



PRODUCT CASE STUDY · EDUCATION · SELF-BUILT

The school that runs on one person's code.

A 14-module ERP and a full-parity staff mobile app for a 900-student institution — conceptualized, architected, built, deployed, and operated by a single person.

Harsh Kumar Arya · harsharya.com · 2026

THE STARTING LINE

A 27-year-old institution, running entirely on paper.

27

years of operation — a trusted neighbourhood institution

900+

students, KG-12, with a team of 50 staff

49

pages of handwritten ledgers — the entire financial record

0

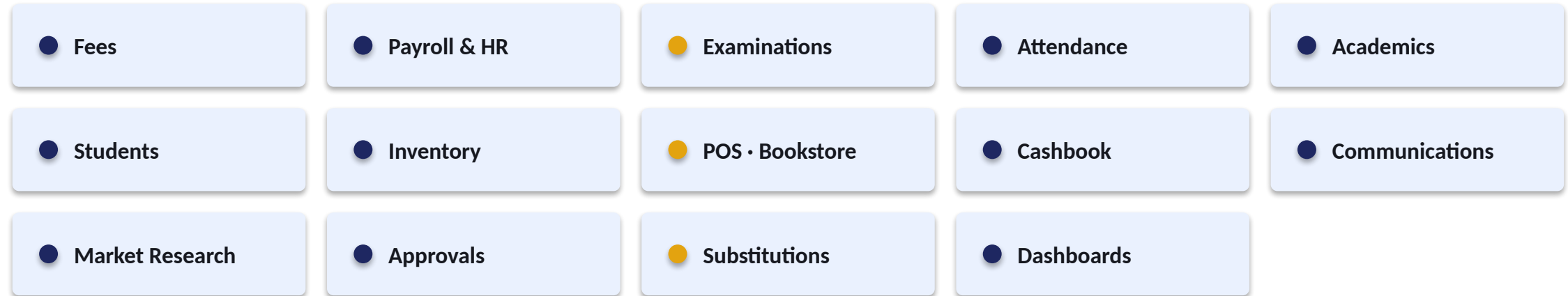
digital systems. No statements, no database, no dashboards

Fee collection in notebooks. Payroll from memory. Marks in registers. **No financial statement had ever been produced in the school's history.**

The brief I gave myself: modernise everything — without hiring anyone.

THE SYSTEM

Fourteen modules. One source of truth.



UNDER THE HOOD

Django + DRF backend · Next.js web client · self-hosted on dedicated European cloud
Role-based access on every feature · every rupee, mark, and attendance event in one database

Not an off-the-shelf ERP with the school's name on it. **Every module was designed around how this school actually works** — then the school was retrained to work digitally.

Full ERP parity, in every staff pocket — shipped in five weeks.

REACT NATIVE • EXPO

From first commit to a pilot build on staff phones in ~5 weeks.

Android-first rollout, mirroring the web client's API contracts so the server stays the single source of truth.

● Geofenced attendance

GPS check-in/out with live-selfie verification and a consent gate

● Fees on the spot

Student lookup, defaulter tap-to-call, payments with immutable receipts

● Exams & report cards

Per-student marks entry with validation; report-card generation

● Payroll & HR

Leave workflows, salary structures, payroll generate → finalize → slips

● AI bill capture

Photograph a bill → OCR → structured expense, reviewed and posted

● Role-based everything

Every screen gated by the same permission grid as the web ERP

AI here isn't a feature. It's headcount the school never hired.

It listens.

Recorded parent conversations are transcribed (Hindi + English) and scored for NPS automatically — every voice becomes structured data.

SARVAM + GEMINI PIPELINE

It reads.

A photo of a supplier bill becomes a categorised, reviewable expense entry. The inventory module runs on camera + OCR.

VISION OCR, HUMAN-REVIEWED

It writes.

Report cards generate themselves. Parent SMS/WhatsApp go out from templates. The back office types less every month.

TEMPLATES + GENERATION

Every pipeline runs inside the school's own system — built once, working every day since.

An ERP that does market research.

113

recorded parent interviews —
226% of the phase target

77

households reached across
two field waves

8

teachers trained and QA'd
as a field-research force

100%

transcribed and qualitatively
coded, Hindi + English

The research module captures, transcribes, and NPS-scores parent conversations natively — feeding a full **5C → STP → 4P strategy pipeline**: 16-attribute preference framework, 6 candidate segments, a 12-hypothesis test plan headed for conjoint analysis and perceptual mapping.

No school in the ~49-school catchment has anything like this.

From a drawer of handwritten pages to audited-grade statements.

BEFORE

49 pages of handwritten ledgers

Combined months, no categories, no verification

No financial statements — ever

Surplus, payroll, and fee leakage invisible

Paper fee ledgers per family

No reconciliation against student rolls

AFTER

4 fiscal years reconstructed (FY23–FY26)

Page-by-page transcription of every entry

First formal statements in school history

6-statement workbook • 475 formulas, zero errors

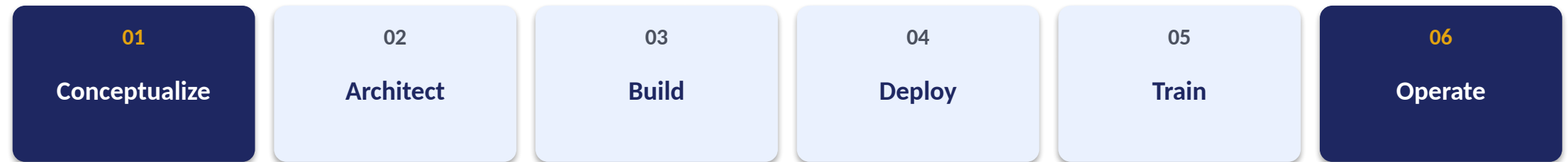
~370 ledger records reconciled to student rolls

Daily accounting now lives in the ERP cashbook

Transcribed page by page, verified formula by formula — the baseline every future decision now stands on.

THE OPERATING MODEL

No team. No vendors. No consultants.



Every stage of the lifecycle — product thinking, systems architecture, code, infrastructure, change management, and daily operations — **executed hands-on by one person, working AI-natively.**

The role AI plays: a build partner that multiplies one operator into a full delivery team — while ownership, judgment, and accountability stay human.

WHAT THIS PROVES

The constraint on institutional software is no longer engineering headcount.

Full-stack ownership

One operator carried product, engineering, data, and change management from zero to production — with real users, every day.

AI-native leverage

Modern AI collapses the cost of building serious systems. The scarce inputs are taste, domain judgment, and follow-through.

Documented, not described

Every number in this deck traces to working software, source ledgers, and field data — receipts, not claims.