CS 169 HW 8 (Group 16)

Final Code (since we can't submit both GitHub and the PDF at the same time) linked below:

- Commit: https://github.com/AlanHe-Xiaoyu/audience1st/commit/6a20b026c12862411369540abd16acf1d716236d
- Feature branch code base: https://github.com/AlanHe-Xiaoyu/audience1st/tree/hw8_improve_accessibility

For each person on your team who wants credit, answer this with your name:

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?

CiCi Huang
I think a screen reader will really help people with disability to navigate through modern websites. It permits people with visual disabilities to better understand the contents presented on a web page by describing the content of an image and reading out loud even a table. I learnt how to access a web page from the perspective of a disabled person by using specific voiceover and keyboard commands. I found it very hard to navigate through it because I am too used to navigate a website by just using a mouse. I tried audits on https://rocky-ravine-27276.herokuapp.com/store/1, which is a website to buy theatre tickets. It was hard to navigate because of insufficient accessibility-related HTML tags. Sometimes, when I use tab or space to jump around, the tool will miss out a few blanks in which I need to fill in information. This badly hurts the performance of our ticket-buying system.

Alan He
The screen reader was extremely frustrating, as it mostly just works with the toolbar and not with the interactive elements on the site itself. I was navigating through GitHub.com and through my own repos; it would read over the things I hover, but I don't actually think it helps with the navigation that much - considering how we normally navigate

through it. GitHub was better than some, but definitely still rather hard to navigate, as we need to listen to the long unnecessary text explaining what's going on.
On our own project site, it's really difficult to navigate as it's not setup yet, i.e. https://rocky-ravine-27276.herokuapp.com/store/1 is really hard to navigate. Afterwards, it becomes significantly better as we incorporated the change.

Autumn
I tried screen reader with gradescope for cs169. I found that screen reader will tell me information about both the content and type of elements on a website. I can imagine this being useful for people with visual impairment to better understand websites. For me, it was kind of hard to navigate. I have to listen to the instruction depending on which area I'm on the webpage and press certain keys on keyboard to change to another part. I then tried screen reader on our project https://rocky-ravine-27276.herokuapp.com/store/1 and screen reader doesn't seem to be that useful other than telling me information about tabs and some limited areas on the website.

Arthur (Hang Yin)
I tried the windows screen reader on the site https://support.microsoft.com/en-gb/help/17173/windows-10-hear-text-read-aloud

The experience is actually pretty bad. The narrator read loud the size of the text I selected and slowly read out the content on the page. I think this might help people with eye impairment reading the site but definitely inefficient in a working environment.

I tried the screen reader on our own project

https://rocky-ravine-27276.herokuapp.com/store/4

It is also very hard to input the value we need just with the help of the narrator. It is very hard to navigate.

Casper Yang
I tried using a screen reader on Google and YouTube, two of my most used websites, and I found it to be extremely useful for people with disabilities when the website is somewhat simple and a result of what the user is looking for. But for a website that is somewhat more complex and filled with random data and content, I find screen reader to be a little too much because it does not filter out unwanted data and content, and it is hard to navigate with such massive content. It would read all the things that I have my

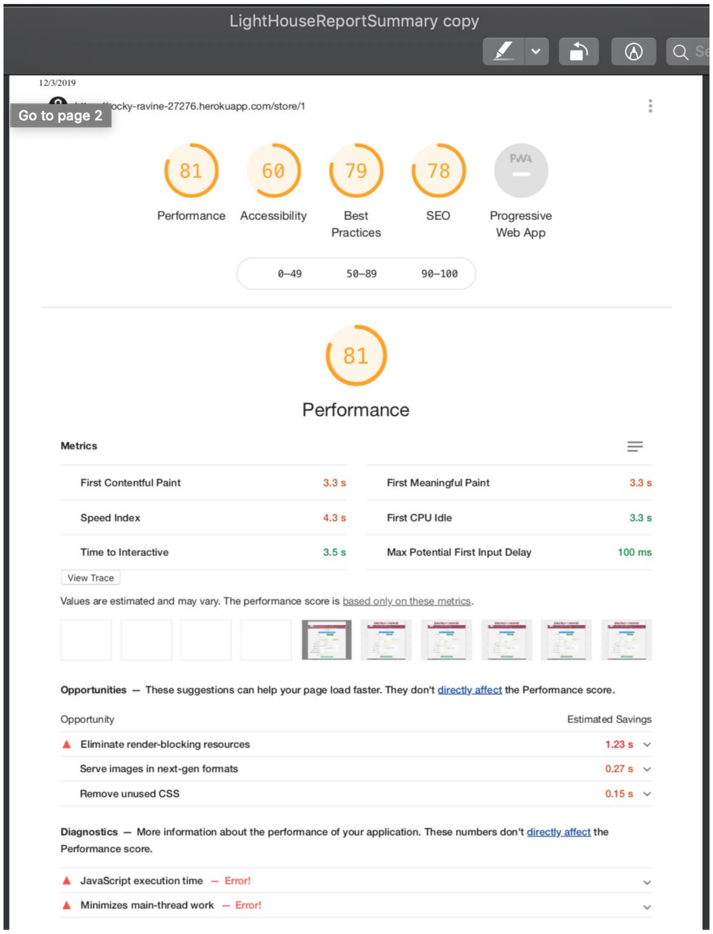
mouse over, but since youtube’s layout is so dense and it is not that much easier to find what I was looking for. And afterward, I tried with our own app website https://rocky-ravine-27276.herokuapp.com/customers and find it a little frustrating as the page is made up of a spreadsheet of data that is closely next to each other. I would always find myself waiting for the voice over to be completed before moving to the next data.

Do these as a team. For each test you should try to use a different route in your application.

1. **What did the accessibility tests from Audits tab in Chrome report? * Include the route in your report along with screenshots.**

The Audits tab reported that contrast is not large enough, id attributes are not unique, <html> element does not have a [lang] attribute, and that the form elements do not have associated labels.

The route is: https://rocky-ravine-27276.herokuapp.com/store/1



LightHouseReportSummary copy

12/3/2019

Go to page 2

...ocky-ravine-27276.herokuapp.com/store/1

Performance

Metrics

First Contentful Paint	3.3 s	First Meaningful Paint	3.3 s
Speed Index	4.3 s	First CPU Idle	3.3 s
Time to Interactive	3.5 s	Max Potential First Input Delay	100 ms

View Trace

Values are estimated and may vary. The performance score is based only on these metrics.

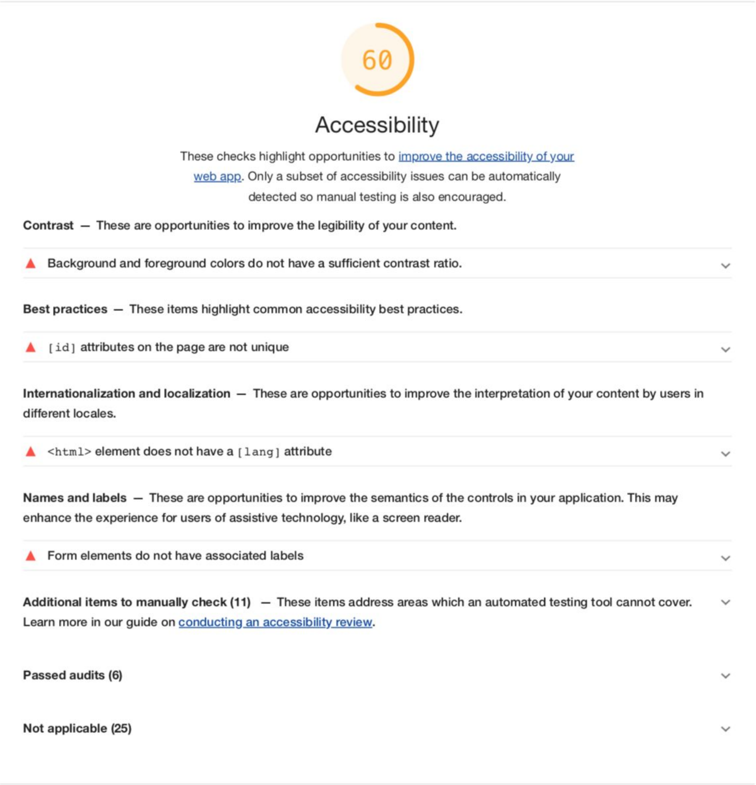
Opportunities — These suggestions can help your page load faster. They don't directly affect the Performance score.

Opportunity	Estimated Savings
▲ Eliminate render-blocking resources	1.23 s
Serve images in next-gen formats	0.27 s
Remove unused CSS	0.15 s

Diagnostics — More information about the performance of your application. These numbers don't directly affect the Performance score.

▲ JavaScript execution time — Error!

▲ Minimizes main-thread work — Error!



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.

Best practices — These items highlight common accessibility best practices.

▲ `[id]` attributes on the page are not unique

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

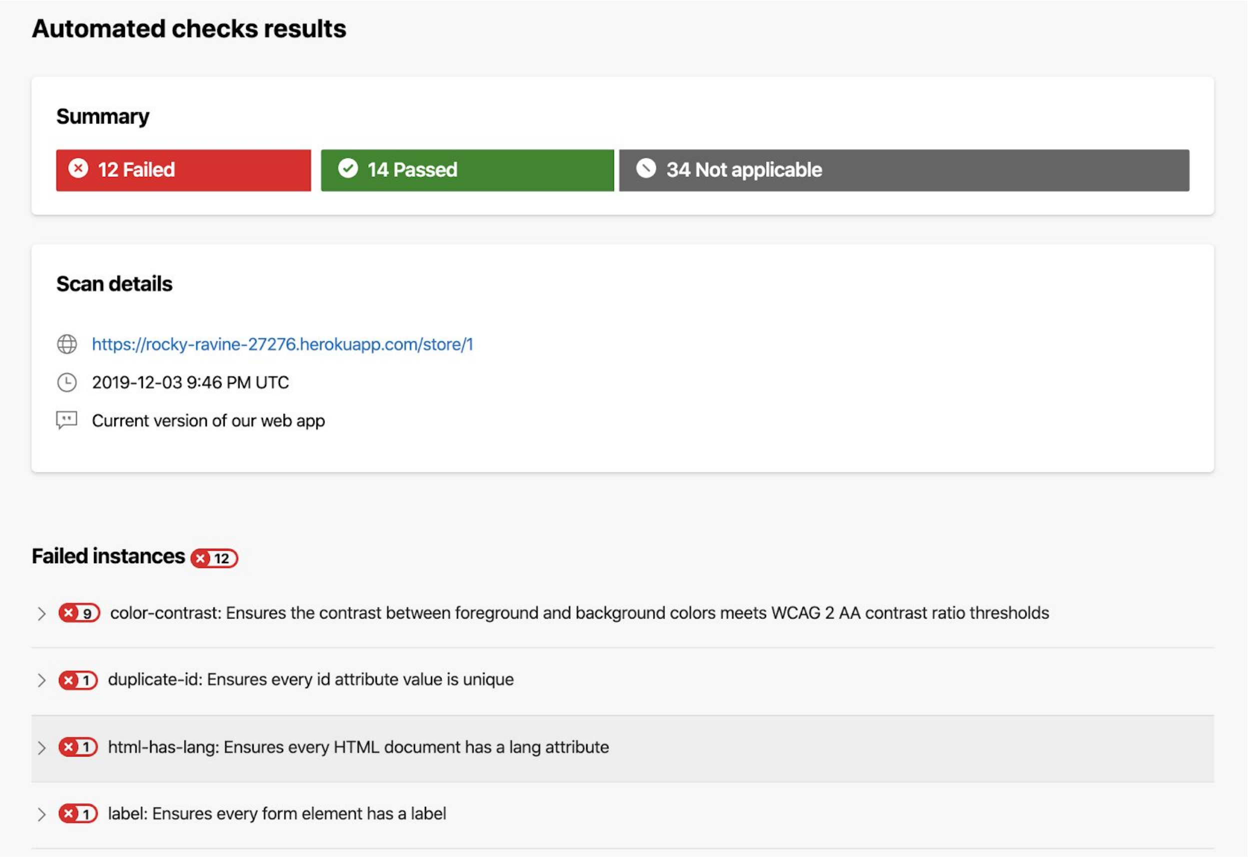
Additional items to manually check (11) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (6)

Not applicable (25)

What did the tests from Accessibility Insights "Fast Pass" report?

The tests from Accessibility Insights "Fast Pass" show a combination of passed and failed instances of the accessibility of our app website. Some failed instances include “color-contrast”, which can be solved by changing the contrast ratio; “incomplete label”, which can be fixed filling in corresponding tags; “html does not have lang element”, which can be fixed by adding a lang attribute.



Automated checks results

Summary

☒ 12 Failed | ✓ 14 Passed | ⊘ 34 Not applicable

Scan details

https://rocky-ravine-27276.herokuapp.com/store/1

2019-12-03 9:46 PM UTC

Current version of our web app

Failed instances ☒ 12

- ☒ 9 color-contrast: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
- ☒ 1 duplicate-id: Ensures every id attribute value is unique
- ☒ 1 html-has-lang: Ensures every HTML document has a lang attribute
- ☒ 1 label: Ensures every form element has a label

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?

We were not surprised by the tests. We haven’t covered most test categories (95%) when developing the app therefore most categories are shown incomplete. For the ones that failed, we highly agree that they need some improvement. We agree with the results below:

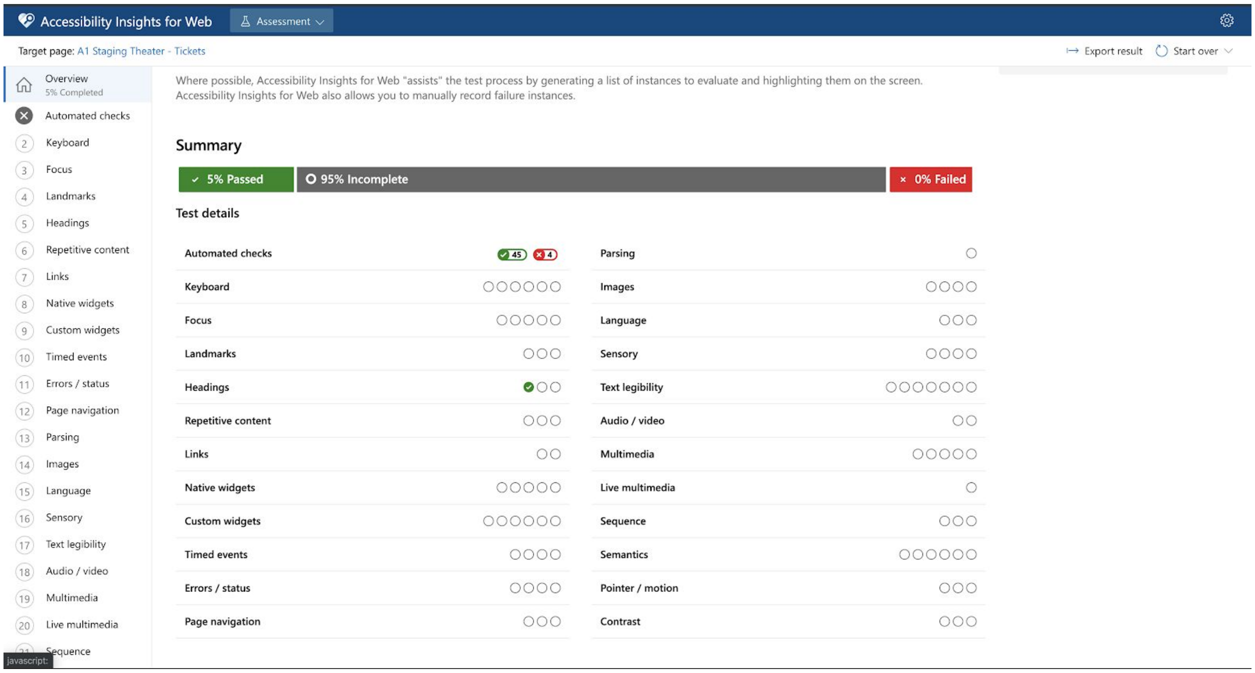
1. Color-Contrast: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds

2. Duplicate-id: Ensures every id attribute value is unique
3. Html-has-lang: The <html> element does not have a lang attribute
4. Label: Ensures every form element has a label

Most of the failed test are due to lack of certain tags in html file and color contrast ratio too low. Those common errors do appear on other pages.

1. For failed color-contrast tests, we decided to standardize color usage, so the color-contrast test would meet the requirement.
2. For duplicate-id, we decided to deduplicate ids for unique items
3. For Html-has-lang, we decided to add lang='en' to every haml file so each page has the lang attribute
4. For label in form, we decided to add a label to each form row

The route link is still: https://rocky-ravine-27276.herokuapp.com/store/1



Target page: A1 Staging Theater - Tickets

Where possible, Accessibility Insights for Web "assists" the test process by generating a list of instances to evaluate and highlighting them on the screen. Accessibility Insights for Web also allows you to manually record failure instances.

Summary

✓ 5% Passed | ○ 95% Incomplete | × 0% Failed

Test details

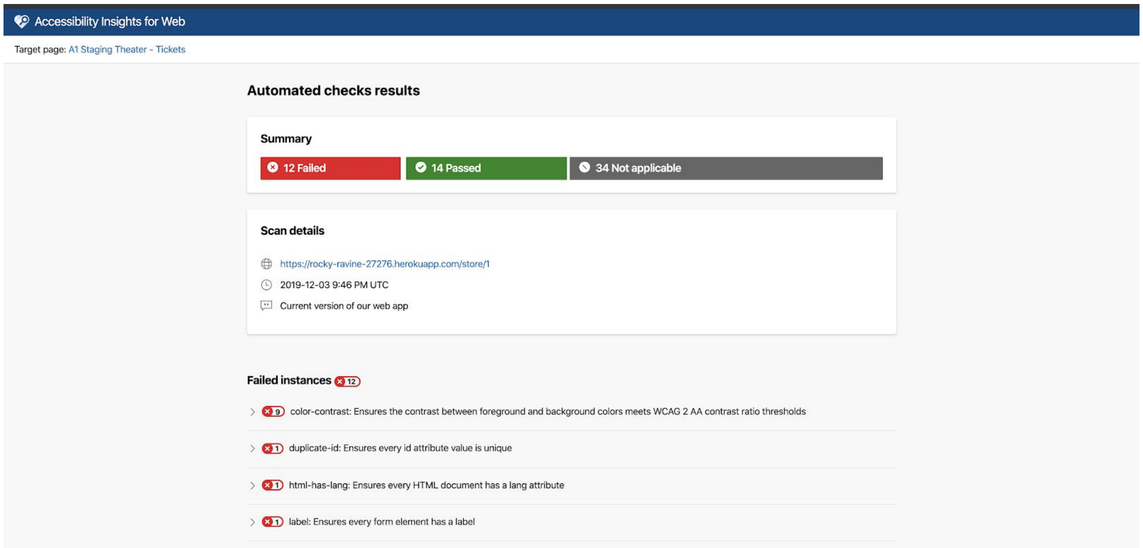
Automated checks	✓ 45 × 4	Parsing	○
Keyboard	○○○○○○	Images	○○○○
Focus	○○○○○	Language	○○○
Landmarks	○○○	Sensory	○○○○
Headings	✓○○	Text legibility	○○○○○○○
Repetitive content	○○○	Audio / video	○○
Links	○○	Multimedia	○○○○○
Native widgets	○○○○○	Live multimedia	○
Custom widgets	○○○○○○	Sequence	○○○
Timed events	○○○○	Semantics	○○○○○○
Errors / status	○○○○	Pointer / motion	○○○
Page navigation	○○○	Contrast	○○○

Pick 2 (non alt-text related) errors to address.

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

The route link is: https://rocky-ravine-27276.herokuapp.com/store/1

Before: (4 tests failed)



Accessibility Insights for Web

Target page: A1 Staging Theater - Tickets

Automated checks results

Summary

12 Failed | 14 Passed | 34 Not applicable

Scan details

https://rocky-ravine-27276.herokuapp.com/store/1

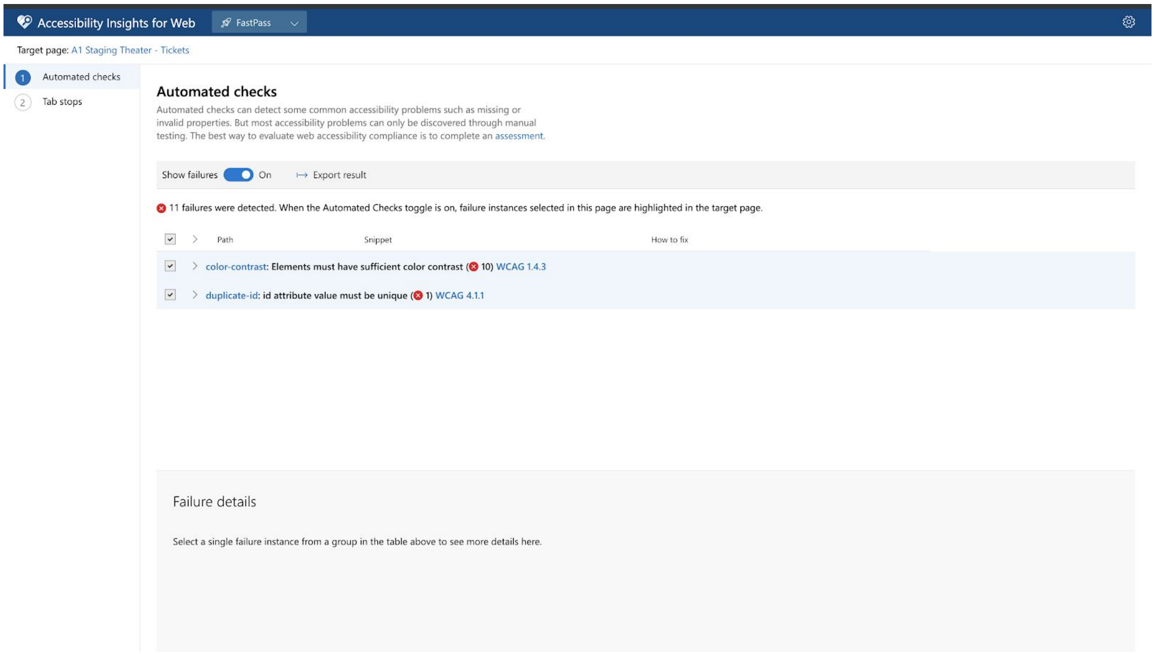
2019-12-03 9:46 PM UTC

Current version of our web app

Failed instances 12

- 9 color-contrast: Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
- 1 duplicate-id: Ensures every id attribute value is unique
- 1 html-has-lang: Ensures every HTML document has a lang attribute
- 1 label: Ensures every form element has a label

After: (2 tests failed)



Target page: A1 Staging Theater - Tickets

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

11 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 (10) WCAG 1.4.3		
☑	duplicate-id: id attribute value must be unique (1) WCAG 4.1.1		

Failure details

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

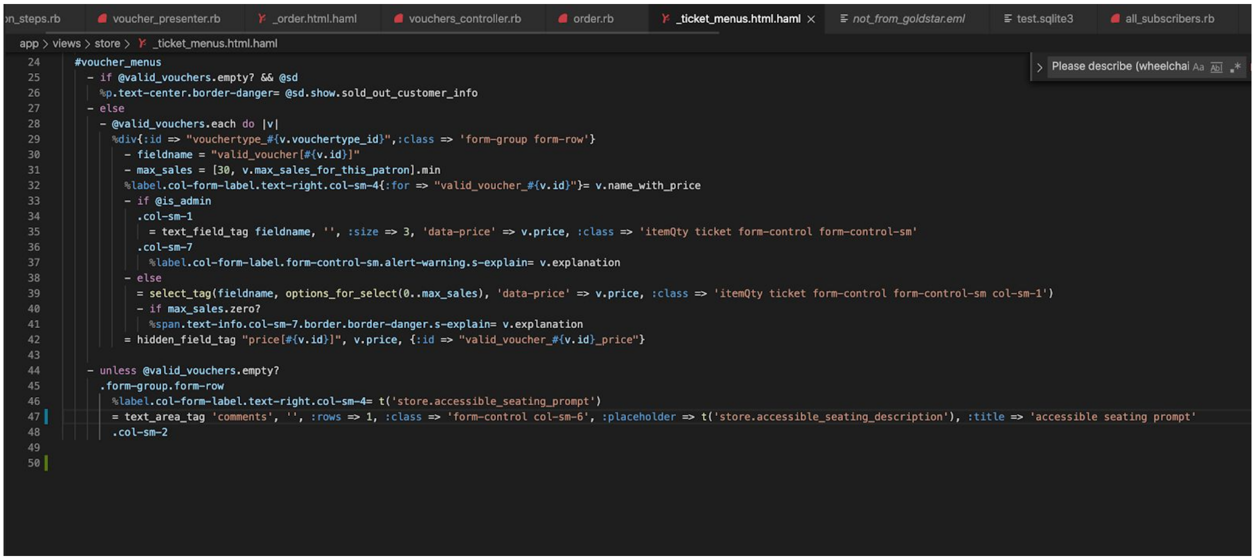
We decided the fix the 3rd and the 4th error appeared:

3. Html-has-lang: The <html> element does not have a lang attribute

4. Label: Ensures every form element has a label

For 3. Html-has-lang: The <html> element does not have a lang attribute, we added :title => ‘accessible seating prompt’ to one form-row. Thus the form-row has the label attribute, becoming accessible.

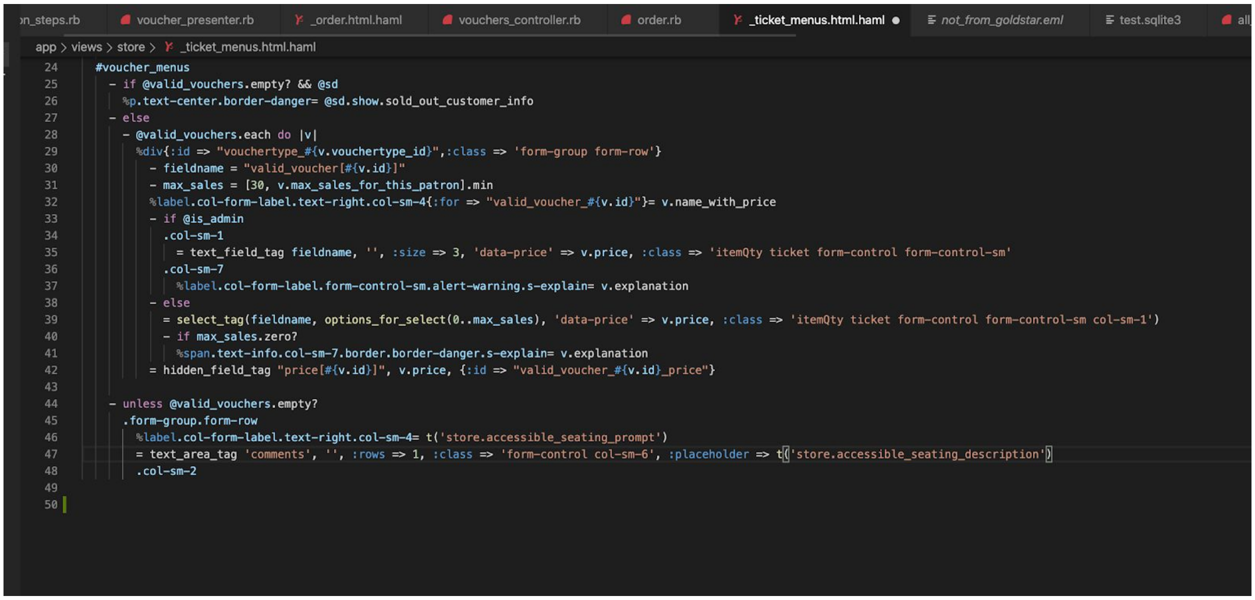
Before:



```
#voucher_menus
  - if @valid_vouchers.empty? && @sd
    %p.text-center.border-danger= @sd.show.sold_out_customer_info
  - else
    - @valid_vouchers.each do |v|
      %div{:id => "vouchertype_#{v.vouchertype_id}",:class => 'form-group form-row'}
        - fieldname = "valid_voucher[#{v.id}]"
        - max_sales = [30, v.max_sales_for_this_patron].min
        %label.col-form-label.text-right.col-sm-4{:for => "valid_voucher_#{v.id}"}= v.name_with_price
        - if @is_admin
          .col-sm-1
            = text_field_tag fieldname, '', :size => 3, 'data-price' => v.price, :class => 'itemQty ticket form-control form-control-sm'
          .col-sm-7
            %label.col-form-label.form-control-sm.alert-warning.s-explain= v.explanation
        - else
          = select_tag(fieldname, options_for_select(0..max_sales), 'data-price' => v.price, :class => 'itemQty ticket form-control form-control-sm col-sm-1')
          - if max_sales.zero?
            %span.text-info.col-sm-7.border.border-danger.s-explain= v.explanation
        = hidden_field_tag "price[#{v.id}]", v.price, {:id => "valid_voucher_#{v.id}_price"}

  - unless @valid_vouchers.empty?
    .form-group.form-row
      %label.col-form-label.text-right.col-sm-4= t('store.accessible_seating_prompt')
      = text_area_tag 'comments', '', :rows => 1, :class => 'form-control col-sm-6', :placeholder => t('store.accessible_seating_description'), :title => 'accessible seating prompt'
      .col-sm-2
```

After:

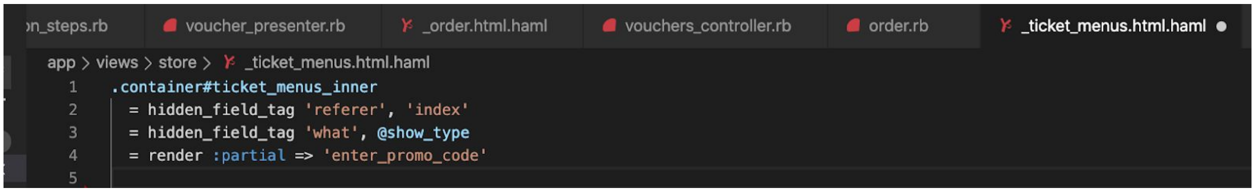


```
#voucher_menus
  - if @valid_vouchers.empty? && @sd
    %p.text-center.border-danger= @sd.show.sold_out_customer_info
  - else
    - @valid_vouchers.each do |v|
      %div{:id => "vouchertype_#{v.vouchertype_id}",:class => 'form-group form-row'}
        - fieldname = "valid_voucher[#{v.id}]"
        - max_sales = [30, v.max_sales_for_this_patron].min
        %label.col-form-label.text-right.col-sm-4{:for => "valid_voucher_#{v.id}"}= v.name_with_price
        - if @is_admin
          .col-sm-1
            = text_field_tag fieldname, '', :size => 3, 'data-price' => v.price, :class => 'itemQty ticket form-control form-control-sm'
          .col-sm-7
            %label.col-form-label.form-control-sm.alert-warning.s-explain= v.explanation
        - else
          = select_tag(fieldname, options_for_select(0..max_sales), 'data-price' => v.price, :class => 'itemQty ticket form-control form-control-sm col-sm-1')
          - if max_sales.zero?
            %span.text-info.col-sm-7.border.border-danger.s-explain= v.explanation
        = hidden_field_tag "price[#{v.id}]", v.price, {:id => "valid_voucher_#{v.id}_price"}

  - unless @valid_vouchers.empty?
    .form-group.form-row
      %label.col-form-label.text-right.col-sm-4= t('store.accessible_seating_prompt')
      = text_area_tag 'comments', '', :rows => 1, :class => 'form-control col-sm-6', :placeholder => t('store.accessible_seating_description')
      .col-sm-2
```

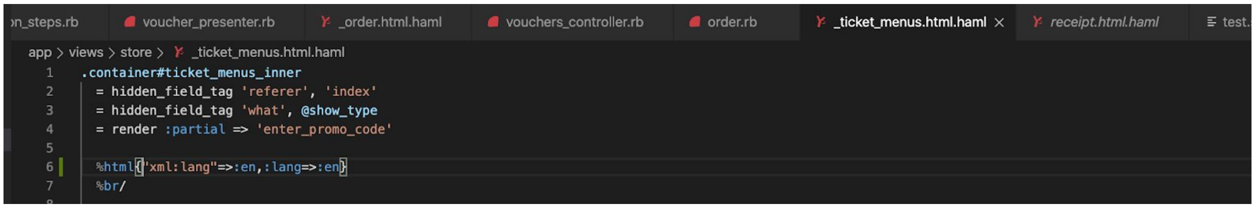
For 4. Label: Ensures every form element has a label , we added %html{"xml:lang"=>:en,:lang=>:en}
Thus the html page has the lang attribute as ‘en’

Before:



```
.container#ticket_menus_inner
  = hidden_field_tag 'referer', 'index'
  = hidden_field_tag 'what', @show_type
  = render :partial => 'enter_promo_code'
```

After:



```
.container#ticket_menus_inner
  = hidden_field_tag 'referer', 'index'
  = hidden_field_tag 'what', @show_type
  = render :partial => 'enter_promo_code'

  %html{"xml:lang"=>:en,:lang=>:en}
  %br/
```

The commit is below:
https://github.com/AlanHe-Xiaoyu/audience1st/commit/6a20b026c12862411369540abd16acf1d716236d

The feature branch is below:
https://github.com/AlanHe-Xiaoyu/audience1st/tree/hw8_improve_accessibility