

Neeraj Sharma

Toronto, ON | neerajsharma.ca | LinkedIn

Professional Summary

SOFTWARE ENGINEER II | PYTHON & JAVA · REACT · REST APIs · DISTRIBUTED DATA · FINTECH

Software engineer with 6 years shipping production features end to end in Python and Java. Works across both stacks daily on a multi-tenant financial SaaS — a Python accounts-receivable application and a Java/Spring cash-application microservice suite — owning ingestion pipelines, relational schema evolution, REST API contracts, and the React data-grid surfaces those APIs serve. Strongest in data integrity, query-layer performance, and diagnosing production incidents escalated by named enterprise tenants.

Core focus: REST API design · OpenAPI contracts · idempotent ingestion · deterministic hashing · EAV-to-relational normalization · React data grids and server-side query compilation · connection-pool and failover diagnosis · field-level audit history · service decoupling · CI/CD quality gates

Experience

Serrala | Software Engineer II (SDE II), Alevate AR & CashApp (Sep 2025–Present) · Toronto, ON

- **Rebuilt deduction CSV ingestion** to stream files directly from AWS S3 via boto3, removing the web-tier FileStorage dependency and retaining uploaded files for audit — decoupling imports from HTTP request lifetime.
- **Standardized deterministic record hashing** on (Reference, Class, Cheque, Date) to stop duplicate deductions across re-uploads, then found legitimate line items colliding on those fields and specified a CSV row-number extension to the key.
- **Promoted `Class` and `Account Manager`** out of the polymorphic custom-fields (EAV) store I had earlier extended to deduction types and statuses, into dedicated indexed PostgreSQL columns while preserving API and UI compatibility — enabling database-level filtering the key-value model could not support.
- **Built field-level audit history** capturing exact old/new values for modified fields plus record-creation events across deduction and dispute entities, using ORM lifecycle hooks and explicit emission on the bulk-update path those hooks do not intercept.
- **Driving resolution of production `MySQL --read-only (1290)` write failures** on AWS Aurora, where pooled connections survive a reader/writer failover and break Stripe customer-ID updates and user-settings writes — separating read/write engines and adding pool liveness checks with retry-on-failover.
- **Own the deductions module end to end — REST API through React grid** — restored bidirectional sorting across five columns, fixed composite multi-filter aggregation, added min/max amount range search, shipped CSV export, resolved a column-chooser crash in the listing view, corrected source-date and negative-amount handling for accounting accuracy, and own the module's regression suite.
- **Eliminated a recursive role-update loop** emitting 4+ GB/day of Loggly error spam from `create_or_update_user_role_for_ar_rep`, using traceback analysis and set-based identity comparison — restoring usable log signal and cutting ingestion cost.
- **Modernized the Java/Spring CashApp microservices** — replaced hand-written RestTemplate calls with OpenAPI-generated typed clients, retired duplicated model classes for shared DTOs, removed a proxy controller coupling PostingRuleService to ConditionsService, and capped @Cacheable TTLs at two minutes to prevent stale financial reads.
- **Hardened build and deploy hygiene across Python and Maven repos** — fixed 27 Lombok constant-case compilation failures, applied Spotless Palantir and pre-commit Black/Flake8/isort gates, migrated deprecated GitHub Actions workflows, and tuned ArgoCD/Kubernetes memory limits for Currency Service.

Dailzer | Software Engineer (Nov 2022–Sep 2025)

- **Built Python microservices** for automated expense categorization and spending recommendations, integrating LLM models behind REST interfaces consumed by the core platform.
- **Led backend and AI integration** for a GPT-powered expense tracker using Java, Spring Boot, and OpenAI GPT — delivering the feature set behind a **30% increase in user engagement**.
- **Designed a microservices architecture** decoupling ingestion from processing with Kafka and Docker — **cutting API latency 40% under peak traffic**; secured services with Spring Security and JWT.
- **Optimized data design** across PostgreSQL and a NoSQL store, applying CAP trade-offs on read-heavy paths — **reducing query latency 25%**; automated deployments with Docker Compose and CI/CD, and mentored 2 junior engineers.

Larsen & Toubro (L&T) Technology Services | Software Development Engineer (Apr 2018–Dec 2019)

- **Delivered enterprise Java/Spring Boot applications supporting 100K+ active users**, leading a team of 4 engineers.
- **Improved performance 30% and cut latency 40%** through multithreading, JVM garbage-collection tuning, and query optimization; built Jenkins CI/CD enabling bi-weekly releases with zero rollbacks.

Skills

Backend & APIs: REST API design · OpenAPI/Swagger · microservices · DTO contracts · service decoupling · idempotency · schema migration · JWT · OAuth 2.0 · system design

Python: Flask · FastAPI · SQLAlchemy · Alembic · Pydantic · boto3 · Black/Flake8/isort

Java: Spring Boot · Spring Security · Hibernate/JPA · Lombok · Maven · OpenAPI codegen

Data: PostgreSQL · MySQL/Aurora · MongoDB · schema design · indexing · query optimization · connection pooling · EAV-to-relational normalization

Cloud & DevOps: AWS (S3, RDS Aurora, EC2, Lambda) · Docker · Kubernetes · ArgoCD · GitHub Actions · Jenkins · CircleCI · Git · Linux/Unix

Quality & observability: end-to-end and integration testing · pre-commit hooks · Spotless Palantir · SonarQube · Loggly · incident response · code review · Agile/Scrum

Frontend: React · TypeScript · JavaScript · data-grid and list-view engineering · component state debugging · Tailwind

Languages & other: Java · Python · SQL · TypeScript · C/C++ · Kafka · OpenAI/Gemini LLM integration

Education

Master of Management — York University, Toronto (2022) · **PG Diploma, Advanced Computing** — C-DAC (2018) · **B.Tech** — KIIT University (2017)