Navya Jakkula
Mobile: (469) 630 1465 | Email: navya292700@gmail.com
LinkedIn | GITHUB |

SUMMARY:

- Java Full Stack Developer with 7+ years of experience delivering scalable enterprise applications across finance, healthcare, retail, media, and e-commerce using Java, Spring Boot, React.js, Angular, Kafka, and AWS/Azure.
- Modernized legacy systems, led FedEx logistics platform migrations, and transformed monolithic apps into cloud-native microservices; implemented event-driven architectures with Kafka, RabbitMQ, and JMS.
- Secured applications using Spring Security, OAuth 2.0, JWT, SSO (OKTA, Azure AD), RBAC, and MFA, and optimized database performance through DB2/PostgreSQL to Azure SQL migrations and advanced query tuning.
- Built CI/CD pipelines with Jenkins, Docker, and Kubernetes; deployed solutions on AWS (EKS, Lambda, S3) and Azure (AKS, Event Hubs); automated ETL processes with Spring Batch and Apache POI.
- Delivered high-quality solutions with JUnit, Mockito, Cypress, Gatling, and real-time monitoring via Splunk; practiced Agile (Scrum/Kanban) using JIRA, Git, and Confluence to ensure on-time, collaborative delivery.

TECHNICAL SKILLS:

Programming Languages	Java (8-21), J2EE, Python, SQL, PL/SQL, C, C++
Frameworks	Hibernate, JPA, Microservices, Servlets, Spring MVC, Spring Boot, Spring Web-Flux, Log4J.
Web Service Technologies	SOAP, REST (JAX-RS), XML, JSON, Graph-QL.
Messaging Technologies	Kafka, Rabbit MQ
Application Servers	WebLogic, JBoss, Tomcat, WebSphere.
Build Tools	Ant, Maven, Gradle.
Cloud Computing	AWS (EBS, EC2, S3, RDS, SNS, SQS, Lambda, SAM), Azure, GCP.
Frontend Technologies	JavaScript, Python, Typescript, Bootstrap, HTML5, CSS3, React.js, Angular, jQuery
Databases	MongoDB, Oracle, DB2, MySQL, PostgreSQL.
IDEs and Development Tools	Eclipse IDE, IntelliJ, WebStorm, Visual Studio, JIRA
Testing	Junit, Mockito, Jasmine, Jest, karma,
Source Control	Maven, Gradle.
Development Methodologies	Waterfall, Agile, Scrum, TDD
Caching Technologies	Redis, Hazel cast, Memcached
Operating Systems	Windows, Unix, Linux, Mac
DevOps & CI/CD	GitLab CI/CD, Jenkins, Docker, Kubernetes, Terraform, GitHub Actions

WORK EXPERIENCE:

Sr. Software Engineer – FedEx
FedEx Rewards (Ship & Earn) — Cloud-Native Loyalty Platform
January 2024 - Present

- Architected cloud-native **microservices** in **Java 17/Spring Boot** using **DDD** with bounded contexts (Member, Offer, Points Ledger, Redemption, SLA Credit).
- Designed **versioned REST APIs** with **OpenAPI/Swagger**, request validation, and a consistent error codex, enforced **idempotency keys** on enroll/earn/redeem to stop duplicates.
- Ingested shipment events (**Delivered**, **SLA_BREACH**) from **Kafka/Amazon MSK**; normalized payloads and published domain events (**Points Accrued**, **Credit Approved**) for downstream consumers.

- Implemented a rule-based **Offer Engine** using precomputed lookups and **Redis** caching to keep evaluation latency <20ms during month-end spikes.
- Modeled a **double-entry points ledger** in **PostgreSQL** (partitions, composite indexes, ACID transactions) to guarantee accurate balances and reversals.
- Coordinated cross-service updates with **Saga + Outbox** patterns to avoid dual-write inconsistencies and enable safe retries with **DLQ** and replay.
- Added selective **Spring WebFlux** for I/O-heavy aggregations to external services, improving throughput and reducing thread blocking.
- Secured APIs with **Spring Security**, **OAuth2/JWT**, and **RBAC**; masked PII in logs and enforced method-level authorization for finance/admin operations.
- Integrated third-party **gift card** provider via signed REST and verified **webhooks** with signature checks and replay protection, stored fulfillment receipts for audit.
- Externalized configuration via **Kubernetes Config-Maps/Secrets** and **AWS Secrets Manager**; service discovery through **Kubernetes DNS** behind **API Gateway/ALB.**
- Wrote **JUnit 5/Mockito** unit tests and **Test containers** integration tests (PostgreSQL, Redis, Kafka); defined **k6/JMeter** load tests and API SLOs.
- Automated schema changes with **Liquibase** and backward-compatible migrations; enforced gating in **Jenkins** CI pipelines.
- Containerized services with **Docker**; deployed to **AWS EKS** using **Helm** (readiness/liveness probes, **HPA**, Pod Disruption Budgets, **blue/green** rollouts
- Monitored and logged microservice performance using tools like Prometheus, Grafana, and ELK stack (Elasticsearch, Logstash, Kibana).
- Delivered **Angular 17** portals (Member + Ops) in **TypeScript** with **NgRx Store/Effects** and **RxJS** streams for near real-time balances and redemption status.
- Built complex **Reactive Forms** (custom validators, conditional steps, autosave/draft) for offer activation, redemption, and **SLA credit** submissions.
- Standardized UI with **Angular Material** + SCSS design tokens; enforced **WCAG 2.1 AA** accessibility; documented components in Storybook.
- Hosted the SPA on **Amazon S3** and served via **CloudFront** with cache-busted assets and environment runtime configs; images stored in **ECR** for cluster pulls.
- Hardened cloud security with private subnets, least-privilege **IAM**, **KMS** encryption, TLS everywhere, and audited access policies.
- Set actionable alerts for consumer lag, ledger mismatches, webhook failures, and SLO breaches; published runbooks for **DLQ replay**, cache invalidation, and rollback.

Environment: Agile Methodology, Java 17, Spring Boot 3.x, Spring Security (OAuth2/JWT), Spring Data JPA, Spring Web-Flux, Microservices, REST, PostgreSQL (AWS RDS), Redis, Apache Kafka (Amazon MSK), Angular 17, TypeScript, HTML5, SCSS, Docker, Kubernetes (AWS EKS), Helm, Amazon ECR, S3, CloudFront, API Gateway/ALB, AWS Secrets Manager, KMS, CloudWatch, Prometheus, Grafana, Splunk, Jenkins, Git, Maven, JUnit 5, Mockito, Test containers, SonarQube, IntelliJ.

Sr. Software Developer- REI Systems
Client: US Food and Drug Administration
November 21 – September 23

- Participated in all phases of the Software Development Life Cycle (SDLC), including requirements gathering, system design, development, testing, and deployment for efficient project delivery.
- Designed and implemented a Java rule engine to determine whether items imported to the USA required manual inspection.

- Built **Spring Boot (Java 17)** microservices and a rules engine to automate import-inspection decisions with **OpenAPI** contracts and request validation
- Re-architected the legacy UI as **Angular 17** modules with **JWT/OAuth2**, route guards, and role-based navigation for secure user flows.
- Implemented **RBAC** and tamper-evident **audit logging** across admin, analyst, and reviewer roles to meet agency compliance.
- Exposed **REST** APIs with versioning and **idempotency** on write paths to prevent duplicate adjudications under retries.
- Modeled data in **Oracle 12c/19c** using **JPA/Hibernate ORM**; optimized with **composite indexes** and **table partitioning**; automated **schema migrations** via **Liquibase.**
- Worked on **Kafka** for event-driven processing of adjudication outcomes and audit events, including **DLQ** and replay tooling.
- Externalized business rules and thresholds in the database; enabled analyst-driven updates without code deploys
- Containerized services with **Docker** and deployed to **AWS EKS** using **Helm** (readiness/liveness probes, **HPA**, PDBs).
- Hosted the **Angular** SPA on **Amazon S3** and served via **CloudFront** with cache-busted assets and environment configs.
- Automated CI/CD with **Jenkins**: build, unit/integration tests, security scans, image sign/push to **ECR**, and **blue/green** rollouts
- Implemented observability with **Open Telemetry** traces, **CloudWatch**/**Prometheus** metrics, **Grafana** dashboards, and **Splunk** logs.
- Wrote **JUnit 5/Mockito** unit tests and **Test containers** integration tests; added **Katalon (Groovy)** suites for smoke/regression and API health checks.
- Hardened security with least-privilege **IAM**, **KMS**-encrypted secrets (**AWS Secrets Manager**), TLS, CSP, and PII field masking.
- Optimized front-end performance using **Reactive Forms**, lazy routes, **OnPush** change detection, trackBy, and virtual scroll.
- Collaborated in Agile sprints; led code reviews and enforced **SonarQube** quality gates to maintain reliability and release cadence.

Environment: Java 17, Spring Boot 3.x, Spring Security (OAuth2/JWT), Spring Data JPA (Hibernate), Microservices, REST/OpenAPI, Oracle 12c/19c, PL/SQL, Liquibase, Apache Kafka, Angular 17, TypeScript, HTML5, SCSS, RxJS, Angular Material, Docker, Kubernetes (AWS EKS), Helm, Amazon ECR, S3, CloudFront, AWS Secrets Manager, CloudWatch, Agile Methodology, Prometheus, Grafana, Splunk, Jenkins, GitLab CI, Git, Maven, JUnit 5, Mockito, Test containers, Jasmine, Karma, SonarQube, IntelliJ IDEA.

Sr. Frontend Developer - BankUnited
Credit Card Installment Service
January 2020 – October 2021

- Developed **Spring Boot (Java 17)** microservices for **Eligibility**, **Installment Plan**, **Repayment Schedule**, and **Ledger**, exposing **versioned REST APIs** with **OpenAPI** contracts, validation, and a clear error model.
- Implemented interest/fee and **tenor schedule** logic (promo windows, rounding rules, early-payoff) with **idempotency keys** so retries never create duplicate plans.
- Modeled data in **MariaDB** using **Spring Data JPA/Hibernate** (composite indexes, projections, pagination) and shipped **Liquibase** migrations for safe, backward-compatible releases.
- Added **Redis** caching for eligibility rules and product catalogs, cutting p95 decision latency during statement spikes and reducing database load.
- Secured services with **Spring Security (OAuth2/JWT)** and **RBAC**; masked PAN/PII in logs and enforced audit trails on plan create, reschedule, and payoff.
- Integrated with the bank's **card processor / payment gateway** via signed REST and verified **webhooks** (HMAC signatures, replay protection) to keep schedules and ledger entries in sync.
- Published domain events (**Plan Created, Payment Applied, Plan Closed**) to **Apache Kafka** for **asynchronous** updates to statements, collections, and analytics.

- Instrumented **Open Telemetry** traces and **request correlation IDs** across APIs, Kafka consumers, and gateway callbacks to speed production debugging.
- Integrated backend with **Angular 14** UI: **HTTP Client** interceptors for JWT and correlation IDs, **Reactive Forms** with custom validators (eligibility, amount limits, tenor windows), and clear, accessible errors.
- Optimized the UI with **On Push** change detection, **lazy routes**, and reusable components (schedule preview, total-cost disclosure, early payoff).
- Wrote **JUnit 5/Mockito** unit tests and **Test containers** integration tests (MariaDB, Redis, Kafka); added **Jasmine/Karma** and **Cypress** for component and end-to-end journeys; enforced coverage gates in CI.
- Deployed apps on **Pivotal Cloud Foundry (Tanzu Application Service)** using build packs and manifests; executed **blue/green** cutovers for zero-downtime releases; externalized config via **Spring Cloud Config** and **CredHub/Vault**.
- Automated CI/CD with **Jenkins/GitLab CI** (build → tests → SAST/DAST → artifact publish → PCF deploy → post-deploy smoke tests). Monitored runtime with **Dynatrace/AppDynamics** APM and **Splunk** logs; produced actionable dashboards for API latency, error rate, and Kafka consumer lag tied to **SLO** alerts.

Environment: Java 17, Spring Boot 3.x, Spring Security (OAuth2/JWT), Spring Data JPA (Hibernate), REST/OpenAPI, Microservices, MariaDB, Liquibase, Redis, Apache Kafka, Angular 14, TypeScript, HTML5, SCSS, RxJS, Reactive Forms, JUnit 5, Mockito, Test containers, Jasmine, Karma, Cypress, Pivotal Cloud Foundry (Tanzu Application Service), Spring Cloud Config, Jenkins, GitLab CI, Git, Maven, Dynatrace/AppDynamics, Splunk, Open Telemetry.

Jr. Backend Engineer
Conquerors Software Technologies (Client: Metro-Goldwyn-Mayer - Media & Entertainment)
Payroll and Budget Management System – Financial Planning & Revenue Tracking
April 2018 – December 2019

- Participated in the full Software Development Life Cycle (SDLC), from requirements gathering to Object-Oriented Analysis and system design, ensuring well-structured and efficient application development.
- Developed a payroll and budget management system for the entertainment industry, enabling financial allocation, real-time crew tracking, and automated payroll processing for seamless operations.
- Built an interactive Angular UI with lazy loading, RxJS, and Material UI, improving usability, system performance, and user experience.
- Designed and optimized the PostgreSQL database, ensuring data consistency, indexing, and high-speed financial transactions for accurate payroll and budget management.
- Automated payroll calculations and revenue tracking across OTT, VOD, and SVOD platforms, providing real-time financial insights for better decision-making.
- Implemented AWS Lambda for serverless payroll execution, integrating with S3, DynamoDB, and SAP, ensuring Realtime data synchronization and processing.
- Deployed the system on AWS (EC2, RDS, IAM roles), ensuring scalability, high availability, and security compliance for enterprise-grade financial management.
- Developed custom SQL queries and batch processing, optimizing payroll calculations, transaction handling, and financial reporting performance.
- Integrated AWS SNS & SQS for event-driven processing, enabling real-time notifications and task queueing for seamless payroll operations.
- Secured financial data using OKTA API & Role-Based Access Control (RBAC), ensuring data privacy, regulatory compliance, and controlled user access.
- Automated CI/CD pipelines using Jenkins, Docker, and Kubernetes, streamlining seamless deployments, version rollback mechanisms, and system updates.
- Integrated Splunk for real-time system monitoring, ensuring high availability, proactive issue detection, and performance optimization.
- Conducted rigorous unit, integration, and performance testing using JUnit and Mockito, guaranteeing system stability

and reliability under high transaction loads.

Environments: Spring Framework, Spring Boot, Java 17, Microservices, Angular, Hibernate, Spring Data JPA, AWS, Apache POI, OKTA, MySQL, HTML5, Typescript, JavaScript, CSS, and Visual studio code, IntelliJ, Eclipse.

EDUCATION:

Master's in computer and information science | University of North Texas, USA.
Bachelor of Technology |JAWARLAL NEHRU UNIVERSITY, Hyderabad, India.