

INDRANEEL SARODE

San Jose, CA

+1-669-360-7809

✉ indraneelsarode@gmail.com

🌐 [linkedin.com/in/indraneel-sarode-587178206](https://www.linkedin.com/in/indraneel-sarode-587178206)

🐙 github.com/neels22

🌐 indraneelsarode.com

Summary

MS Software Engineering student focused on enterprise architecture and distributed systems. Built and documented REST services (FastAPI/Next.js), CI/CD with GitHub Actions, and system integrations (Slack/AI services). Comfortable producing C4/UML diagrams, OpenAPI specs, runbooks, ADRs, and release notes. Availability: Summer 2026.

Education

San Jose State University

Master of Science in *Software Engineering* GPA: 3.56/4.0

Jan 2025 – Dec 2026

San Jose, CA, USA

University of Pune

Bachelor of Engineering in *Computer Engineering* GPA: 3.87/4.0

Aug 2020 – Jun 2024

Pune, India

Internships

Software Engineer Intern

Jan 2024 – Jun 2024

Coditas Solutions LLP

Pune, India

- Shipped a **LangChain**-based **RAG** service for **PDF/YouTube/web** with source-linked citations; on a seeded corpus, median answers consistently included **1 relevant citation**.
- Built **RAG-on-SQL**: NL→SQL translation with schema-aware prompts and guardrails (**read-only classification**, query linting, retries); malformed queries were auto-corrected or routed to a clarifying question *in staging*.
- Benchmarked vector DBs/retrievers and tuned chunking/overlap; **hybrid retrieval** with lightweight re-ranking **improved factual grounding** in evaluator spot-checks versus baseline.
- Versioned the **Python (FastAPI) REST API** (v1→1.1) and published **release notes + implementation plan** (roll-out/rollback, health checks); added caching/connection pooling to cut p50/p95 latency under like load.
- Built a Next.js **observability dashboard** (requests, latency bands, citation rates) so stakeholders could self-serve; reduced manual report tickets.

Projects

Aiventory Link | *FastAPI, Next.js, Gemini API, MCP client-server*

2025

- Designed a **FastAPI + PostgreSQL** backend with **JWT** auth and typed **Pydantic** models for products, orders, and inventory; implemented validation and pagination for clean CRUD.
- Delivered a **Next.js** admin dashboard centralizing product/order workflows with **server-side** search, filters, and bulk actions to streamline operations.
- Implemented an async **MCP** client-server with **Gemini** for natural-language lookups (e.g., “stock for SKU 123?”), including tool routing, parameter checks, and safe fallbacks.
- Hardened security with **RBAC** and rate limiting; added **audit logs** for sensitive actions and **request IDs** for traceability.
- Containerized with **Docker** and wired **CI/CD** (multi-stage images, env configs) for build/test/deploy, enabling smooth rolling updates behind a reverse proxy.
- Shipped OpenAPI 3.0 spec, **RBAC policy doc**, and runbook (alerts, dashboards, on-call steps) for the inventory backend.

BookIt Link | *Next.js, React, Docker, AWS EC2*

2025

- Delivered a **Next.js/React** portal for browse→reserve; reduced the demo booking flow from **7→4 steps** with validated forms and REST integrations.
- Deployed a **Dockerized** stack on **AWS EC2** behind a reverse proxy; containers **auto-restart on failure** and pass health checks in staging.
- Implemented **JWT** auth and route-level **RBAC**; customer/manager/admin dashboards provide guarded API access with **audit logs**.
- Documented JWT/RBAC routes and access patterns; wrote **integration notes** for frontend/backend handoffs.
- Worked in short sprints with daily check-ins; tracked issues to closure and documented integration decisions to unblock frontend↔backend handoffs.

Perplexity-2.0 Link | Demo | *Next.js 14, TypeScript, FastAPI, LangGraph, Google Gemini, SSE*

2025

- Architected a full-stack AI chatbot with **Server-Sent Events** for progressive, word-by-word **streaming**; added **stage indicators** (**search** → **read** → **write**) to reduce perceived latency.
- Built an agentic **LangGraph** workflow with **conditional tool-calling**; integrated **Tavily web search** + **Gemini** to auto-generate queries and synthesize **source-grounded answers**.
- Implemented **stateful conversations** with **MemorySaver** checkpointing across sessions; exposed a **FastAPI REST API** with streaming + **CORS** and containerized it for deployment (**Docker**→**Render**).

Technical Skills

Languages & Frameworks: Python, JavaScript/TypeScript, Java, C++, FastAPI, Node.js/Express, Next.js/React, Flask, Tailwind CSS, Bash, HTML/CSS

Integration & APIs: REST, Webhooks, OAuth2/OIDC, JWT, RBAC, API gateways/reverse proxy, Versioning & changelogs

Platforms, Ops & Observability: Docker, CI/CD (GitHub Actions), AWS (EC2, S3, Lambda, ECS), Grafana, Prometheus, OpenTelemetry (basic)

Data & ML: PostgreSQL, MySQL, MongoDB, DynamoDB, Pandas, NumPy, PyTorch, Keras, Vector DBs

Architecture & Documentation: C4 model, Sequence diagrams, ADRs (Architecture Decision Records), OpenAPI/Swagger, Interface catalogs, Implementation plans, Release notes, Runbooks, Mermaid/PlantUML