

Designing Accessible Digital Products



- Training for UX/UI designers and product creators on accessible interfaces



- Business, legal, and ethical drivers: reach, revenue, risk, and inclusion



- Course roadmap: from POUR principles to design-system integration



- 2026 landscape: EAA enforcement, rising ADA litigation, WCAG 2.2 AA standard

The Business Case for Accessibility



Market reach: Over 1 billion people with disabilities, \$13 trillion in disposable income



Revenue impact: 89% say accessibility provides competitive advantage; 75% tie it to improved revenue



Legal risk: 59% of organizations feel at risk of legal action; 80% in U.S. public sector



ROI evidence: Legal & General achieved 100% ROI in 12 months; Tesco saw £13M annual revenue from a £35K investment



Procurement requirements: 74% of organizations now require documented accessibility proof when purchasing digital products



Legal and Standards Landscape



- EAA enforcement began June 28, 2025; covers e-commerce, banking, and transport



- Over 4,000 ADA digital accessibility lawsuits filed annually in the US



- WCAG 2.2 is the current W3C Recommendation; adds 9 new cognitive and motor criteria



- Conformance levels: A (minimum), AA (industry standard), AAA (highest)

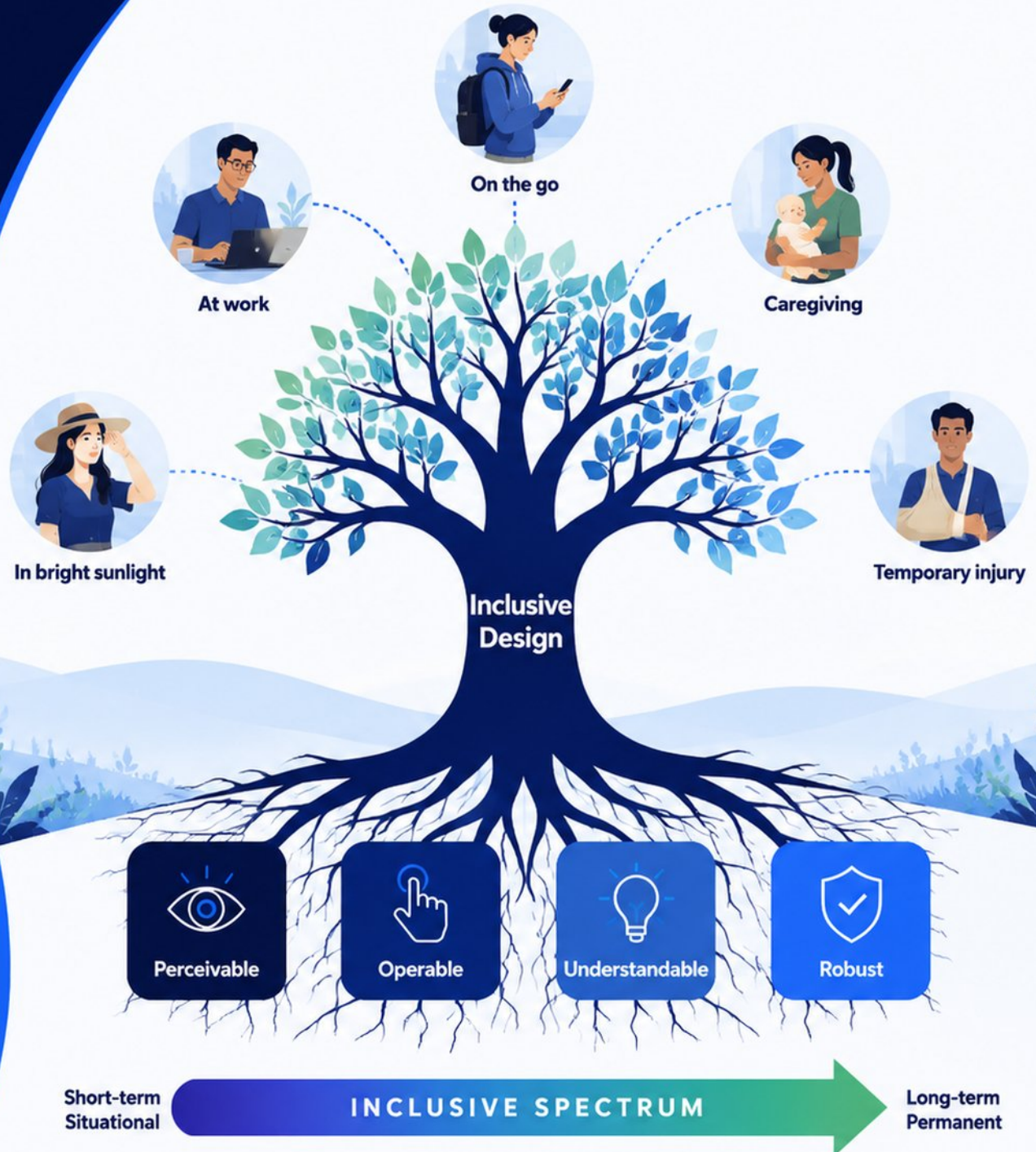


- EN 301 549 harmonized with EAA; creates presumption of conformity



Core Principles of Inclusive Design

- Distinguish accessibility, usability, and inclusive design
- The POUR framework: Perceivable, Operable, Understandable, Robust
- Shift from a medical model to a social and situational model of disability
- Temporary and situational impairments: broken arm, bright sunlight, holding a baby
- Inclusive design benefits everyone: curb cuts, closed captions, voice assistants



The POUR Framework in Practice

The POUR Cascade

Accessibility is built in layers.



P

Perceivable

- alt text, captions, sufficient contrast, adaptable content



O

Operable

- full keyboard access, visible focus, no seizure triggers



U

Understandable

- readable language, predictable behavior, error assistance



R

Robust

- semantic HTML, proper ARIA, assistive technology compatibility



WCAG 2.2 adds:



Focus
Not Obscured



Target
Size



Dragging



Accessible
Authentication



Consistent
Help



Redundant
Entry

Designing for Screen Reader Users



- Screen readers navigate via headings, landmarks, links, and forms mode



- NVDA and JAWS dominate Windows desktop; VoiceOver leads iOS, TalkBack leads Android



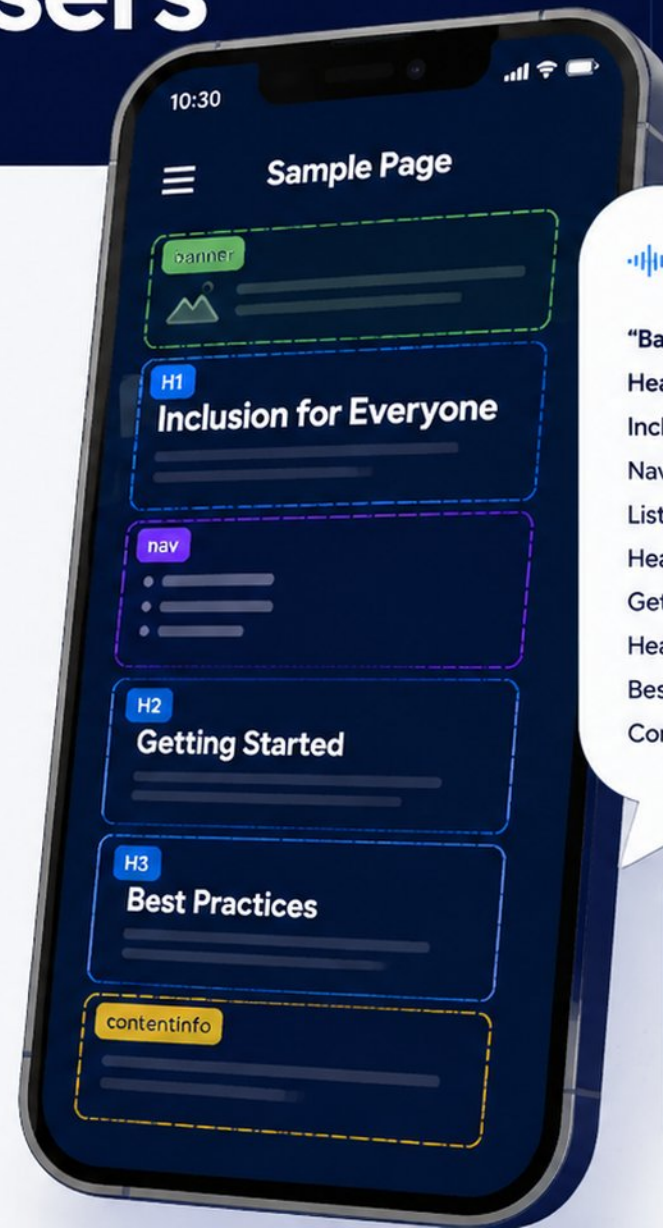
- 91.3% of screen reader users now use a mobile device; 71% use iOS



- 71.6% of users rely on headings to find information on a page



- Testing minimum: NVDA + VoiceOver; add TalkBack + JAWS for high confidence



Screen Reader Output

"Banner. Sample Page.
Heading level 1.
Inclusion for Everyone.
Navigation landmark.
List with 3 items.
Heading level 2.
Getting Started.
Heading level 3.
Best Practices.
Content info."

Landmark Key

- Banner
- Navigation
- Headings
- Content Info

Keyboard Accessibility and Focus Management



- Every interactive element must be reachable and operable by keyboard alone



- Focus order must follow visual layout; skip links bypass repetitive navigation



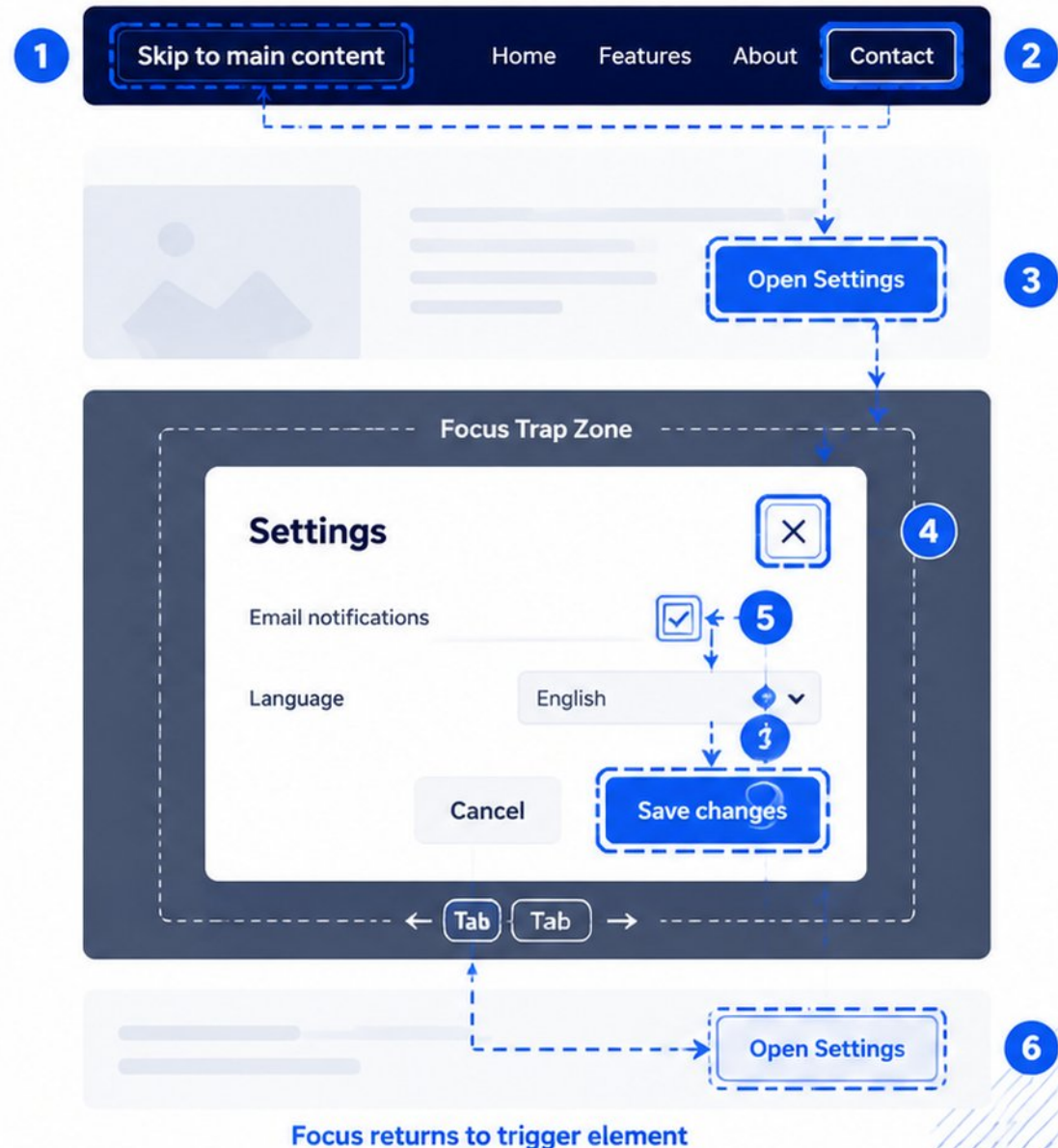
- Focus indicators must be visible, at least 2px thick, with 3:1 contrast ratio



- Modal dialogs must trap focus until dismissed; focus returns to trigger element



- WCAG 2.2: Sticky headers, cookie banners, and chat widgets must not cover focused elements



Visual Design: Color, Contrast, and Typography



- **Meet WCAG contrast:** 4.5:1 for normal text, 3:1 for large text and UI components



- **Never rely on color alone:** add icons, patterns, or text labels



- **Use accessible typography:** adequate font size, line-height, and responsive scaling



- **Support text resize** up to 200% without loss of content or functionality



- **Watch for failures:** text on hero images without contrast, placeholder text as labels



BARRIER (Hard to Use)

My Account

Username

Enter username

Password

Enter password

SIGN IN

Forgot password?

Discover your next adventure

LEARN MORE

- Low contrast text
- Color-only error states
- Placeholder used as labels
- Missing icons and indicators



ENABLER (Accessible & Usable)

My Account

Username

 Enter username

Password

 Enter password 

SIGN IN

Forgot password?

 Discover your next adventure

LEARN MORE

- ✓ Meets contrast standards
- ✓ Icons + text labels
- ✓ Clear error and focus states
- ✓ Supports text resize up to 200%

Accessible Content and Microcopy



- Write clear, scannable labels, instructions, and error messages — not just "Error"



- Alt text: descriptive for informative images; alt="" for decorative; full text for complex charts



- Link text must describe the destination: never "click here" or "learn more"



- Video needs captions; audio needs a transcript; both need audio descriptions for visual content

Quarterly Website Traffic

Quarter	Traffic (K)
Q1	15K
Q2	25K
Q3	35K
Q4	45K

Alt text for chart
A bar chart showing website traffic by quarter. Traffic increases from 15K in Q1 to 45K in Q4.

Unlimited access to our accessibility training library

Explore training topics by skill level →

Descriptive link text
Clearly describes the destination and what users will find there.

Video with captions
Captions shown on video. Transcript available. Audio description explains key visuals.

Designing Accessible Forms

- 1 Every input needs a visible, programmatic label; prefer explicit `<label>` elements
- 2 Required fields: indicate with both visual marker (*) and screen reader text
- 3 Error identification: describe what is wrong and how to fix it, not just "Error"
- 4 Inline validation: provide real-time feedback without disrupting user flow
- 5 WCAG 2.2 Redundant Entry: auto-populate previously entered info; offer "Same as shipping" options

WHAT USERS SEE

Visible Label
(with required marker)

Email Address *

Input Field
(user entry)

alex@example.com|

Helper Text
(guidance)

We'll never share your email.

Error Message
(descriptive)

⚠ Please enter a valid email address.
Example: name@example.com

Inline Validation
(real-time feedback)

✓ Looks good!

Layered
User
Experience

UNDER THE SURFACE

Programmatic Label
(for assistive tech)

```
<label for="email">Email Address (required)</label>
```

Accessibility
(linked programmatically)

```
<input type="email" id="email1" required  
aria-describedby="email-help email-error email-status">
```

Programmatic
Connection

THE POUR
CASCADE



Perceivable

Information is presented so users can perceive it.



Operable

Interface components and navigation are usable by everyone.



Understandable

Information and operation of the UI are clear and predictable.



Robust

Content works reliably with assistive technologies.

Designing for Neurodiversity and Cognitive Load



- **Reduce cognitive load:** predictable layout, progressive disclosure, consistent navigation



- **Support reading:** adequate line spacing, optimal reading width, clear visual hierarchy



- **Respect motion sensitivity:** honor prefers-reduced-motion, add pause/stop controls



- **WCAG 2.2 cognitive criteria:** Consistent Help, Redundant Entry, Accessible Authentication



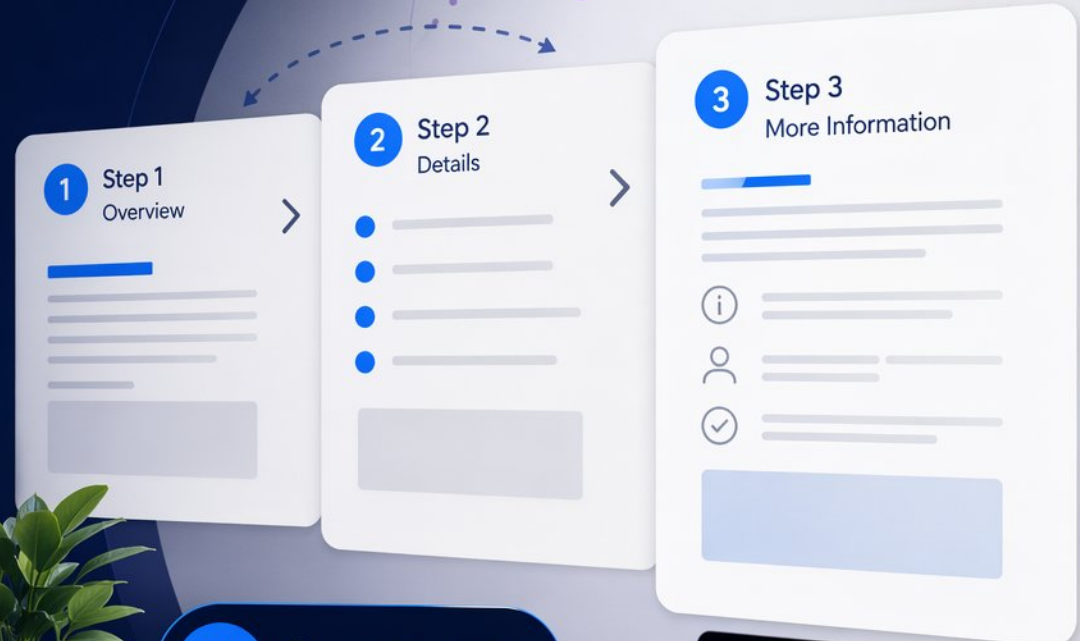
- **13.9%** of US adults report cognitive disabilities — the most common disability category



13.9% of US adults report cognitive disabilities — the most common disability category



INCLUSIVE
SPECTRUM



Pause Animations
You're in control



Touch Targets and Input Modalities



- Interactive targets must be at least 24x24 CSS pixels (WCAG 2.5.8)



- Spacing exception: smaller targets allowed if a 24px circle won't overlap adjacent targets



- Any drag interaction needs a single-pointer alternative: click, tap, or button (WCAG 2.5.7)



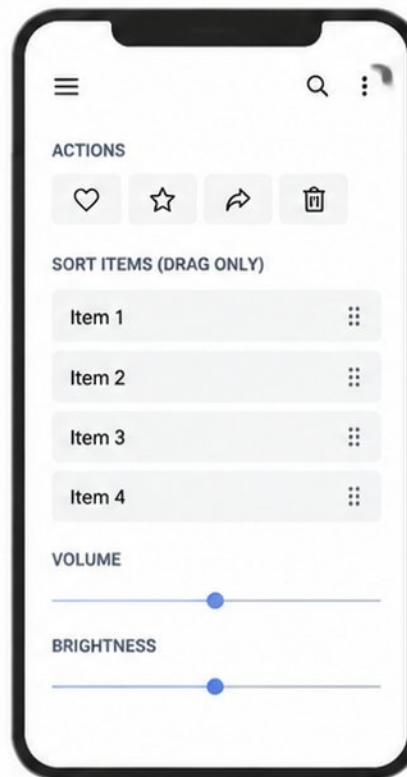
- Design for voice control, switch devices, and screen magnification



- Common failures: tiny icon buttons, drag-only sortable lists, sliders without +/- buttons

✗ BARRIER

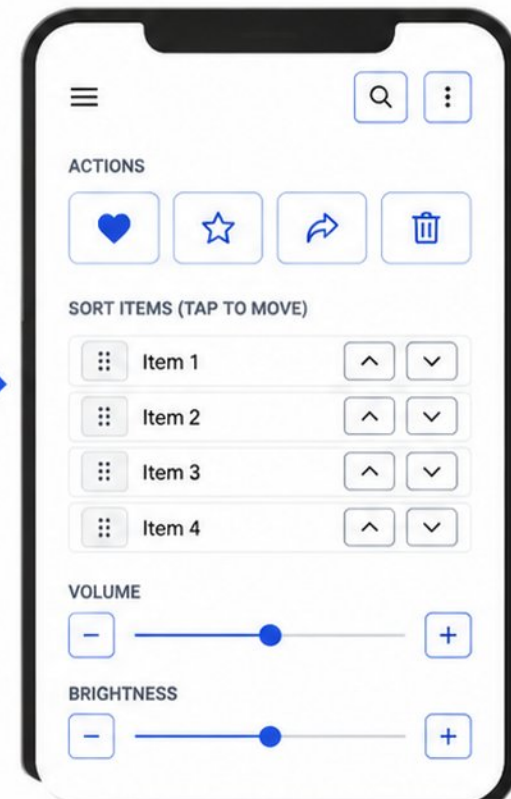
Hard to Use



Tiny targets, drag-only interactions, no alternatives

✓ ENABLER

Accessible & Usable



24px+ targets, tap alternatives, +/- controls, accessible for all



Accessible design benefits everyone.



Meet WCAG 2.5.7 and 2.5.8.



Usable today. Accessible tomorrow.

Accessible Authentication



- No cognitive test without an alternative (3.3.8 AA)



- Support password managers: never disable paste



- Offer alternatives: magic links, passkeys, biometrics



- CAPTCHAs: object recognition allowed at AA; removed at AAA



- Design login flows for memory, dyslexia, and cognitive disabilities

Start: Login

Email address

user@example.com

Password

• • • • •



Sign in

Prefer an alternative?



Use a passkey

Sign in with your device securely



Send me a magic link

Email a link for passwordless sign in



Use biometrics

Sign in with your fingerprint or face



Continue with OAuth

Use your account from a trusted provider

Accessibility in Your Design System



- Embed accessibility into component specs, tokens, and docs from day one



- Use design tokens for focus ring styles, min target sizes, and contrast pairs



- Build buttons, form fields, modals, tabs, and accordions as accessible defaults



- Prefer native HTML; use ARIA only when no native equivalent exists



- Document keyboard interactions, focus management, and screen reader announcements per component

COMPONENT: BUTTON



The POUR Cascade



Perceivable
Information is available to users in more than one way.



Operable
Interface components and navigation are usable.



Understandable
Content and operation of the interface are clear and predictable.



Robust
Content is reliable enough to be interpreted by a wide variety of tools.

Annotating Designs for Accessibility



What to annotate:

headings, landmarks, focus order, reading order, alt text, ARIA properties, and keyboard interactions



Annotation tools:

GitHub Annotation Toolkit, CVS Health Web Accessibility Annotation Kit, and the Include plugin



48% of GitHub's accessibility issues could have been prevented in the design phase



Annotation frameworks:

document focus order, semantic roles, states, and accessible names



Handoff best practices:

pair annotations with component documentation for developer clarity

Alt text
alt="Mountain landscape"

Landmark
role="banner"

ARIA property
aria-label="Search"

Semantic role
role="region"
aria-labelledby="features-heading"

ARIA property
aria-describedby="newsletter-desc"

Landmark
role="contentinfo"

Focus order
Logical tab sequence for keyboard users.

1 → 2 → 3 → 4 → 5 → 6

Design QA and Testing Tools

- Figma plugins: Stark for contrast & focus order; A11y Smart Lens for token-aware fixes; Access Lens for EAA conformance checks
- Automated tools catch ~57% of issues; manual keyboard and screen reader testing is essential
- Testing checklist: keyboard navigation, screen reader walkthrough, zoom to 200%, color contrast, touch targets
- Vision simulators: test designs under color blindness, low contrast, and blurred vision conditions
- Integrate accessibility checks into design review and sprint planning



Building a Testing and Prioritization Strategy



- Integrate usability testing with people with disabilities into design



- Recruit via disability organizations and remote testing platforms



- Prioritize fixes with impact vs. effort matrix; start with authentication and forms



- Shift-left: fixing in design costs 1x; in production, 30-100x



- Sustain practice with champions, training, and continuous monitoring



Inclusive Spectrum

Design for one, extend to many.

Better for everyone.



Cognitive



Visual



Hearing



Motor



Speech



Neurodiverse



Age Diversity

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/3676c0bb/public