *(1)*
- ☑ Declined by staff *(1)*
- ☑ Declined by customer *(1)*
- ☑ Declined by customer available next sem *(3)*
- ☑ Back up *(2)*

Apps per page | 10 | 50 | 100 | All

First | Previous | Page 1 | Next | Last

ID	Name	Organization	Description	
84	Language Exchange Program Algorithm	Berkeley Language Exchange Program	Pairing website to match students by language 90 (S15-26): Robyn Zhang, Shana Hu, Seungwoo Choi, Tsion Behailu, Iris Wang, Jason Jia 239 (S15-26): Robyn Zhang, Shana Hu, Seungwoo Choi, Tsion Behailu, Iris Wang, Jason Jia 388 (S15-26): Robyn Zhang, Shana Hu, Seungwoo Choi, Tsion Behailu, Iris Wang, Jason Jia	Request Change
15	BirdGo	BirdGo	An app that visualizes the professional data of birds on map and	Request Change

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Contrast — These are opportunities to improve the legibility of your content.

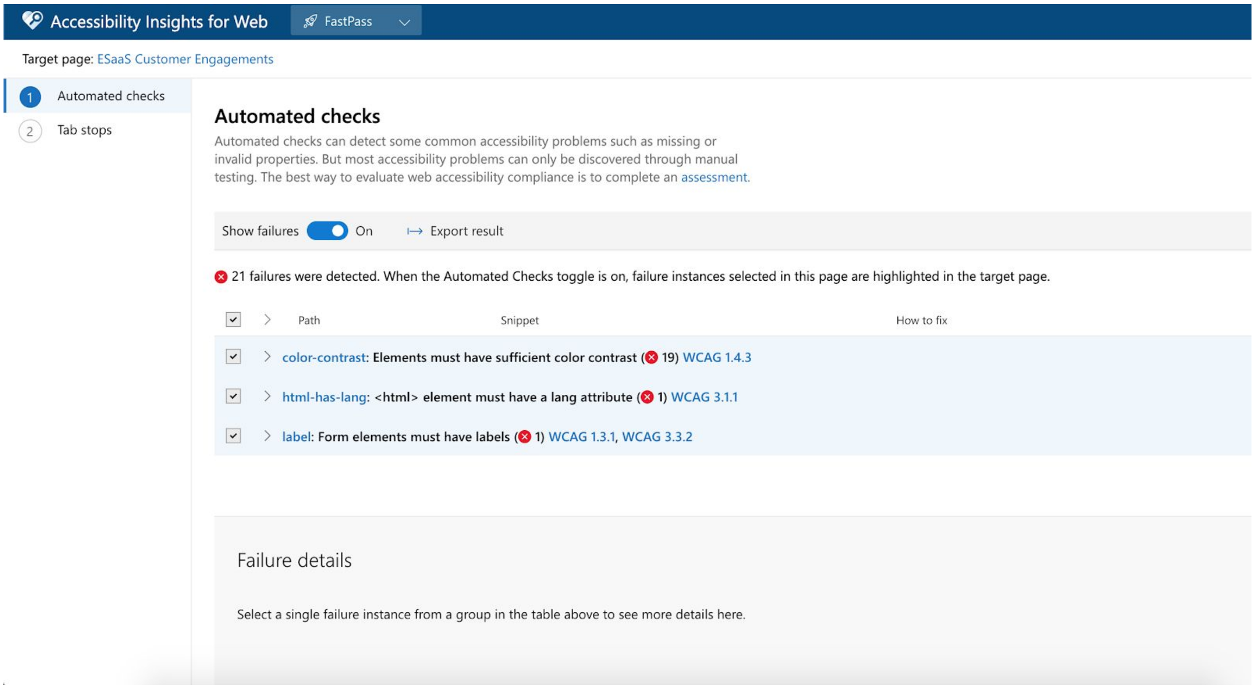
▲ Background and foreground colors do not have a sufficient contrast ratio.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels



Target page: ESaaS Customer Engagements

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

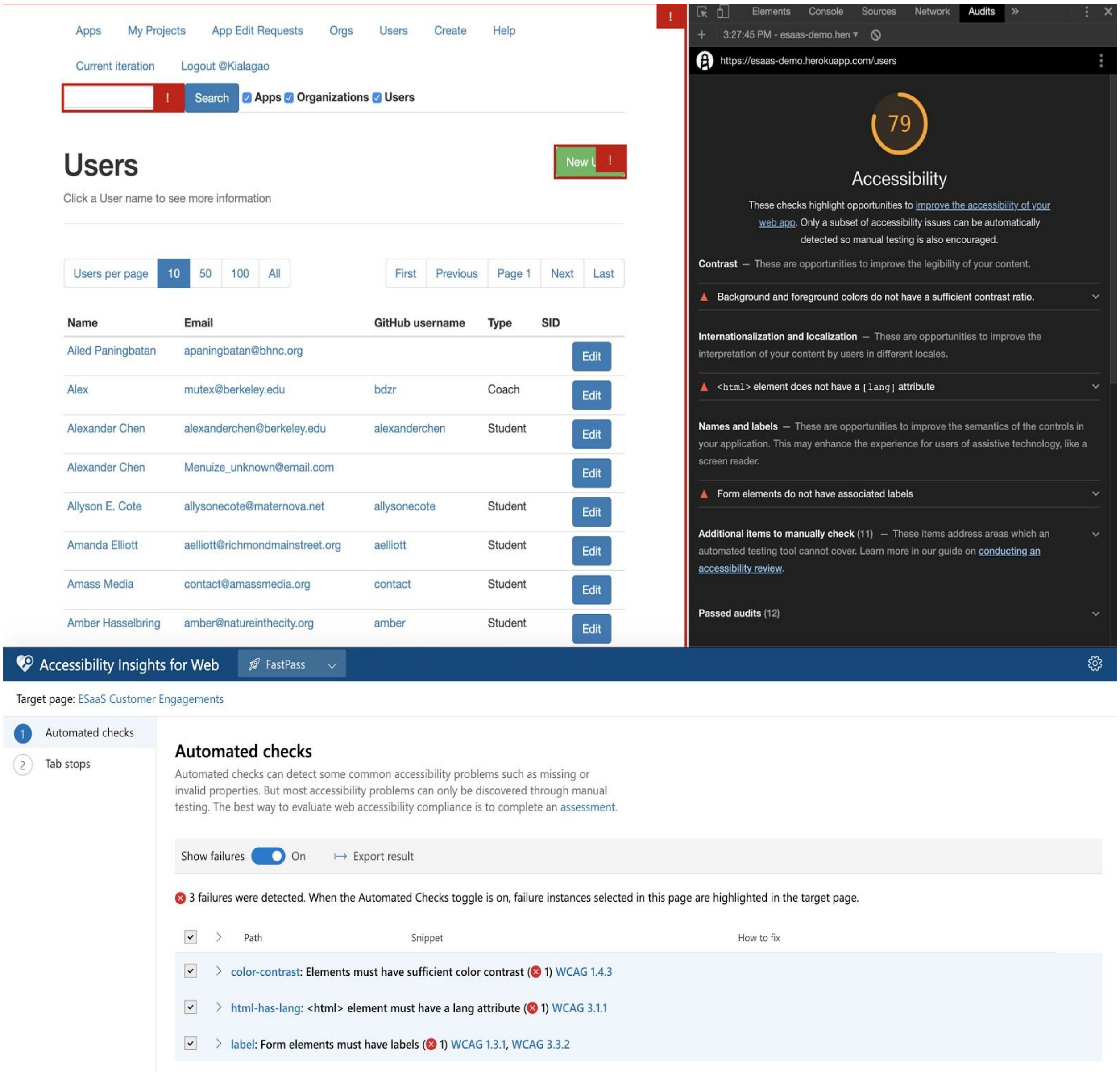
21 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

☑	Path	Snippet	How to fix
☑	color-contrast: Elements must have sufficient color contrast (☒ 19) WCAG 1.4.3		
☑	html-has-lang: <html> element must have a lang attribute (☒ 1) WCAG 3.1.1		
☑	label: Form elements must have labels (☒ 1) 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.

/users : Chrome Audits - Accessibility score: 79 /Microsoft Accessibility Insight - contrast is not sufficient, and form elements (search box) do not have associated labels



Users

Click a User name to see more information

Name	Email	GitHub username	Type	SID	
Ailed Paningbatan	apaningbatan@bhnc.org				Edit
Alex	mutex@berkeley.edu	bdzr	Coach		Edit
Alexander Chen	alexanderchen@berkeley.edu	alexanderchen	Student		Edit
Alexander Chen	Menuize_unknown@email.com				Edit
Allyson E. Cote	allysonecote@maternova.net	allysonecote	Student		Edit
Amanda Elliott	aelliott@richmondmainstreet.org	aelliott	Student		Edit
Amass Media	contact@amassmedia.org	contact	Student		Edit
Amber Hasselbring	amber@natureinthecity.org	amber	Student		Edit

Elements
Console
Sources
Network
Audits
3:27:45 PM - esaas-demo.her ▾
https://esaas-demo.herokuapp.com/users
79

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels

Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (12)

Accessibility Insights for Web

Target page: ESaaS Customer Engagements

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

⊗ 3 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

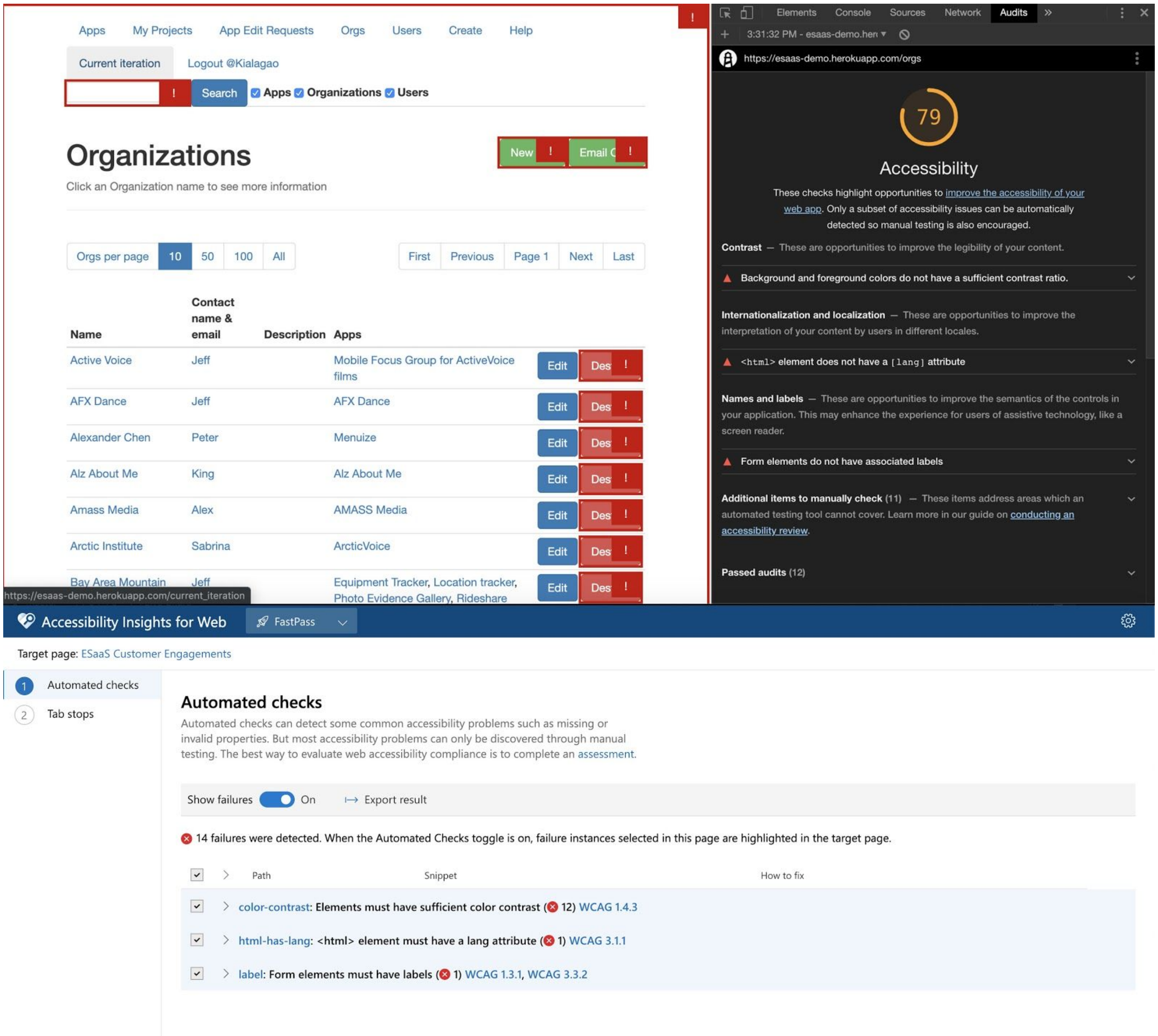
☑ > Path Snippet How to fix

☑ > color-contrast: Elements must have sufficient color contrast (⊗ 1) WCAG 1.4.3

☑ > html-has-lang: <html> element must have a lang attribute (⊗ 1) WCAG 3.1.1

☑ > label: Form elements must have labels (⊗ 1) WCAG 1.3.1, WCAG 3.3.2

/orgs: Chrome Audits - Accessibility score: 79 /Microsoft Accessibility Insight - contrast is not sufficient, and form elements (search box) do not have associated labels



Organizations

Click an Organization name to see more information

Name	Contact name & email	Description	Apps		
Active Voice	Jeff		Mobile Focus Group for ActiveVoice films	Edit	Des
AFX Dance	Jeff		AFX Dance	Edit	Des
Alexander Chen	Peter		Menuize	Edit	Des
Alz About Me	King		Alz About Me	Edit	Des
Amass Media	Alex		AMASS Media	Edit	Des
Arctic Institute	Sabrina		ArcticVoice	Edit	Des
Bay Area Mountain	Jeff		Equipment Tracker, Location tracker, Photo Evidence Gallery, Rideshare	Edit	Des

https://esaas-demo.herokuapp.com/current_iteration

Elements
Console
Sources
Network
Audits
3:31:32 PM - esaas-demo.her
https://esaas-demo.herokuapp.com/orgs
79

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels

Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (12)

Accessibility Insights for Web

Target page: ESaaS Customer Engagements

Automated checks

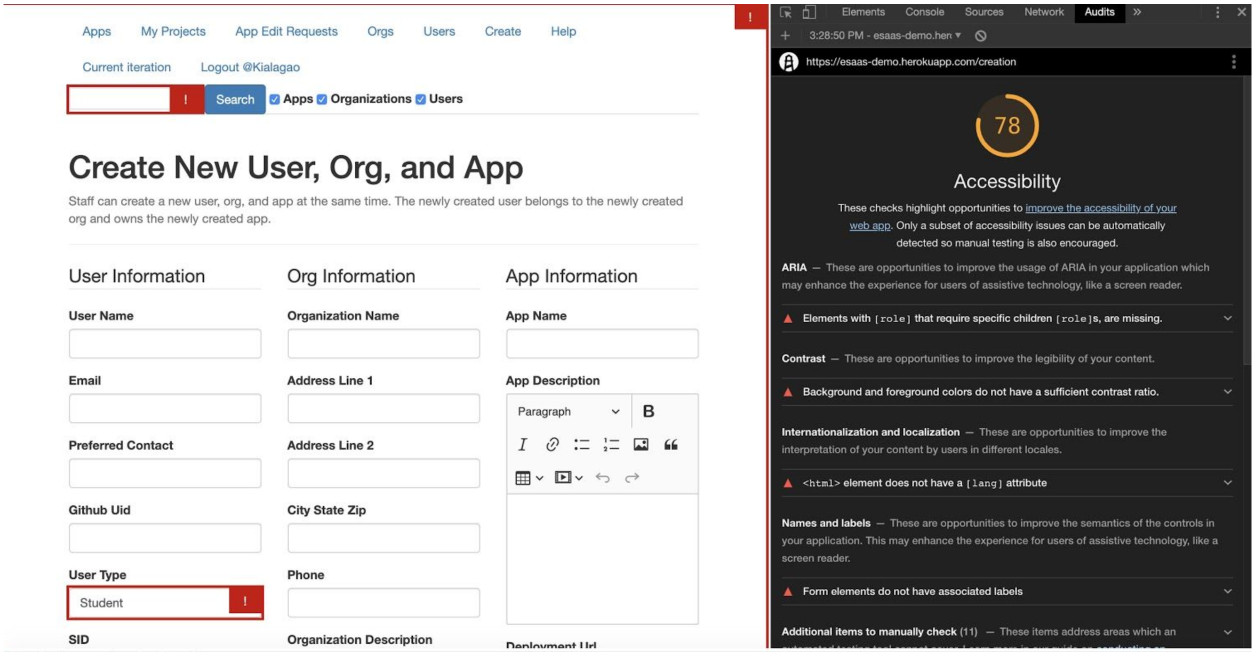
Automated checks can detect some common accessibility problems such as missing or invalid properties. But most accessibility problems can only be discovered through manual testing. The best way to evaluate web accessibility compliance is to complete an assessment.

Show failures On

⊗ 14 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

☑	Path	Snippet	How to fix
☑	color-contrast: Elements must have sufficient color contrast (⊗ 12) WCAG 1.4.3		
☑	html-has-lang: <html> element must have a lang attribute (⊗ 1) WCAG 3.1.1		
☑	label: Form elements must have labels (⊗ 1) WCAG 1.3.1, WCAG 3.3.2		

/creation: Chrome Audits - Accessibility score: 78 /Microsoft Accessibility Insight - contrast is not sufficient, and form elements (search box) do not have associated labels



Create New User, Org, and App

Staff can create a new user, org, and app at the same time. The newly created user belongs to the newly created org and owns the newly created app.

User Information	Org Information	App Information
User Name	Organization Name	App Name
Email	Address Line 1	App Description
Preferred Contact	Address Line 2	
Github Uid	City State Zip	
User Type: Student	Phone	
SID	Organization Description	Deployment Url

https://esaas-demo.herokuapp.com/creation

78

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ARIA — These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

▲ Elements with `[role]` that require specific children `[role]`s, are missing.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

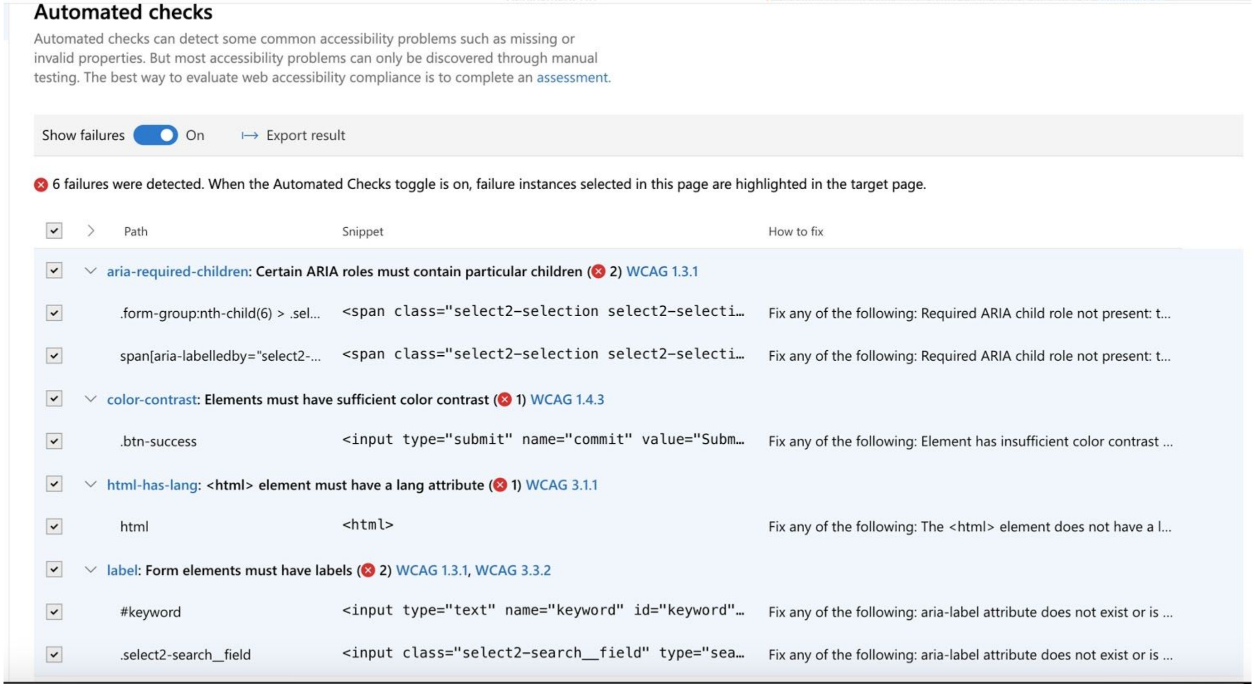
Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels

Additional items to manually check (11) — These items address areas which an



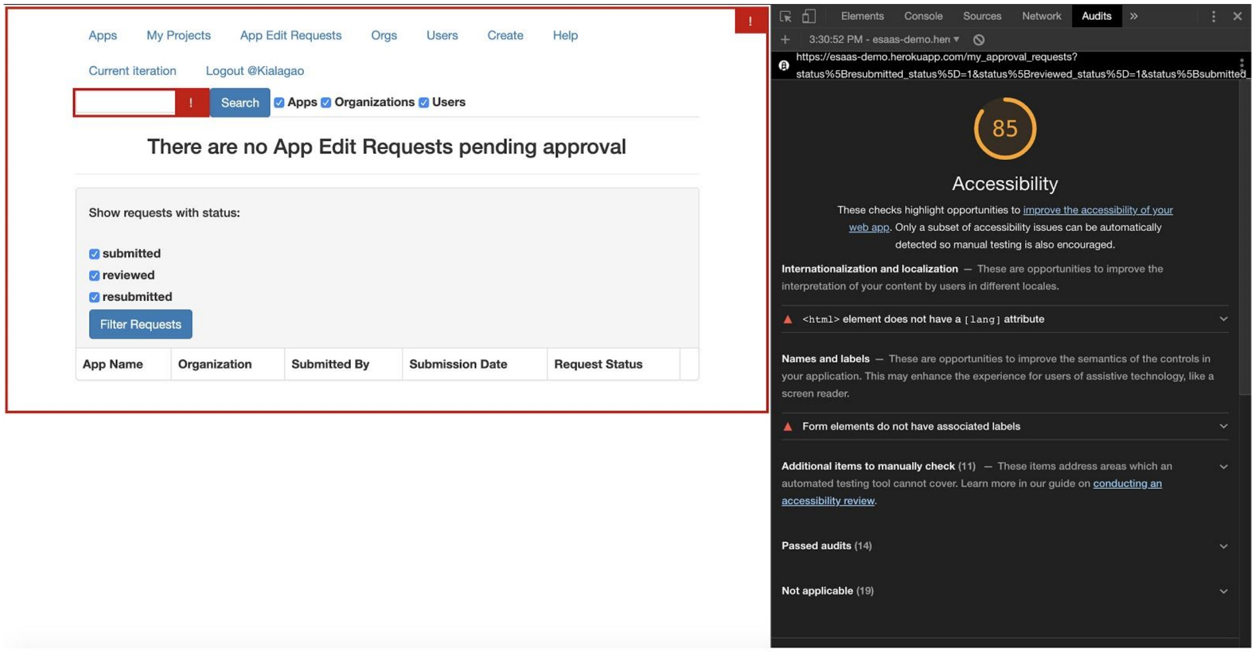
Automated checks

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.

⊗ 6 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

Path	Snippet	How to fix
aria-required-children: Certain ARIA roles must contain particular children (⊗ 2) WCAG 1.3.1		
.form-group:nth-child(6) > .sel...	`<span class="select2-selection select2-selecti...`	Fix any of the following: Required ARIA child role not present: t...
span[aria-labelledby="select2-...	`<span class="select2-selection select2-selecti...`	Fix any of the following: Required ARIA child role not present: t...
color-contrast: Elements must have sufficient color contrast (⊗ 1) WCAG 1.4.3		
.btn-success	`<input type="submit" name="commit" value="Subm...`	Fix any of the following: Element has insufficient color contrast ...
html-has-lang: <html> element must have a lang attribute (⊗ 1) WCAG 3.1.1		
html	`<html>`	Fix any of the following: The <html> element does not have a l...
label: Form elements must have labels (⊗ 2) WCAG 1.3.1, WCAG 3.3.2		
#keyword	`<input type="text" name="keyword" id="keyword"...`	Fix any of the following: aria-label attribute does not exist or is ...
.select2-search__field	`<input class="select2-search__field" type="sea...`	Fix any of the following: aria-label attribute does not exist or is ...

/appeditrequest: Chrome Audits - Accessibility score: 85 /Microsoft Accessibility Insight - Form elements (search box) do not have associated labels and missing attr.



There are no App Edit Requests pending approval

Show requests with status:

☑ submitted
☑ reviewed
☑ resubmitted

Filter Requests

App Name	Organization	Submitted By	Submission Date	Request Status

3:30:52 PM - esaas-demo.herokuapp.com/my_approval_requests?status%5Bresubmitted_status%5D=1&status%5Breviewed_status%5D=1&status%5Bsubmitted

85

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

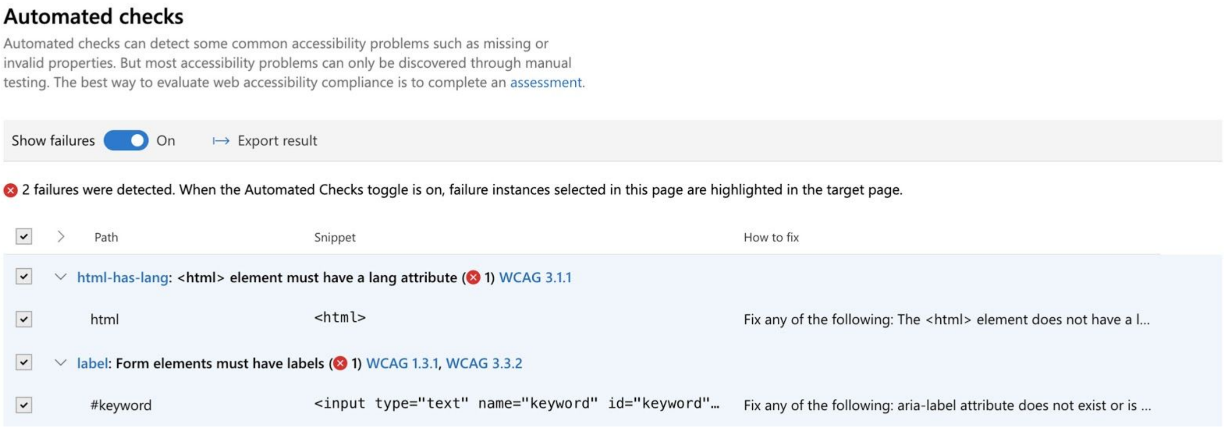
Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels

Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (14)

Not applicable (19)



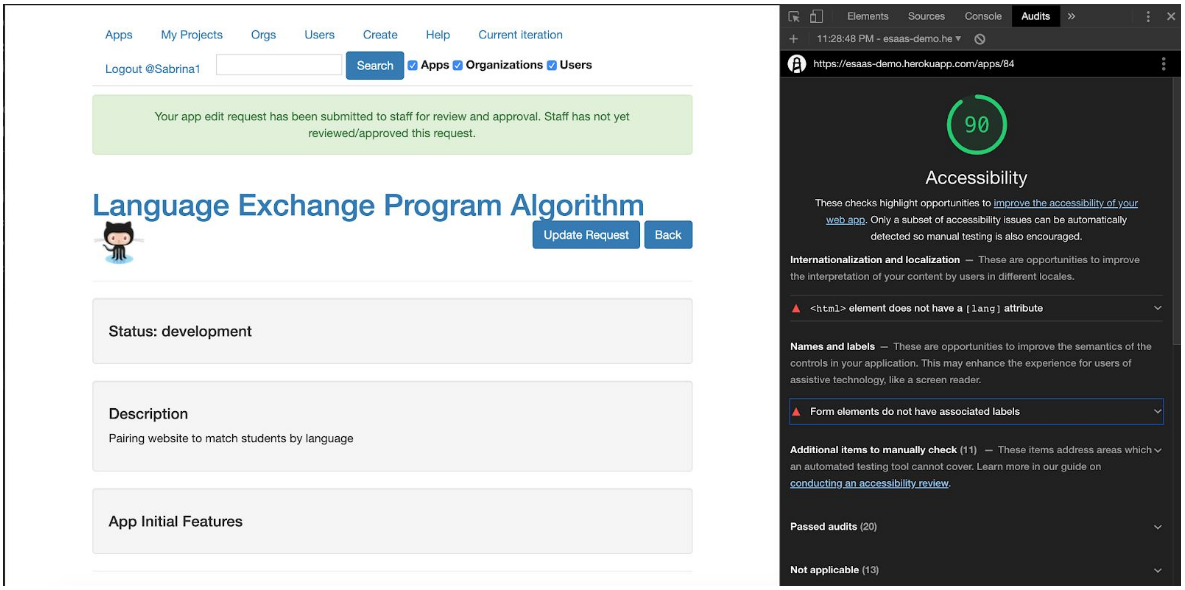
Automated checks

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.

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

☑	Path	Snippet	How to fix
☑	html-has-lang: <html> element must have a lang attribute (1) WCAG 3.1.1		
☑	html	`<html>`	Fix any of the following: The <html> element does not have a l...
☑	label: Form elements must have labels (1) WCAG 1.3.1, WCAG 3.3.2		
☑	#keyword	`<input type="text" name="keyword" id="keyword"…`	Fix any of the following: aria-label attribute does not exist or is ...

/apps/:app_id: Chrome Audits - Accessibility score: 90 /Microsoft Accessibility Insight - Form elements (search box) do not have associated labels and missing [lang] attr.



Your app edit request has been submitted to staff for review and approval. Staff has not yet reviewed/approved this request.

Language Exchange Program Algorithm

Update Request | Back

Status: development

Description

Pairing website to match students by language

App Initial Features

90

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

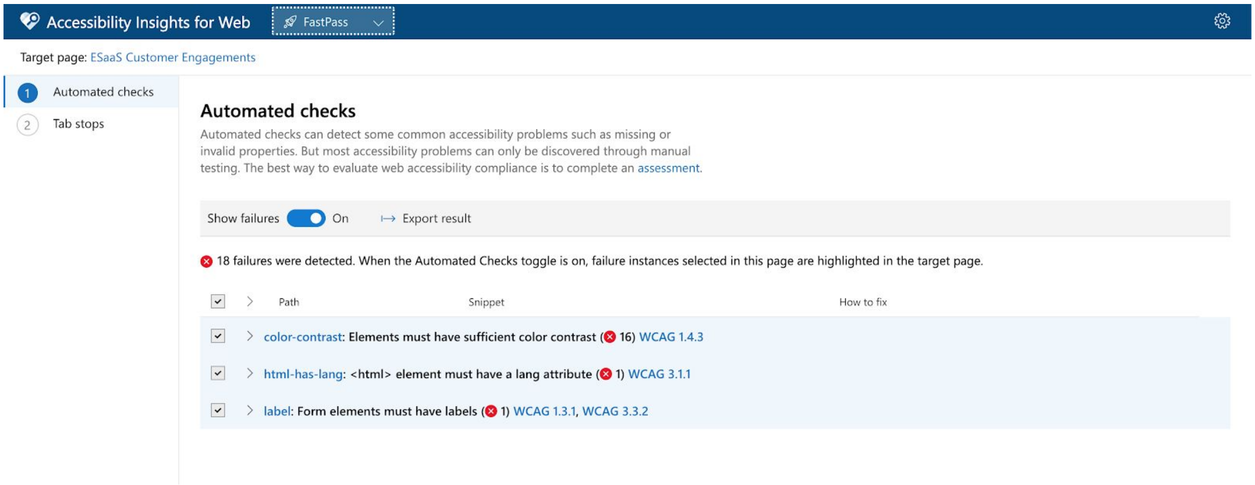
Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Form elements do not have associated labels

Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (20)

Not applicable (13)



Target page: ESaaS Customer Engagements

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

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

	Path	Snippet	How to fix
☑	color-contrast: Elements must have sufficient color contrast (16) WCAG 1.4.3		
☑	html-has-lang: <html> element must have a lang attribute (1) WCAG 3.1.1		
☑	label: Form elements must have labels (1) WCAG 1.3.1, WCAG 3.3.2		

Run the more complex "Assessment" from Accessibility Insights. Run this on a different page, if possible. Ideally, the most complex page in your application. This will aid you in manually assessing the compliance of your page. Were you surprised by these tests? Do you disagree with any of the results? Are there any errors that appear on multiple pages? What are they? How might you fix them?

/engagements/23/iterations/2/edit

I was surprised by the “Repetitive content” tests. I agree with the result while I hadn’t thought about it before the test. The tests were designed to check coherency between similar pages, and for a few pages we were not positioning the buttons at the same or relatively similar location. Indeed, I realized accessibility should be enforced across the pages not only on a single page. I would fix the issue by repositioning the common buttons at the same or relatively similar locations between pages.

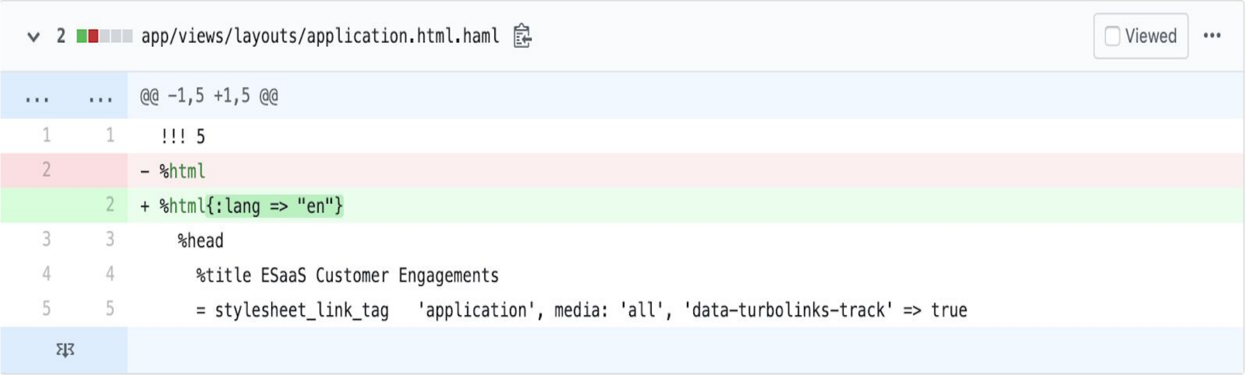
<u>Pick 2 (non alt-text related) errors to address.</u>

<u>What was the error you are addressing? How did you address it? In your report, include a sample of before-and-after code.</u>

Error #1: html element missing a lang tag

Fix: We added a lang tag and assigned “en” to its attribute.

Before-and-After Code

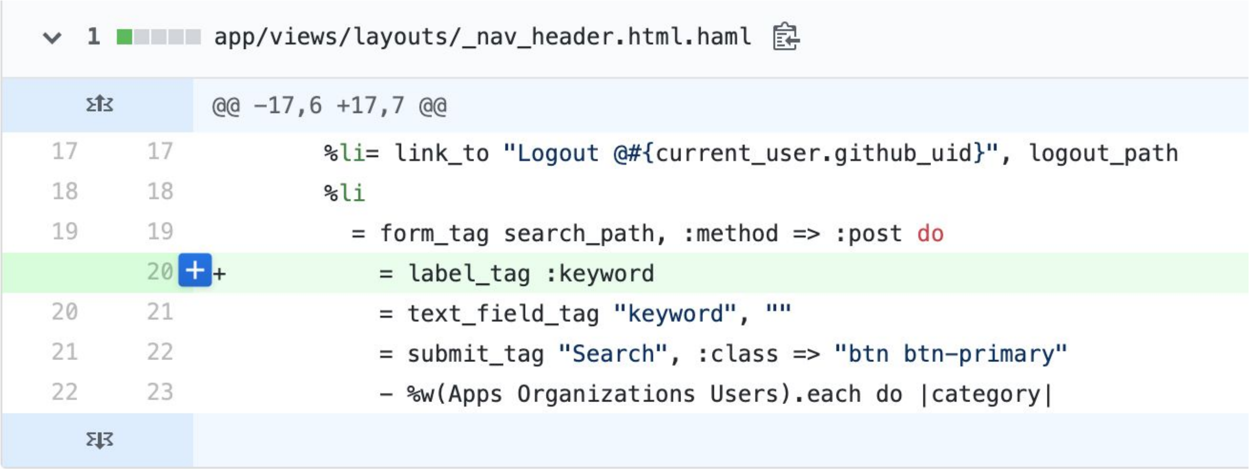


```
@@ -1,5 +1,5 @@
  !!! 5
- %html
+ %html{:lang => "en"}
    %head
      %title ESaaS Customer Engagements
      = stylesheet_link_tag   'application', media: 'all', 'data-turbolinks-track' => true
```

Error #2: A text field in one of our html files was missing a label tag

Fix: We added a label tag and assigned the id “keyword” to its attribute

Before-and-After Code

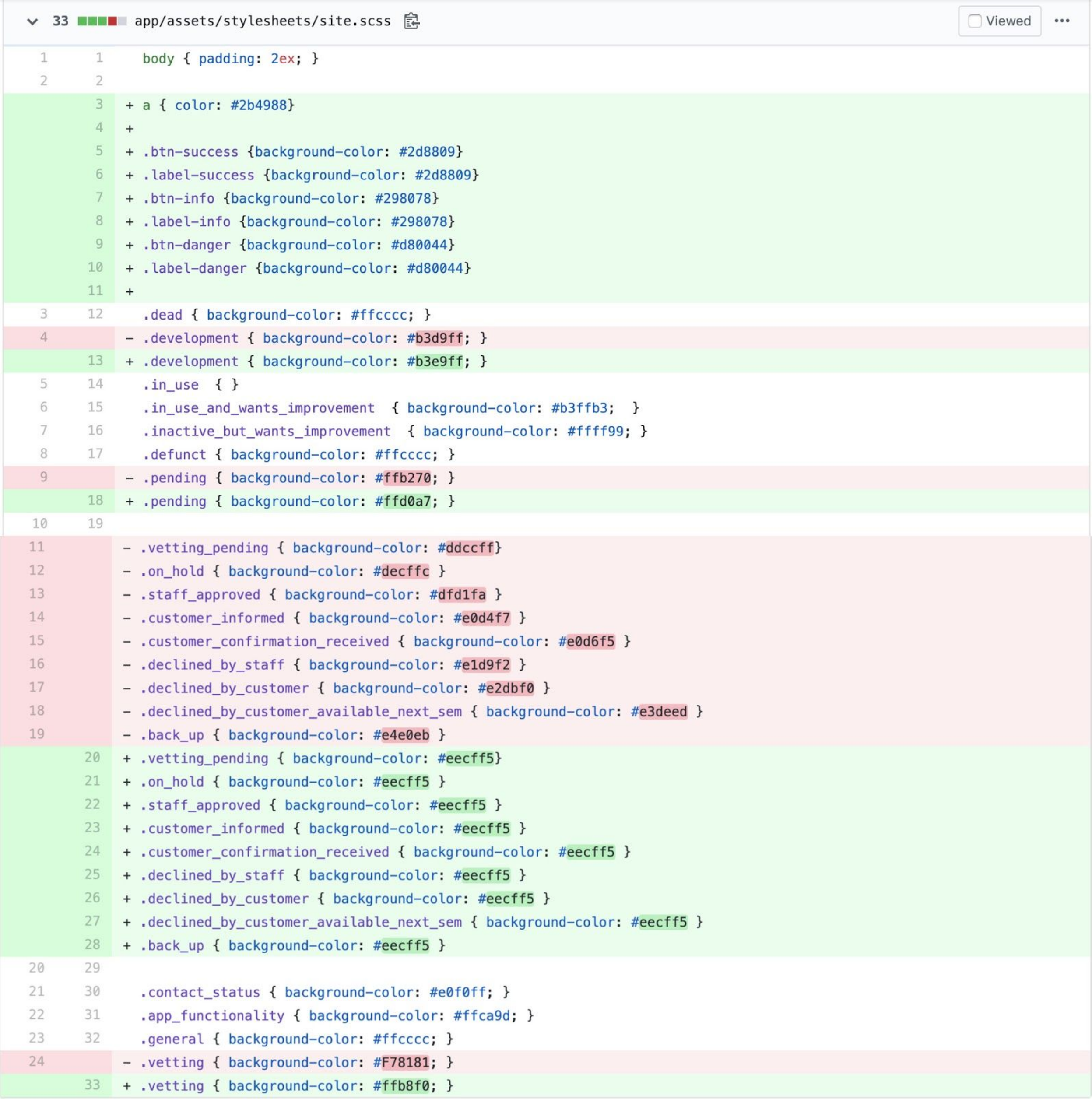


```
@@ -17,6 +17,7 @@
            %li= link_to "Logout @#{current_user.github_uid}", logout_path
            %li
              = form_tag search_path, :method => :post do
+               = label_tag :keyword
                = text_field_tag "keyword", ""
                = submit_tag "Search", :class => "btn btn-primary"
                - %w(Apps Organizations Users).each do |category|
```

Error #3: The color contrast ratio on our application was low

Fix: Changed the background color of table rows and buttons.

Before-and-After Code



```
  body { padding: 2ex; }

+ a { color: #2b4988}
+
+ .btn-success {background-color: #2d8809}
+ .label-success {background-color: #2d8809}
+ .btn-info {background-color: #298078}
+ .label-info {background-color: #298078}
+ .btn-danger {background-color: #d80044}
+ .label-danger {background-color: #d80044}
+
  .dead { background-color: #ffcccc; }
- .development { background-color: #b3d9ff; }
+ .development { background-color: #b3e9ff; }
  .in_use  { }
  .in_use_and_wants_improvement  { background-color: #b3ffb3;  }
  .inactive_but_wants_improvement  { background-color: #ffff99; }
  .defunct { background-color: #ffcccc; }
- .pending { background-color: #ffb270; }
+ .pending { background-color: #ffd0a7; }

- .vetting_pending { background-color: #ddccff}
- .on_hold { background-color: #decffc }
- .staff_approved { background-color: #dfd1fa }
- .customer_informed { background-color: #e0d4f7 }
- .customer_confirmation_received { background-color: #e0d6f5 }
- .declined_by_staff { background-color: #e1d9f2 }
- .declined_by_customer { background-color: #e2dbf0 }
- .declined_by_customer_available_next_sem { background-color: #e3deed }
- .back_up { background-color: #e4e0eb }
+ .vetting_pending { background-color: #eecff5}
+ .on_hold { background-color: #eecff5 }
+ .staff_approved { background-color: #eecff5 }
+ .customer_informed { background-color: #eecff5 }
+ .customer_confirmation_received { background-color: #eecff5 }
+ .declined_by_staff { background-color: #eecff5 }
+ .declined_by_customer { background-color: #eecff5 }
+ .declined_by_customer_available_next_sem { background-color: #eecff5 }
+ .back_up { background-color: #eecff5 }

  .contact_status { background-color: #e0f0ff; }
  .app_functionality { background-color: #ffca9d; }
  .general { background-color: #ffcccc; }
- .vetting { background-color: #F78181; }
+ .vetting { background-color: #ffb8f0; }
```

We were not able to get the axe-core dependency working on our project since rails was not able to recognize the node_modules directory (we tried adding it in many other ways), which made it impossible for us to use the axe-matchers. For that reason, we have not added the hookup tools; however, we did create unit/cucumber scenarios which *would* check our accessibility scores if axe worked.

We were able to manually increase our scores for many pages to 100 on the Chrome audits for our indexes, and above 80 for the other pages.

The following cucumber scenarios should be in features/accessibility.feature:

```
Feature: Accessibility for each page

Background: seed data and logged as coach

  Given the following apps exist:
    | id  | name  | description | org_id |
    | 1   | app1  | test1       | 1      |

  And the following orgs exist:
    | id | name | contact_id |
    | 1  | org1 | 1          |

  And the following users exist:
    | id | name  | github_uid      | email          | user_type    | developing_engagement_id |
    | 1  | user1 | esaas_developer | test@user.com  | coach        |       1                  |

  And the following App Edit Requests exist:
    | id | description  |
    | 1  |    app1      |

  And the following iterations exist:
    | id | engagement_id | end_date   |
    | 1  | 1             | 2017-04-14 |

  And I am logged in

Scenario: App page
  When I follow "Apps"
  Then the page should be accessible
  When I follow "New App"
  Then the page should be accessible
  Given I press "Back"
  When I follow "Edit"
  Then the page should be accessible
  And I press "Update App"
  When I follow "app1"
  Then the page should be accessible
  When I press "New Engagement"
  Then the page should be accessible
  And I press "Back"
  And I press "Request Feedback"
  Then the page should be accessible
  When I press "Create Iteration"
  Then the page should be accessible
  When I press "Edit"
  Then the page should be accessible

Scenario: Project page
  When I follow "My Projects"
  Then the page should be accessible
  When I follow "Request Change"
  Then the page should be accessible

Scenario: App Edit Requests page
  When I follow "App Edit Requests"
  Then the page should be accessible
  When I follow "View Request"
  Then the page should be accessible

Scenario: Orgs page
  When I follow "Orgs"
  Then the page should be accessible
  When I follow "org1"
  Then the page should be accessible
  When I follow "Edit Org"
  Then the page should be accessible

Scenario: Users page
  When I follow "Users"
  Then the page should be accessible
  When I follow "user1"
  Then the page should be accessible
  When I follow "Edit User"
  Then the page should be accessible
```

```
Scenario: Create page
  When I follow "Create"
  Then the page should be accessible

Scenario: Current Iteration page
  When I follow "Current iteration"
  Then the page should be accessible

Scenario: Login page
  When I follow "Logout"
  Then the page should be accessible
```