Research Information Systems

Accessibility - [done]

How one web development team includes Accessibility in their Definition of Done

May 20, 2021, GAAD
Who We Are

An Agile Web Development Team Delivering Solutions for Sponsored Programs and Compliance Offices
Agile Team

- Continuous development approach
- Cross-functional group with all necessary roles for product development
- Small team, open structure, iterations
Projects

→ Web Applications
  • System collecting compliance information, forms, html elements
  • User interface(s) with multiple components, reports

<table>
<thead>
<tr>
<th>New Animal Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User/Account Details</td>
</tr>
<tr>
<td>Principal Investigator</td>
</tr>
<tr>
<td>Account-Protocol</td>
</tr>
<tr>
<td>3. Location Information</td>
</tr>
<tr>
<td>Please select species to select location.</td>
</tr>
<tr>
<td>5. Quantity Details</td>
</tr>
<tr>
<td>Quantity Either Sex</td>
</tr>
<tr>
<td>Quantity Male</td>
</tr>
<tr>
<td>Quantity Female</td>
</tr>
<tr>
<td>Please select species and vendor to enter quantity details.</td>
</tr>
</tbody>
</table>
Projects
More samples

Information on this slide does not represent an actual researcher or actual research activity.
Technologies

- Ruby On Rails, JavaScript, Vue.js, Durandal.js, Bootstrap, PHP
Why we focus on Accessibility
For the Users
Accessibility

Accessibility helps people who use screen readers, people who have various degrees of color discernment.
Accessibility is Usability

- Developing with Accessibility in mind means more people can use your application productively
Accessibility is Usability

Keyboard Access

- Consider keyboard access
Accessibility is Inclusion

- By the time someone complains about an issue, they've already been made to feel like the application is not intended to be used by people like them
Accessibility is Inclusion
Our Philosophy

• We have to do everything we can to make our applications accessible to everyone
Incorporating Accessibility

Definition of Done and Tool Selection
Definition of Done

When is a unit of work complete?

- Consistency in how we determine that is important

Definition Of Done
Versioning - [NA]
Security - [NA]
Reporting - [NA]
Validations - [DONE]
Good Design - [DONE]
Accessibility - [DONE]
Browser Testing Is Successful - [DONE]
Automated tests written for Rails apps - [DONE]
Automated Build Passes - [DONE]
Deploy to demo - [eiacuc-demo]
Pull Request Made – [https://git.uiowa.edu/RIS/eiacuc/-/merge_requests/747]
Definition of Done
Expanded

Reviewed *(check on this and cleanup)*
All stories/defects must be reviewed by at least 2 sets of eyes (In mingle speak this is before the card is moved to the reviewed column)

**Reviewer Checklist** - things to be reviewed

Accessibility
- Fix all errors
- Fix all warning as seems necessary

Listen to the page with NVDA on Windows *(http://www.nvaccess.org/)*
- Remote into one of our RIS Managed physical Windows boxes as listed in the Operations page
- If necessary, use Jaws, Mac VoiceOver for the testing

Browser Testing Is Successful
- Testing must be complete in all applicable browsers (Chrome, Firefox, IE, etc.)
  - Browsers that need to be tested will be determined by the project
  - Does this look and feel like the rest of the site?
  - Does the browser console display any error?

Automated tests are written for Rails apps
- Tests should cover changes in business logic, tests do not need to be written to cover items that are part of frameworks functionality
- Testing current PHP applications should be done manually
Tool Selection – Frameworks and libraries

• Maintainability, tool community (for feature request/fixes)

• Accessibility features

• Quick demo before finalizing
Accessibility Testing

Using the tools to make sure we are done
Axe from Deque

HTML validation *Deluxe Edition*
Axe

• **Pros**
  • Fast
  • Easy to use
  • Provides hints for how to resolve issues
  • Issue highlighting
  • Only tests visible or screen reader accessible content

• **Cons**
  • Possible contrast issues are flagged for further review
  • Only tests visible or screen reader accessible content

---

**DevTools axe-core 4.1.3**

- URL: [https://uiris-project1.research.uiowa.edu/animal_pub/dist/index.html#/](https://uiris-project1.research.uiowa.edu/animal_pub/dist/index.html#/)

- **Total Issues**: 25
  - **Automatic Issues**: 25
  - **Guided Issues**: 0

**ALL ISSUES: 25**

- Certain ARIA roles must be contained by particular parents: 12
- `<ul>` and `<ol>` must only directly contain `<li>`, `<script>` or `<template>` elements: 1
- Elements must have sufficient color contrast: 12
Axe - Suggestions

Certain ARIA roles must be contained by particular parents

```
a[href="#/settings/buildings"]
```

Element source

```
<a href="#/settings/buildings" class="dropdown-item" role="menuitem" target="self"> Buildings </a>
```

To solve this violation, you need to...

Fix the following:

Required ARIA parents role not present: menu, menubar
Using Axe

In practice
Install the extension

chrome web store

axe DevTools - Web Accessibility Testing
Offered by: www.deque.com

4.6 stars 86 | Developer Tools | 100,000+ users

Runs offline
Setting up Axe

Draft DEI Action Plan - Introduction
DRAFT content for review and prioritization - October 2020

Goal 3: Enhance campus-wide DEI accountability, effectiveness, and collaboration.
DRAFT content for review and prioritization - October 2020
# Reviewing Findings

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><img src="image.png" alt="Image of DevTools interface" /></td>
</tr>
</tbody>
</table>

### ALL ISSUES: 3

- `<ul>` and `<ol>` must only directly contain `<li>`, `<script>` or `<template>` elements  
  - 1
- All page content should be contained by landmarks  
  - 2

- `<ul>` and `<ol>` must only directly contain `<li>`, `<script>` or `<template>` elements  
  - 1

- ![Highlight icon](highlight.png)  
  - `highlight`  
  - ![Inspect icon](inspect.png)  
  - ![More info icon](more_info.png)  
  - ![Share issue icon](share_issue.png)  

- ![More details icon](more_details.png)
identifying issues
Rescan

You are here

Vous êtes ici

Upgrade  /  Sign in

Re-run automatic scan

3

0
Rescan results

URL: https://oneit.uiowa.edu/draft-diversity-equity-and-inclusion-action-plan

TOTAL ISSUES

AUTOMATIC ISSUES

GUIDED ISSUES

Critical

0

Serious

0

Moderate

2

Minor

0

OneIT

Home / Strategic Plans

DRAFT Diversity, Equity, and Inclusion Action Plan
Lighthouse From Chrome

Chrome’s testing powerhouse
Lighthouse

- **Pros**
  - Comprehensive report
- **Cons**
  - No issue highlighting
  - Solutions aren't as apparent
  - Reloads pages to test.
Lighthouse Audit Report

Detail explanation of passed audits & non-passed audits

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.

Additional items to manually check (10) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (19)

- [aria-\(\text{attr}\)] attributes match their roles
- [aria-hidden="true"] is not present on the document \(<\text{body}\>\)
- [role]s have all required [aria-\(\text{attr}\)] attributes
- Elements with an ARIA [role] that require children to contain a specific [role] have all required children.
- [role]s are contained by their required parent element
- [role] values are valid
- [aria\-\(\text{attr}\)] attributes have valid values
- [aria\-\(\text{attr}\)] attributes are valid and not misspelled
- Buttons have an accessible name
- The page contains a heading, skip link, or landmark region
Using Lighthouse

In practice
Setting up Lighthouse

Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more

Categories
- Performance
- Progressive Web App
- Best practices
- Accessibility
- SEO

Device
- Mobile
- Desktop

Community Plugins (beta)
- Publisher Ads

Research Information Systems
Reading the report

https://oneit.uiowa.edu/draft-diversity-equity-and-inclusion-action-plan

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.

Tables and lists — These are opportunities to improve the experience of reading tabular or list data using assistive technology, like a screen reader.

⚠️ Lists do not contain only <li> elements and script supporting elements (<script> and <template>).
Details

⚠️ Lists do not contain only `<li>` elements and script supporting elements (`<script>` and `<template>`).

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more.

Failing Elements

```
ul.breadcrumb
```
Guidance for what else to test

Additional items to manually check (10) — These items address areas which an automated testing tool cannot cover.

Learn more in our guide on conducting an accessibility review.

- The page has a logical tab order
- Interactive controls are keyboard focusable
- Interactive elements indicate their purpose and state
- The user's focus is directed to new content added to the page
- User focus is not accidentally trapped in a region
- Custom controls have associated labels
- Custom controls have ARIA roles
Screen Readers

Not experts and we don’t even play experts on tv
Choosing One

• Options
  • Jaws
  • NV Access (NVDA)
  • Voice Over
  • Narrator

• A lot of options to consider
NVDA

• Run NVDA voice over tool on application screen(s)

• Listen the page flow by accessing various page elements

• Appropriate screen reader text, Aria roles & essential accessibility elements should be present on the screen(s)
Listening to NVDA

**Manage Orders**

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Account Owner</th>
<th>Account</th>
<th>Protocol Number</th>
<th>Vendor Name</th>
<th>Species</th>
<th>Quantity Ordered</th>
<th>Expected/Received Date</th>
<th>Housing Location</th>
<th>Order Status</th>
<th>Order ID</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>247182</td>
<td>Bartoletti, Enrique</td>
<td>068E</td>
<td>9031670</td>
<td>EXEMPLAR GENETIC</td>
<td>Pig</td>
<td>8</td>
<td>03/31/2021</td>
<td>817 hold - PEDB</td>
<td>Placed with Vendor</td>
<td>247182</td>
<td></td>
</tr>
<tr>
<td>247467</td>
<td>Bartoletti, Enrique</td>
<td>068E</td>
<td>9031670</td>
<td>EXEMPLAR GENETIC</td>
<td>Pig</td>
<td>16</td>
<td>05/05/2021</td>
<td>817 - BREF</td>
<td>Placed with Vendor</td>
<td>247467</td>
<td></td>
</tr>
<tr>
<td>247475</td>
<td>Bartoletti, Enrique</td>
<td>068E</td>
<td>9031670</td>
<td>EXEMPLAR GENETIC</td>
<td>Pig</td>
<td>8</td>
<td>04/23/2021</td>
<td>817 hold - PEDB</td>
<td>Placed with Vendor</td>
<td>247475</td>
<td></td>
</tr>
<tr>
<td>247155</td>
<td>Bartoletti, Enrique</td>
<td>7</td>
<td>8102176</td>
<td>JACKSON LABS</td>
<td>Mouse</td>
<td>24</td>
<td>04/05/2021</td>
<td>408 - ML</td>
<td>Placed with Vendor</td>
<td>247155</td>
<td></td>
</tr>
</tbody>
</table>

Rows per page: 25
What are we listening for

• Can I tell what to do next without looking at the screen?
  • When I get to a control can I understand what it is for?

• Do I know when things change on the page as I interact?
  • Like the results announcement as I change filters

• Are all the controls in a logical order?

• Can I get to everything?
More Keyboard Navigation

Where should the focus go after we perform an action?

<table>
<thead>
<tr>
<th>Expiration Date</th>
<th>Active</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Add a New Account
Current Month Charges
Inventory
Enable Online Ordering
Lessons Learned

Things for you to consider if you add this to your process
Part 1

- Return the user back to a reasonable location when a modal closes

- Make sure you can't navigate out of a modal with the keyboard when it is open

- When adding content to a page dynamically, make sure you announce the new content

- Use screen reader text to announce search results

- Don't be afraid to alter the code a framework generates to make things more accessible.
More Lessons Learned

- Create and reuse components for consistency
- Retest, even when you don’t think you broke anything
- Standards and recommendations change over time
- Accessible-first development
Challenges

Lessons yet to be learned
Things that could be better

- Screen Readers
  - Don't all work the same
  - Don't work the same across browsers
  - None of us on the team use them a regular basis
- Windows vs Mac
  - NVDA only runs on PC
  - Voice Over (mac) and Narrator (pc) work differently
- Live regions might jump in early
- Consistency across solutions
Accessibility - attempted

How one web development team includes Accessibility in their Definition of Done

May 20, 2021, GAAD
Research Information Systems

Questions?
IOWA