

SAI VIVEKANAND REDDY VANGALA

saivivekred@gmail.com | linkedin.com/in/vangalas | github.com/sai-vivekanand | saivivek.com

EDUCATION

Northeastern University

Master of Science, Computer Software Engineering

BITS Pilani

Bachelor of Engineering, Electronics Engineering

Boston, USA

Sept 2023 – Dec 2025

Pilani, India

Aug 2016 – June 2020

TECHNICAL SKILLS

Languages & Frontend: Python, C++, Go, Java, SQL, TypeScript, JavaScript, React

Backend & Distributed Systems: Kafka, Spring Boot, Node.js, Redis, PostgreSQL, OpenSearch, TCP, Concurrency

AI/ML Systems: PyTorch (DDP, AMP), CUDA, NCCL, LangGraph, RAG, Ollama

Cloud & Infrastructure: Kubernetes, Docker, Terraform, GCP, AWS (EKS), Helm, Prometheus, Grafana, CI/CD

EXPERIENCE

IpsierLab

Software Engineer Intern

Remote, USA

April 2025 – present

- Built a provider-agnostic LLM abstraction layer that swaps Claude for local Ollama models behind a single environment variable, so AI parsing isn't locked to one vendor.
- Prevented abuse of auth, OCR, and AI-suggestion routes with a Redis-backed sliding-window rate limiter that falls back to in-process limiting if Redis becomes unavailable, keeping the API online.
- Shipped one React Native/Expo codebase to iOS and Android, closing platform gaps like date pickers and camera permissions, and signed releases through EAS build profiles.

Humanitarians AI

AI Engineer Co-op (Backend & Infrastructure)

Boston, USA

Jan 2025 – May 2025

- Owned ingestion, chunking, and metadata-aware ChromaDB indexing for a Gemini Vision OCR pipeline reaching **~95%** extraction accuracy (reviewed eval set) across 10+ document formats.
- Designed the state-passing contracts and episodic-memory indexing that multi-step agent workflows relied on, with OpenSearch-backed audit logs for traceability.
- Automated containerized rollouts on GCP with GitHub Actions, Terraform, and Kubernetes; health-check-based validation held availability at **~99.9%** during the monitored period.

Puddl

Founding Engineer (Volunteer)

Bangalore, India

June 2022 – Aug 2023

- Built a React/TypeScript analytics SPA that turned raw provider usage metadata (cost, latency, tokens) into project-level dashboards, syncing sanitized summaries to a Node.js backend for persistence.
- Kept provider API keys in-browser with a privacy-first client-side credential flow, so the backend never received or stored raw credentials.

Blue Yonder

Software Engineer (Full-Stack)

Hyderabad, India

July 2020 – May 2022

- Reduced forecast workbook load times **40%** for enterprise retailers by replacing greedy data fetching with a dynamic SQL query builder in the Java Spring backend.
- Added a server-side session-caching layer so planners could run rapid “what-if” simulations without repeated database writes, lifting forecasting throughput **25%**.
- Fixed **70+** customer-reported bugs a year across the Spring MVC backend, JSP/JS frontend, and Oracle 18c queries, backed by JUnit regression suites holding **95%+** coverage.

PROJECTS

Kubernetes Platform on EKS | Terraform, Kubernetes, Go, Kafka

[\[GitHub\]](#)

- Created a custom **Kubernetes operator in Go** with Kubebuilder that reconciles hourly CVE releases into processing Jobs using CRDs and finalizers.
- Provisioned a multi-AZ EKS platform end-to-end with Terraform (VPC, IAM/OIDC, KMS, managed node groups), with Prometheus/Grafana metrics and Fluent Bit shipping logs to CloudWatch.

Kafka From Scratch | Go, TCP, Kafka Protocol, Concurrency

[\[GitHub\]](#)

- Built a Kafka-compatible broker from scratch in Go: length-prefixed binary request parsing, correlation IDs, and concurrent client handling over a goroutine-based TCP server.
- Metadata-discovery and read/write protocol flows parse KRaft cluster metadata and pass the CodeCrafters test harness.

Distributed Training Systems | Python, PyTorch DDP, AMP, CUDA, NCCL

[\[GitHub\]](#)

- Accelerated CNN training **3.7×** with PyTorch DDP over NCCL across 4 GPUs on a single node.
- Cut GPU training time **66%** and memory footprint **31%** while retaining **93%** accuracy by integrating Automatic Mixed Precision with bfloat16 autocast.

Text-to-SQL Distillation Engine | Python, Ollama, DeepSeek-V3, Qwen

[\[GitHub\]](#)

- Improved schema-grounded Text-to-SQL execution accuracy from **36% to 74%** on a held-out query set by distilling DeepSeek-V3 outputs into a compact Qwen 0.6B student model.
- Reduced serving latency and cost by deploying the quantized student model through Ollama GGUF, enabling local SQL generation over a Pandas/SQLite schema layer without remote teacher inference.