

Summary

Software engineer with 3+ years of experience building scalable backend systems and frontend interfaces across e-commerce, EdTech, and aerospace domains. Proficient in backend development with modern Java frameworks, complemented by hands-on experience in frontend engineering using component-based UI libraries. Skilled in working with both relational and NoSQL databases, designing secure, cloud-native architectures, and implementing automated CI/CD pipelines. Experienced in deploying, monitoring, and troubleshooting production systems on AWS, with a proven ability to diagnose and resolve critical system failures under pressure. Leverages AI-assisted engineering practices and agentic workflows, including prompt-driven development and LLM-based tooling, to accelerate feature delivery, improve code quality, and support debugging and test generation in complex distributed systems. Familiar with the full software development life cycle and Agile delivery.

Technical Skills

Languages: Java (8//17/21), Python, TypeScript, JavaScript **Backend & Frameworks:** Spring Boot, Flask, Django, Node.js, Express.js, REST APIs, GraphQL
Databases: PostgreSQL, MySQL, MariaDB, MongoDB, Redis **Distributed Systems:** Apache Kafka, Event-Driven Architecture, Microservices
AI-Augmented Engineering: AI-assisted development (Cursor, GitHub Copilot), LLM API Integration, Agentic Workflows, Retrieval-Augmented Generation
Dev Tools & Testing: Git, Jenkins, GitHub Actions, Selenium, Jest, Swagger
Cloud & Infrastructure: AWS (EC2, S3, RDS, Lambda, CloudFront), Docker, Kubernetes, Apache Spark

Education

Purdue University M.S. in Computer and Information Technology	West Lafayette, IN	Aug. 2022 - Dec. 2024
North China Electric Power University B.E. in Software Engineering	Beijing, China	Sep. 2018 - Jun. 2022

Work Experience

American Express Software Engineer II, Sunstone Team	Sunrise, FL	June 2026 - Present
---	-------------	---------------------

American Express is a global financial services company serving consumers and businesses worldwide. The Sunstone team develops backend and data-platform services that maintain and distribute customer demographic and legal entity information across enterprise systems. As a part of the engineering team, I build asynchronous processing workflows, real-time event integrations, data-quality analysis tools, automated tests, and CI/CD pipelines that support reliable customer-data propagation and production delivery.

- Developed asynchronous customer-data processing components in **Java and Spring Boot**, using **Vert.x Verticles** to execute non-blocking request workflows and propagate eligible customer-record updates across distributed services.
- Implemented field-level validation, conditional update logic, survival check, and cross-service data mapping to prevent missing or ineligible demographic values from contaminating valid customer data.
- Evolved shared Java domain models and service contracts across multiple repositories, maintaining compatibility between request-processing, data-access, and downstream integration services.
- Designed and built a team-facing data-quality analysis platform in **Python** that retrieves correlated event data from **OpenSearch and Couchbase**, compares customer fields across processing stages, and visualizes field-level differences, before-and-after values, and affected events for repeatable regression analysis.
- Built and deployed part of a real-time customer-demographic event pipeline using **Apache Kafka**, implementing event subscriptions, schema contracts, payload parsing and mapping, idempotent processing, and downstream delivery across distributed services.
- Added unit and integration tests and expanded **Cucumber regression suites** with cases covering new customer-data features and edge conditions. Used **Postman** for end-to-end API testing and supported Product UAT before phased production releases.
- Built and maintained **GitHub Actions** workflows for application builds, automated testing, pipeline orchestration, and deployment through staged and production environments.
- Monitored production services with **OpenSearch and Kibana**, tracking 4xx and 5xx error rates, average processing time, and p90, p95, p99, and p99.9 latency to validate releases and investigate anomalous system behavior.
- Used **GraphQL** interfaces to support typed data access and cross-service integration while coordinating schema and payload changes with data-platform and downstream engineering teams.
- Applied AI-assisted engineering workflows with **Codex** to trace distributed code paths, review implementation designs, expand test coverage, diagnose errors, and produce technical documentation while retaining responsibility for implementation and release quality.

eBay Backend Engineer, eBay For Charity	Remote	Aug. 2025 - Jan. 2026
--	--------	-----------------------

eBay for Charity is a global initiative to streamline charitable giving by embedding donation workflows into the eBay commerce platform. The Charity team is responsible for developing the whole donation pipeline to accommodate the donation workflow from buyer/seller to the charity organizations. As part of the backend engineering team, I worked on modernizing donation infrastructure, ensuring financial accuracy, regulatory compliance, and system resiliency across global markets.

- Migrated legacy charity donation workflows to an automated **Donation Netting** architecture, enabling upfront deduction of pledged funds from seller payouts and supporting post-checkout reconciliation during service outages across US, UK, CA, and FR markets.
- Designed and implemented backend donation pipelines for **Give at Checkout** using **Spring Boot**, **RESTful APIs**, **Jackson serialization**, and **YAML-based global configuration**, integrating with regional donation partners including **PayPal Giving Fund (US)** and **Benevity (UK/CA/FR)**.
- Built end-to-end **forward and reverse money movement (FMM/RMM) flows** across distributed systems (OPS, NCS, SVLS, e4coms, e4cdash, e4cpayout) using **Kafka-driven event orchestration**, **Oracle/Vertica persistence**, and **immutable audit logging** to ensure financial accuracy and traceability.
- Implemented efficient queries and indexing strategies to support large-scale donation workflows. Managed data consistency and lifecycle states for charity organizations, sellers, and item-level donation records using **Oracle (SQL)** and **MongoDB (NoSQL)** hosted on **eBay Cloud**, optimizing for lookup performance and transactional integrity across asynchronous pipelines.
- Migrated some traditional database querying with the **Unified Listing Platform (ULP)** by consuming **Federated GraphQL (FGQL)** APIs, adhering to schema governance requirements, and correctly handling buyer-context-aware versus context-agnostic listing data access patterns.
- Instrumented system observability using **Prometheus**, and **Elasticsearch + Kibana** on **eBay Cloud**, exposing custom metrics and application-level probes for system health, event throughput, and back-pressure alerts.
- Developed a comprehensive **regression testing** framework for mission-critical donation pipelines, integrated into **Jenkins CI/CD**. Utilized **WireMock** for **service stubbing** and **Gson** for mock payload de/serialization, covering edge cases such as delayed payouts, partial refunds, and outage-triggered backfills, significantly reducing production regressions.
- Managed end-to-end deployment lifecycle via **Jenkins declarative pipelines**, **automating build, test, Docker image packaging**, and **ECS task definition updates** for staging and production environments.
- Leveraged **AI-assisted development tools** (e.g., GitHub Copilot, Cursor, Claude) to accelerate backend feature implementation, generate service scaffolding, and detect edge-case regressions early in high-throughput financial pipelines.
- Applied Cursor-based **agentic workflows** and **Vibe Kanban** to support architectural refactoring and complex project/issue management across distributed donation services. Utilized **structured prompting** and context-aware workflows (e.g., **Model Context Protocol**) to best sync with AI Agent and guide large-scale refactoring, unit test generation, GraphQL Schema Design, and code review across Kafka-driven microservices.
- Actively participated in **Agile Scrum rituals**, including sprint planning, daily stand-ups, backlog grooming, and retrospectives.

NASA Full Stack Engineer, School of Mechanical Engineering

West Lafayette, IN Jan. 2024 - Aug. 2025

NASA RETHi-DRDS is a systems engineering initiative focused on resilient space habitat systems, where I contributed as a full-stack developer building scalable, real-time telemetry platforms deployed on AWS. As part of the systems engineering software team, I contributed to both frontend and backend development for real-time data aggregation, telemetry processing, and cross-platform communication.

- Designed and implemented reactive microservices using **Java 17**, **Spring Boot**, and **Spring WebSocket** to ingest and broadcast high-frequency telemetry streams. Achieved low-latency, bi-directional communication between telemetry collectors and mission dashboards.
- Built **RESTful APIs** using **Spring MVC** and **Spring Data JPA/Hibernate**, powering mission dashboards, configuration interfaces, and subsystem diagnostics tools. Persisted time-series telemetry data in **PostgreSQL** with **time-based partitioning**, improving query throughput for datasets exceeding 10K records/day.
- Applied **JWT-based** authentication with **Spring Security** to enforce role-based access control for engineers, analysts, and simulation subsystems.
- Integrated **Apache Kafka** to stream mission telemetry events across subsystems, with topic partitioning and consumer groups enabling real-time alerting and downstream analytics.
- Developed rich, responsive interfaces using **React.js** and **Tailwind CSS**, supporting real-time mission control workflows. Utilized **React Hooks** (useState, useEffect, useRef) for lifecycle-aware data rendering and performance tuning.
- Structured dynamic UI flows with **React Router**, enabling multi-view transitions between dashboards, telemetry graphs, and simulator control panels.
- Managed global state using **Redux Toolkit** to maintain global telemetry snapshots and alerting states, supporting shared session info, data subscriptions, and mission timelines across components.
- Streamed backend Flux/Mono responses to the browser using **Axios + RxJS** pipelines, maintaining backpressure compatibility and live update flows.
- Containerized services with **Docker** and deployed to **AWS ECS**, using **EC2** for compute hosting, **S3** for telemetry storage, **RDS PostgreSQL** for mission data, and **IAM** for secure resource access.
- Configured real-time monitoring and observability using **Spring Boot Actuator**, **Prometheus**, and **Grafana**, enabling health checks, latency tracking, and automated alerts via CloudWatch and Slack.
- Participated in **Agile** ceremonies with mission scientists and system engineers, helping translate complex telemetry workflows into user-friendly interfaces and robust backend systems.
- Employed **Cursor** to streamline development in a multi-language telemetry stack, using in-context LLM suggestions for real-time refactoring, API contract drafting, and protocol debugging across Java, Python, and Go services—enabling rapid “vibe coding” without sacrificing architectural rigor.