SAI VIVEKANAND REDDY

857-465-9966 | vangala.sa@northeastern.edu | Boston, MA | LinkedIn | GitHub | Portfolio Site

Education

Northeastern University, Boston, USA *September 2023 - December 2025*

Master of Science, Computer Software Engineering

- **Relevant Courses:** Data Science Engineering, MLOps, Networks & Cloud Computing, Intelligence Analytics

BITS Pilani, Pilani, India *August 2016 - June 2020*

Bachelor of Engineering, Electronics Engineering

- **Relevant Courses:** OOPs, AI, DBMS, Algorithms, Operating Systems, Neural Networks, Distributed Systems

Technical Skills

- **Languages:** Python, Java, Go, SQL, Javascript, HTML5/CSS, TypeScript
- **AI/ML:** Gen AI, LangChain, LangGraph, NLP, PyTorch, TensorFlow, scikit-learn
- **Cloud & DevOps:** GCP, AWS (S3, EC2, Lambda, RDS), Docker, Kubernetes, Terraform, CI/CD, Packer
- **Frameworks & Tools:** Spring Boot, FastAPI, React, Git, Spark, Airflow, MySQL, PostgreSQL, Redis

Work Experience

Humanitarians AI Boston, USA

AI Engineer *Jan 2025 - May 2025*

- Achieved over 95% accuracy in multi-modal document processing by engineering a content analysis engine with the Gemini Vision API to handle 10+ file formats, including image-based table extraction.
- Architected a 4-tier persistent memory system for AI agents, enabling semantic search and long-term knowledge retention by integrating ChromaDB for vector storage and OpenSearch for immutable, large-scale audit logging.
- Deployed highly reliable AI services with 99.9% uptime by orchestrating a complete CI/CD pipeline using Terraform, Docker, and Kubernetes for automated, zero-downtime deployments.

Puddl Bangalore, India

Member of Technical Staff *June 2022 - August 2023*

- Engineered a client-side analytics platform for OpenAI API monitoring, ensuring 100% data privacy by architecting a React and TypeScript Single Page Application (SPA) that processed sensitive API keys exclusively in the browser.
- Translated complex API cost and token data into actionable insights by developing an intuitive dashboard with Recharts that processed and visualized usage data directly from OpenAI's API.

Blue Yonder Hyderabad, India

Software Engineer *July 2020 - May 2022*

- Reduced initial load times by 40% for data-heavy users by re-architecting the data-loading mechanism, implementing dynamic SQL query construction in Java to selectively fetch only user-requested measures.
- Enhanced user productivity by 25% by re-engineering a monolithic 'Save & Calc' feature into discrete actions, leveraging server-side HTTP session caching to enable rapid, database-free "what-if" forecast simulations.
- Diagnosed and resolved over 70+ customer-reported bugs annually by debugging a full-stack Java Spring MVC application and implementing robust fixes with comprehensive JUnit tests.

Projects

Cloud-Native Web App on GCP (Java, Spring Boot, Terraform)

- Engineered a scalable, 3-tier web application on GCP, achieving high availability by deploying a Spring Boot API to an auto-scaling instance group and a private Cloud SQL DB.
- Automated infrastructure provisioning for 15+ GCP resources (VPC, KMS, LB, SQL), reducing manual deployment time by 100% using Terraform and Packer for immutable, version-controlled infrastructure.
- Designed an event-driven, asynchronous email verification system using GCP Pub/Sub and Python Cloud Functions, decoupling the service from the main Spring Boot application to improve resilience.

Multi-Agent RAG Research System (Python, LangGraph, Airflow)

- Architected a multi-agent research system that synthesizes information from parallel web, academic, and RAG agents by orchestrating the workflow using Python, LangGraph, and a FastAPI backend.
- Automated a multi-modal document ingestion pipeline using Apache Airflow to process and index diverse file formats (PDFs, images) into a Pinecone vector database for efficient semantic retrieval.

Distributed Deep Learning Training Pipeline (PyTorch, HPC)

- Achieved a 3.7x speedup on a U-Net text-to-image model by designing and implementing a multi-GPU parallel processing pipeline using PyTorch Distributed Data Parallel (DDP) on an HPC cluster.
- Reduced training time by an additional 66% by integrating Automatic Mixed Precision (AMP), optimizing hardware utilization and memory efficiency.