Affan Arif Khamse

Software Engineer | Distributed Systems | AI/ML Applications | NYU CS 2025

New York City, NY · (516) 853-4972 · khamseaffan@gmail.com · github.com/khamseaffan · linkedin.com/in/affan-khamse

Education

Master of Science in Computer Science (GPA: 3.8/4.0) — New York, USA
New York University — *May 2025*

Bachelor of Engineering in Computer Engineering (GPA: 9.00/10.0) — Mumbai, India
University of Mumbai — *June 2023*

Technical Skills

Programming and Scripting Languages: Java, Python, JavaScript, C++, TypeScript
Frameworks and Libraries: Spring Boot, FastAPI, Flask, Django, ReactJS, LangChain, Node.js
Cloud & Infrastructure: AWS (EC2, DynamoDB, S3, Lambda, SQS, SES), Azure, Docker, Kubernetes
Databases: SQL (MySQL, PostgreSQL), NoSQL (MongoDB, DynamoDB), Redis, Firebase
Testing & Monitoring: JUnit, PyTest, Django Test, CloudWatch, Chrome DevTools
DevOps & CI/CD: GitHub Actions, Travis CI, Docker Compose, AWS SAM, Git, Swagger/OpenAPI
System Design: Microservices, Event-Driven, Distributed Systems, RESTful APIs, WebSockets

Work Experience

AI Software Engineer — New York, USA
Novum AI — Jun 2025 – Present

- Architected a **real-time telephony system** using Telnyx and **AWS serverless architecture** (Lambda, DynamoDB, API Gateway, WebSockets) to manage inbound call routing, queuing, and live agent availability with **99.9% uptime**
- Implemented **dual-stream audio processing pipeline** integrating AssemblyAI for live transcription and **AWS Bedrock** for sentiment analysis, processing real-time analytics on agent dashboard with **500ms or less latency**
- Built **event-driven microservices** with AWS SAM and CI/CD pipeline, implementing **JWT authentication**, RBAC, and **WebSockets** for real-time dashboard updates across 50+ concurrent sessions

Technical Lead — New York, USA
InquisAI (Startup Project) — Jun 2024 – Mar 2025

- Spearheaded an **AI assistant platform** using LangChain and OpenAI Embeddings for document-based question answering with **Chroma vector store** for semantic search
- Redesigned backend by migrating Flask to FastAPI with asynchronous processing and optimized database queries, reducing API latency by 30% through performance testing
- Led **Agile development** in a 3-person team, using Azure DevOps to manage sprints and implementing **CI/CD pipelines** for automated testing and deployment to AWS

Projects

Home Store – Multi-Tenant E-Commerce Platform — Jan 2025 - Present
Spring Boot, Spring Cloud, Eureka, Docker, PostgreSQL, React Router, Firebase, SwaggerUI — GitHub

- Engineered a **microservices ecosystem** using Spring Boot with **Eureka service discovery**, centralized configuration, and API Gateway for a scalable multi-tenant e-commerce platform
- Designed modular services with clear separation of concerns for core functionalities (product, cart, order, store management), enabling independent deployment and scaling
- Containerized services with **Docker/Docker Compose** and implemented **RESTful APIs** with SwaggerUI documentation, enabling consistent testing and parallel development

FlashBids - Anti-Sniping Auction Platform — Sep 2024 - Dec 2024
AWS EC2, DynamoDB, S3, Redis, WebSockets, CloudWatch, Flask, Python, REST APIs, DevOps — Demo | GitHub

- Created a **real-time auction platform** using Flask and WebSockets to prevent last-second sniping, implementing a fair bidding mechanism with automatic time extensions
- Constructed a **scalable backend** using AWS EC2 with Auto Scaling and DynamoDB, designed to handle 1000+ of concurrent users through elastic infrastructure
- Engineered **non-blocking APIs** and Redis pub/sub event pipeline, achieving **30% improved latency** and sub-second response times verified with 25-30 concurrent users

VibeCheck - Social Music Platform — Sep 2023 - Dec 2023
Django, Python, Bootstrap, AWS Elastic Beanstalk, PostgreSQL, Redis, TravisCI — Demo | GitHub

- Built a social platform that paired users based on **real-time Spotify API integration** and music preference similarity
- Established a **reactive messaging framework** using Redis Pub/Sub, reducing end-to-end chat latency by **30–40%**
- Deployed via **AWS Elastic Beanstalk** with **CI/CD pipeline** (Travis CI), achieving **87% test coverage** ensuring reliability