

READ THIS FIRST (MANDATORY)

You are no longer testing code in isolation.

You are now testing the product as a living system.

This task is about:

- Discovering where Gurukul breaks under real usage
- Identifying demo risks before an external observer does
- Hardening behavior on free-tier infra constraints
- Eliminating anything that feels “unfinished” or “fragile”

You are NOT:

- Adding new features
- Refactoring architecture
- Changing product scope

You ARE:

- Acting as the first hostile user, investor, school admin, and auditor combined

INTEGRATION BLOCK

You will coordinate lightly with:

- Yashika — Backend / Deployments — confirm backend failure behaviors
- Rukayya — PRANA / Guardrails — validate non-invasive telemetry + safety posture
- Vinayak — QA / Validation — cross-check reproducibility of issues

No blocking dependencies. You own this task.

TIMELINE

Execution Effort: ~12–15 focused hours

Target Completion: 1.5 calendar days
Daily EOD WhatsApp summary required.

DAY-BY-DAY BREAKDOWN

DAY 1 — Product Stress & Breakpoint Discovery

Scope:

Test Gurukul exactly as an external user would.

You must test:

- Student flows (login → subject → lesson → chat → quiz → reflection)
- Teacher dashboards
- Admin dashboards
- EMS integration points
- Multilingual flows (at least 2 languages)
- TTS playback under slow network
- Backend cold starts (Render wake-ups)
- Vercel frontend behavior under refresh, route jumps, idle time

Explicitly test:

- Free-tier timeouts
- Inactive buttons
- Broken routes
- Silent failures
- Loading states that never resolve
- Auth edge cases (expired token, refresh, logout/login loops)

Deliverables:

- breakpoints_report.md
 - dead_links_and_ui_risks.md
 - infra_constraint_findings.md
-

DAY .5 — Demo Hardening & Guardrail Alignment

Scope:

Turn findings into demo-safe behavior, not feature work.

You must:

- Identify what must be hidden, disabled, or frozen for demo
- Propose safe fallback behaviors (do nothing gracefully)
- Align with existing demo_safety_profile.json and guardrails
- Validate that no crash or broken state is visible in demo paths

Also include:

- What works reliably even under stress
- What must NOT be shown in any public or investor demo

Deliverables:

- demo_safe_paths.md
 - demo_exclusion_list.md
 - failure_behavior_matrix.md
 - final_demo_readiness_note.md
-

LEARNING KIT (MANDATORY)

Keywords:

- “Product stress testing SaaS free tier”
- “Demo hardening checklist startups”
- “UX failure modes under latency”

LLM Prompts:

- “List all ways a user can break a SaaS demo unintentionally”
 - “What signals make a product feel unfinished during a demo?”
-

DELIVERABLES (NON-NEGOTIABLE)

- Markdown reports listed above
 - Short screen recording (5–7 min):
 - Showing real breakpoints
 - Showing hardened flows
 - Clear YES / NO on: “Is Gurukul demo-safe today?”
-