


Challenger settings


Challenge intensity 

- ☐ mild
- ☒ spicy

Agent Selector

Tip: You can also explicitly type an agent tag in your message, e.g., `@novelty_radar` or `@bug_logger`.

Preferred Agent (optional)

Auto (Smart Intent D... 

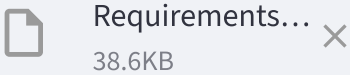
Upload Document

Requirements / Logs / Specification
(txt / pdf)

Drag and drop file here

Limit 200MB per file • TXT, PDF

Browse files



Preview of Uploaded

> Document (first 1000 characters)

Defying Gravity in Testing



@AssumptionBuster To test state leaks, I’m keeping both panels (admin and normal user) visible and do reserve cancel then flip filters and sort. Identify what could cause false alarms or false confidence in my observations, then rewrite my test approach with corrections to reduce both.



- Logic and assumptions
 - The approach assumes keeping admin and normal user panels visible will reveal cross-role state leaks.
 - The approach assumes reserve/cancel plus filter/sort actions synchronously propagate to both panels and the in-memory model.
- Adversarial critique — assumption weaknesses
 - The approach treats panels as independent contexts, but requirements tie admin visibility and active User ID together (FR-33, FR-4).
 - Asynchronous UI updates, event propagation gaps, admin-specific ordering, and filter persistence (FR-19/20/38, FR-12) can produce transient differences that mimic leaks.
- Persona-based challenge
 - As an attacker, could race conditions or delayed event handling make admin-only changes momentarily appear in a user view?
 - As a regular user, could filter or sort state be local to one panel, causing false confidence about synchronized state?
- Corrected logical framing (non-prescriptive)
 - Reframe approach: treat each visible panel as reflecting the single active global context, not separate users.
 - Expect asynchronous propagation, UI rendering delays, and view-specific ordering or filter state as normal behaviours.
- Reflective challenge

- How will the approach distinguish genuine cross-context data corruption from transient UI lag or view-scoped state?
- Which observable invariants would falsify the assumption that both panels represent independent, simultaneously active identities?

Type here... (e.g., '@Brainstormer Find unseen evaluation paths in the unlabeled GPC')>