Mokshit Jain

Hyderabad, India | mokshitjain18@gmail.com
linkedin.com/in/mokshit-jain | github.com/mokbhai | mokshit.jainparichay.in

SUMMARY

AI Engineer with 1.5 years of experience building backend-heavy products, applied-AI pipelines, and workflow automation for startup clients. I own problems end to end, from requirements and architecture to APIs, deployment, and production support, mostly in Python/FastAPI and TypeScript/NestJS with PostgreSQL, Redis, Temporal.io, and Docker on AWS and GCP.

EXPERTISE

Languages:	Python, TypeScript/JavaScript, SQL, Rust (Tauri), Java [Spring Boot] (basics)
Backend & APIs:	FastAPI, NestJS, Node.js, Express, Flask, REST, tRPC, Microservices, Temporal.io
Applied AI:	OpenAI [GPT-4o, Vision, Realtime, embeddings], Gemini, LangChain/LangGraph, LiteLLM, RAG, pgvector, ChromaDB, Pinecone, ElevenLabs, whisper-rs
Databases:	PostgreSQL [indexing, pgvector], MongoDB, Redis, SQLite, MySQL; Prisma, TypeORM, SQLAlchemy
Cloud & DevOps:	AWS [S3, EC2, SQS], GCP [Cloud Run, Compute Engine], Docker, Nginx, GitHub Actions, Jenkins, Cloudflare
Frontend:	React, Astro, Next.js, Three.js, Tailwind CSS
Testing & Observability:	Pytest, Playwright, Vitest, Sentry, PostHog, OpenPanel
Practices & Tools:	Agile/Scrum, code review, OOP, Low-Level Design, DSA [350+ solved], Claude Code, Codex, Cursor

WORK EXPERIENCE

Johnson & Johnson - AI Engineer (Contract) | Nov 2026 – Present | Hyderabad, India

DRCODE (AIRIA) - Software Developer | Apr 2025 – Nov 2026 | Remote, India

Python, TypeScript, FastAPI, NestJS, PostgreSQL/pgvector, Redis, Temporal.io, Docker, AWS, GCP, OpenAI

- **YourMitra** (client project, sole engineer): built a live AI storybook platform with paying customers, from a **200+ endpoint** FastAPI/PostgreSQL/Redis backend to an Astro/React frontend. **30+ Temporal.io workflows** parallelize image, audio, and video generation, cutting story generation from **20 minutes to 2** while keeping page loads under 500ms. [case study]
- On YourMitra, also built a Redis token-bucket rate limiter across OpenAI, ElevenLabs, and Replicate (admin-tunable per provider), six operational feature flags, a 6-character short-link service, Razorpay payments, and a multi-provider notification engine (email, SMS, WhatsApp, RCS).
- Replaced fragile cron pipelines with parallel Temporal.io workflows: **0 production incidents** in the 2 months after the migration.
- **FDC** (team project): built the FastAPI WhatsApp cheque bot (Evolution API + GPT-4o Vision) and worked across the pipeline: compared 10+ OCR approaches, RAG catalog search, and settling bills in Busy accounting. A **10-minute** manual entry now takes **under 1 minute**, at **95%+ overall and 99% party-resolution accuracy** over ~300 cheques in its first 10 days, measured through the human-in-the-loop approve/disapprove step I built. [case study]
- **UPSCash**: built cash-reconciliation automation for a US department-store chain (NestJS, React, SQLite). It pulls end-of-day entries from **QuickBooks Online** over Intuit OAuth, checks each store against $100/$500 variance thresholds, and emails the owner on a mismatch; AES-256-GCM token storage, HMAC-verified webhooks, RBAC, audit logging. [case study]
- **Wall decor** (client project, with three other engineers): worked inside an existing **~129k-line** React/Three.js + headless Blender codebase. Added color-differentiated materials and photorealistic rendering, fixed resizing and the 22-rule cost engine's recompute after a resize, and split multi-thousand-line files into modules. [case study]
- Built the **UI-testing toolkit** for DRCODE's AI testing product from the ground up: a Playwright MCP agent that runs user-prompted flows on staging with test credentials and saves reusable scripts; also fixed bugs and improved test-case generation in the API end-to-end flow. [case study]

- Built **Run App** from the ground up: an Astro/React + 5-service NestJS fitness platform with a Flask ML pace predictor, **Strava and Garmin OAuth** sync, and Gemini run analysis; 9 Docker Compose services shipped to AWS EC2 through GitHub Actions. [case study]
- Built the Judge0 code-execution layer and Monaco editor for the DRCODE AI Interview platform; wrote a Pytest regression suite covering **120+ endpoints** on every PR, and kept ~90% automated test coverage in CI. [case study]
- Worked with the engineering team through code reviews and sprint planning, and led a small team of interns.

Innovation Studio LPU × Ministry of Defence - Freelance Developer | Nov 2024 – Jan 2025

Python, YOLOv5, Flask, NestJS, Next.js, PostgreSQL, Prisma

- Sole developer of a vehicle-checkpoint system, from gathering requirements with the Army unit to building, testing, and training its users: YOLOv5 ANPR + OCR plate recognition, live dashboards, RBAC and MFA. It tracks **10,000+ daily movements across 100+ checkpoints** and cut manual processing from **48 hours to under 5 minutes**; recognized by the Indian Army.

Innovation Studio LPU - Student Project Lead | Jun 2024 – Aug 2024

React, Node.js, MongoDB, WhatsApp bot

- Led a small student team building an event-registration platform with payments and a WhatsApp query/notification bot; it handled INR 1.5 lakh in registrations.

PROJECTS

KEA - Cross-platform AI desktop tool (tested on macOS)

github.com/mokbhai/KEA | Rust, Tauri 2, React, TypeScript, SQLite, OpenAI, whisper-rs

- Menu-bar app for global-hotkey text rewriting, push-to-talk dictation, and meeting notes with pluggable cloud or local AI engines; **63 Tauri commands** and a 3-OS CI matrix with signed releases. Saves me ~50 minutes a day. [case study]

orca - Multi-agent coding orchestrator

github.com/mokbhai/orca | Python, pipx

- CLI that fans work out to parallel headless AI coding agents with pluggable adapters, telemetry, and resumable sessions; used it to modularize ~14k lines that one agent session could not hold. [case study]

AutoTube - AI text-to-video pipeline

youtube.com/@mokbhaimj | Python, FFmpeg, Edge TTS, Selenium

- Scrapes a novel, cleans and translates the text, generates narration and images, assembles the video with FFmpeg, and uploads through the YouTube API, automating ~80% of the content work. [case study]

Thrombocizer - IoT leg-exercise device | 1st Runner-Up, MEDHA 2024

ESP32, Node.js, WebSockets

- Prototype that moves paralyzed patients' legs to cut DVT/blood-clot risk (~65%); I wrote the ESP32 firmware, did the wiring, and built the live monitoring dashboard in a team of four. [case study]

ACHIEVEMENTS

- **Certificate of Appreciation**, Indian Army (387 (I) Sup Pl ASC, Feb 2025), for delivering the Ministry of Defence Vehicle Impressment automation project.
- **Certificate of Recognition**, DRCODE: ownership, reliability, and production incident resolution.
- **1st Runner-Up**, MEDHA 2024 (Top 24 nationally); **1st Place**, CipherThon 2.0; solved **350+ DSA problems**.

EDUCATION

Bachelor of Technology, Computer Science & Engineering (minor in IoT)
Lovely Professional University | Aug 2022 – May 2026 | CGPA 8.1/10