

Priyadarshan Ghosh Hazra

Bengaluru, India

priyadarshanghoshhazra908@gmail.com | +91-8100357319 | GitHub: Priyadarshan | LinkedIn: Priyadarshan

SKILLS

Programming Languages: Java, Typescript, SQL, PL/SQL, Golang
Frameworks & Libraries: Spring Boot, Node.js, React, Next.js
Tools & Platforms: Docker, Git, Jenkins, Postman, GitHub, Kubernetes
Databases: PostgreSQL, Redis, Elasticsearch
Soft Skills: Communication, Teamwork, Adaptability, Problem Solving

WORK EXPERIENCE

Tata Consultancy Services

Bengaluru, India

Fullstack Developer / Full-time

Dec 2024 – Present

- Optimized frontend performance and reduced server load by architecting a global data-caching layer using React Context API, consolidating 8 redundant page-level API calls into a single application-wide request.
- Engineered serverless email schedulers using AWS Lambda and Amazon EventBridge to automate daily operational reporting and real-time system alert dispatches.
- Designed a centralized Event Orchestrator using Spring Boot, Spring Integration, and Kafka, reducing service-to-service calls by 60% and duplicate transformations by 45%.
- Implemented an Event Reconciliation Module using ELK to auto-detect missing or partial events, improving recovery from 0% manual handling to 85% automated remediation.
- Built a fault-tolerant resilience layer with intelligent retries and fallbacks, reducing incident-driven downtime by 35% during DB or Kafka outages.
- Accelerated large-scale ETL processing via Remote Batch Partitioning across JVMs (Spring Batch + Kafka), cutting execution time from 4 hours to under 1 hour in production.
- Technologies / Skills Used:** Java, Spring Boot, React.js, Kafka, ELK, Redis, AWS

PROJECTS

SnapGo | GitHub | Go (Golang), P2P Gossip, Consistent Hashing, Concurrency

- Architected a masterless, decentralized P2P distributed LRU cache in Go featuring Gossip discovery and Consistent Hashing (50 virtual nodes) to ensure uniform data distribution with zero SPOF.
- Implemented an $O(1)$ thread-safe LRU eviction engine using `sync.RWMutex`, doubly linked list, and active background TTL sweeping via ticker goroutines.
- Engineered a Double-Checked Locking read path to eliminate hot-key contention, achieving **13,139+ req/s** read throughput (6.4ms median latency) and **1,877+ req/s** write throughput.
- Prevented proxy broadcast storms via internal routing flags and verified fault tolerance through chaos testing, ensuring dynamic ring re-balancing and zero data loss on node dropout.

ApexLedger | GitHub | Java, Spring Boot, Kafka, PostgreSQL, Redis

- Architected a high-throughput core banking backend using Spring Boot microservices to process asynchronous, event-driven financial transactions.
- Integrated Apache Kafka for streaming transaction events with strict sequencing, loose service coupling, and fault-tolerant message delivery.
- Implemented low-latency caching and distributed locking in Redis to prevent double-spending and race conditions during concurrent account operations.
- Developed resilient background workers supporting retries, de-duplication, and exactly-once task execution, processing over 1K simulated events in local benchmarks.

EDUCATION

Heritage Institute of Technology

Kolkata, India

B.Tech in Electrical Engineering, CGPA: 8.52

Jul 2020 – Jun 2024

AWARDS & ACHIEVEMENTS

LeetCode: Solved 800+ problems, Rating 1691