Nikhil Gautam

Backend Software Engineer | Java, Distributed Systems and Search | Nearly 6 Years of Experience
nikhilgautam1729@gmail.com | linkedin.com/in/nikhilgautam96 | nikhil-gautam.com

SKILLS

Languages and Paradigms: Java, SQL, TypeScript, Java concurrency, reactive programming
Backend: Spring (Boot, Cloud, Batch, Security, Data JPA), NestJS, microservices, multitenancy
Distributed Systems: OpenSearch (hybrid and vector search), Kafka, event-driven architecture, CQRS read models
Data and Cloud: PostgreSQL, Oracle, Redis, Amazon Athena, AWS (ECS, Batch, S3, CloudWatch)
Testing: JUnit 5, Mockito, Karate, WireMock, Pact, PIT, Playwright, JMeter
DevOps and Observability: Docker, Terraform, Flyway, Jenkins, OpenTelemetry, Splunk, ELK, New Relic, PagerDuty
Frontend: Angular, React, RxJS

EXPERIENCE

Senior Software Engineer, Wissen Technology (Client: Capital One) – Bengaluru Dec 2025 – Present

- Built global search from inception for Capital One's multi-tenant risk and controls SaaS platform across 15 asset types (controls, requirements, FMEAs, risk issues); it became leadership's primary way to find and preview assets
- Implemented hybrid ranking in Spring Boot by running OpenSearch keyword and k-NN vector queries in parallel, normalizing each result list and fusing scores with config-driven 70:30 semantic-to-lexical weights, selected over 60:40 and 65:35 in product-owner relevance reviews
- Designed a recursive boolean filter language for the Search API (nested AND, OR and NOT; 6 comparison operators; nested fields) and an Aggregate API returning per-asset-type counts through OpenSearch aggregations, with guardrails on nesting depth, filter size and query length
- Cut Search History P99 latency from 800 ms–2 s to 80 ms with PostgreSQL indexes shipped as Flyway migrations to every tenant; held Search API P99 under 40 ms by tuning OpenSearch mappings, analyzers and filters
- Developed a CQRS read model indexing 100K+ documents in near real time from Kafka (sized for 1M+), reconciling missed events and out-of-band DB changes, with zero-downtime Spring Batch reindexing via aliases
- Contributed to 45% faster backend-for-frontend (BFF) responses by replacing per-document fan-out calls in 2 NestJS APIs with concurrency-limited RxJS batching, then one payload-driven POST endpoint
- Interviewed 30+ candidates with 4–8 years of experience (DSA, SQL, Java concurrency, OOP), making hire calls at about a 5% pass rate; mentored a junior engineer through onboarding, code reviews and debugging

Technologies: *Java, Spring Boot, Spring Batch, Spring Security, OpenSearch, Kafka, PostgreSQL, Flyway, Redis, NestJS, RxJS, Angular, Playwright, JMeter, Terraform, AWS (ECS, CloudWatch), OpenTelemetry, Splunk, ELK (Kibana), New Relic, PagerDuty*

Senior Analyst, Accenture (Client: Barclays) – Pune Jul 2024 – Oct 2025

- Implemented Under-18 onboarding and fraud detection journeys for retail banking, with downstream fraud service integration, in a new Spring Boot gateway layer during a migration off a 16-op-code monolithic gateway
- Parallelized sequential blocking downstream calls with CompletableFuture on bounded thread pools, so journey latency was bounded by the slowest call instead of the sum of all calls
- Added Pact contract tests to protect service boundaries during the migration, plus Karate, WireMock and PIT mutation tests; migrated JMock tests to JUnit 5 and Mockito, reaching 95% coverage (JaCoCo)
- Delivered a React proof of concept for a Tesco Bank onboarding journey (3 pages, domain-driven micro-frontends), demoed to the product owner, VP and directors

Technologies: *Java, Spring Boot, Spring Cloud, Oracle, JUnit 5, Mockito, Pact, Karate, WireMock, PIT, JMeter, React*

System Engineer, Tata Consultancy Services (Client: New York Life) – Kolkata Feb 2021 – Jun 2024

- Built Spring Batch pipelines for multi-million-row insurance, tax and regulatory reports, using chunked, partitioned steps and cursor reads to keep memory flat; ran them via AWS Batch job queues on ECS
- Developed insurance calculation and reporting logic in Java over an S3 data lake queried through Athena, with chunk-level transactions so failed jobs resumed from the last committed chunk
- Created Apache Camel routes for file ingestion, JSON, XML and CSV transformation, PGP encryption and SFTP delivery of regulatory reports stored on S3; scheduled workflows in Stonebranch with automated retries
- Wrote multi-stage Dockerfiles for batch jobs; automated CI/CD with Jenkins and UDeploy

Technologies: *Java, Spring Batch, Apache Camel, Athena, AWS Batch, S3, ECS, Stonebranch, Docker, Jenkins, UDeploy*

EDUCATION

Shri Shankaracharya Institute of Professional Management and Technology Oct 2020
B.Tech. in Computer Science and Engineering

- CODEREX 2019 winner (1st of 200+ participants); CodeChef 3-Star