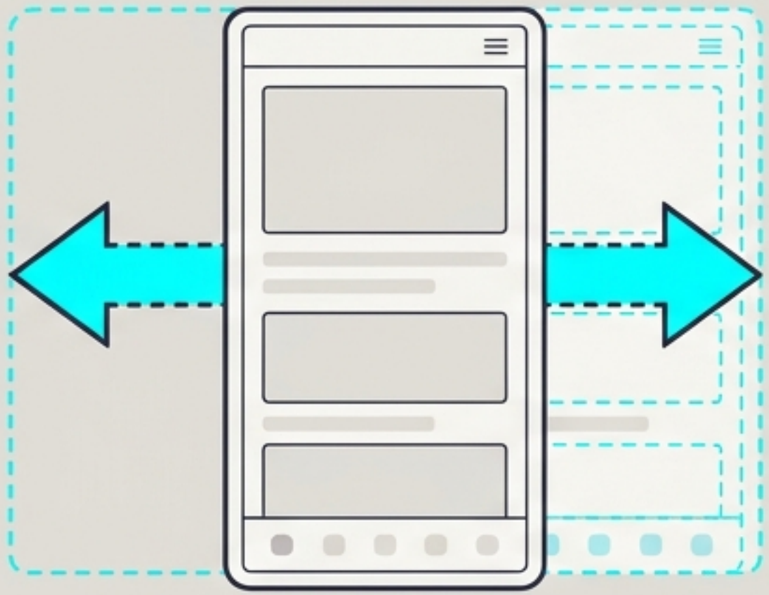
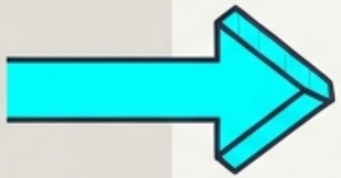
Series 10 / Digital Accessibility / S10E00

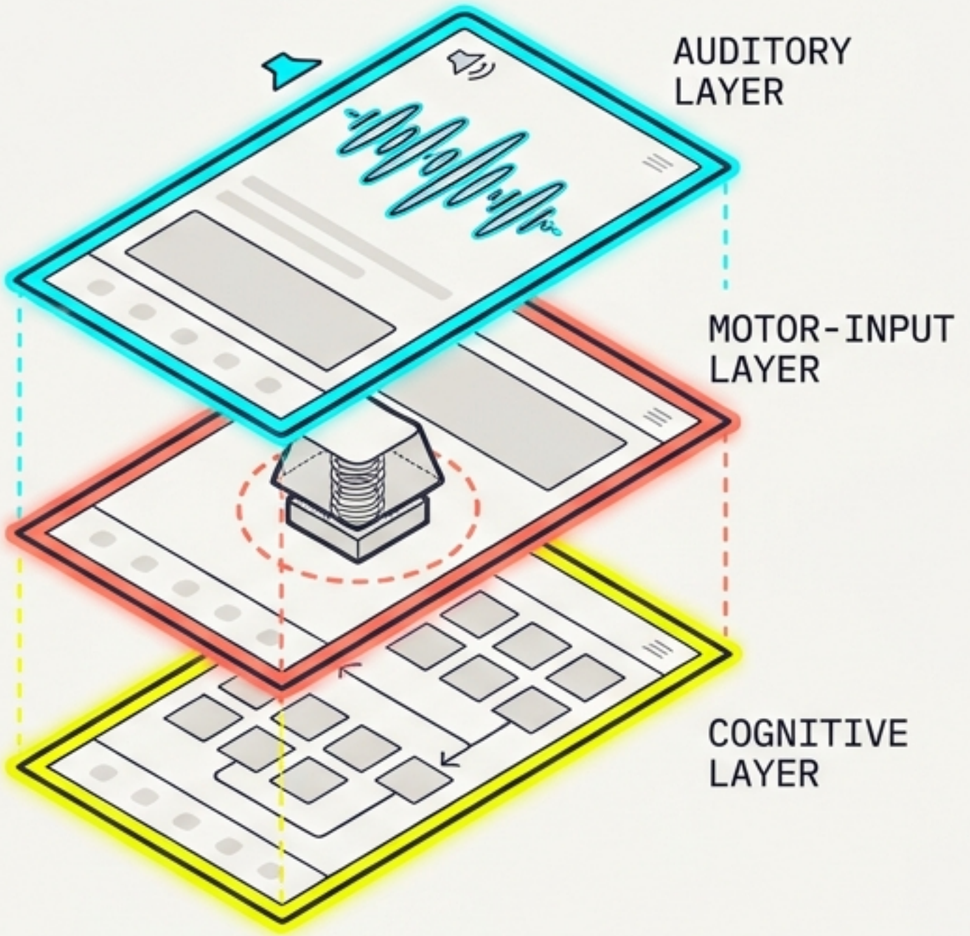
Series Overview: Engineering for Independence

From responsive layouts to full-stack, multi-sensory product accessibility.

Gemini Notebook







Responsive design adapts to the viewport. Accessible design adapts to the human.

In S09E11, we mastered visual foundations. Now, we expand into multi-sensory engineering. Digital accessibility means removing design-induced barriers across perception, motor input, auditory access, and cognition.

The Goal: Disabled users must complete journeys with comparable independence, time, and effort.

Gemini Notebook



Perceivable

Users can perceive the information.

`text_alternatives, media_captions, semantic_html`



Operable

Users can operate the interface.

`keyboard_access, predictable_focus, ample_time`



Understandable

Users comprehend the UX.

`readable_text, predictable_behavior, error_correction`



Robust

Content works across current/future tools.

`valid_markup, aria_roles, AT_compatibility`

Gemini Notebook

Designers

Define focus states, color contrast, and semantic hierarchy.

Vibe Coders (AI-Assisted)

Ensure LLM-generated UI components are contextually accessible, moving beyond visually plausible to structurally sound.



FE / Mobile Engineers

Implement native markup, ARIA properties, and robust focus management.

QA

Execute 5-modality smoke tests and validate with screen readers.

PMs

Prioritize inclusive user stories and defend accessibility in the backlog.

Gemini Notebook

CH 2 | The WebAIM Million: A Baseline of Broken Web

95.9%

of the top 1,000,000 home pages fail WCAG 2.0 A/AA standards.

Avg errors/page: 56.1

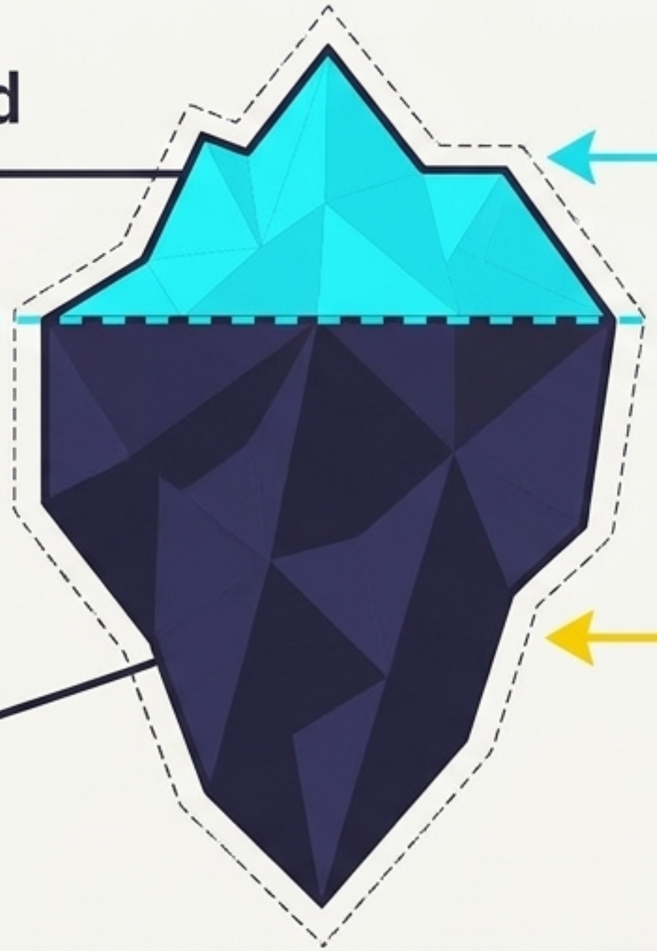
1. Low contrast text (83.9%)
2. Missing image alt text (53.1%)
3. Missing form labels (51.0%)
4. Empty links (46.3%)
5. Empty buttons (30.6%)
6. Missing document language (13.5%)

Warning: Increasing page complexity (avg. 1,437 elements/page) directly correlates with accessibility regression.

Gemini Notebook

CH 2 | Why Automated Linters Fall Short

Automated Scanners (30-40%)



Context & Usability (60-70%)

Linters easily catch missing `alt` attributes, poor contrast ratios, and missing `<html lang>`.

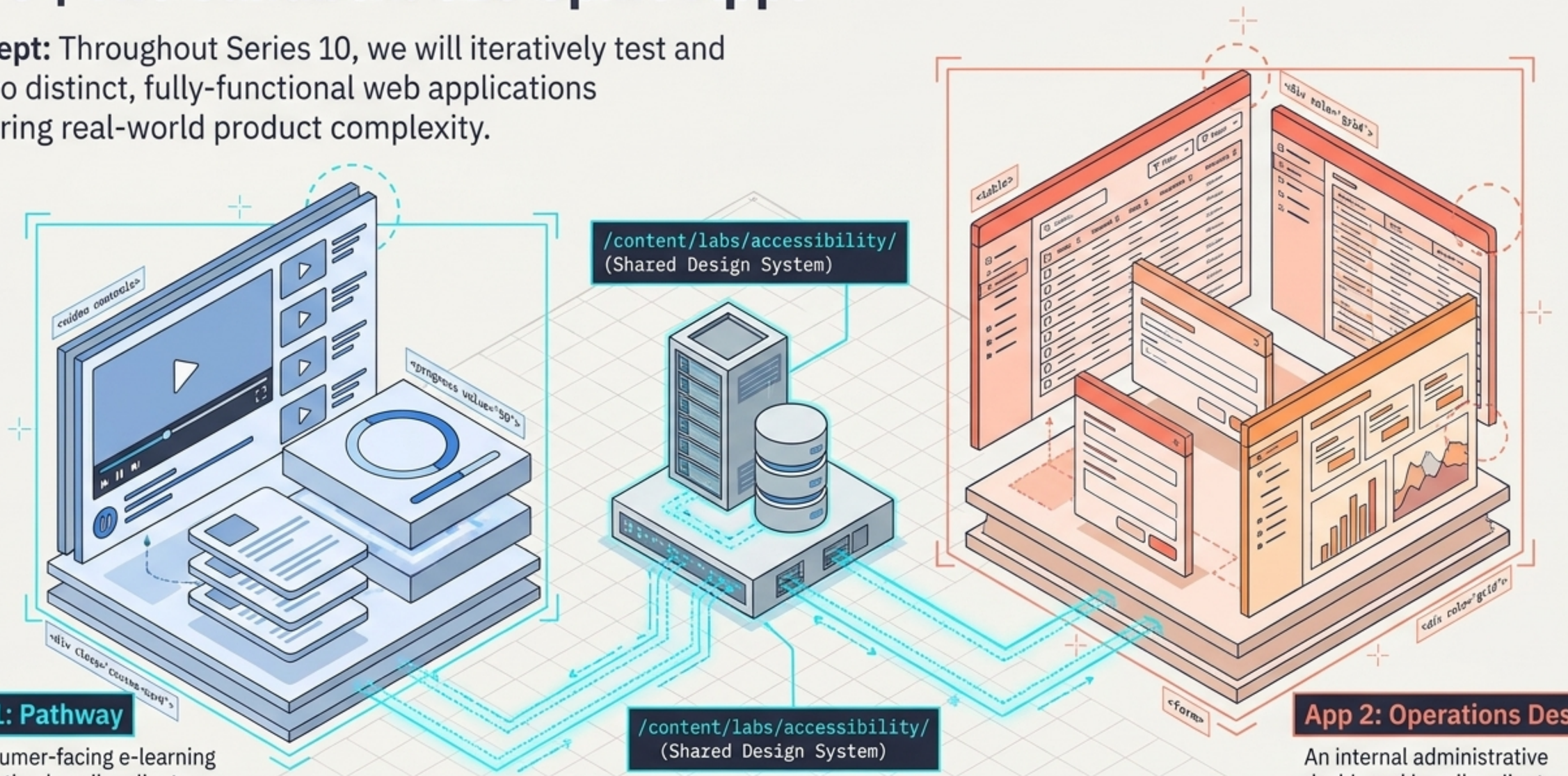
- Linters cannot tell if `alt='image'` is a meaningful description.
- They cannot test if a custom `<div>` click handler traps keyboard focus.
- They cannot verify if a complex dynamic DOM update makes sense when read aloud by VoiceOver.

The Rule: Automation is your baseline, manual testing is your reality.

Gemini Notebook

CH 3 | The Sandbox: S10 Spine Apps

Concept: Throughout Series 10, we will iteratively test and fix two distinct, fully-functional web applications mirroring real-world product complexity.



App 1: Pathway

A consumer-facing e-learning application heavily reliant on media and dynamic progress updates.

App 2: Operations Desk

An internal administrative dashboard heavily reliant on complex forms, data tables, and overlay states.

Gemini Notebook

CH 3 | Pathway: End-to-End LMS Journey



Step 1: Browse Catalog

Semantic heading stzucture and card navigation.



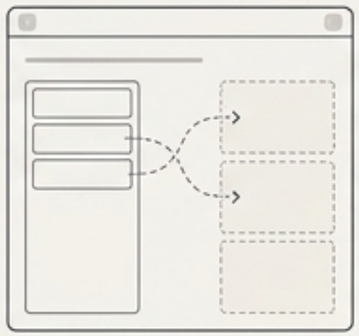
Step 2: Open Lesson

Keyboard-accessible routing and focus management.



Step 3: Play Video

Captions, transcripts, and audio descriptions.



Step 4: Interactive Exercise

Operable drag-and-drop or select components.



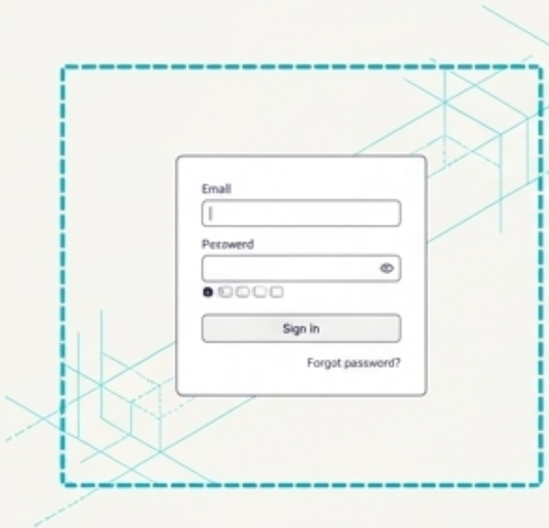
Step 5: Progress Save

Async live-region (aria-live) announcements without stealing visual focus.

Gemini Notebook

CH 3 | Operations Desk: Admin Dashboard Journey

Step 1: Sign-In



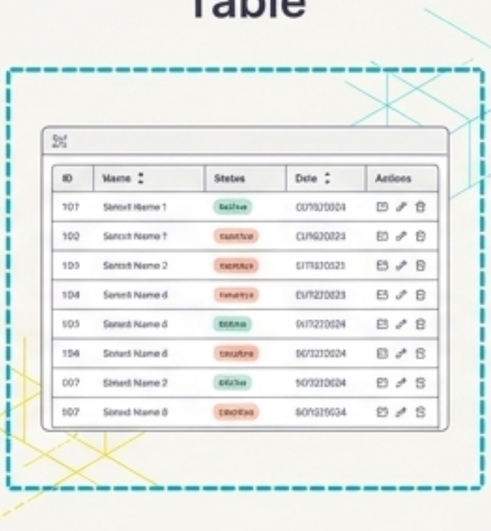
Accessible authentication with error recovery and input masking.

Step 2: Multi-Step Enrollment



Labeled form inputs, clear validation, and persistent context.

Step 3: Cohort Table



Fully sortable/filterable data grids with correct aria-sort and table headers.

Step 4: Overlays



Native <dialog> elements for modals/drawers featuring robust focus trapping and Esc key closures.

Gemini Notebook

CH 4 | The 5-Modality Broken → Fixed Matrix (Part 1)

	Broken State	Fixed State
Visual Modality	Pastel CTA & red-only error borders (fails contrast & color-reliance).	High-contrast design tokens + icon indicators for state changes.
Keyboard Modality	Tab-skipped `<div class='card-btn'>` & drawer focus traps.	Native `<a href>` / `<button>`, Enter/Space activation, and native Escape focus return.

Gemini Notebook

CH 4 | Matrix: Screen Reader, Zoom, Mobile/Switch

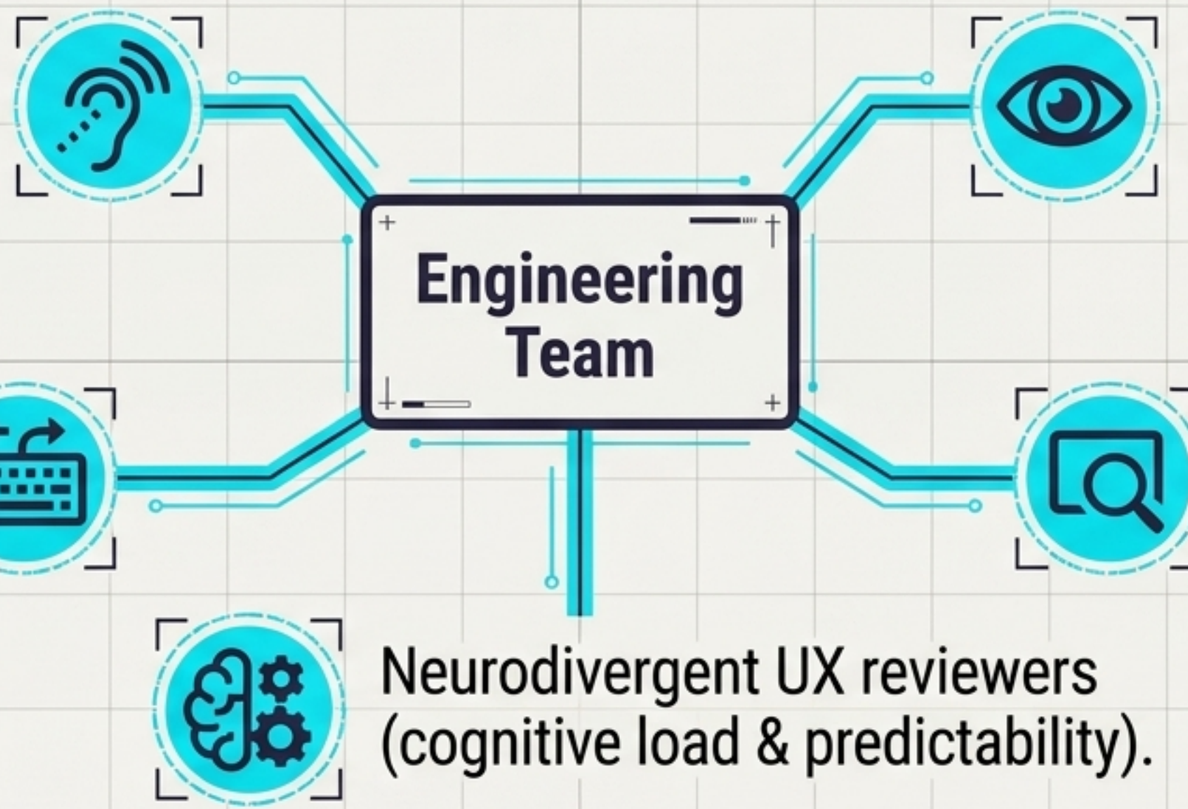
	Broken State	Fixed State
Screen Reader	`clickable, clickable, button, graphic icon_play_24, link`	WCAG 4.1.3 `Main landmark, Pathway Course Catalog, heading level 1. **Start Lesson 2: Landmark Regions**, button. Progress saved: Lesson 2 bookmarked, polite status`
400% Zoom	Clipped sticky toolbars, 2D stepper scroll locking (320 CSS px).	WCAG 1.4.10 Single-column reflow & fluid rem sizing (1.4.4/1.4.12).
Mobile / Voice / Switch	18x18px unnamed icons.	WCAG 2.5.8 >= 24x24 CSS px targets & matching Voice Control names.

Gemini Notebook

CH 5 | Co-Design: Engineering with Lived Experience

GOV.UK / W3C
WAI Protocol

Accessibility is a human right, not just a checklist. We design with, not for.



d/Deaf caption and transcript researchers.

Blind screen-reader software engineers.

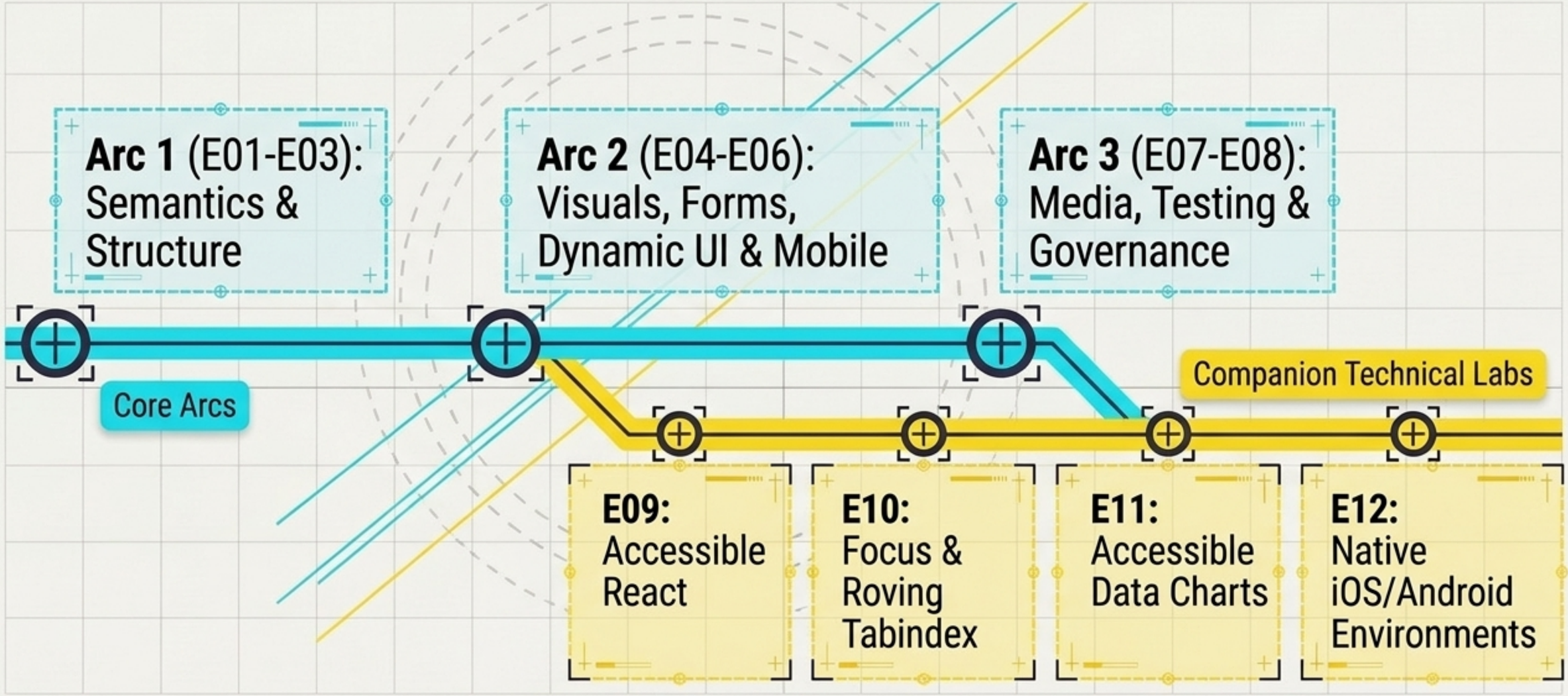
Voice/Switch navigation power-users.

Low-vision screen magnifier users.

Neurodivergent UX reviewers (cognitive load & predictability).

Gemini Notebook

CH 5 | Complete Season 10 Road Map (E01–E12)



Gemini Notebook

CH 6 | Code Diff: Semantic Action & Async Feedback

Broken Markup

```
<div class='btn' onclick='saveProgress()'>Save</div>
<div class='btn' onclick='saveProgress()'>Save</div>
```

Result: Silent visual toast, invisible to screen readers, un-tabbable.

Fixed Markup

```
<button type='button' class='btn' aria-describedby='lesson-meta'>Save Lesson Progress</button>
<div role='status' aria-live='polite' class='sr-status'>Saved Lesson 2 progress</div>
```

Polite status messages (WCAG 4.1.3) inform AT users of async updates without interrupting their current workflow.

Gemini Notebook



Step 1: Throw away your **mouse**. Can you reach and activate every control using only `Tab`, `Shift+Tab`, `Space`, `Enter`, and `Esc`?

Step 2: Turn on your screen reader (VoiceOver/NVDA). Close your eyes. Can you understand your location and actions?

Step 3: Zoom browser to 400%. Does the layout reflow into a single column without horizontal scrolling?

Step 4: Emulate a mobile viewport.
Are all touch targets >= 24x24 CSS pixels?



Step 5: Trigger a dynamic update (save/error).
Did the screen reader announce it?

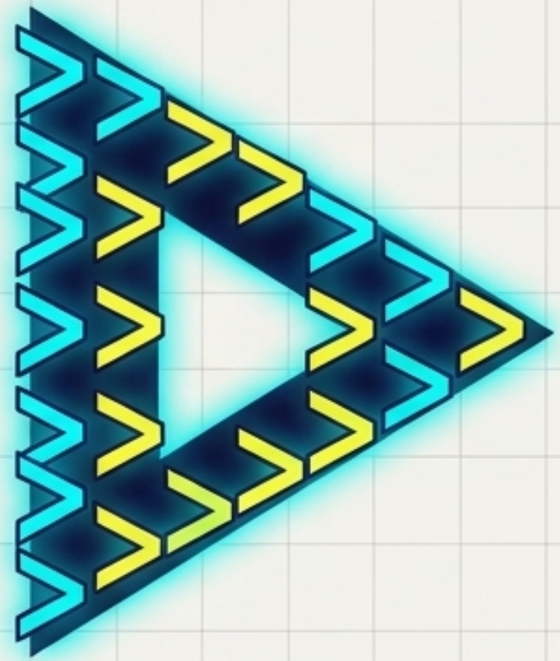


Gemini Notebook

CH 7 | Starter Testing Checklist: 4-Point Baseline

- **Reach & Visibility:** 100% keyboard reachability with a highly visible focus ring.
- **Reflow & Clipping:** Zero horizontal scrolling or text clipping at 320 CSS px width & 200% text spacing.
- **Semantic Context:** Meaningful roles/names and role=‘status’ speech for async updates.
- **Media & Targets:** Accurate captions/transcripts present, and touch targets are >= 24x24 CSS px.

Gemini Notebook



S10E01: People, Barriers and Assistive Technology

Focus: We will dive deep into how real users utilize screen readers, magnifiers, and voice control, moving beyond theory into live demonstrations.

Call to Action:

1. Bookmark the `/content/labs/accessibility/` repo.
2. Run the 10-minute smoke test on your current project today.

Gemini Notebook