

Karthik Uppari

Software Engineering Intern Candidate · Backend & AI Products

upparikarthik505@gmail.com

github.com/karthik1122-code

linkedin.com/in/karthik-uppari-4005a7373

Hyderabad, India

SUMMARY

B.Tech CS (AI/ML) student who builds and tests backend and AI systems: a streaming AI workspace, an exactly-once payment-recovery service, an evaluated RAG pipeline and a policy gate for AI agents.

PROJECTS

CogniSpace AI-enhanced workspace: notes, docs & sprint board

cognispace-sigma.vercel.app

React 19 · TypeScript · Vite · Tailwind · Framer Motion · Express 5 · MongoDB Atlas · Gemini API

- Built a Notion-style workspace with a TipTap block editor, "/" command menu, toggles and to-dos, plus an AI Copilot that improves or summarizes the text you select. Streamed Gemini responses over **Server-Sent Events** with stop and undo.
- Designed autosave that shows "Saved" only after server confirmation, retries offline edits with backoff, and uses version checks to offer "keep mine" or "load theirs" on conflicts.

RecoverFlow Razorpay AI Buildathon (Track 03)

recoverflow-ten.vercel.app

Node.js 22 · PostgreSQL · Razorpay webhooks · HMAC · Policy engine · Docker · CI

- Built a policy-first engine for failed payments: the AI proposes a diagnosis, a deterministic policy layer decides what is allowed, and a human approves one bounded action. Verified **HMAC signatures** and rejected forged webhooks.
- Made webhook handling **exactly-once**: the idempotency key, event and hash-chained audit record commit in one Postgres transaction; 200 simultaneous deliveries produced one record.
- Wrote a 47-case hard evaluation set; fixing what it exposed raised accuracy from **72.3% to 97.9%** with 0 unsafe actions (author-written cases, used as a regression gate). 49 tests, CI, role-based sign-in, Razorpay Test Mode only.

DocuMind Evaluation-first RAG over the Express 5 docs

github.com/karthik1122-code/documind

Node.js 22 · BM25 · LSA · rank fusion · reranking · SSE

- Wrote a 50-question test set first, then compared chunking and retrieval methods: heading-based chunks beat fixed windows at equal context (recall@1 **65.7% vs 51.4%**); the best local pipeline reached recall@3 **85.7%**.
- Built answer abstention: refused all 15 unanswerable questions (0 false answers) but answered only 29% of answerable ones correctly, which I report as a limit. Model-based results are pending.

AgentGate Policy gate and approval runtime for AI agents

github.com/karthik1122-code/agentgate

Node.js 22 · PostgreSQL · event sourcing · hash chain · policy engine

- Built a default-deny gate for agent tool calls: canonicalise first, then judge; approvals are bound to one call hash, need a different person, expire and are re-checked at execution; effectively-once execution, with role-based sign-in.
- Adversarial suite of 56 cases: **37 of 44 attacks denied, 7 held for a human, 0 executed**; 0 of 12 legitimate calls wrongly denied (author-written policy and attacks; scripted agents, simulated tools).

EXPERIENCE

IoT & Robotics Trainee · T-Works

Aug – Sep 2024

Worked on AI-powered smart street lights: hardware integration, sensor calibration and IoT connectivity.

Student Intern · Airbac Labs

Apr – Jul 2024

Learned Python and machine learning and applied them to practical projects, including stock-market prediction models.

Community Volunteer · Rubaroo

Aug 2024

Led a STEM-career awareness project that engaged 100+ students.

EDUCATION

B.Tech, Computer Science (AI/ML) · NxtWave Institute of Advanced Technologies (NIAT), Hyderabad

2025 – 2029

SKILLS

Languages

C++, Python, TypeScript, JavaScript

Frontend

React, Next.js, Tailwind CSS, TipTap, Framer Motion

Backend

Node.js, Express, MongoDB, Mongoose, JWT, SSE, REST, webhooks

AI

Gemini API, LLM streaming, prompt engineering, machine learning

Tools

Git, GitHub Actions, Docker, Vercel, Render