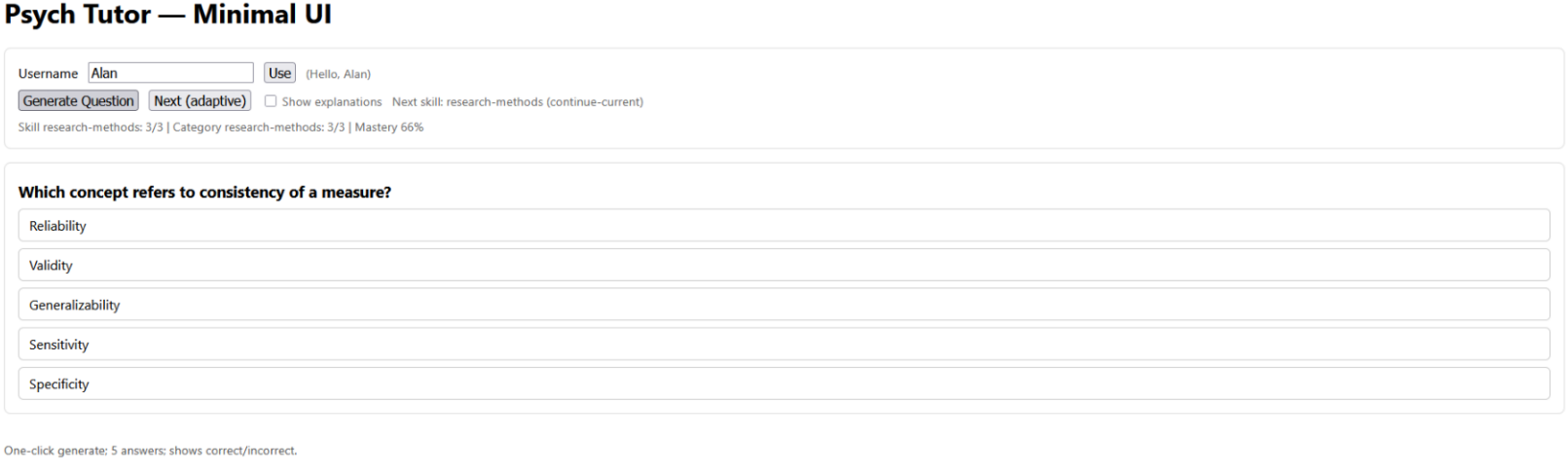
Psych Tutor — Validity & Usefulness Roadmap

Actionable improvements to make the minimal UI psych tutor more psychometrically sound and effective.

Project: Real-Time LLM-Powered Psychology Tutor (Minimal UI)

Date: 2025-08-31
Generated by ChatGPT

Current Minimal UI (reference screenshot)



Executive Summary

This roadmap focuses on two outcomes: (1) increase **validity** (item quality, answer-key correctness, construct coverage), and (2) increase **usefulness** (learning impact, accessibility, ease of use). It keeps your constraints intact: a fixed OpenAI model with strict JSON responses, a minimal web UI, and lightweight adaptivity with mastery decay and misconception counters.

Deliverables are grouped as Quick Wins (1–3 days), Validity/Reliability (1–2 weeks), Usefulness & Learning Impact, Engineering Hooks, and Success Metrics. Where relevant, the recommendations are grounded in your README and Agent Runbook.

A. Quick Wins (1–3 days)

1) Turn on explanations by default (toggle remains): Generate items with rich rationales and misconception tags; show them after answer submission.

2) Add a 1–5 confidence check: Capture self-rated confidence per item and compute a calibration metric (e.g., Brier score).

3) Keyboard + accessibility polish: number-key selection, Enter to submit; proper ARIA roles and an aria-live region for results.

4) Item quality guardrails (rule-based): single best answer, no duplicates, avoid “All/None of the above”, balanced option length, non-trivial stems.

5) “Report a problem” button: users can flag ambiguous/incorrect items; log item hash and prompt seed for audit.

B. Strengthen Validity & Reliability (1–2 weeks)

6) Template-driven item generation per skill: craft 2–3 templates per skill with distractor patterns keyed to common misconceptions; validator enforces fill quality.

7) Two-stage generator → validator (same model): stage 1 creates MCQ + rationale + tags; stage 2 independently answers and verifies key, uniqueness, and plausibility; regenerate once on failure to improve answer-key accuracy.

8) Seed an item fingerprint: include item_id (hash), prompt_id, and seed; support reproducibility, audit, and de-duplication.

9) Basic item analysis: track difficulty (p-value), point-biserial discrimination, and flag rate; retire low-discrimination or high-flag items after ≥100 responses.

10) Construct coverage checks: enforce rotation across subtopics inside a skill before advancing mastery.

11) Misconception-targeted remediation: use per-learner misconception counters to select the next contrasting template (e.g., validity vs reliability).

C. Improve Usefulness & Learning Impact

12) Short, source-linked feedback: attach a one-line “Learn more” to a cached OER snippet for each skill.

13) Spaced review you can feel: surface a “Due for review” banner and a small daily goal; keep the existing 7-day half-life policy visible.

14) Study modes: Practice (immediate feedback) and Check-up (feedback after a 10-item block with a mini-report of strengths and misconceptions).

15) Tiny teacher/admin dashboard: show most-missed skills, most-flagged items, and mastery trends to guide template curation.

D. Engineering Hooks (Where to change things)

Server / item flow: add validate_mcq() after generation to run an LLM self-check and local rule checks; only return items that pass.

Storage: persist item_id, difficulty, discrimination, confidence, flag counts; roll into existing per_skill/per_category aggregates.

API: extend /api/record with item_id, confidence, time_to_answer_ms, and misconception_tag_if_wrong; keep payloads minimal.

UI: implement keyboard controls, report-problem button, explanations toggle default-on, and the confidence widget without increasing page weight.

E. How We’ll Know It’s Better (Success Metrics)

• Item correctness (content validity): ≥ 98% keys correct on a 100-item random audit; < 1% duplicate/ambiguous by rules.

• Discrimination: median point-biserial ≥ .20 for items with ≥ 150 responses.

• Learning: +10–15 percentage-point gain from first-seen to first-review on the same skill (spaced effect).

• Confidence calibration: lower Brier score month-over-month.

• Usefulness: ≥ 75% thumbs-up on explanations; flag rate < 1% of delivered items.

F. Grounding, Constraints, and Operational Notes

Grounded Constraints

• Fixed OpenAI model with strict JSON outputs and minimal UI design.

• Lightweight adaptivity with mastery (0..1), 7-day half-life decay, and misconception counters.

Operational Notes

• Keep the JSON contract strict; include item_id, prompt_id, and seed for auditability.

• Enforce content guardrails before display; log flags for review.

• Use empirical item analysis to curate the live item pool over time.

Appendix: Stricter Question JSON (back-compatible)


```
{
  "type": "mcq",
  "skill_id": "cog-research-methods",
  "question": {
    "item_id": "sha256:...",
    "stem": "Which concept refers to consistency of a measure?",
    "options": ["Reliability", "Validity", "Generalizability", "Sensitivity", "Specificity"],
    "correct_index": 0,
    "rationales": [
      "Correct: reliability is about consistency across time/items/raters.",
      "Validity concerns accuracy, not consistency.",
      "Generalizability is about external applicability.",
      "Sensitivity is detection of signal/change.",
      "Specificity is correct rejection."
    ],
    "misconception_tags": ["validity_vs_reliability", "validity", "external_validity", "sensitivity", "specificity"],
    "template_id": "rm-def-contrast-01",
    "prompt_id": "mcq-minimal-v3",
    "seed": 4219
  },
  "validation": {
    "rule_checks": {"single_key": true, "no_duplicates": true, "no_all_none": true},
    "model_self_answered": true
  }
}
```