

UI Design Examples: Patterns, Strengths, and Pitfalls



Build a repeatable critique habit —
not opinions about visual taste



For designers, developers, students,
educators, PMs, and reviewers



Three lenses per example: patterns,
strengths, pitfalls



Read context, user goal, and
constraints before judging



In 2026: calm density, inline AI,
accessibility as infrastructure



Leave with a critique template,
shared vocabulary, and critique card



Patterns



Strengths



Pitfalls



A Working Critique Framework

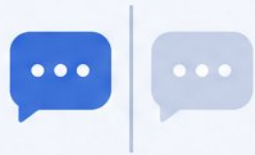
FROM LOOKING
TO BETTER UX

1. Context



Start with context: user, task, platform, constraints, business goal.

2. Observe vs. Interpret



Separate observation from interpretation so every claim stays checkable.

3. Four Axes



Tag findings to four axes: clarity, efficiency, accessibility, trust.

4. Heuristics and Severity



Use Nielsen's 10 heuristics, severity 0–4, with 3–5 independent evaluators.

5. Evidence and Fix



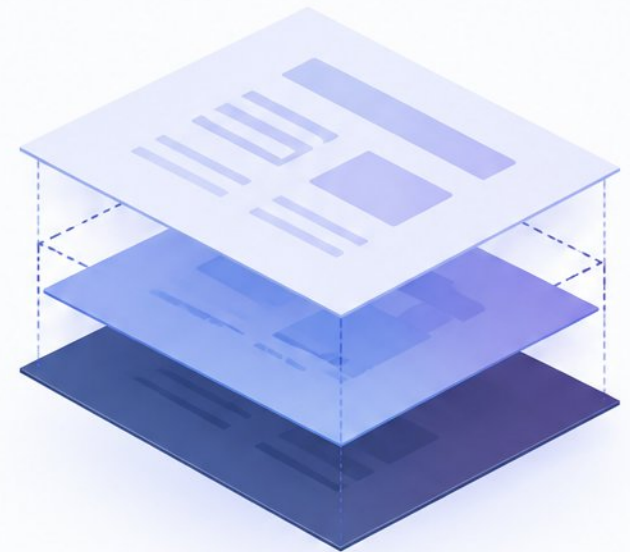
Rate evidence, not preference — pair each finding with a targeted fix.

6. Complement, Don't Replace



Heuristic review complements user testing but never replaces it.

- 1 Start with context: user, task, platform, constraints, business goal.
- 2 Separate observation from interpretation so every claim stays checkable.
- 3 Tag findings to four axes: clarity, efficiency, accessibility, trust.
- 4 Use Nielsen's 10 heuristics, severity 0–4, with 3–5 independent evaluators.
- 5 Rate evidence, not preference — pair each finding with a targeted fix.
- 6 Heuristic review complements user testing but never replaces it.



Navigation and Information Architecture Patterns



Core patterns

- Core patterns: top bar, sidebar, bottom tabs, breadcrumbs, mega menu, command palette, fat footer



Rule of thumb

- Rule of thumb: 4–6 desktop categories, 3–5 mobile destinations, max two levels deep



Strengths

- Strengths: predictable orientation, lower search cost, clear scope, spatial permanence



Pitfalls

- Pitfalls: 7+ top items, primary destinations buried in hamburger, vague labels, no active state



Labels

- Labels come from search queries, support tickets, and analytics — never internal team names

Forms, Input, and Error Prevention

Help people provide the right information, with less effort and fewer errors.

1 Full name
Your full name

2 Email address
you@example.com

3 Shipping address
Street address

4 Payment method
Card number

Continue

Validation timing

- On blur
for format
- On submit
for required
- On change
only after error
- Never
placeholder-as-label



Patterns

- Patterns: single-column, labels above placeholders, smart defaults, autofill, review-before-submit



Validation timing

- Validation timing: on blur for format, on submit for required, on change only after error



Strengths

- Strengths: less friction, fewer errors, input preserved, context fresh while fixing



Pitfalls

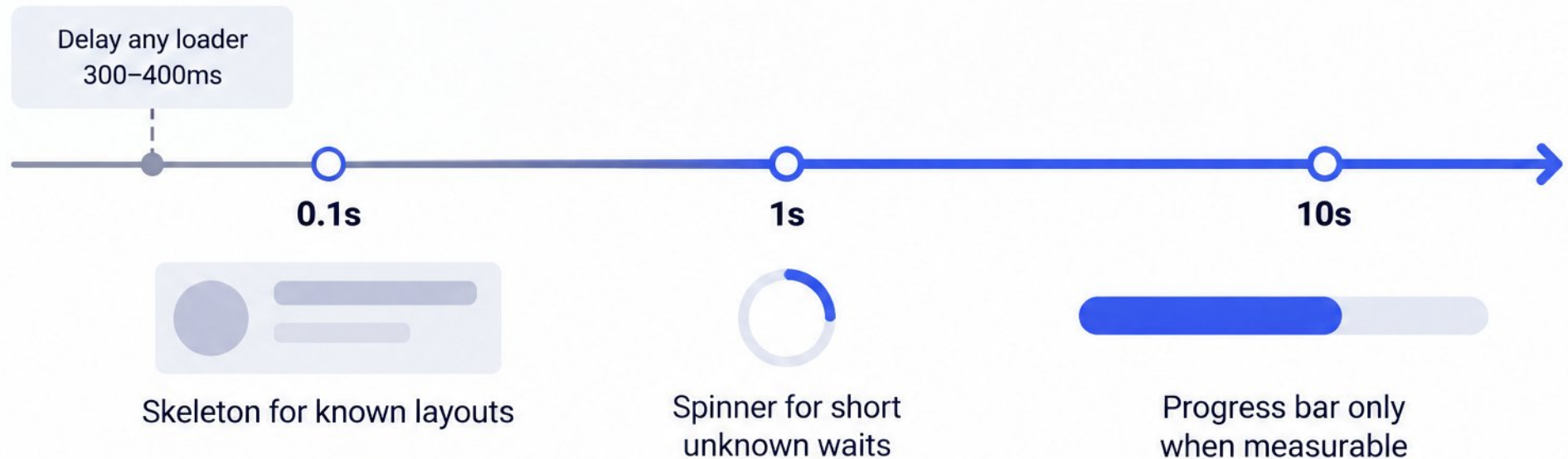
- Pitfalls: premature scolding, “Invalid input”, disabled submit, wiped values, placeholder-as-label



Access

- Access: real labels, aria-invalid, polite live regions, focus first error, never color alone

Feedback, Status, and System Visibility



Under 0.1s feels instant;
past 10s attention has left the task



Skeleton for known layouts,
spinner for short unknown waits,
progress bar only when measurable



Delay any loader 300–400ms;
reserve final dimensions to avoid
layout shift



Tier feedback by attention:
inline, ambient, transient, blocking



Pitfalls: endless spinner,
fake bar stalling at 90%,
toast-only failures



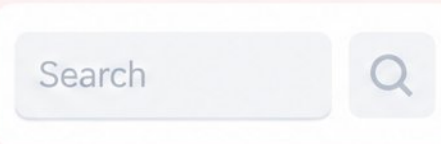
Every pending path needs
a matching failure path with retry

Accessibility as a Quality Lens

- Treat **accessibility** as infrastructure, not a post-launch audit
- Generalizes: contrast, visible focus, semantics, keyboard operability, target size
- WCAG 2.2 AA adds six audit criteria — focus obscured, dragging, target size, help, redundant entry, auth
- Common failures: low-contrast text (83.9%), missing labels, paste blocked in password fields
- Pitfalls: color-only meaning, removed outlines, focus lost in dialogs, ARIA duplicating errors
- Automation catches roughly 30–57% — keyboard, screen reader, and reduced-motion passes are manual
- Mini-example: review one component for focus order, target size, and announcement



Pitfalls



- Low-contrast text
- Removed focus outline
- Undersized icon button



Pattern



- Visible focus ring
- 24px-plus targets
- Announced status changes



Platform Conventions and Cross-Platform Critique



Core patterns: bottom tabs, drawers, pull-to-refresh, swipe actions, bottom sheets



Touch targets: 44×44 pt on iOS, 48×48 dp on Android; primary actions bottom third



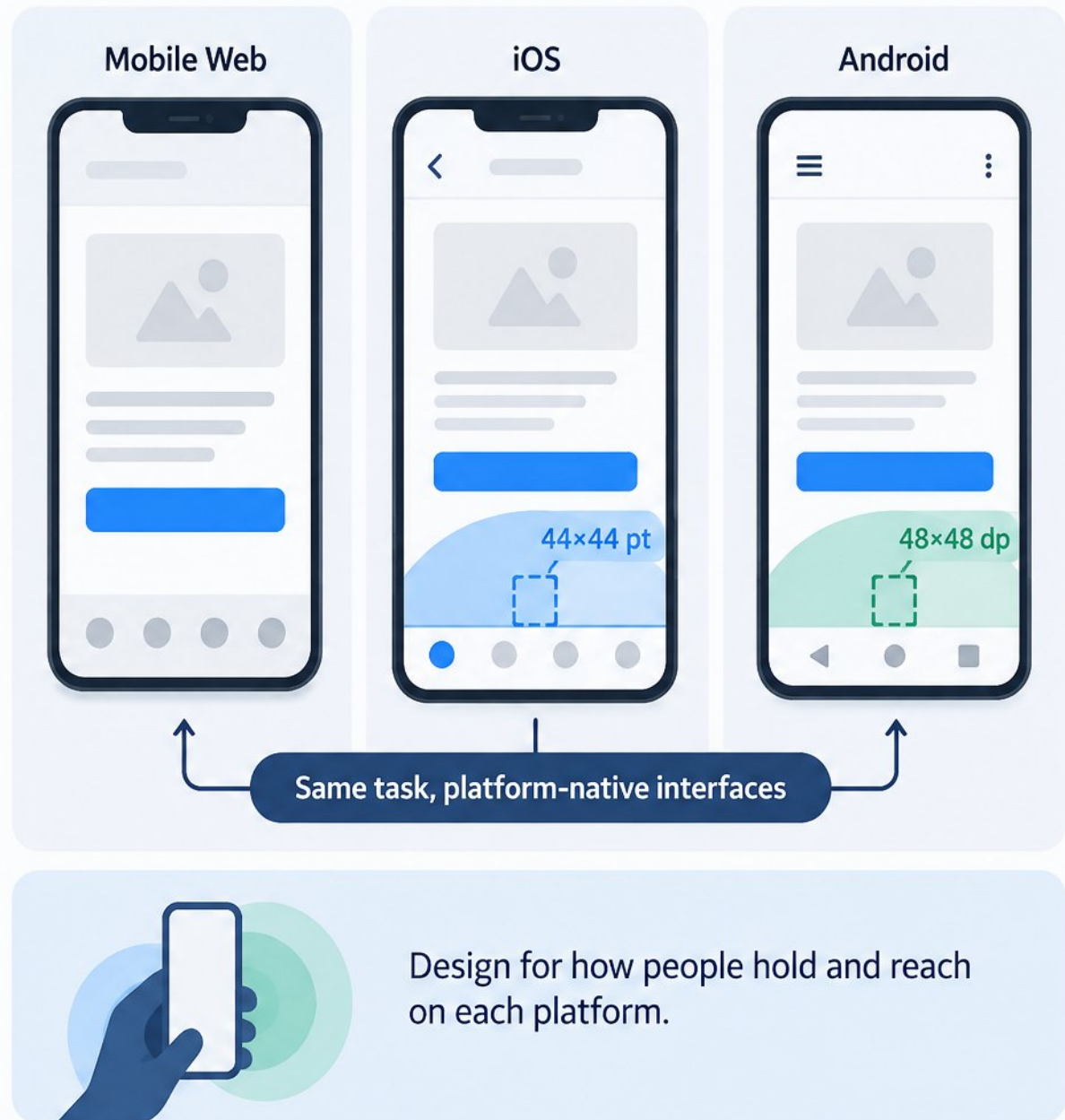
iOS expects swipe-back and tab bars; Android expects Material and system back



Shipping one platform's UI on the other reads as broken, not consistent

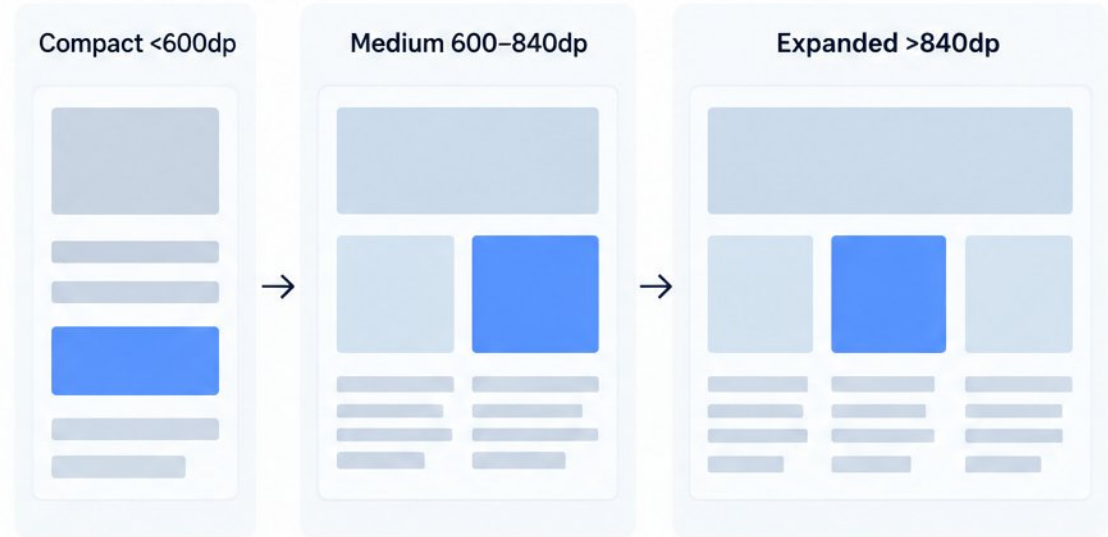


Be consistent in behavior, not layout; design mobile nav as 3–5 destinations



Responsive and Adaptive Layout Decisions

- Responsive flexes one layout with width; adaptive swaps distinct layouts at breakpoints
- Mix both: content reflows while structural layouts switch
- Use 3–5 semantic breakpoints; pick mobile-first min-width, never mix directions
- Window size classes: compact <600dp, medium 600–840dp, expanded >840dp
- Component responsiveness lives in container queries, not hardcoded pixel widths
- Pitfalls: too many breakpoints, letterboxed phone layouts, lost state on unfold



Component responsiveness with container queries



Container query boundary

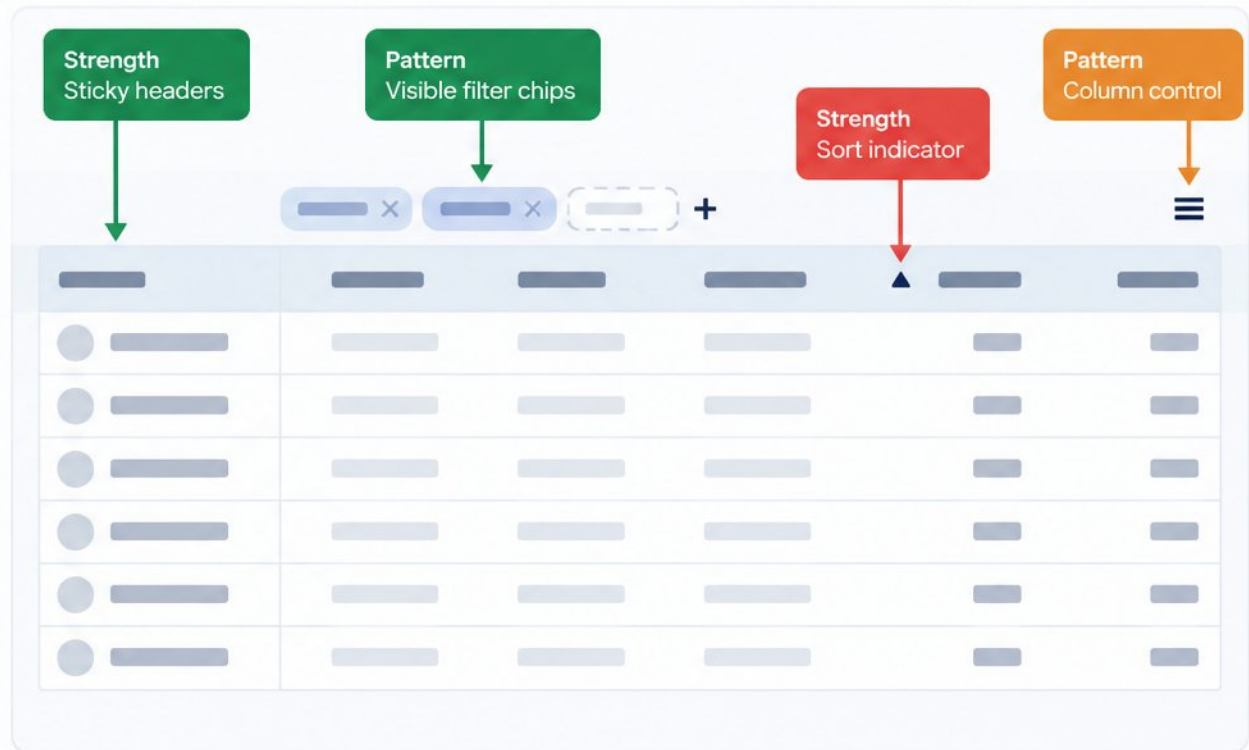
Layered states (example)



Data-Dense Dashboards and Tables

UI DESIGN TRAINING

- 1 Lead with the identity column; keep 3–7 default columns; right-align numerics
- 2 Density is a task decision: 32–36px rows to scan, 44–56px to inspect
- 3 Sticky headers, column control, and remembered preferences scale one table
- 4 Sorting needs a persistent indicator; filters need visible chips and one-click clear
- 5 Pitfalls: every field as a column, hover-only row actions, pagination with no total
- 6 Design true empty, no-results, loading, and error states alongside the table



Design these states

Empty state



No-results state



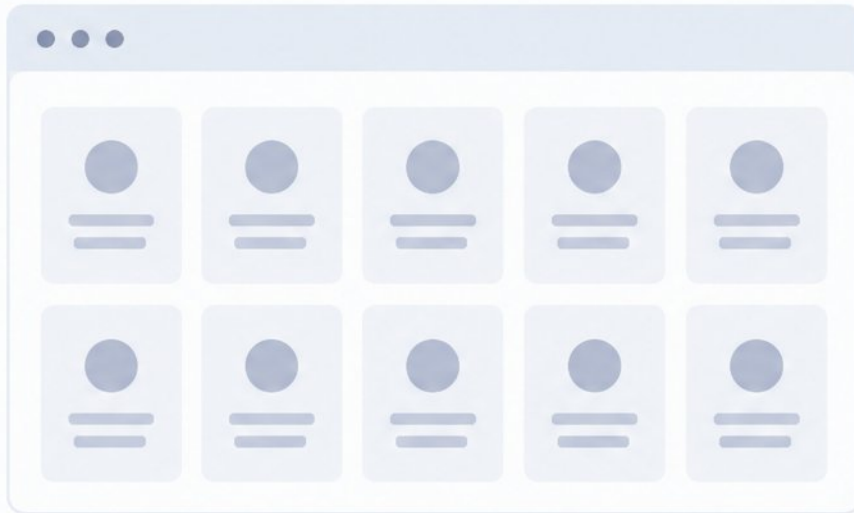
Loading state



Error state

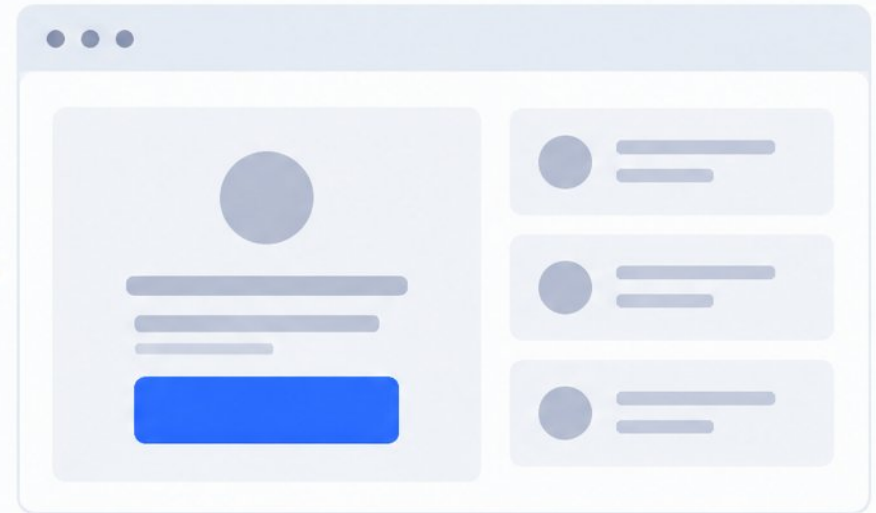


Visual Hierarchy, Density, and Prioritization



Ten equal-weight metrics compete

Everything looks important.



One primary action, clear priority

Main action stands out; secondary actions recede.



Calm density is the 2026 direction:
more information, quieter treatment



One primary action per screen;
secondary actions visually subordinate



Hick's and Miller's Laws set ceilings:
nav ≤ 7 items, forms ~ 5 sections



Contrast and visual weight carry
priority: strongest weight for the main CTA



Pitfalls: gradients instead of density,
ten equal-weight metrics, tiny type
faking density



Test every element: if removed, what
user understanding is lost?

AI-Touched and Generative Interfaces



Transparency baseline: show why a suggestion appeared, its confidence, and how to override



Inline AI in the editor, cell, or canvas has displaced the side chat bubble



Keep predictable envelopes: navigation, headers, and canonical actions stay fixed



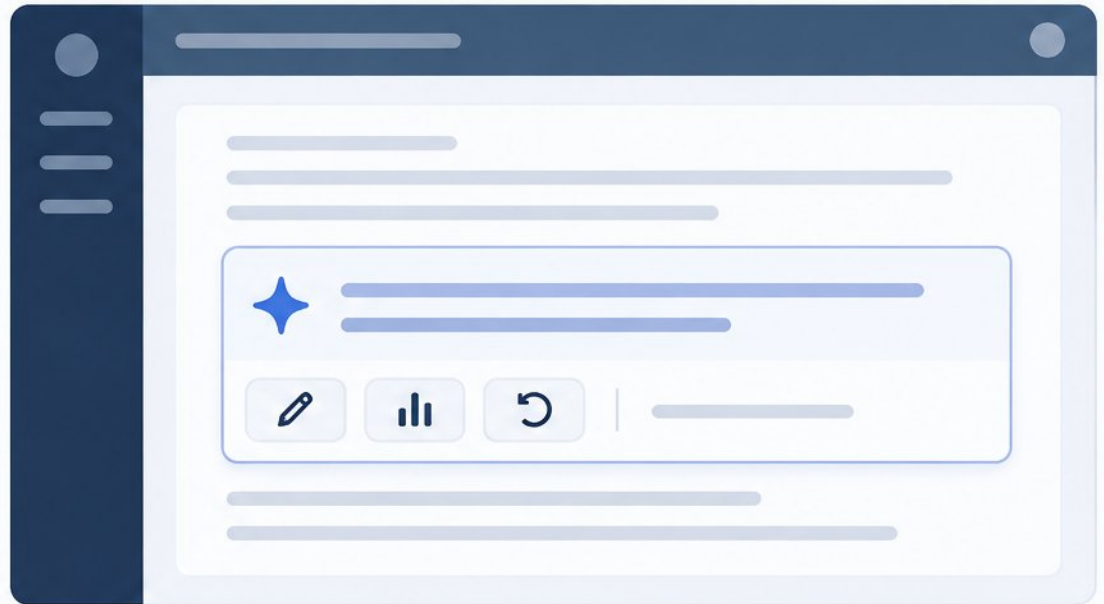
Every generated component needs an edit affordance, deterministic fallback, and error states



Pitfalls: inconsistent sessions, unpredictable DOM breaking screen readers, AI code bypassing the component library



Critique question: does the affordance clarify the user's task or serve the platform?



Pitfall Clinic: Dark Patterns and Recurring Anti-Patterns

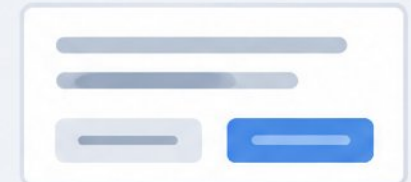
- Dark patterns deceive on purpose; anti-patterns are well-intentioned mistakes
- Deceptive set: obstruction, sneaking, interface interference, forced action, social engineering
- Confirmshaming, preselection, hidden costs, and trick wording still work on most users
- Education and cool-downs rarely counter them — regulation is the real fix
- Accidental anti-patterns: hamburger overuse, carousel blindness, icon-only buttons, modal overload
- Ask: would users still act this way if fully informed?

Common Patterns in the Wild

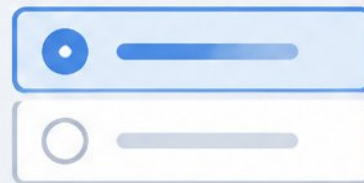
Cancellation obstacle



Confirmshaming



Preselection



Fake urgency

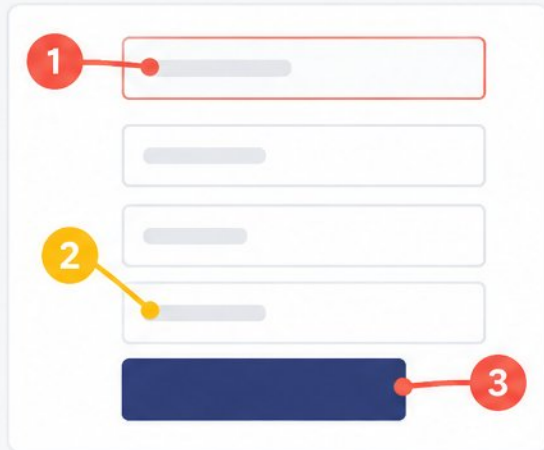


Ask: would users still act this way if fully informed?

Critique Practice: Diagnosing Real Examples

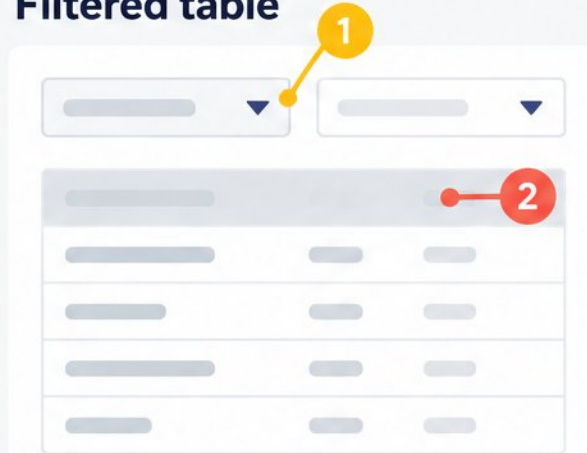
Look closely. Find the issues. Explain the impact. Propose a fix.

Checkout



A diagram of a checkout form with three input fields and a dark blue submit button. Annotation 1 (red circle) points to the first input field. Annotation 2 (yellow circle) points to the second input field. Annotation 3 (red circle) points to the submit button.

Filtered table



A diagram of a table with a header row and several data rows. Annotation 1 (yellow circle) points to a dropdown filter in the header. Annotation 2 (red circle) points to a row in the table.

Consent flow



A diagram of a consent flow with two input fields and a dark blue submit button. Annotation 1 (red circle) points to the first input field. Annotation 2 (yellow circle) points to the submit button.

-  **Checkout:** validate-on-submit, missing labels, blocked paste (WCAG 3.3.1, 1.3.1, 3.3.7)
-  **Filtered table:** missing sort indicator, filters reset on back-navigation
-  **Consent flow:** separate deliberate dark patterns from accidental anti-patterns
-  **Four-part write-up:** context, pattern, strength, pitfall, targeted fix
-  **Rank by impact:** blocks conversion > adds friction > reduces trust > minor polish
-  **Review logistics:** 3–5 evaluators, 1–2 hours each, independent, then consolidate

Building a Critique Portfolio and Review Habit

- Build a reusable critique card: context, pattern, strength, pitfall, evidence, fix
- Rotate device, platform, and data volume so examples stay varied
- Timebox reviews, require evidence, and end with action items
- Re-run the checklist after fixes; revisit across releases



Critique Card Template

Context

Pattern

Strength

Pitfall

Evidence

Fix



Mobile Example

Context

Pattern

Strength

Pitfall

Evidence

Fix



Desktop Example

Context

Pattern

Strength

Pitfall

Evidence

Fix



Tablet Example

Context

Pattern

Strength

Pitfall

Evidence

Fix



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/166b0424/public