PARTHO MAL

Parthomal7@gmail.com
github.com/Partho-Mal | linkedin.com/in/parthomal | parthomal.vercel.app

Summary

Software engineer experienced in building scalable distributed systems and low-latency services using Go, Python, and Java. Strong background in system design, data structures, and performance optimization, with hands-on experience deploying production workloads using PostgreSQL, Redis, and Docker.

Education

Bachelor of Engineering – Information Technology **June 2022 - June 2026**

Vasantdada Patil Pratishthan's College of Engineering, Mumbai University, India (CGPA: 8.0)

Projects

Shortly - URL Shortener & Analytics Platform **May - July 2025**

Go (Gin), PostgreSQL, Redis, JWT, React

- Engineered a **high-performance URL shortening service** handling **13K+ requests/sec**, validated via load testing with hey.
- Reduced redirect latency by **~70%** by introducing Redis caching and optimizing database access patterns.
- Implemented **JWT-based authentication**, QR code generation, and a **real-time analytics dashboard** for link usage insights.
- Designed stateless backend services to enable horizontal scalability.
- Deployed at: shortly.streamlab.in

Distributed Transaction Ledger System **Dec 2025 - Jan 2026**

Java, Spring Boot, Apache Kafka, Redis, PostgreSQL

- Designed and implemented a **fault-tolerant, event-driven payment ledger** processing **~2,300 TPS** in local benchmarks using Apache Kafka.
- Eliminated double-spending and race conditions by enforcing **idempotency keys** and **optimistic locking**, ensuring deterministic and ACID-compliant transaction processing across **180k+ stress-test requests**.
- Achieved **p95 latency <190 ms** under **500 concurrent users** by optimizing Redis caching strategies and tuning HikariCP connection pooling for high-throughput workloads.
- Source: fintech-transaction-service

Real-Time Fraud Detection Engine **Jan 2026**

Python, FastAPI, XGBoost, ONNX Runtime, Docker

- Built a **low-latency fraud detection microservice** performing real-time transaction classification as a downstream consumer of a payment ledger.
- Reduced model inference latency by **~95%** by migrating XGBoost artifacts from pickle to **ONNX Runtime** (~0.45 ms to ~0.02 ms per inference), achieving **<25 ms p95 end-to-end latency** in local load tests.
- Implemented **Shadow Mode inference** to safely evaluate model variants on live request payloads, validating a **~15% recall improvement** without impacting production decisions.
- Containerized and deployed the service using Docker for reproducible inference environments.
- Source: fraud-detection-service

Technical Skills

Languages: Python, Go, Java, C, C++, TypeScript, JavaScript, SQL
Frameworks & Technologies: Spring Boot, FastAPI, Gin, React, Apache Kafka, gRPC, XGBoost, ONNX Runtime, PostgreSQL, Redis
Developer Tools & Platforms: Docker, AWS, Git, Linux, Jenkins, Postman
Core Concepts: Data Structures and Algorithms, Object-Oriented Programming (OOP), Microservices, Operating System, Computer Networks