PRATEEK N JANAGOUDA

Bangalore, India • +91-8088293981 • prateekjanagouda1008@gmail.com • linkedin.com/in/prateek-janagouda • github.com/PrateekJanagouda

I build production LLM systems — RAG, agentic tooling, and getting inference to run efficiently. Nearly two years of that has been at Synchronoss, and the work is in daily use by our engineering team rather than sitting in a notebook. I care a lot about whether what I build holds up once real users hit it, so I lean heavily on evaluation and grounding.

PROFESSIONAL EXPERIENCE

Synchronoss Technologies, Bangalore *Oct 2024 — Present*
Software Engineer Trainee, Machine Learning *Nov 2025 — Present*
Machine Learning Intern *Oct 2024 — Oct 2025*

Code Brain — Agentic Code-Intelligence RAG

An internal assistant that lets developers ask plain-English questions about our Java microservices and get back answers backed by real citations from the codebase.

- Built an agentic RAG across **61 Spring Boot microservices** (~50K Java files, ~1.2M call-graph edges, ~350K symbols). The agent runs on vLLM with Qwen3-30B-A3B and picks from 20+ retrieval tools to answer with citations. The engineering team uses it regularly.
- Went with two stores instead of one: Postgres holds the exact structural facts (symbols, call graph, beans, endpoints, properties) plus raw-file grep, and Qdrant handles semantic search over ~1M chunks in 9 collections (jina-code/bge embeddings, bge-reranker-v2-m3).
- Added a `javap` bytecode pass on top of tree-sitter so the call graph resolves across repos, which parsing alone can't do. Every answer has to cite its evidence, checked by a citation validator and confidence scoring.
- Wrote an eval harness that auto-generates questions from the index (106 of them) and scores answers against it. It passes 95%.

Tech: Python, Qdrant, PostgreSQL, vLLM, Qwen3-30B, jina/bge embeddings, tree-sitter, FastAPI

Helm Config Reconciliation Agent

An agent that keeps customer Helm deployments in sync with a source-of-truth environment. That propagation used to be slow, manual, and easy to get wrong.

- Built an agent that pushes Helm chart upgrades from the source-of-truth env out to **50+ microservices** across customer environments. It catches both version drift and missing config keys, then reconciles each customer's overrides.
- Instead of letting the model rewrite YAML, I had it emit only **RFC 6902 JSON Patch ops** against a fixed schema, grounded in the actual git diff. That keeps it from hallucinating changes on config that can't afford mistakes.
- Every change runs through `helm template` + kubeconform and the repo's own pre-commit hooks (detect-secrets, yamllint) before it ships, with auto-fix and rollback. Comments and secret pragmas survive the patching so nothing downstream breaks.

Tech: Python, ruamel.yaml, jsonpatch, GitPython, vLLM/Qwen, Helm, kubeconform

LLM JUnit Test Generation & Inference Optimization

A Co-Pilot project to figure out which open-source LLM was best at writing JUnit tests for Java, then make it fast enough for the whole team to use at once.

- Tested 5+ open-source models (LLaMA, Starcoder, CodeGen) for Java test generation. Moved serving from Ollama to vLLM once load testing showed Ollama choking — it dropped every request at 20 concurrent users.
- Benchmarked it with an async load script from 1 to 2,000 concurrent. vLLM held **1,000 at 100% success** and ~790 tok/s thanks to continuous batching.
- Hooked the Continue plugin into IntelliJ so developers could generate tests right inside their editor.

Tech: Python, LLaMA, vLLM, Ollama, Continue IDE, IntelliJ

PERSONAL PROJECTS

MiniRAG — RAG Pipeline from Scratch *github.com/PrateekJanagouda/minirag*

A full RAG pipeline with no LangChain or ChromaDB, just numpy and FastAPI. I built it to understand every step myself: chunking, nomic-embed-text embeddings, cosine-similarity retrieval, then generation.

PyTestGen — AI Unit Test Generator *github.com/PrateekJanagouda/pytestgen*

A CLI that writes unit test cases for any Python file using a local LLaMA through Ollama. Runs fully offline with no API keys, has a syntax-highlighted terminal UI, and saves the test files for you.

TECHNICAL SKILLS

LLM & Inference	vLLM, Ollama, LLaMA, Qwen3-coder, continuous batching, guided / structured decoding
RAG & Vector DBs	Qdrant, agentic / tool-calling RAG, cross-encoder reranking, jina / bge / nomic embeddings, semantic search
AI / ML	TensorFlow, Keras, Scikit-learn, OpenCV, MediaPipe
Web & APIs	FastAPI, Flask, REST APIs, streaming responses
Languages & Tools	Python, Java, C, Git, Docker, Linux, PostgreSQL, Helm

EDUCATION & CERTIFICATIONS

B.E. Artificial Intelligence & Data Science *2021 — 2025*

· SG Balekundri Institute of Technology, VTU **CGPA 8.48 / 10**

Pre-University *2019-2021*

· ICS Mahesh PU college, Dharwad **Grade: 87.16**

SSLC

· Holy Cross Convent School, Bidi **Grade: 84.64**

CERTIFICATIONS

- Claude Code in Action, Anthropic (2026)
- Data Structures & Algorithms in Java, Great Learning
- Python (Basic), HackerRank
- Problem Solving Through Programming in C, NPTEL / IIT