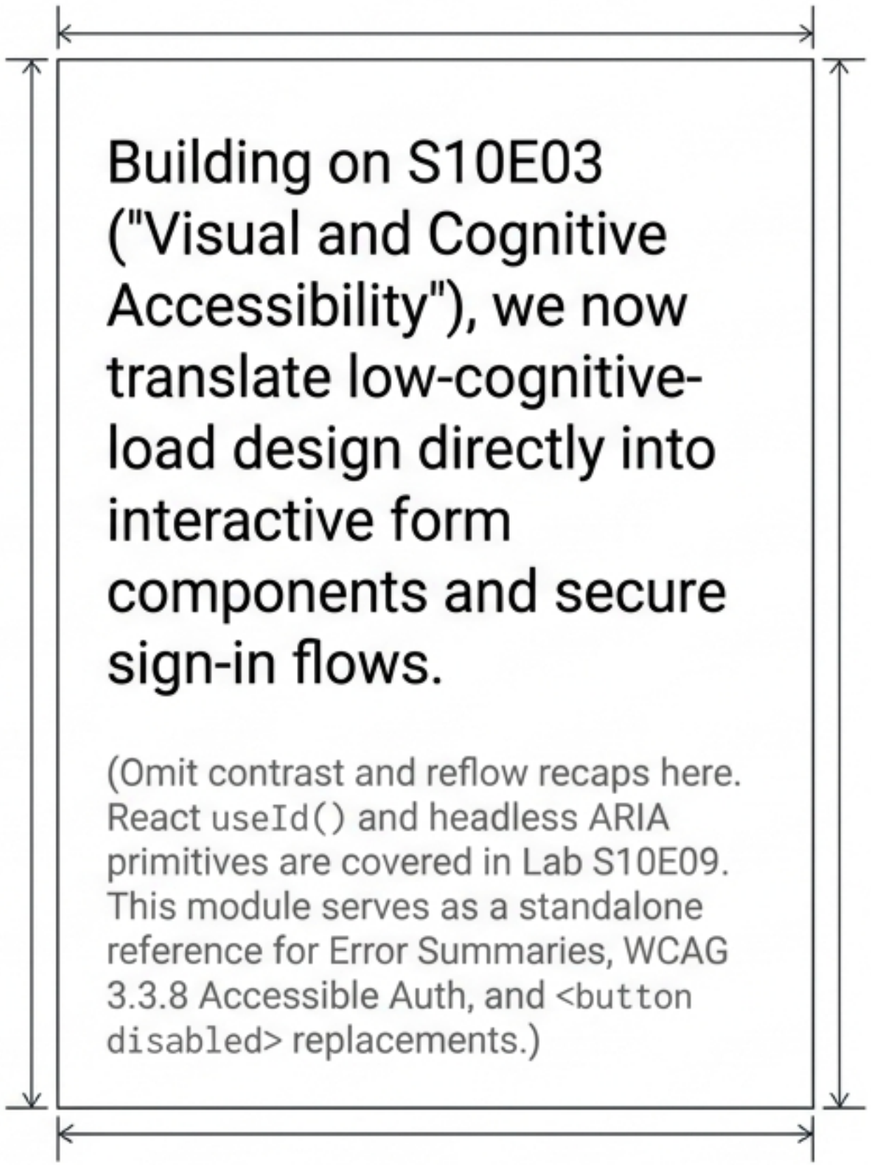
Digital Accessibility: S10E04

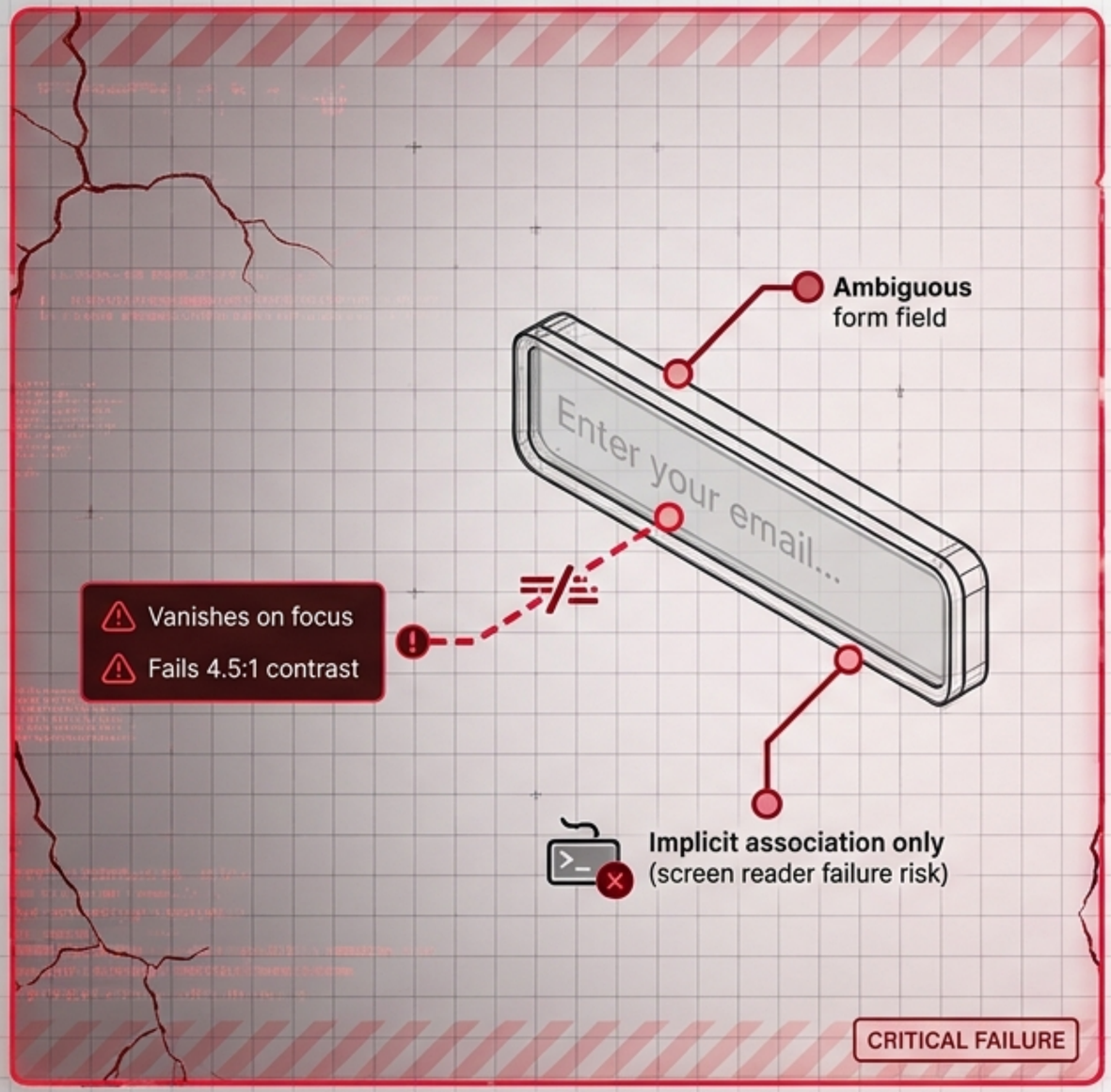
Forms, Controls, and Authentication

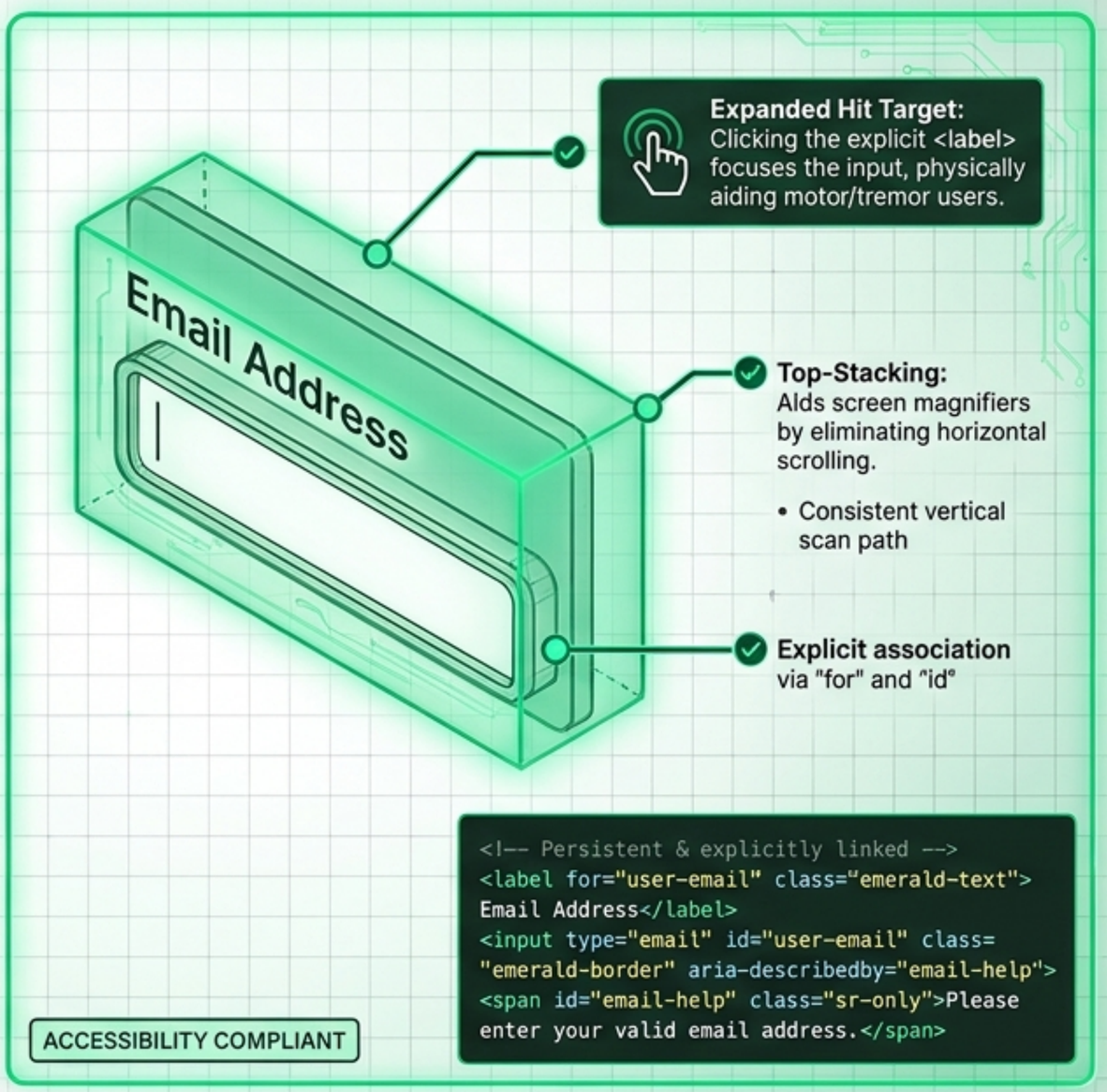
Building Implementation-Ready, Low-Friction Interactive Experiences.



Gemini Notebook

The Anatomy of an Accessible Field





Gemini Notebook

Grouping Complexity: Radios, Checkboxes, and Multi-Part Fields

Bold, fake heading div

Date of Birth

DD ▾ MM ▾ YYYY ▾

Context is entirely lost.

Screen Reader Output:
Combo box, 01.
Combo box, January.
Combo box, 1990.

CRITICAL FAILURE

Date of Birth (Optional)

DD ▾ MM ▾ YYYY ▾

Screen Reader Output:
Date of Birth, group.
Day, combo box, 01...

Rule: Use `<fieldset>` to programmatically bind controls. The `<legend>` acts as the overarching label announced upon entry.

Indicator Rule: Use explicit text like '`(Optional)`' inside the `<legend>` rather than relying on ambiguous, verbose red asterisks.

ACCESSIBILITY COMPLIANT

Gemini Notebook

Upfront Hints & The Automation of autocomplete

WCAG 3.3.2: Labels or Instructions

Password

MM/YY

```
<label for="password">Password</label>
<input type="password" id="password"
<input type="password" id="password"
  aria-describedby="hint-id">
<p id="hint-id">MM/YY</p>
```

Link format rules programmatically *before* the user makes a mistake. Do not rely on disappearing placeholders.

WCAG 1.3.5: Identify Input Purpose

Email

Jane Doe
jane@example.com

Monospace Tokens Box

```
email
username
current-password
one-time-code
```

By identifying the input purpose, you offload the user's cognitive and motor load to a single click via password managers and browser autofill.

Gemini Notebook

The “Disabled Submit” Trap

Screen Reader User

Tab-navigates right past the button. Since it is disabled, it is removed from the focus order.
Result: “Where did the submit button go?”

Keyboard-Only User

Hits “Enter” or tries to tab to the button. Nothing happens.
Result: Trapped without knowing which field is missing.

Low-Vision User

Encounters a low-contrast grey rectangle that fails the 4.5:1 contrast ratio.
Result: Unreadable control.

Name

Email

BRICK WALL

DISABLED

```
<button disabled="true">
    Submit</button>
```

J a

The Nagging Effect: Avoid onInput keystroke nagging that screams “Invalid!” before the user has finished typing.

Gemini Notebook

Calm Validation & The State of `aria-invalid`

The Calm Validation Lifecycle

Step 1: User Entry (Focus)

```
<input type="email"
 aria-invalid="false">
```

Action: User types. No validations fire on keystroke.

Step 2: Validation Trigger (Blur / Submit)

Action: Validation scripts run ***only*** when the user tabs away (blur) or attempts to submit.

Step 3: State Toggle & Chaining

```
<input type="email"
aria-invalid="true"
aria-describedby="hint-id err-id">
```

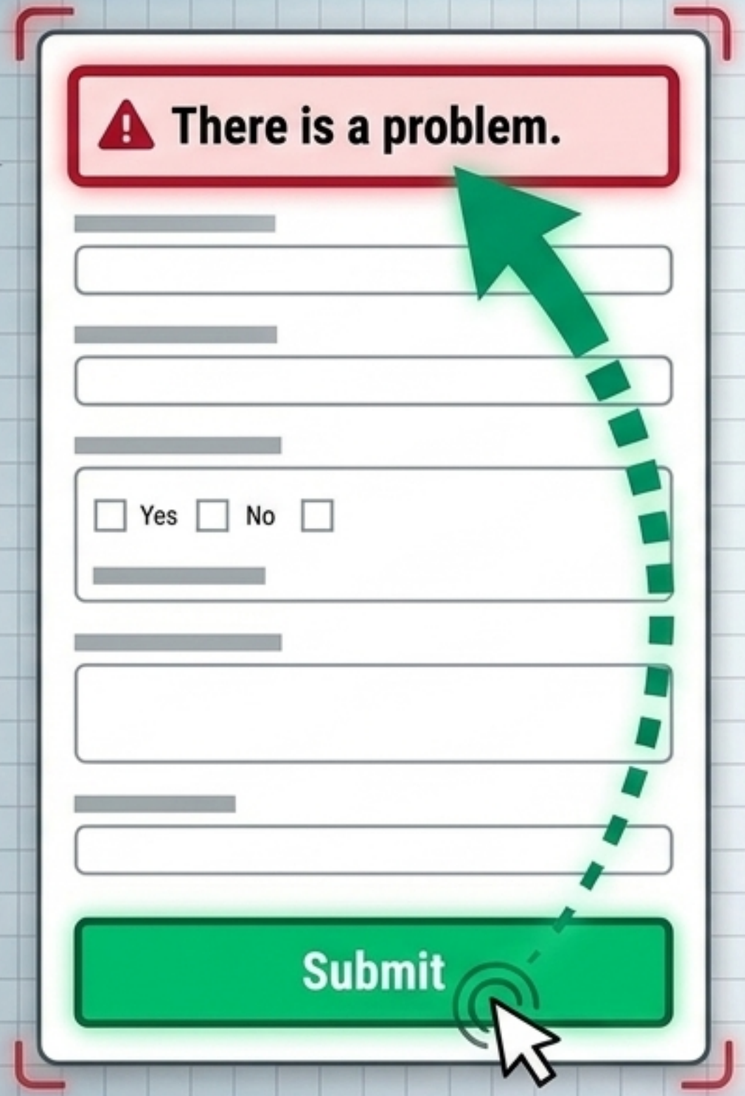
Only applied *after* validation fails.

Dynamically chains the new error message ID to the existing hint ID.

Design Note: Don't set **aria-invalid='true'** on empty required elements upon page load. Users are still working.

Gemini Notebook

The GOV.UK Error Summary Pattern

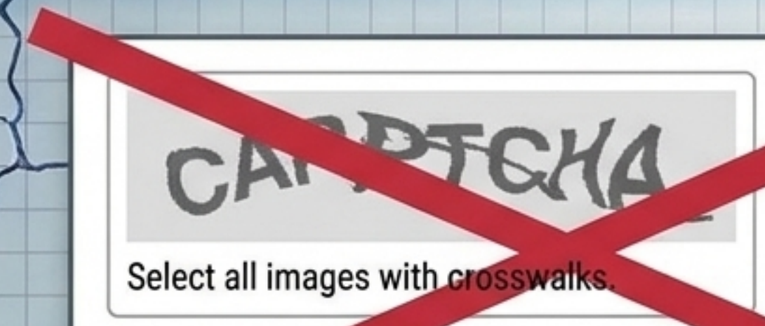


Anatomy of the Error Summary

1. **Always Enabled:** The submit button never disables.
2. **Focus Handoff:** On invalid submit, intercept the form, render the top summary, and forcibly move keyboard focus to it using `tabindex='-1'`.
3. **Anchor Jump Links:** Inside the summary, provide direct `<a href='#field-id'>` links to jump focus straight to the offending inputs.
4. **Error Prevention (WCAG 3.3.4):** Add a review step before finalizing critical legal or financial data.

Gemini Notebook

Eliminating Cognitive Function Tests at Sign-In



Distorted CAPTCHAs: Blocks blocks visually impaired and cognitively fatigued users.



6-Box OTP Grids: `<input maxlength='1'>` chained together. Breaks screen reader flow and blocks paste mechanics.

WCAG 2.2 (3.3.8 / 3.3.9):

Authentication cannot rely on memorization, transcription, or puzzle-solving without an accessible, non-cognitive alternative.



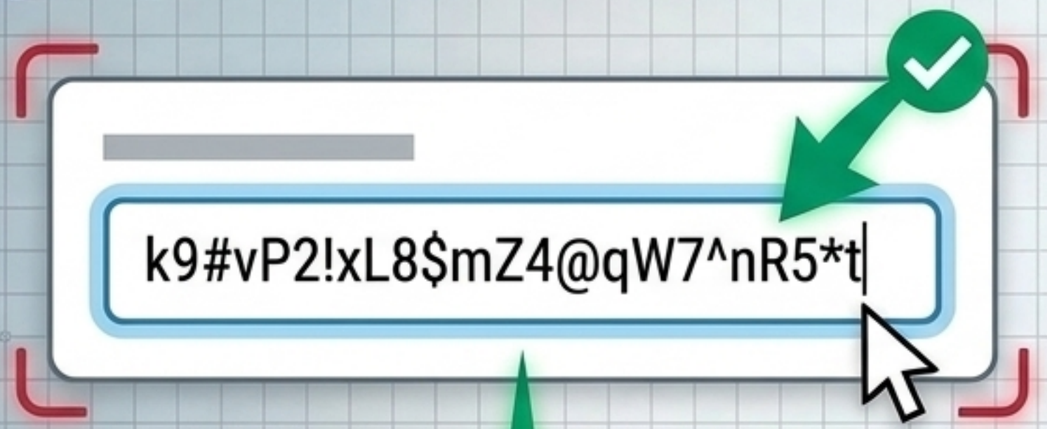
Blocked Paste: <input onpaste='return false'>. Forces dangerous password memorization.



Memory Transcription: Forces arbitrary recall and counting.

Gemini Notebook

Accessible Passwords & Unrestricted Paste Mechanics



Never block paste. Support standard `type='password'` with `autocomplete='current-password'` to harmonize with 1Password, Bitwarden, and OS keychains.

Hidden State

```
<button type='button'
aria-pressed='false'
aria-label='Show password'>
```

Visible State

```
<button type='button'
aria-pressed='true'
aria-label='Hide password'>
```

Use a real `<button>`, not a clickable `<i>` or `<span>`, to ensure native keyboard activation (Space/Enter).

Gemini Notebook

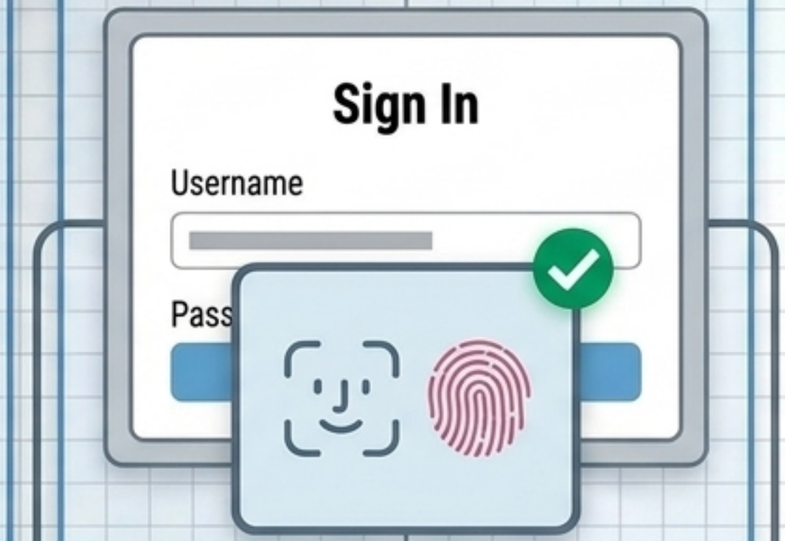
Modern Auth: Single-Field OTPs, Passkeys & Redundant Entry




```
<input
 autocomplete='one-time-code'
  inputmode='numeric'>
```

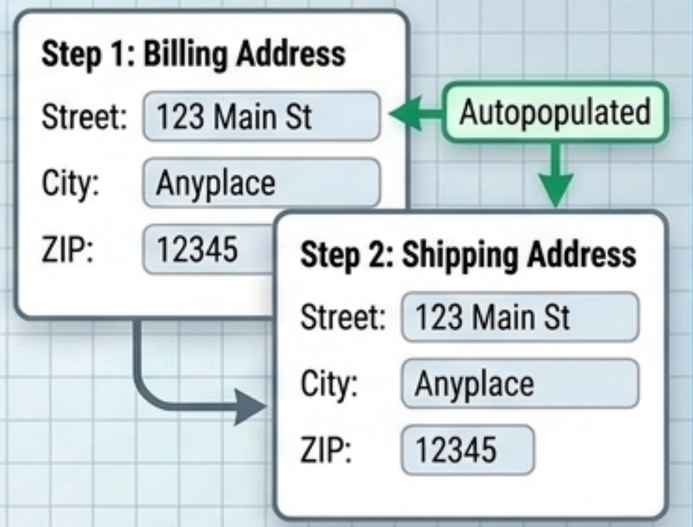
The Single-Field OTP

Allows the OS to read the incoming SMS/Email and offer a single-tap autofill.



WebAuthn / Passkeys

Phishing-resistant, zero-memory authentication. The ultimate accessible login.



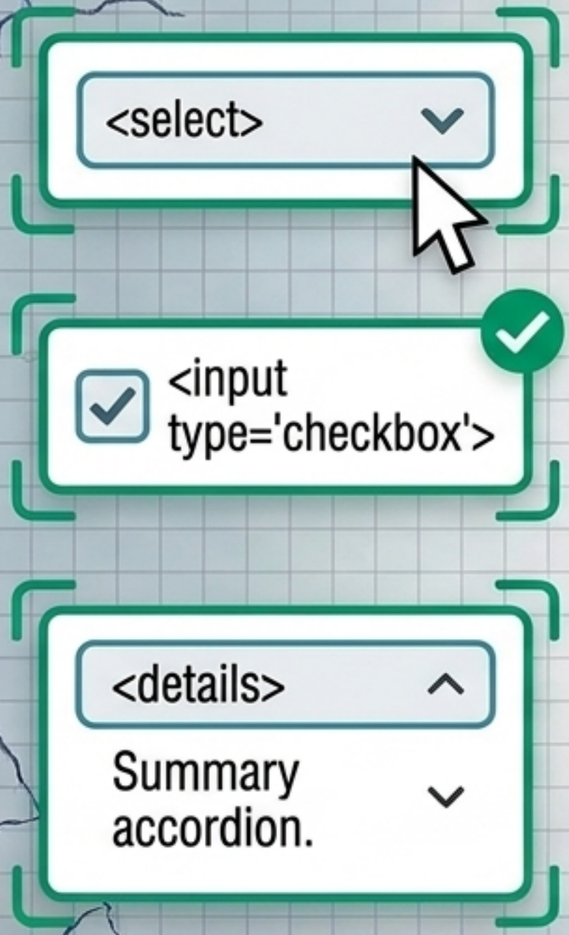
Redundant Entry (WCAG 3.3.7)

Users should never have to enter the same information twice in a single session.

Gemini Notebook

Native HTML Controls vs. The Soup

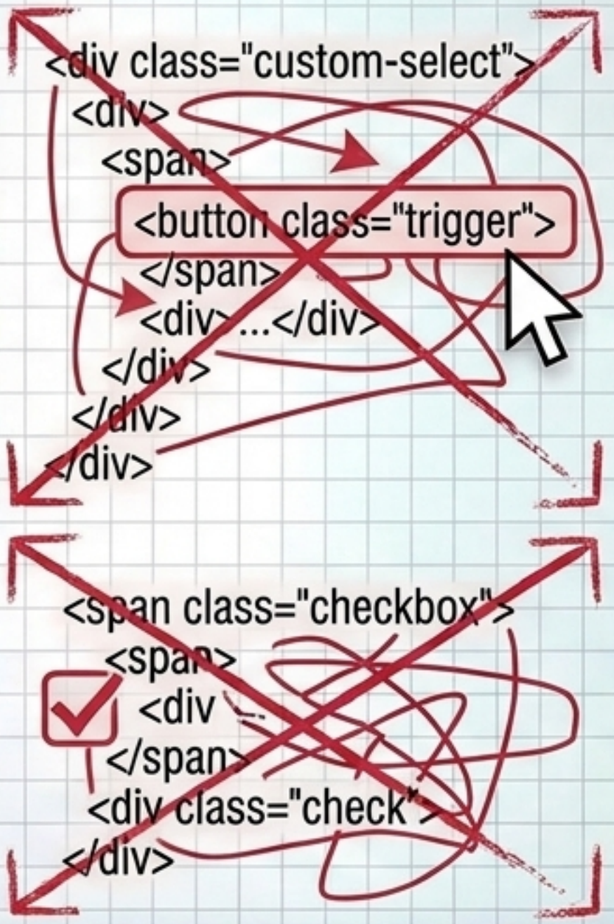
Native HTML – The Emerald Path



The Free Features List:

- Native Keyboard Navigation (Arrows, Space, Enter)
- 400% Zoom Reflow without breaking
- High-Contrast Mode (Windows HCM) support
- Mobile OS touch-optimized picker interfaces

Custom ARIA - The Crimson Path



The Debt List:

- You must write JavaScript for every keypress.
- You must manually manage focus.
- You must manually toggle all states.
- It will likely break on mobile screen readers.

Gemini Notebook

The Five Rules of ARIA & The Widget Contract

The 5 Rules (W3C Standard) – Contractual Clauses

Clause 1: Use native HTML whenever possible. (e.g., use <button> for buttons, not).

Clause 2: Do not change native semantics (e.g., don't make an <h2> a tab).

Clause 3: All interactive ARIA controls must be usable with the keyboard. (e.g., focusable, operable via Enter/Space).

Clause 4: Do not use role='presentation' or role='none' on focusable elements.

Clause 5: All interactive elements must have an accessible name (via label, aria-label, or aria-labelledby).

⚠ WARNING: The Widget Contract

If you build a custom ARIA widget, you legally bind yourself to manually manage role, tabindex, aria-expanded/checked, and focus handoffs.

Don't sign the contract unless you have to.

X _____

NO ARIA IS BETTER THAN BAD ARIA

Gemini Notebook

5-Modality Matrix: Operations Desk Pathway (Part 1)

Modality	Broken State	Fixed Pathway
Visual Experience	Vanishing placeholders, grey <button disabled>, distorted puzzle CAPTCHAs.	Top-stacked <label> + explicit hint, high-contrast Error Summary, Show Password toggle + Passkey biometric prompt.
Keyboard Experience	Blocked paste forcing manual typing, custom trapping focus, 6-box OTP trap requiring 6 tab stops.	Unrestricted paste allowed, native <select> utilizing arrow keys, single OTP input, programmatic focus handoff to the Error Summary on failure.

Gemini Notebook

5-Modality Matrix: Operations Desk Pathway (Part 2)

Modality	Broken State	Fixed Pathway
Screen Reader	‘Edit text, blank... Invalid entry... Button, dimmed, Submit.’ (User is lost).	‘Notification preference, group. Email updates, radio button, 1 of 2. Institutional email, edit text, required, invalid entry. Error: Enter an institutional email address...’
400% Zoom (320px viewport)	Wide side-label gutters force horizontal scrolling; native tooltips get clipped off-screen.	Single-column, top-stacked alignment: Label -> Hint -> Error -> Input, ensuring zero horizontal scroll.
Mobile / Voice / Switch	Unclickable custom elements; no OS integration.	Voice command ‘Tap Institutional email’ works perfectly; OS-level one-time-code and Passkey autofill eliminate typing.

Gemini Notebook

Annotated Code Diff: The Accessible Input Contract

```
<label for="org-email"

<label for="org-email">Institutional Email</label>
<p id="org-email-hint">Use your .edu or organization address.</p>
<p id="org-email-err" class="error">Error: Enter a valid institutional email.</p>

<input type="email"
       id="org-email"
       autocomplete="email"
       required
       aria-invalid="true"
       aria-describedby="org-email-hint org-email-err">
```

Strict matching binds label to input.

Satisfies WCAG 1.3.5.

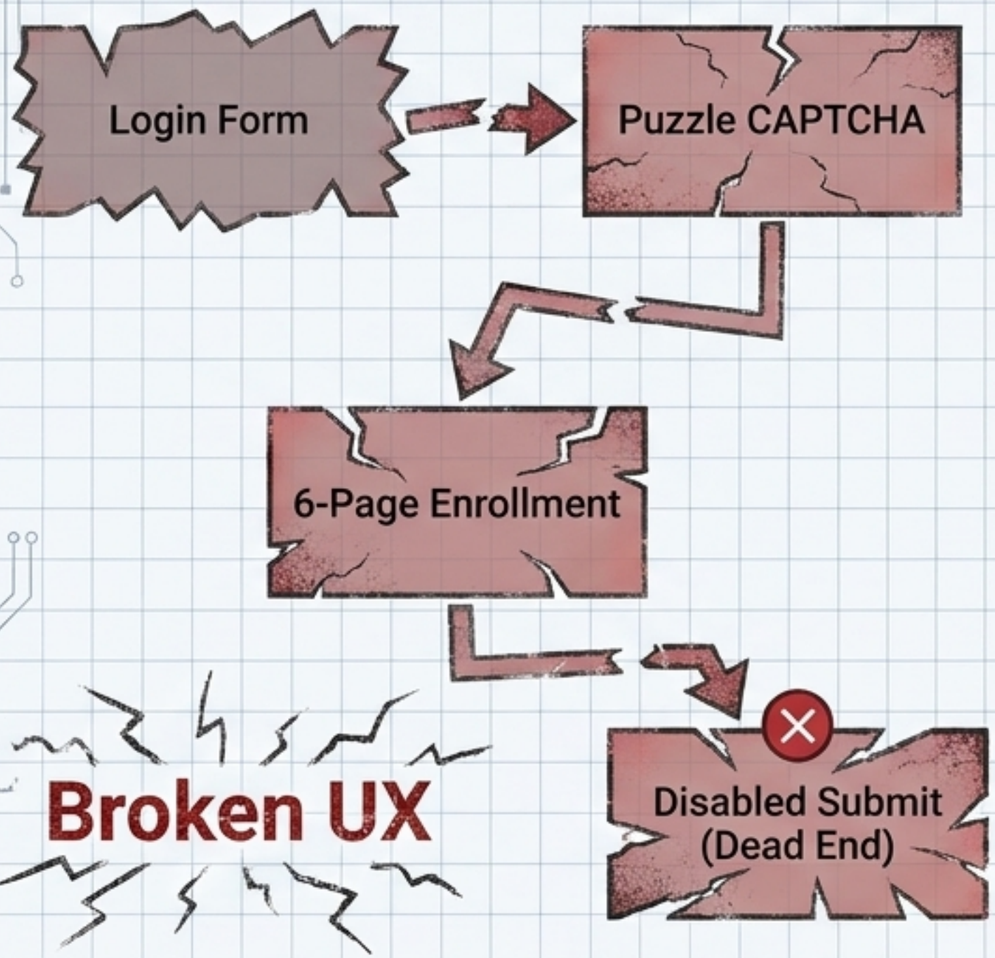
Toggled dynamically only after validation.

The Chain: Links both the hint ID and the error ID via a space-separated list, ensuring the screen reader announces everything in one cohesive string.

Gemini Notebook

Hands-On: Refactoring the Operations Desk Sign-In Flow

Broken UX



Unified Pathway

Step 1: Stripped out puzzle CAPTCHAs, replaced with WebAuthn/Passkey options.

Step 2: Replaced the 6-box OTP grid with a single autocomplete="one-time-code" field.

Step 3: Step 2 Enrollment data auto-populates via localStorage, satisfying WCAG 3.3.7 (Redundant Entry).

Step 4: Replaced the disabled submit button with an **active, high-contrast button that intercepts failures and triggers a top-focused Error Summary.**

Gemini Notebook

The 5-Point Form & Authentication Checklist

- **Structure & Grouping: <label>** explicitly tied to inputs; **<fieldset>** & **<legend>** used for multi-part questions and groupings.
- **Automation (1.3.5 & 3.3.7):** autocomplete tokens deployed appropriately; multi-step flow data auto-populated (no redundant entry).
- **Accessible Auth (3.3.8):** Paste mechanics function normally; autofill works; WebAuthn/Passkeys supported; NO cognitive tests (CAPTCHAs).
- **Error Handling (3.3.1 - 3.3.4)**: Submit button remains ENABLED; validation failures trigger a top-focused Error Summary and toggle aria-invalid.
- **Semantics First**: Native HTML controls (<select>, <details>) are exhausted before custom ARIA is ever considered. ('No ARIA is better than Bad ARIA').

Gemini Notebook

Next Up in the S10 Series...

[✅] S10E04: Forms, Controls, and Authentication Mastered.

S10E05: Dynamic Interfaces and Mobile Apps

Tackling single-page application (SPA) routing, focus management during dynamic DOM updates, and the nuances of iOS VoiceOver vs. Android TalkBack.

Ready for the code?
Scan to access Lab S10E09 covering React **useId()** implementation and headless ARIA primitives.

Gemini Notebook