

Name: Navya Jakkula
Mail: navya292700@gmail.com
Phone: 4696301465
LinkedIn: <https://www.linkedin.com/in/navya-jakkula2925/>



Summary

Accomplished Senior Software Engineer with 7+ years of experience in designing, developing and optimizing high-performance, scalable, and secure web applications. Expertise in Java, Spring Boot, Spring Cloud, and Spring WebFlux. Proficient in modern web technologies like Angular, React.js, HTML5, CSS3, JavaScript, and Redux. Strong background in microservices architecture, RESTful and SOAP web services, and containerization with Docker and Kubernetes. with a strong background in building distributed systems and working with various databases, including Oracle, Cassandra, and MongoDB. Proficient in implementing RESTful services, messaging systems such as Apache Kafka and AWS SQS/SNS, and caching solutions like Redis and Hazelcast. Demonstrated experience in software security, PCI-DSS compliance, and CI/CD pipelines. Proven ability to deliver robust solutions in fast-paced environments and collaborate effectively with cross-functional teams.

Technical skills

Category	Technologies/Tools
Frontend Technologies	React.js, JavaScript (ES6+), TypeScript, HTML5, CSS3, Redux, Bootstrap
Backend Technologies	Node.js, Express.js, RESTful APIs, GraphQL, Microservices, Spring Boot, Spring Cloud, Spring WebFlux, Java, SQL
Messaging Servers	RabbitMQ, Apache Kafka, AWS SQS, AWS SNS
Web Technologies	HTML5, CSS3, JavaScript, React.js, Angular(2/4/6/8/9/14,17), TypeScript, Bootstrap, jQuery
Databases	MySQL, Oracle (10g/11g), MongoDB, Cassandra, Couchbase
Web/App Servers	Apache Tomcat, WebSphere
Caching Technologies	Redis, Hazel cast, Memcached
Architectures	Multi-Tier Enterprise Application, MVC, MVVM, Microservices
Web Services	SOAP, RESTful, JAX-RS, XML, JSON
Frameworks	Spring Framework, Hibernate, JUnit, Mockito, AngularJS
Version Control Tools	GitHub, Bitbucket
IDE/Tools	Eclipse, IntelliJ IDEA, Visual Studio, Sublime Text, WebStorm
Build and Deployment Tools	Maven, Jenkins, Docker, Kubernetes
Testing	JUnit, Selenium, Mockito
CI/CD Tools	Jenkins, GitLab CI/CD
Cloud Platforms	AWS (EC2, S3, RDS, Lambda, EKS, CloudFormation), Microsoft Azure
DevOps Tools	Docker, Kubernetes
Operating Systems	Windows, Linux, Unix, macOS
SDLC Methodologies	Agile, Scrum, Waterfall, Test-Driven Development (TDD)
Monitoring and logging	Splunk, Log4j, SonarQube

Professional Experience

Senior Software Developer

January 2024 - Present

FedEx - Memphis, TN, US

Led the Development of a cloud-native logistics optimization platform at FedEx aimed at improving shipping cost efficiency and real-time package tracking. The backend microservices were engineered using **Java**, **Spring Boot**, and **Spring WebFlux**, followed a **microservices-based architecture with integrating Apache Kafka** for asynchronous communication between pricing, routing, and shipment modules. REST APIs were secured using Spring Security and **JWT authentication** and deployed as Docker containers to AWS EKS via **Helm charts**. Delivered a responsive frontend interface using **Angular** based dashboards with RxJS and Bootstrap to visualize KPIs, route intelligence, and dynamic pricing updates for internal stakeholders. Ensured performance, fault tolerance, and regulatory compliance through **CI/CD pipelines** were built using Jenkins and GitLab CI with integrated unit testing (JUnit, Mockito) and UI validation (Jasmine, Karma). Logging and monitoring were handled through **AWS CloudWatch**, **Log4j2**, and **Splunk**, while **Redis** was implemented for caching high-frequency configuration lookups. Collaborated with cross-functional DevOps and security teams to manage IAM roles, secrets, and FedEx-compliant encryption using **AWS KMS** and **private S3 buckets**.

Backend:

- Designed and implemented highly scalable **microservices** using **Spring Boot** and **Spring Web Flux** to handle real-time pricing and routing logic based on live factors like delivery zones, urgency, weight, traffic, and weather conditions.
- Built and secured **RESTful APIs** using **Spring Security** and integrated **JWT-based authentication**, ensuring role-based access control for internal services and partner integrations.
- Developed asynchronous processing pipelines with **Apache Kafka**, enabling decoupled communication between Pricing, Notification, and Dispatch modules to support peak-hour load handling.
- Created optimized data access layers using **Spring Data JPA** and **Hibernate**, writing complex **PostgreSQL** queries for shipment history, pricing rules, and delivery estimates.
- Containerized services using **Docker** and deployed to **AWS ECS**, enabling efficient CI/CD workflows with **Jenkins** for automated testing, build, and deployment pipelines.
- Ensured backend stability by writing comprehensive **unit and integration tests** using **JUnit 5** and **Mockito**, maintaining over 90% code coverage across services.
- Documented all service APIs using **Swagger/OpenAPI**, improving team collaboration and accelerating integration with downstream services.
- Monitored system performance and errors using **Log4j** and **Splunk**, proactively addressing issues and ensuring 24/7 service availability.

Frontend:

- Developed and maintained scalable UI modules using **Angular 17**, integrating **TypeScript**, **HTML5**, and **SCSS** to build operational dashboards for pricing, route intelligence, and shipment tracking.
- Engineered reusable, component-driven UI elements like **pricing sliders**, **route maps**, **delivery status indicators**, and integrated them with backend APIs via **JWT-secured RESTful endpoints**.
- Applied **RxJS** for managing asynchronous streams and real-time updates on shipment and pricing metrics. Implemented **route guards**, **interceptors**, and **centralized error handling** to improve security and user experience across modules.
- Designed responsive layouts with **Bootstrap** and Angular best practices, ensuring optimal performance across desktop and tablet environments.
- Used **Angular CLI** to enforce modular architecture and DRY principles for scalable frontend development.

- Collaborated with UI/UX designers and product teams to translate business KPIs into functional, user-friendly dashboards.
- Wrote **unit tests with Jasmine/Karma** and participated in E2E validations during major sprint releases, improving reliability and test coverage.

Cloud and Integration:

- **Containerized** Spring Boot microservices using **Docker** and orchestrated deployments to **AWS EKS (Elastic Kubernetes Service)** for scalable, resilient infrastructure.
- Used **Helm charts** to templatize Kubernetes configurations, enabling **configurable**, secure, and repeatable deployments across dev, QA, and production environments.
- Automated infrastructure provisioning via **AWS CloudFormation**, spinning up resources like **EC2, S3, RDS, and IAM roles/policies** in line with FedEx's security compliance.
- Developed and maintained **CI/CD pipelines** using **Jenkins** and **GitLab CI**, incorporating automated build, test, artifact storage, and blue-green deployments.
- Integrated **Apache Kafka** for asynchronous, high-throughput communication between pricing, dispatch, routing, and shipment services.
- Implemented **Redis** caching for zone and weight-based pricing data, reducing load on PostgreSQL and improving API latency under high traffic.
- Deployed **AWS Lambda** functions for background tasks like audit trail logging, cache refreshes, and real-time pricing recalculations.
- Monitored infrastructure and application health using **AWS CloudWatch, Splunk, and Log4j**, with proactive alerting integrated with Microsoft Teams and PagerDuty.
- Ensured secure access to cloud resources using **IAM roles, KMS encryption, private S3 buckets, and security group configurations** adhering to FedEx policy.

Environment: Java 17, Spring Boot, Spring MVC, Spring WebFlux, Spring Security, Hibernate (JPA), PostgreSQL, Angular 17, TypeScript, RxJS, Bootstrap, Apache Kafka, RabbitMQ, Redis, Docker, Kubernetes, AWS (EC2, RDS, S3, EKS), Jenkins, GitLab CI/CD, Swagger/OpenAPI, JUnit 5, Mockito, Jasmine, Karma, Git, IntelliJ IDEA, Agile/Scrum.

Senior Software Developer

November 2021 – September 2023

REI Systems - Sterling, VA-US

Contributed to REI's cloud modernization of the **DoD Small Business Engagement Portal**, used by thousands of vendors and contracting officers. The platform digitized the full lifecycle of small business onboarding, proposal submission, tracking, and engagement analytics for defense agencies. The solution replaced legacy .NET and SharePoint systems with a secure, scalable, **Angular + Spring Boot** Microservices architecture on **AWS GovCloud**.

Backend:

- Developed secure and scalable **Spring Boot microservices** to support business registration, proposal submissions, review workflows, and audit logging.
- Used **Spring MVC** and **Spring WebFlux** to build RESTful APIs and non-blocking endpoints for high-throughput modules like search, messaging, and file intake. Implemented **Spring Security** with **JWT/OAuth2** authentication to secure APIs and enforce fine-grained access control per DoD user role.
- Wrote complex business logic in the Spring Service Layer for registration, proposal evaluation, and approval workflows; handled exceptions with `@ControllerAdvice` and implemented custom annotations like `@ValidDunsNumber`, `@RequiresAgencyRole`, and `@AuditableAction` to modularize validation, access control, and user activity auditing.

- Designed and tuned relational data models with **JPA/Hibernate**, performing advanced joins, fetch optimizations, and batch inserts in **PostgreSQL**.
- Integrated **Apache Kafka** for asynchronous communication between services, enabling event-driven workflows such as submission confirmation, user notifications, and audit trails.
- Built reusable API utilities and integrated **Swagger/OpenAPI** to document endpoints for frontend and external partner teams.
- Authored comprehensive **unit and integration tests** using **JUnit 5**, **Mockito**, and embedded test containers to validate persistence and service layers.
- Implemented **Spring Batch jobs** for nightly data sync, cleanup tasks, and status propagation across distributed services.
- Applied layered architecture principles, separation of concerns, and service decoupling patterns to enable maintainable, extensible microservices.

Frontend:

- Developed dynamic and responsive Angular 14 components for dashboards, forms, and application tracking modules using **Reactive Forms**, **RxJS**, and **Bootstrap**.
- Built configurable, reusable UI libraries and shared modules (e.g., modals, alert banners, pagination) to streamline code reuse across the portal.
- Integrated real-time notifications and updates using **RxJS Subjects and BehaviorSubjects**, enhancing user engagement and system responsiveness.
- Implemented **custom form validators** for critical fields (DUNS, SAM.gov ID, NAICS codes), improving data integrity and submission success rate.
- Applied **route guards and permission-based navigation logic** using JWT claims to restrict access for different user roles (Admin, DoD Officer, Applicant).
- Built complex **nested tab structures** and wizard-style forms with conditional rendering and dynamic field binding.
- Designed and optimized data pipelines using **Spring Cloud Data Flow** for efficient stream processing and batch jobs. Worked with **ngx-bootstrap** and **Angular Material** to build visually consistent, WCAG-compliant UIs. Collaborated with backend and QA teams to define API contracts and implement **consistent error handling** across API layers and UI
- Wrote **unit tests using Jasmine/Karma**, achieving >85% frontend coverage across services, pipes, and UI logic.
- Conducted **performance profiling and optimization** using **Chrome DevTools**, reducing first-contentful paint (FCP) and page rendering times. Refactored legacy jQuery-based code to Angular services and components, simplifying code maintenance and reducing tech debt. Integrated user session management (idle timers, warning modals, session expiration) to meet DoD security and timeout policies.
- Participated in UX design sessions and stakeholder demos, gathering feedback and iteratively improving component design.

Cloud and Integration:

- Dockerized all backend services and deployed them to **AWS EKS** clusters using **Helm charts** and environment-specific values.yaml configurations.
- Collaborated with cloud and DevSecOps teams to automate provisioning of **EC2, RDS, S3, and IAM policies** using **CloudFormation**.
- Implemented end-to-end **CI/CD pipelines** using **Jenkins, GitLab CI**, and GitHub Actions for build, test, artifact versioning, and deployment workflows.
- Set up application-level logging with **Log4j2** and centralized log monitoring via **AWS CloudWatch** and **Splunk** for audit and incident management.
- Integrated **Redis** as a caching layer to reduce DB hits for reference lookups (NAICS codes, agency metadata) and speed up repeated API calls.

- Designed **Lambda-based utilities** for background tasks like scheduled alerts, email dispatching, and file format conversions. Created S3-based document handling pipeline for proposal attachments, with encryption, access logging, and size validation.
- Monitored service performance, resource utilization, and pod health using **Prometheus, Grafana, and CloudWatch dashboards**. Supported disaster recovery testing and RPO/RTO policy verification by contributing to infrastructure snapshots and database backup strategies. Participated in vulnerability scanning and patching cycles using **SonarQube** and REI DevSecOps toolkits, addressing CVEs and misconfigurations.

Environment: Java 17, Spring Boot, Spring Cloud, Spring Security, Spring WebFlux, Hibernate (JPA), PostgreSQL, Angular 14, TypeScript, RxJS, Bootstrap, Apache Kafka, Redis, Docker, Kubernetes (AWS EKS), AWS (EC2, RDS, S3, Lambda, CloudWatch, IAM, CloudFormation), Jenkins, GitLab CI/CD, Swagger/OpenAPI, JUnit 5, Mockito, Jasmine, Karma, Git, IntelliJ IDEA, SonarQube, Splunk, Agile (SAFe), RESTful APIs, JWT, OAuth2.

Software Developer

January 2020 – October 2021

BankUnited - Miami Lakes, FL, USA

At BankUnited, I contributed to the development and enhancement of a **Credit Card Installment Service** that enabled customers to convert eligible credit card transactions into structured EMIs. The solution was built using a **Spring Boot microservices architecture**, exposing secured **RESTful APIs** for real-time eligibility validation, interest calculation, and repayment scheduling. I was actively involved in implementing backend services, integrating with **MariaDB**, and deploying microservices on **Pivotal Cloud Foundry (PCF)** as part of a highly secure, scalable environment.

Frontend:

- Collaborated with senior frontend developers to integrate backend APIs with the **Angular 11** UI for EMI booking and repayment screens.
- Assisted in building and validating dynamic forms using **Reactive Forms** and basic Angular validators. Created **custom Angular pipes and directives** to handle dynamic formatting of payer codes and eligibility status. Built reusable components and lazy-loaded modules for performance optimization.
- Created interactive and data-driven UI elements with **Angular Material** to deliver an intuitive and engaging user experience.
- Employed **CSS Flexbox** and **Grid Layouts** for designing adaptable and flexible layouts, catering to various screen sizes and devices.
- Conducted frontend testing using **Jasmine** and **Karma**, achieving >85% coverage on all major components.

Backend:

- Developed **Spring Boot** microservices for eligibility validation, claim intake, and audit logging. Integrated **Spring Security** with OAuth2 and JWT for secure, role-based API access.
- Utilized **Spring Data JPA** and **PostgreSQL** to manage payer rules, member coverage data, and claim adjudication workflows.
- Created **custom annotations** like `@ValidNPI`, `@CoveredByPolicy` for validation of request payloads across services.
- Implemented asynchronous processing with **Apache Kafka**, handling claim lifecycle events (submission, review, response).
- Wrote **JUnit 5** and **Mockito** test suites with >90% method-level coverage; applied best practices for TDD. Used **Swagger/OpenAPI** for auto-generating interactive API documentation.

Cloud and Integration:

- Containerized microservices with **Docker** and deployed them to **AWS EKS (Kubernetes)**. Managed cloud infrastructure with **CloudFormation** (EC2, S3, RDS, IAM, Lambda) for scalable deployments.
- Integrated **Redis** as a caching layer to store frequent payer metadata and eligibility rules. Set up **GitLab CI/CD** pipelines for automated build, test, and deployment workflows.
- Enabled real-time monitoring with **Prometheus** and **Grafana** dashboards, triggered alerts for service latency and API failures.
- Ensured HIPAA compliance via encrypted RDS snapshots, private S3 buckets, and secure audit trail logging. Participated in **disaster recovery (DR) testing** with RPO/RTO validation and infrastructure snapshots.

Environment: Angular , TypeScript, HTML5, CSS3, RxJS, Java 11, Spring Boot, Spring Security, Spring Data JPA, RESTful APIs, PostgreSQL, Apache Kafka, Redis, OAuth2, JWT, Docker, Kubernetes (AWS EKS), AWS (EC2, S3, RDS, Lambda, CloudFormation), Jenkins, GitLab CI/CD, JUnit 5, Mockito, Jasmine, Karma, Swagger, Prometheus, Grafana, SonarQube, Splunk, Agile (Scrum).

Software Developer

April 2018 – December 2019

Conquerors Software Technologies, Hyderabad, India

Contributed to the end-to-end development of a **cloud-native, HIPAA-compliant platform** designed for validating **patient insurance eligibility** and **automating claims processing** for a healthcare provider. The system aimed to reduce manual adjudication, minimize claim denials, and ensure compliance with U.S. healthcare data standards such as **FHIR** and **HL7**. Patient Eligibility & Claims Processing Platform integrated with major **Electronic Health Record (EHR)** systems (e.g., Epic, Cerner) and health insurance payers via secure **RESTful APIs**, supporting real-time eligibility checks, policy coverage retrieval, and automated claim adjudication. The backend was designed using a **microservices architecture**, enabling scalable, event-driven communication between services using **Kafka**. The system featured robust audit logging, role-based access control (RBAC), and encryption at rest and in transit to ensure data privacy and HIPAA compliance.

Frontend:

- Developed and maintained responsive UI using **React.js**, **HTML5**, **CSS3**, and **TypeScript** for claim submission, validation results, and benefit summaries.
- Integrated frontend with REST APIs using **Axios** and **React Query/Fetch API**, managing real-time claim status updates and asynchronous data flows.
- Implemented **React Hook Form** with validation using **Yup** for complex insurance data, including coverage dates, dependents, and out-of-pocket limits.
- Built custom **React hooks** and **reusable components** for dynamic formatting of payer codes and eligibility status. Applied **code-splitting** and **lazy loading** with React. **Lazy** and **Suspense** for performance optimization.
- Designed intuitive and data-driven UI components using **Material-UI (MUI)** to deliver an engaging user experience. Utilized **CSS Flexbox** and **Grid Layout** to create responsive designs adaptable to various screen sizes and devices.
Performed frontend testing using **Jest** and **React Testing Library**, achieving over 85%-unit test coverage on major components.

Backend:

- Developed **Spring Boot** microservices for eligibility validation, claim intake, and audit logging. Integrated **Spring Security** with OAuth2 and JWT for secure, role-based API access

- Utilized **Spring Data JPA** and **PostgreSQL** to manage payer rules, member coverage data, and claim adjudication workflows.
- Created **custom annotations** like `@ValidNPI`, `@CoveredByPolicy` for validation of request payloads across services.
- Implemented asynchronous processing with **Apache Kafka**, handling claim lifecycle events (submission, review, response).
- Wrote **JUnit 5** and **Mockito** test suites with >90% method-level coverage; applied best practices for TDD. Used **Swagger/OpenAPI** for auto-generating interactive API documentation.

Cloud and Integration:

- Containerized microservices with **Docker** and deployed them to **AWS EKS (Kubernetes)**. Managed cloud infrastructure with **CloudFormation** (EC2, S3, RDS, IAM) for scalable deployments.
- Integrated **Redis** as a caching layer to store frequent payer metadata and eligibility rules. Set up **GitLab CI/CD** pipelines for automated build, test, and deployment workflows.
- Enabled real-time monitoring with **Prometheus** and **Grafana** dashboards, triggered alerts for service latency and API failures.
- Ensured HIPAA compliance via encrypted RDS snapshots, private S3 buckets, and secure audit trail logging. Participated in **disaster recovery (DR) testing** with RPO/RTO validation and infrastructure snapshots.

Environment: Angular 10, TypeScript, HTML5, CSS3, RxJS, Java 11, Spring Boot, Spring Security, Spring Data JPA, RESTful APIs, PostgreSQL, Apache Kafka, Redis, OAuth2, JWT, Docker, Kubernetes (AWS EKS), AWS (EC2, S3, RDS, Lambda, CloudFormation), Jenkins, GitLab CI/CD, JUnit 5, Mockito, Jasmine, Karma, Swagger, Prometheus, Grafana, SonarQube, Splunk, Agile (Scrum).

Education

Master's in computer and information Science
University of North Texas, USA

Bachelor's in information technology
Jawaharlal Nehru Technological University, Hyderabad, India