**Responsive Navigation Design & Testing**

**Objective:**

Test a navbar’s usability across screen sizes. Students will be able to:

* Design user-friendly navigation systems.
* Implement mobile-first responsive navigation using HTML/CSS.
* Test usability and responsiveness across devices.
* Explain their design decisions using UX vocabulary.

Use this layout:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Responsive Nav Demo</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<header>

<nav class="navbar">

<div class="brand">MySite</div>

<button class="menu-toggle" aria-label="Toggle Menu">☰</button>

<ul class="nav-links">

<li><a href="#home">Home</a></li>

<li><a href="#services">Services</a></li>

<li><a href="#about">About</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

</header>

<main>

<section id="home"><h2>Home</h2></section>

<section id="services"><h2>Services</h2></section>

<section id="about"><h2>About</h2></section>

<section id="contact"><h2>Contact</h2></section>

</main>

</body>

</html>

**Part 1: Mobile-First Navigation**

1. Create a styles.css file and apply Flexbox to arrange the nav.
2. Hide the .nav-links on mobile using media queries.
3. Use JavaScript to toggle .nav-links on and off when the menu button is clicked.
4. Use ARIA attributes and tabindex to ensure accessibility.

**Part 2: Desktop Navigation**

1. Use a media query to display .nav-links inline on screens wider than 768px.
2. Remove the menu toggle on desktop and ensure hover/active states are visible.

**Part 3: Usability Testing**

Test the site using:

* **Chrome DevTools** in mobile and responsive view.
* **Keyboard navigation**: Can users tab through the links easily?
* **VoiceOver/NVDA**: Are the navigation elements announced properly?
* How to record short usability feedback with questions, or give them a task:
  + “Was it easy to find each section?”
  + “Did anything confuse or frustrate you?”

**Part 4: Analysis & Iterative Approach**

After testing:

* Refine padding, colors, and hover styles for clarity.
* Add aria-current="page" to indicate the active section.

Were the issues you found usability or accessibility?

**Usability:** How easy, intuitive, and efficient it is for any user to accomplish a task. Example: A confusing "Go" button label hinders task completion

**Accessibility:** Whether people with disabilities can perceive, navigate, and interact. Example: Missing alt text prevents screen reader users from understanding an image.

**Example Usability Issues:**

* "Go" button label is vague – users don’t know what will happen
* Duplicate “Learn More” links – creates cognitive load
* Navigation disappears on mobile – users can’t navigate at all
* Placeholder text used instead of labels – hard to recover from input mistakes
* Unclear hierarchy in layout or actions (no visual clues for scanning)

**Example Accessibility Issues:**

* Low contrast text in nav – visually impaired users can’t read it
* Missing alt text – screen reader users get no context
* No keyboard focus styling – keyboard users lose track of navigation
* No form labels / ARIA – screen readers can’t identify fields
* Missing ARIA roles / skip links – affects assistive tech experience

**Summary**

* Usability affects everyone
* Accessibility ensures inclusion of users with disabilities

**Usability and accessibility might overlap:**

* A vague button (usability issue) also affects users with cognitive disabilities (accessibility).
* Poor contrast affects both low-vision users (a11y) and users on sunlit screens (usability).

**When to Use A/B Testing**

* CTA wording or color
* Headline variations
* Layout options (e.g., card vs. list view)
* Form length (short vs. long)
* Navigation styles