ENGINEERING CASE STUDY · GURUKUL AI

Twelve Weeks, One Builder, 679 Commits

Gurukul AI is a bilingual AI tutor for India's competitive exams — RRB, SSC, banking, NDA, JEE, NEET. It went from an empty repository to a working product with real students in twelve weeks, built by one founder working with Claude. This is the honest account of how.

23	4,943	2	Free
EXAMS SUPPORTED	AUTOMATED TESTS	PLATFORMS	FOR ALPHA TESTERS, RIGHT NOW

Why It Needed Building

Twelve million students sit these exams every year. Private coaching runs Rs. 5,000–Rs. 20,000 a month — out of reach for most Tier 2 and Tier 3 families, the ones these exams exist to give a way up. The tutoring itself isn't the hard part; knowing exactly where a student is weak is. Not “weak in Maths” — weak in Number System, specifically, and strong in Percentage. Gurukul AI runs a full diagnostic paper, scores every topic on its own, and builds a plan from that map instead of a guess.

The Shape of the Build

Commits by month tell the real story better than a roadmap slide would:



July was the skeleton: the FastAPI backend, the agent architecture, a Flutter app with feature parity from day one. August is where the curve breaks — thirteen times July's pace — because that's where the standing rule this project runs on actually paid for itself: **design first, frozen, then code.** Every feature gets a written spec, every open question gets closed before anyone builds, and a change after that point is a new, explicitly-agreed revision — never something absorbed silently mid-build. September kept that pace on hardening: security audits, a bilingual push through every question the app serves, and an alpha cohort now sending real feedback back in.

Four Things That Actually Happened

Not features — incidents. The kind of story that shows whether an engineering process holds up under a real bug, not just on a slide.

SECURITY · DAY ONE, NOT AN AFTERTHOUGHT

A Spine Before a Single Feature Shipped

Every piece of text a student sends passes an input guard before it reaches an AI agent, and every reply passes an output guard before it reaches the student. Every interaction is logged into a hash chain — each entry carries the previous entry's hash, so an edit or deletion anywhere breaks the chain and shows exactly where. Built before the first exam's content was, not bolted on once something went wrong.

7 whitelisted routes · 3-layer guard stack · SHA-256 chained audit log

ROOT CAUSE · 7 SEPTEMBER

When the Audit Log Forked Itself

Two parts of the app each kept their own logger, and each worked out “what's the last entry's hash?” once, at startup, then cached the answer. When both wrote around the same moment, the second chained its entry to a hash that was no longer current — a fork, twice, four minutes apart. The fix makes every logger ask the file itself, under one lock, on every write. The two historical forks weren't quietly edited away — a tamper-evident log that gets silently “corrected” isn't tamper-evident anymore. They're recorded, by hash, as acknowledged exceptions the checker verifies around.

2 forks found → 0 possible now · acknowledged, never erased

MEASURED, CORRECTED, MEASURED AGAIN · AUG–SEP

The Free Tier's Real Size Took Three Tries

Google's free AI tier looked like roughly 500 requests a day per model — measured by testing one model at a time. Running two together suggested they shared one ~1,000-call bucket. A full day's real logs settled it for good: one model gets ~500, and two others split a second ~500 between them — a different, smaller shape than either earlier guess. What mattered in the end wasn't any of the three estimates. It was the discipline behind them: trust a full day's run over an inference from a smaller one, every time.

~500 / ~1,000 / ~500 — three measurements, one method survives

FOUND AND FIXED · 29 SEPTEMBER, THIS WEEK

One Slow Answer Was Freezing the App for Everyone

The server runs as a single process handling many students' requests at once — the design only works if nothing blocks it for long. One overlooked line meant that whenever the AI tutor took its time answering one student — measured at up to two and a half minutes on the slowest model — every other student's request froze behind it: logins, page loads, everything, on the same instance. Proven with a real reproduction before touching any code, fixed at fourteen call sites, and confirmed live against the production service after deploying.

14 call sites fixed · verified live, not assumed · 4,943 tests still green

> *"Believe the run over the arithmetic" turns out to be a rule that applies to a lot more than one quota measurement.*

Where It Stands

Gurukul AI runs today on Google Cloud Run, serving both a React web app and a sideloadable Android APK from one FastAPI backend. Every question a student sees can be answered in Hindi or English. A named human mentor — not a chatbot pretending to be one — reviews everything the AI proposes before a student ever sees it: study plans, curated videos, interventions when someone is stuck. Nothing the AI decides reaches a student unreviewed.

23 Exams

Railway, defence & entrance, banking, SSC — each with its own real syllabus, sourced from the actual notification where one exists

Bilingual Throughout

Hindi first, then English, stacked — the way the real exam paper prints it, not a toggle that swaps one for the other

A Human in the Loop

NAGA reviews every AI-generated study plan, note and intervention before a student sees it

Told Straight

The newest feature — a try-before-signup demo — is web-only, on purpose. Mobile inherits from web, not the other way round, so a web feature doesn't owe the app an immediate port.

This account is built from the project's own commit history, its internal engineering log, and a live check of the production service on the day it was written — not from memory or a summary written after the fact.

Gurukul AI · Sajana Tech