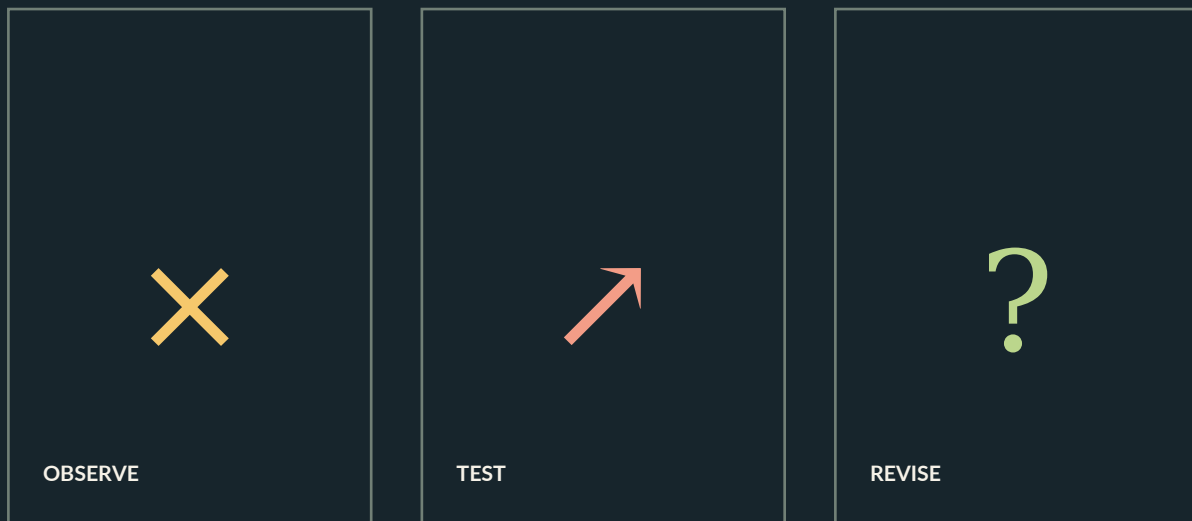


THE GLITCH CATALOG

Beautiful mistakes. Honest revisions.



SIX EXHIBITS / TWO WORKSHEETS / ONE PRACTICAL ETHIC

Field notes for people who publish, repair and keep learning.

01 / INTRODUCTION

A museum with visible seams

A first version is a hypothesis about how the world will respond. A second version is evidence that someone was willing to look again. The Glitch Museum makes that difference legible: not by celebrating failure, but by describing the move from an unsupported assumption toward a testable revision.

Six cases in the museum are illustrative constructions. They are not authenticated incidents from Sahil Tayal's projects, not a promise that a repair will work everywhere, and not data about real users. The benefit lies in practising an honest method.

The four-sentence discipline

1 / OBSERVE

State what happened under what conditions. Avoid moral conclusions about the creator.

2 / HYPOTHESE

Name a possible mechanism. Mark it as a hypothesis until there is evidence.

3 / TEST

Specify the smallest experiment that could challenge the explanation.

4 / REVISE

Change the minimum needed, retest the original failure, then check adjacent tasks.

Three questions to take into any review

What failed? A behaviour or outcome — not a person's identity.

How do we know? A screenshot, reproduction, log, reader observation, or an honest admission that evidence is incomplete.

Who benefits from the fix? Name the reader, visitor or future maintainer and the reduced cost to them.

02 / GALLERY LABELS, PART ONE

From fragility to a usable record

Exhibit 01 / Broken Deploy

Illustrative flaw. A beautiful page depends entirely on a script loading. The path is wrong, leaving a blank shell. Before: an empty root node waits for code. After: the heading, primary navigation and main explanation already exist as HTML; optional scripts add richer interactions.

Unexpected lesson

A build can complete successfully while first paint is unusable. Test the deployed folder, not merely the source project.

Exhibit 02 / Too Much Motion

Illustrative flaw. Cards drift, rotate and shimmer indefinitely. The spectacle has no explanatory job and cannot be paused. Before: one looping animation applies to every card. After: movement reveals a transition and respects a Pause control and the operating system's reduced-motion preference.

Unexpected lesson

A stop control is part of authorship. The visitor owns their attention.

Exhibit 03 / The Unread Essay

Illustrative flaw. A grand headline opens into an undifferentiated wall of writing. The reader cannot tell the question, the evidence or what to read next. Before: a vague heading and long body. After: one clear question, summary, navigable headings and reading paths.

Unexpected lesson

Editorial clarity is not simplification at the expense of depth. It gives the depth a door.

Gallery labels are models, not claims about specific live pages. To turn one into a documented case, preserve the original artifact, reproduce the symptom and identify author-approved context.

03 / GALLERY LABELS, PART TWO

Four, five, six: missing contracts

Exhibit 04 / It Worked on My Laptop

Illustrative flaw. A desktop hover reveals the answer, but on touch or with keyboard navigation the result is unreachable. Before: visibility depends on `:hover`. After: focus, click/tap activation and an explicit open state all expose the same content.

Unexpected lesson

A good desktop demo can conceal an untested input assumption.

Exhibit 05 / The Dead Button

Illustrative flaw. “Make magic” looks tappable but has no event, destination or state change. Before: the call to action is a promise without a consequence. After: a descriptive button announces what it will reveal, produces an actual local result and handles unavailable states.

Unexpected lesson

The label, action and observable outcome should tell one coherent story.

Exhibit 06 / The Missing Source

Illustrative flaw. A claim is stated with absolute confidence but no test or source trail. Before: “This method always works.” After: a scoped observation, reproducible steps, uncertainty and a link to methods and limitations.

Unexpected lesson

Provenance is a form of hospitality toward the sceptical future reader.

A curator’s caution

Publishing a failure story can be helpful without publishing sensitive data, confidential code or another person’s identifiable mistakes. Remove secrets, obtain permission for names and third-party artifacts, and prefer mechanism over blame. A generous archive shows what changed without inventing a heroic timeline.

04 / PRACTICAL FRAMEWORK

Anatomy of a respectful postmortem

A good postmortem treats the artifact seriously and the people humanely. It is specific enough to reproduce, modest about causes, and complete enough to help a stranger make a better choice.

01 / Context	What was the intended task, audience, device and environment?
02 / Observation	What happened? What did you expect? Include the exact failure condition.
03 / Evidence	Screenshots, build revision, steps, console note or approved artifact.
04 / Hypotheses	Which cause is plausible? Which alternative explanation still fits?
05 / Intervention	What changed, and why should that change affect the symptom?
06 / Retest	Does the original test now pass? Which neighbouring paths also need checking?
07 / Open questions	What remains unknown, untested or intentionally out of scope?

Exercise A / Write the smaller, testable claim

Vague verdict: "My website is broken." Better observation: "At 360px, the menu overlay blocks the primary link; keyboard activation cannot reach the next section."

Now write yours. Keep it factual; do not guess at causes yet.

Observed result: _____

Conditions / steps: _____

Expected result: _____

One hypothesis worth testing: _____

05 / REPRODUCIBILITY

Give the next person a route back

A reliable bug story should survive the disappearance of the author's memory. Someone else needs enough context to repeat or reject the result. Preserve a simple path before preserving a dramatic screenshot.

Minimum reproducible path

- [] Record the exact page or file and the tested version, not just the product name.
- [] State operating system, browser family, viewport and input method.
- [] List the steps from a clean load (including login assumptions, if any).
- [] Separate expected outcome from observed outcome.
- [] Record frequency: every time, intermittently, or only once.
- [] Attach only safe artifacts; remove personal data, keys and tokens.
- [] Record what was NOT tested. Incomplete coverage is normal when disclosed.

Worked example / navigation hidden by a layout edge

Observation

At a narrow viewport, opening the menu overlays the primary action. Focus can move behind the overlay, but the action is not visible.

Hypothesis

The menu was positioned relative to a desktop-sized header. This is a guess, not the established cause.

Test

Reload at 360px; open and close the menu; tab through all reachable controls; repeat with touch and disabled animation.

Revision. Place the menu in document flow at narrow widths; ensure escape and close states are visible. Regression check. Test headings, links, 768px and 1440px layouts, text zoom and keyboard reading order. A passing 360px test does not prove all devices pass.

06 / FOUR FUNCTIONAL REPAIRS

Before / after is a behaviour, not a filter

Each pair below is a teaching example. A visual slider is useful only when the revised state changes what the visitor can actually do, understand or verify.

BUTTON OUTCOME	Before: styled control, no result. After: labelled action changes a live status message. Test: activate with mouse, Enter and Space.
MOTION CHOICE	Before: indefinite movement with no alternative. After: a pause path and reduced-motion behaviour. Test: activate the pause, then refresh with motion reduced.
EVIDENCE TRAIL	Before: an absolute claim without a method. After: scoped observation, method and explicit limits. Test: ask an unfamiliar reader to reproduce the evidence path.
DISCOVERABLE TASK	Before: ambiguous icon and no destination. After: labelled control with three steps. Test: complete the task without hover or visual guessing.

The real improvement contract

Before: Write what a visitor can observe, not only what a screenshot looks like.

Change: Name the rule, state, accessible name or content structure you changed.

After: Perform the action and record the result with the same conditions.

Regression: Verify the change did not remove a related path, such as keyboard support, printability or no-JS readability.

A nice after-image can still be a fake fix if the action remains inert. In the online Conservation Lab, each slider alters the local demonstration rule used by the Test button; the written comparison is available without needing to see animation.

07 / DEPLOYMENT PREFLIGHT

Publish for the imperfect internet

The aim is resilient delivery, not a declaration that all environments are covered. Run these tests against the exact archive that will be deployed. Write down untested cases rather than substituting confidence for evidence.

- [] Unzip into a clean empty folder: is index.html at the root?
- [] Open index.html with JavaScript disabled: can a stranger read the main purpose and core content?
- [] Confirm that every relative CSS, JS, SVG, PDF and favicon path resolves from the root.
- [] Click every CTA, nav anchor and footer destination. Does each action have an outcome?
- [] Run at 360px, 768px and 1440px; check horizontal overflow and visual clipping.
- [] Try keyboard-only completion from the skip link through forms and reset actions.
- [] Test prefers-reduced-motion and Focus mode; no important information should disappear.
- [] Check loading error paths and user-visible states rather than relying only on console silence.
- [] Check media rights, alt text, metadata, canonical address and correct copyright identity.
- [] Check robots.txt, sitemap.xml, 404 page and relevant static-host headers.
- [] Open the real PDF, select text and inspect page breaks and links.
- [] Record the tested browser, version and any remaining limitations.

A failure worth preventing

An absolute asset path can work on one host and fail in another. Relative runtime paths make a static ZIP more portable. However, portability is not a substitute for testing actual deployment and refresh behaviour. A green local test is evidence about a local test.

08 / INCLUSION REGRESSION SHEET

The fixes must include people

An accessible interface is more than a contrast checker. It helps different visitors achieve the same purpose through distinct input methods. Review the completed task, not just individual components.

Input and structure

- [] One clear H1 and a logical heading hierarchy.
- [] Keyboard can reach, activate, exit and reset every meaningful interaction.
- [] Focus is visible and order follows the reading task.
- [] Buttons have honest names; status changes are announced where useful.
- [] Drag or scrub interactions provide native range input or equivalent buttons.

Motion, perception and alternatives

- [] Reduced motion removes unnecessary transitions, never the explanation.
- [] Any persistent optional movement has a visible Pause control.
- [] Text alternatives describe the function of art when the art conveys meaning.
- [] High contrast and 200% zoom preserve comprehension and usable controls.
- [] Do not require sound, hover, colour alone or precise pointer movement.

Devices and resilience

- [] Complete the key flow at 360px, 768px and 1440px.
- [] Check touch target size, field labels and validation error recovery.
- [] Test a no-JS path for essential content, even if generators require scripting.
- [] Print view contains understandable content and avoids chopped-off essential text.

Reference point: WCAG 2.2 (W3C). A checklist is a review aid, not proof of conformity; real conformance needs targeted evaluation and, when possible, testing with assistive technology users.

09 / READY-TO-USE TEMPLATES

Two sheets to copy into real work

Template B / Issue report

Title: A symptom and condition, not a judgment

Artifact: Exact URL / file / revision

Environment: Browser / viewport / input / accessibility setting

Steps: 1 → 2 → 3, beginning at a clean load

Expected: What the visitor should have been able to do

Actual: What happened; frequency and safe evidence

Impact: Which visitor task was blocked or degraded

Unknown: What you have not tested or reproduced

Template C / Small postmortem

Observation (not identity): _____

Hypothesis + alternative: _____

Smallest test and result: _____

Change + reason: _____

Regression checks: _____

Principle for the next person: _____

Do not write “fixed everywhere” unless the evidence warrants it. A truthful closeout might be: “The original issue no longer reproduces in the listed tests; other devices remain unverified.”

10 / CLOSING NOTES

A century of useful corrections

An archive of only finished objects may look impressive but can hide the craft of getting there. A documented correction allows a future reader to recover the original assumptions, inspect the new evidence, and understand why a design changed. That is often more useful than a flawless origin myth.

Being willing to revise does not mean glorifying errors, creating problems for drama or treating creativity as a competition in visible output. Mistakes can cost real time and trust. Repair is a responsibility to the person affected. The humane record names the mechanism, preserves privacy and declines to turn a flawed draft into a judgment about a human being.

Limitations of this catalog

The museum cases are hypothetical teaching instruments; the website's local interactions are not validated behavioural assessments, empirical tests of users or verified release histories. The checklists do not establish formal accessibility conformance, universal performance or long-term archival preservation. No backups, hosting guarantees or automated monitoring are supplied by this PDF.

Annotated routes for further exploration

W3C / WCAG 2.2: [w3.org/TR/WCAG22](https://www.w3.org/TR/WCAG22) — normative accessibility success criteria; distinguish requirements from aspirational tips.

MDN Web Docs / progressive enhancement: [MDN glossary](#) — begin with resilient content and layer on optional capabilities.

web.dev / Core Web Vitals: web.dev/articles/vitals — performance metrics and practical measurement, not substitutes for task testing.

GitHub Docs / issue templates: [Issue template guidance](#) — ways to standardise reproducible reports in team projects.

The museum's final label

"We observed this. We suspect that. We tested it. Here is what changed. Here is what we still do not know."

© 2026 Sahil Tayal / Sahilverse · The Glitch Catalog · Original educational material. Copy exercises for your own learning or project documentation; credit appreciated.