**Exercise: Evaluating & Improving Accessibility**

Use audit tools and manual checks to evaluate and improve accessibility in your responsive layout. Sample Code for a page with accessibility issues:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Accessibility Test Page</title>

<style>

body {

background-color: #fff;

color: gray;

font-family: Arial, sans-serif;

}

.btn {

background-color: green;

color: green;

padding: 1rem;

border: none;

font-size: 1rem;

cursor: pointer;

}

.hidden-label {

display: none;

}

.important {

font-size: 24px;

font-weight: bold;

}

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

table {

width: 100%;

}

</style>

</head>

<body>

<h1>Welcome</h1>

<p>Click the button to go forward</p>

<button class="btn">Click</button>

<form>

<input type="text" name="email" placeholder="Your email">

<br><br>

<input type="password" placeholder="Password">

<br><br>

<input type="submit" value="Submit">

</form>

<img src="logo.png">

<img src="decorative.jpg" alt="">

<h2>Our Features</h2>

<ul>

<li>Easy to use</li>

<li>Fast</li>

<li>Secure</li>

<!-- List not labeled as navigation -->

</ul>

<h2>Employee Table</h2>

<table>

<tr>

<td>Jane</td>

<td>Manager</td>

</tr>

<tr>

<td>John</td>

<td>Engineer</td>

</tr>

</table>

<p style="color: lightgray;">This text has very low contrast</p>

</body>

</html>

**Known Accessibility Issues to Find & Fix:**

|  |  |
| --- | --- |
| **Issue** | **Description** |
| No alt text | logo.png is missing alt="", so screen readers can't describe it. |
| Low color contrast \* | Button text is green on green. The paragraph is light gray on white. |
| No <label> elements | Form inputs are missing accessible <label for=""> associations. |
| Button label unclear | The button just says "Click" – not descriptive. |
| No table headers | Table rows lack <th> for screen reader context. |
| Missing landmarks | No <main>, <nav>, or sectioning elements for semantic structure. |
| Placeholder misuse | Placeholders are used instead of labels (placeholders disappear!). |
| Decorative image not marked as such | One alt="" is correct, but the other image is missing any alt. |

*\* Solution for low color contrast: Ensure text/background contrast meets [WCAG AA](https://www.w3.org/TR/WCAG21/#contrast-minimum): use tools like [contrast-ratio.com](https://contrast-ratio.com/) to adjust colors (e.g., darken green, darken gray text).*

**Suggested Tests (instructions below)**

1. **Run it through WAVE & Lighthouse**
2. List all problems found
3. Fix:
   * Add alt text
   * Improve contrast
   * Label inputs
   * Use proper semantic tags
4. Retest to verify improvements

**Part 1: Evaluate Your Page**

**Tools:**

* WAVE
* Lighthouse (Chrome > DevTools > Lighthouse > Accessibility)
* VoiceOver (Mac) or NVDA (Windows) screen reader (see below for instructions)
* Keyboard-only navigation (Tab, Enter, Space, Esc)

……………………………………………..

Here’s how to use **VoiceOver (Mac)** and **NVDA (Windows)** for accessibility testing, including beginner-friendly keyboard shortcuts and tips:

**VoiceOver (Mac)**

**Enable/Disable**

* **Toggle on/off:** Command + F5
  + (If that doesn't work: System Settings > Accessibility > VoiceOver > Turn On)

**Basic Navigation**

* **Control + Option (VO)** is the VoiceOver modifier key. You’ll use it with arrows and letters.
* **VO + Right/Left Arrow** → Move forward/backward by element.
* **VO + Space** → Click/interact with item.
* **Tab** → Moves between focusable items (like normal tabbing).
* **VO + U** → Opens the rotor (a menu of landmarks, links, etc.)
* **Escape** → Exit menus or interactions.

**Tips**

* Use **Safari or Chrome**—VoiceOver works well with both.
* Test **logical flow**: Does the order of reading match the visual hierarchy?
* Test **focus visibility**: Is the focus indicator visible as you tab?

**NVDA (Windows)**

**Install**

* Download NVDA (Free)

**Enable/Disable**

* After installing, NVDA runs in the background. Use:
  + **Control + Alt + N** to toggle NVDA on/off.
  + Or manually start from your Start menu.

**Basic Navigation**

* **NVDA + Arrow keys** → Read text.
  + NVDA key is usually **Caps Lock** or **Insert**.
* **Tab** → Jump between links/buttons.
* **Shift + Tab** → Reverse.
* **NVDA + Space** → Interact/click.
* **NVDA + T** → Reads window title.
* **NVDA + F7** → Element list (headings, links, landmarks).

**Tips**

* Use **Firefox** (best compatibility with NVDA).
* You can also **simulate tabbing** and form navigation using just Tab, Enter, and arrow keys without the NVDA modifier.

**Testing Checklist with a Screen Reader**

* Can you reach all interactive elements?
* Do links/buttons say what they do?
* Do form inputs have meaningful labels?
* Does reading order match visual layout?
* Can you skip repetitive content (e.g., add skip links)?
* Does modal or dialog content get focus?

**Checklist:**

* Images have meaningful alt text
* Buttons and links have clear labels
* Sufficient color contrast (4.5:1 for text)
* Form fields have <label>s
* Can navigate with keyboard
* No flashing/strobing content

…………………………………………………………

**Part 2: Implement Fixes**

Choose 3–5 accessibility issues from the audit and fix them:

* Replace <div> buttons with real <button> elements
* Add aria-label to ambiguous icons
* Use rem or em units for scalable text
* Ensure focus styles are visible (outline, not removed!)
* Fix contrast issues (adjust text or background color)

**Part 3: Re-Test and Document Improvements**

Re-run Lighthouse and WAVE audits.

**Example table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Fix Implemented** | **Score Before** | **Score After** |
| Missing alt text | Added alt text to all images | 70 | 94 |
| Poor contrast on buttons | Updated button background | 68 | 95 |
| Tab order broken | Added tabindex and labels\* | 60 | 92 |

\*Note: The tabindex attribute in HTML controls the order in which elements receive keyboard focus when the user presses the Tab key. It's commonly used to improve accessibility and user navigation, especially for custom components or interactive UIs.

**Syntax**

<div tabindex="0">Focusable Div</div>

<button tabindex="1">First Tab</button>

<input tabindex="2" />

**What tabindex Values Mean**

|  |  |
| --- | --- |
| **tabindex Value** | **Meaning** |
| 0 | Element is focusable in natural DOM order. Useful for making non-interactive elements (like <div>) focusable. |
| -1 | Element is **focusable via JavaScript** (e.g., .focus()), but **not** in the tab order. |
| >0 | Sets **custom tab order** (rarely recommended). Lower numbers get focus first. |

**tabindex Best Practices**

1. Use tabindex="0" to make custom components keyboard-accessible.  
     
     
   <div role="button" tabindex="0">Click me</div>
2. Avoid tabindex > 0 — it overrides the natural flow and can confuse screen readers and users.
3. Use tabindex="-1" for elements that should be focusable via JavaScript but skipped in tabbing.  
     
   modal.focus(); // works if modal has tabindex="-1"

**Example Use Case: Custom Modal**

<div id="modal" role="dialog" aria-modal="true" tabindex="-1">

<button>Close</button>

</div>

<script>

document.getElementById('modal').focus();

</script>

This ensures the modal receives focus when opened, but doesn’t interfere with normal tabbing when hidden.

**Review:**

* What accessibility issues were most common?
* How did the fixes affect layout or design?
* What surprised you about using a screen reader?