

Nikhil Sai

+91-9438002199 | 2019198@iiitdmj.ac.in | [linkedin.com/in/niku-419](https://www.linkedin.com/in/niku-419) | github.com/niku419

EDUCATION

PDPM IIITDM, Jabalpur, MP

Bachelor of Technology in Electronics and Communication Engineering

Jul 2019 – Jun 2023

GPA: 8.0

EXPERIENCE

Software Development Engineer II, EquityDataScience

Jan 2024 – Present

- Hardened Node.js/TypeORM + PostgreSQL (Aurora/RDS Proxy) against connection-pool exhaustion via idempotent streaming cleanup, bounded connection release, session-state restoration, and transaction timeouts, stopping pooled leaks and client-aborted streams from cascading into instance-wide failures.
- Designed graceful degradation for database overload by rebuilding ECS readiness checks to distinguish pool saturation from database failure and adding per-pool, method-gated load shedding with 503 + Retry-After; shed ~2,500 requests in a live burst while preserving in-flight work, recovering without restart.
- Streamlined secure portfolio ingestion with presigned S3 URLs and multithreaded parsing/validation, reducing processing time for large portfolio uploads by ~60% with clear user-facing error reporting.
- Automated report delivery end to end with AWS Step Functions and Fargate orchestrating headless Puppeteer workflows, generating 1,000+ reports/day while cutting manual effort by 90%.

Software Development Engineer Intern, Amazon

Jan 2023 – Jun 2023

- Architected and provisioned scalable stress-testing infrastructure using AWS CloudFormation (TypeScript/YAML), generating production-scale traffic of up to 1,000 TPS per host to benchmark system performance, identify bottlenecks, and drive targeted optimizations.
- Onboarded clients across 2 regions (FE and NA) with region-specific infrastructure, led migration from deprecated address-upload clients to a new system, and improved reliability through S3-based failed-message handling to reduce DLQ pressure and Sev-2 alarms.

Full Stack Developer, Logicwind Technologies

Jan 2022 – Apr 2022

- Shipped user-facing features for production web applications in Next.js during a 4-month engagement, improving responsiveness and usability.

Full Stack Developer, HIDS Technologies

Oct 2021 – Dec 2021

- Delivered a full-stack MERN application within a 3-month timeline, implementing real-time availability tracking and CI/CD pipelines for faster, more reliable releases.

PROJECTS

4bid — Real-Time Auction Platform | *Java 25, Spring Boot, Kafka, PostgreSQL, Redis, Angular*

Jan 2026 – Present

- Built an event-driven bidding pipeline that returns HTTP 202 with a pre-generated bid ID, then processes asynchronously through Kafka partitioned by auction ID into a row-locked PostgreSQL write, with idempotency keys making redeliveries no-ops and per-auction ordering guaranteed under concurrency.
- Engineered fault tolerance using Resilience4j circuit breakers, retry-with-backoff, a dead-letter queue with an operator replay endpoint, and a Redis distributed lock settling each expired auction exactly once across instances; Redis Pub/Sub fans live updates to a Node.js WebSocket gateway, degrading to REST reconciliation so real-time failure never fails a bid.
- Exposed the platform to LLM clients via an MCP server (14 tools) and an embedded AI chat widget; documented 11 ADRs and covered the async paths with Testcontainers integration tests.

Pulse — Self-Hosted Observability Platform | *Java 25, Spring Boot, Kafka, ClickHouse, Redis*

June 2026 – Present

- Designing a multi-tenant telemetry backend that speaks OpenTelemetry (OTLP) natively as its ingest API, so any application in any language integrates by repointing its exporter — no platform-specific SDK required, and first-party SDKs stay thin configuration wrappers rather than proprietary instrumentation.
- Specified an ingest spine that keeps storage I/O off the hot path: authenticated OTLP requests buffer into Kafka partitioned by application ID, then at-least-once consumers batch-insert into ClickHouse (columnar, TTL retention, pre-aggregated rollups), so a storage outage costs ingest lag instead of telemetry.
- Implemented the control plane: JWT operator auth plus application registration issuing SHA-256-hashed API keys with dual-key rotation grace windows, per-app event quotas and retention overrides; roadmap sequenced logs → traces → metrics → live tail → alerting, each phase gated on its own acceptance test.

TECHNICAL SKILLS

Languages: Java, TypeScript, JavaScript, Python, C/C++

Backend: Node.js, Spring Boot, Kafka, OpenTelemetry, GraphQL, REST APIs

Frontend: Angular, React

Cloud & DevOps: AWS (Lambda, Fargate, S3, Step Functions, SNS), CloudFormation, Docker, CI/CD

Databases & Caching: PostgreSQL, DynamoDB, ClickHouse, Redis

Core Concepts: Data Structures & Algorithms, OOP, Distributed Systems, System Design