

REET NANDY

reet.nandy@nyu.edu | reetnandy.com | github.com/techpertz | linkedin.com/in/reetnandy | +1 (518) 930-6116

EDUCATION

- New York University** – Masters in Computer Science (MS) May 2025
- *Relevant Coursework:* Data Structures and Algorithms, Cloud Computing, Machine Learning, Big Data Analytics, Database Systems, Software Engineering, Object-oriented Design, Probability and Statistics
 - *Graduate Teaching Assistant:* CS GY 6233 - Operating System, CSCI UA 0310 - Algorithms
- Manipal University Jaipur** – Bachelors in Computer Science (B. Tech) May 2023

SKILLS

Languages: Python, JavaScript / TypeScript, SQL, C/C++, Java, HTML / CSS, Bash

Frontend + Backend: React, Next.js, Tailwind CSS | Django, FastAPI, Node.js / Express.js, Spring Boot, Microservices

Cloud + DevOps: AWS (EC2, S3, Lambda), GCP, Kubernetes, Docker, Terraform, Prometheus / Grafana, CI/CD, ELK Stack

Databases + APIs: PostgreSQL, MongoDB, Redis | Airflow, Kafka, RabbitMQ | REST / GraphQL / gRPC | OAuth2, Agile

PROFESSIONAL EXPERIENCE

- Backend Development Intern** June 2024 – December 2024
Mobility Intelligence, New York City, USA *React, FastAPI, AWS, PostgreSQL, Airflow*
- Designed WebSocket-powered visualization dashboard reducing client-side latency from 2.1s to 290ms.
 - Architected hybrid FastAPI deployment (EC2+LAMBDA) with auto-scaling, achieving 99.9% SLA and 90% cost savings.
 - Constructed high-throughput ETL system batch processing 15M+ records/day using Airflow DAGs and S3 data lakes.
- Software Engineering Intern (R&D)** January 2023 – June 2023
Defence Research & Development Organisation, India *PyQT, Redis, PostgreSQL, Docker, Airflow, Threading*
- Engineered multi-threaded architecture for real-time LiDAR processing, handling 50K data points/sec (97% accuracy).
 - Configured Redis-PostgreSQL query system using geospatial indexing to reduce GPS API latency from 1sec to 150ms.
 - Developed ETL pipeline with zero-copy buffers, processing 12GB/min while reducing memory usage by 60%.
- Software Engineering Intern** April 2022 – December 2022
Solar Industries India Ltd, India *Python, AWS, Django, Kafka, Redis, SQL, Docker*
- Led cross-facility automation system deploying 5 Django microservices, standardizing 80% processes.
 - Spearheaded Kafka pipeline with horizontal scaling, processing 2.5M rows/sec (100K daily requests) at 99% uptime.
 - Optimized API from 1200ms to 960ms via Redis cache strategies (cache-aside/write-through) and connection pooling.

PROJECTS

- [AWS / Fullstack] AI-Fitness Analytics Dashboard (Github)** *AWS, ReactJS, Django, Kubernetes, DynamoDB, KNN*
- Deployed Django API on Elastic Beanstalk with Google Fit integration, configured auto-scaling groups + health probes.
 - Orchestrated pipelines with Lambda-SageMaker, deploying KNN models for recommendations at 92% accuracy.
 - Coordinated event-driven metrics processing via SNS/SQS, achieving 800ms p95 latency for real-time health data.
- [Python / Core] Hierarchical Vector Database from Scratch (Github)** *FastAPI, Pydantic, Cohere, AsyncIO, Kubernetes*
- Built embedding database (library - document - chunk) w/ async collection mutexes; 12K ops/sec at <0.1% conflicts.
 - Added 3 indexing algorithms (LinearScan/KD-Tree/LSH) for vector search on 10M vectors in 18ms.
 - Led Kubernetes Helm deployment along with custom made CLI toolkit, reducing onboarding complexity by 100%.
- [AWS / Infra] Cloud Monitoring and Alerting System (Github)** *AWS, Docker, Kubernetes, Prometheus*
- Released Flask - MongoDB on AWS EKS with managed nodes and HPA, enabling rolling updates and health probes.
 - Established Prometheus with SNS-Lambda alerts for Slack, configuring multi-cluster monitoring rules.
- [Java / Fullstack] Real-Time Collaborative Whiteboard (Github)** *Java, Spring Boot, Swing, Websockets, Concurrency*
- Built low-latency collaboration using WebSockets, achieving <150ms sync for concurrent users via binary compression.
 - Implemented vector operations using operational transformation, resolving 98% conflicts in real-time updates.
 - Executed batch processing with PostgreSQL and JSONB storage, achieving 85% network overhead reduction.