Rudraksh Nanavaty

✉ rudrakshnanavaty@gmail.com   🔗 rudraksh.nanavaty.in   github.com/RudrakshNanavaty

M medium.com/@RudrakshNanavaty   in linkedin.com/in/RudrakshNanavaty   ☎ +91 94294 24060

Education

B Tech **Pandit Deendayal Energy University**, *Computer Engineering* 2021 – 2025

Experience

FirstPeak.ai, *Back End Developer* September 2025 – Present

- Built Python-based backend services for a multi-channel conversational AI platform; optimized async workflows to handle 50k+ concurrent requests with minimal latency.
- Implemented end-to-end usage tracking and billing, integrating Stripe subscriptions/checkout + metering to enable paid plans and automated invoicing/reconciliation workflows.
- Designed an agent-to-agent evaluation framework where "judge" agents orchestrate calls to other agents, reducing manual QA cycles by 70% and accelerating release velocity.
- Parallelized automated voice-agent testing by handling multiple concurrent async Twilio calls, cutting regression suite runtime from 4 hours to 25 minutes.
- Integrated Langfuse observability (trace + latency/cost visibility) to debug slow runs and optimize prompts/model routing.

New Engen, *Back End Developer* Mar 2024 – May 2025

- Migrated a JavaScript monolith into a TypeScript + Python microservices architecture, improving deployment frequency by 40% and reducing runtime errors by 25%.
- Built Python GraphQL APIs to power custom client dashboards, sustaining 100ms average response times even during 3x traffic spikes.
- Implemented a fault-tolerant bulk emailing pipeline using RabbitMQ, ensuring 99.99% delivery success across server restarts and network partitions.
- Introduced idempotency + durable job processing for email sends (e.g., dedupe keys, retries, dead-letter handling), preventing duplicate sends and guaranteeing at-least-once processing semantics.
- Established performance baselines (avg/p95) for critical APIs and iterated on async execution paths to hit strict latency targets for customer-facing dashboards.
- Optimized backend APIs via async processing and concurrency, improving response times and throughput under load.

Achievements

- Presented my first solo-authored academic paper as the youngest speaker at the *2024 Stanford Geothermal Workshop*. Paper
- Co-authored 2 academic papers submitted to high impact peer-reviewed journals under *Springer Nature*. Paper 1, Paper 2

Projects

NotebookLM RAG (NextJS, TypeScript, LangChain, Pinecone, PostgreSQL) Demo | GitHub

- Built a Retrieval-Augmented Generation (RAG) chatbot capable of answering domain-specific queries with high accuracy.
- Designed a robust backend with session persistence, vector-based retrieval, and dynamic ingestion of large documents.

ChatGPT Tokenizer (NextJS, ReactJS, TypeScript) Demo | GitHub

- Manual implementation of the Byte Pair Encoding (BPE) algorithm used by OpenAI in GPT-4.

Amazon Price Tracker (Selenium, Python, Go) GitHub | Blog

- Engineered a web scraper with Selenium + Go routines for concurrent price monitoring of multiple SKUs.

Algorithm Visualizers (ReactJS, JavaScript, Go)

- Engineered interactive simulations for OS scheduling and concurrency using React and Go; implemented a deadlock prevention engine for the Dining Philosophers problem.

Skills

AI & LLMs: RAG, LangChain, Pinecone, Prompt Engineering, Langfuse, Vector Databases
Programming Languages: Python, TypeScript, JavaScript, Go
Back End: FastAPI, ExpressJS, tRPC, REST, GraphQL, Gin
Front End: NextJS, ReactJS, Redux Toolkit, TailwindCSS
Databases: PostgreSQL, MongoDB, MySQL, SQLite3, Redis, Firebase
Cloud and DevOps: AWS, GCP, Docker, NGINX, Consul, RabbitMQ, Linux