Gautama Shastry Bulusu Venkata

+1 571-653-0056 | satya.2k02@gmail.com | https://linkedin.com/in/satya2603/ | https://github.com/GautamaShastry |
https://gautamportfolio.com

Professional Summary

Master's Candidate and Software Engineer focused on **Full Stack Development and Generative AI**. Experience designing and deploying secure, scalable microservices and orchestrating multi-agent **LLM workflows**. Proficient in reducing latency and optimizing UI performance using **Java Spring Boot, React, and AWS**. Published researcher in NLP with a focus on cloud-native deployments and RAG pipelines.

Education

George Mason University — Fairfax, VA
Master's in Computer Science - 3.85/4 — *Jan. 2024 – December 2025*

- **CourseWork:** Introduction to Mathematical Foundations of CS, Intro to AI, Systems Programming, Data mining, Machine Learning, Cryptography, Deep Learning, Analysis of Algorithms, DevOps, Natural Language Processing

Andhra University — Visakhapatnam, AP, India
Bachelor's in Computer Science - 8.15/10 — *Aug. 2019 – May 2023*

Technical Skills

Languages: Java, Python, JavaScript/TypeScript, SQL, C/C++, HTML/CSS
Backend: Spring Boot, FastAPI, Node.js, Express, Django, Flask, REST, Microservices, Distributed Systems, Linux/OS
Frontend: React, Redux, Vite, Tailwind CSS
Cloud/DevOps: AWS (EC2, S3, Lambda, RDS, Route 53, CloudFront, IAM, SQS/SNS, CloudWatch), Docker, Kubernetes, Jenkins, Rancher, Git/GitHub
Databases: PostgreSQL, MySQL, MongoDB, JPA/Hibernate, HikariCP
AI/ML: LangChain, LangGraph, Retrieval-Augmented Generation (RAG), Vector DBs (FAISS, Qdrant), HuggingFace, TensorFlow, Keras, spaCy, NLP, Agentic AI, LLM-fine tuning, LLMs, OpenAI, Claude
Tools & Practices: VS Code, IntelliJ, Postman, Maven, Jupyter, Confluence, CI/CD, Agile/Scrum, Version Control (Git)
Other: Data structures and algorithms, Object-oriented programming, Operating systems, Networking, parallel processing

Experience

Associate Software Engineer — Jan 2023 – December 2023
Backflipt — *Hyderabad, India*

- **UI Architecture Modernization**: Refactored a monolithic UI into modular React/Redux feature slices, improving component reusability and increasing user engagement by **20%**.
- **Performance Optimization**: Eliminated render bottlenecks by implementing strategic memoization and granular component splitting, **reducing re-render** overhead by **40%** and page load time by **10%**.
- **Latency Reduction**: Optimized client-server communication by normalizing Redux entities and implementing caching layers, **cutting response latency by 15%**.
- **Developer Experience**: Authored 10+ technical documentation pages on Confluence, standardizing UI patterns used by 50+ developers; this initiative reduced new-hire onboarding ramp-up time by 25% and ensured consistent code quality across squads.

Projects

Support Sage — AI Customer Support Agent | *Python, LangGraph, ChromaDB, FastAPI, React* — Jan 2025-Feb 2025

- **Architected a RAG pipeline** using ChromaDB and OpenAI embeddings to enable semantic search over 8 policy documents; implemented paragraph-based chunking with overlap to preserve context, reducing query latency from 150ms to 15ms (10x improvement) compared to brute-force vector search.
- **Built an agentic workflow** using **LangGraph ReAct** pattern with 9 specialized tools for order management, profile updates, and ticket escalation; designed policy-enforced order cancellation that validates customer reasons against business rules, automating 7 cancellation scenarios without human intervention.
- **Developed production-ready backend** with FastAPI featuring async **LLM orchestration, session-based conversation memory** with TTL expiration, **IP-based rate limiting, and Jira integration** with exponential backoff retry logic for reliable ticket creation
- **Implemented end-to-end observability** including structured JSON logging, request ID tracking, response time metrics, and health check endpoints with database/vector store monitoring; **achieved 100% test coverage** across 32 unit tests

Document Assistant | *FastAPI, LangChain, FAISS, RAG, Ollama, HuggingFace* Aug 2025 - Sept 2025

- **High-Throughput RAG Pipeline**: Developed a FastAPI service integrating **LangChain and Ollama**, capable of processing document ingestion at **5.57 chunks/sec** with an optimized chunking strategy (avg 555 chars) to maximize context window utilization.
- **Vector Search Optimization**: Implemented FAISS indexing with hybrid search capabilities (Similarity + MMR), delivering sub-50ms retrieval speeds ($k = 4$ in 0.038s avg) across dense vector spaces.
- **Production-Grade Lifecycle**: Designed full vector-store administration endpoints (create, load, stats) and integrated citation-enforced prompts to ground LLM responses in retrieved context, significantly reducing generation errors.
- **Quality Assurance**: Established a comprehensive automated test suite achieving a 100% pass rate across ingestion, indexing, and retrieval modules, ensuring regression-free deployments.

Student Survey Application | *SpringBoot, Docker, Kubernetes, Jenkins, AWS, Rancher* March 2025 - May 2025

- **Cloud-Native Architecture**: Containerized a Spring Boot application using Docker and orchestrated a highly available 3-replica deployment on AWS EC2 using Kubernetes and Rancher, achieving 99.9% uptime during the pilot phase.
- **Automated CI/CD**: Engineered a Jenkins pipeline that triggers on Git commits to build Maven artifacts, push versioned Docker images, and execute rolling updates, reducing deployment turnaround time to <5 minutes.
- **Database Tuning**: Provisioned Amazon RDS (MySQL) and optimized HikariCP connection pooling (configured maxPool=20, minIdle=5) to handle concurrent traffic spikes without exhausting database connections.
- **API Reliability**: Designed strictly typed RESTful endpoints using JPA/Hibernate and aggressive input validation (@NotNull, @Email) to prevent bad data ingestion, verified by a suite of 10+ production-grade Postman tests.

Resume Analyzer | *React.js, SpringBoot, Python, Flask, PostgreSQL, GPT-4, AI agents, LangGraph* Feb 2025-Nov 2025

- **Multi-Agent Orchestration**: Architected a sophisticated AI workflow using LangGraph to coordinate 6 specialized autonomous agents (Parsing, Scoring, Skill-Matching), enabling complex reasoning chains that reduce hallucination rates compared to zero-shot prompting.
- **Hybrid Scoring Algorithm**: Engineered a dual-layer evaluation engine combining **Semantic Similarity** (Sentence Transformers, 60% weight) and **Rule-Based NER** (spaCy, 40% weight) to quantify candidate-job fit with high precision.
- **Resilient Microservices**: Built a fault-tolerant Spring Boot backend decoupled from the Flask AI service; implemented Circuit Breaker patterns and retry logic to maintain zero API failures during high-concurrency stress testing.
- **Performance & Output**: Optimized the analysis pipeline to achieve < 1.5s p95 latency per resume while generating detailed visual analytics (Matplotlib) and PDF reports via a dedicated reporting microservice.

TRUST Agents: Multi-Agent Fact-Checking System | *Python,GPT-4, LangGraph, RAG, AI Agents* Sept 2025 - Dec 2025

- **Architected** a scalable, multi-agent fact-checking pipeline using **4 specialized LLM agents** (Claim Extractor, Evidence Retrieval, Verifier, Logic Aggregator) and **Delphi consensus verification** for high-fidelity decision tracing, enhancing system **interpretability** by **30%** compared to monolithic models.
- **Developed** a Claim Decomposition Engine leveraging **Named Entity Recognition (NER)** and dependency parsing to break complex statements into verifiable sub-claims, enabling the formal computation of compound verdicts via custom **AND/OR/IMPLIES logic aggregation**.
- **Benchmarked** multi-agent performance against **7 SOTA methods** on the **LIAR** dataset (**12.8K** political statements), achieving a strong baseline of **65.2% accuracy** with fine-tuned **BERT** while isolating performance gains attributable to the architectural decomposition.
- **Analyzed** agent uncertainty across configurations, identifying **70-82% uncertain prediction rates** and formalizing this model conservatism as an actionable signal for **human-in-the-loop** escalation in production misinformation detection systems.

Certifications

- AWS Cloud Practitioner CLF-C02
- Deep Learning Specialization(Deep Learning.ai)

Publications

- Code-Mixed Telugu-English Hate Speech Detection **https://arxiv.org/abs/2502.10632**
- How Does A Multilingual LM Handle Multiple Languages? **https://arxiv.org/abs/2502.04269**