

Chinmoy Paul

hello.chinmoypaul@gmail.com • +91 9903876744 • [GitHub](#) • [LinkedIn](#) • [Portfolio](#) • [Blog](#)

Final-year BS in Data Science & AI at IIT Guwahati and an AI builder. I ship production-grade AI systems by directing coding agents against written specs — decomposing systems into chunked builds, gating each on adversarial review, and turning ideas into deployed, tested artifacts in days.

EDUCATION

Indian Institute of Technology Guwahati

Sep 2023 – present · Expected 2027

B.S. in Data Science and Artificial Intelligence

Relevant Coursework: Machine Learning, Deep Learning, Artificial Intelligence, Natural Language Processing, Probability & Statistics, Optimization, Data Structures & Algorithms, Databases.

EXPERIENCE

AI Engineer Intern — Nistula (Goa)

May – Aug 2026

Nistula Assistance — WhatsApp AI host [\[repo\]](#) TypeScript, Fastify, Postgres, pg-boss, Claude Sonnet 4.5, WhatsApp Cloud API, eZee PMS

- Built, solo, a production-standard WhatsApp AI host (tagged v1.0.0) driving a seven-tool agentic Claude loop — live-priced quotes, PMS booking lookups, staff job dispatch with SLAs, per-guest memory; 1,813 tests, CI green on Node 22/24, live-verified against a Meta test line and the real eZee PMS (drained a 62-item production backlog, 0 errors).
- Enforced honesty in code rather than the prompt: seven deterministic post-generation guardrails (price and promise integrity, negotiation lock, 24h window, identity, format, leak) so the model structurally cannot state an unbacked price or promise an undelivered action; a pre-merge multi-agent adversarial review gate caught 17 blocker-class defects a green suite had passed.

Nistula Website — direct-booking villa site [\[live\]](#)

Next.js 15, React 19, Drizzle / Neon Postgres, Razorpay, eZee PMS

- Built the full-stack direct-booking site replacing a brochure page that could not take a booking: reads live availability and pricing from the eZee PMS behind a single swappable adapter (15 endpoints, one Zod schema each), takes Razorpay payment and writes the reservation back — the server re-quotes every price so a client value is never trusted, with idempotency keys on all writes.

PROJECTS

Samadhan — AI Civic Resolution Agent

Next.js, TypeScript, Genkit, Gemini 2.5 Flash, Firebase, Cloud Run

Solo build for Vibe2Ship (Coding Ninjas × Google for Developers): an autonomous agent taking a civic complaint from a single photo to verified resolution — a resolution layer, not a reporting app. [\[live\]](#) [\[repo\]](#)

- One Genkit run over Gemini 2.5 Flash (Vertex AI) perceives, reverse-geocodes, de-duplicates against nearby reports by comparing photos, routes to the correct municipal body via a rules table, and files the formal complaint in the citizen's own detected language under a tracking ID.
- Tracks the statutory SLA and autonomously drafts the escalation ladder — reminder → appeal → RTI — on breach via a 10-minute Cloud Scheduler sweep, then verifies the fix from before/after vision (advisory only; the citizen confirms). Deployed as an installable PWA on Cloud Run with green CI, 36 tests, and an Open311 export.

Acumen — Ten-Year NSE Backtester & Live Screener

Python, pandas, pyarrow/Parquet, SmartAPI, pytest

Backtester and live screener that replayed one trader's intraday NSE F&O strategy over ten years of minute data and proved, with a full audit trail, that it loses money. [\[repo\]](#)

- Replayed 188,345 trades across 204 stocks and 495,312 symbol-days from a 4.1 GB local Parquet lake, delivering the unwanted answer in full: ₹1.68 crore net, 11 of 11 calendar years negative, recounted row-by-row with 32 of 32 consistency checks agreeing so the result would be believed.
- The whole strategy is four pure deterministic functions (integer-paise prices, an exact-Fraction volume-profile POC to avoid float error); the same functions drive a no-orders Telegram screener proven byte-identical to the backtester candle-for-candle. 2,621 offline tests, 0 failures; every build chunk sealed by a separate adversarial review that could fail it (28 reports).

TECHNICAL SKILLS

Agentic AI & LLMs: agent orchestration (tool use / function calling, Genkit, Anthropic SDK), Claude Sonnet 4.5, Gemini 2.5 Flash on Vertex AI, prompt caching, deterministic guardrails, model evaluation & benchmarking, LLM deployment (Ollama, Llama 3.2), PyTorch (foundations)

AI-assisted development: spec-driven chunked builds, multi-agent adversarial review gates, Claude Code, Cursor

Frontend: React, Next.js, Vite, TypeScript, Tailwind CSS, Recharts, Framer Motion

Backend: Node.js, Fastify, Express, Spring Boot, REST APIs, PostgreSQL (Prisma, Drizzle), pg-boss, Neon, MySQL, Zod

Languages: Python, TypeScript, JavaScript, Java, SQL, C

Data & Infrastructure: pandas, pyarrow/Parquet, Docker, Kubernetes (kind, Helm), Google Cloud (Cloud Run, Scheduler), Firebase/Firestore, Vercel, Railway

Observability & Tooling: Prometheus (custom exporters), Grafana, Git, Linux, Postman, Vitest, pytest, GitHub Actions

WRITING & CODING PROFILES

Hashnode — chinmoypaul.hashnode.dev: featured “How to deploy an Open-Source LLM reliably?” (3,400 words, Apr 2026), plus posts on Kubernetes observability & production deployment.

CodeChef: chinmoypaul — 950+ problems solved. **Codeforces:** chinmoypaul — 150+ problems solved.