Sandeep Lanke

Phone:+1 (901)480-6309
Email:lankesandeep13@gmail.com

Professional Summary:

- Senior Java Full Stack Developer with 11+ years of experience delivering enterprise-scale, cloud-native, and distributed applications across **finance, retail, insurance, and cloud platforms.**
- Specialized in full-cycle development using **Java 17/11**, **Spring Boot**, **Spring MVC**, **Spring Security**, **Hibernate**, and **Microservices Architecture**.
- Experienced in designing scalable microservices and event-driven architectures using **Java 17**, **Spring Boot**, and **Apache Kafka** for high-throughput systems.
- Skilled in deploying cloud-native applications using **AWS Lambda**, **API Gateway**, and containerization via **Docker**, with **CI/CD pipelines** automated through **Jenkins** and **Azure DevOps**.
- Proficient in designing and consuming **RESTful APIs**, and **SOAP-based services** using **Spring Boot**, **JAX-RS**, **JAX-WS**, **WSDL**, and **Apache Axis**.
- Migrated legacy Java/JSP and EJB-based systems into **Spring Boot Microservices**, improving scalability, maintainability, and CI/CD readiness.
- Integrated complex role-based authorization systems using **OAuth 2.0**, **JWT**, and **Spring Security**, supporting secure authentication flows across microservices.
- Developed multi-tenant backend services with **Spring Data JPA**, **Hibernate**, and **JPA Criteria Builder**, connecting to **PostgreSQL**, **MySQL**, **MongoDB**, and **Oracle**.
- Built and orchestrated backend jobs and ETL operations using **Spring Batch**, **Quartz Schedulers**, and **Shell Scripts** for batch imports and exports.
- Implemented **Spark Streaming** and **Apache Kafka** pipelines to capture real-time data feeds and enrich transactional flows using **HBase** and NoSQL.
- Conducted robust **unit, integration, and UI testing** using **JUnit**, **Mockito**, **Selenium**, **Jasmine**, **Karma**, and **Postman**, achieving high code coverage.
- Experienced in multi-environment configuration management and containerization using **Docker**, **Kubernetes (AKS)**, and **Azure Container Registry (ACR)**.
- Followed Agile/Scrum practices: contributed to daily standups, sprint planning, retrospectives, user story grooming, and technical demos.
- Maintained build automation and artifact versioning via **Maven**, **Ant**, and **Nexus**, managing dependencies across microservices.

- Developed multi-tenant backend services with **Spring Data JPA**, **Hibernate**, and **JPA Criteria Builder**, connecting to **PostgreSQL**, **MySQL**, **MongoDB**, and **Oracle**.
- Built and orchestrated backend jobs and ETL operations using **Spring Batch**, **Quartz Schedulers**, and **Shell Scripts** for batch imports and exports.
- Implemented **Spark Streaming** and **Apache Kafka** pipelines to capture real-time data feeds and enrich transactional flows using **HBase** and NoSQL.
- Conducted robust **unit, integration, and UI testing** using **JUnit**, **Mockito**, **Selenium**, **Jasmine**, **Karma**, and **Postman**, achieving high code coverage.
- Experienced in multi-environment configuration management and containerization using **Docker**, **Kubernetes (AKS)**, and **Azure Container Registry (ACR)**.
- Followed Agile/Scrum practices: contributed to daily standups, sprint planning, retrospectives, user story grooming, and technical demos.
- Maintained build automation and artifact versioning via **Maven**, **Ant**, and **Nexus**, managing dependencies across microservices.

- Applied design patterns such as **Factory**, **Builder**, **Singleton**, **MVC**, **DAO**, and **DTO** across layers to ensure clean architecture and testability.

- Proven expertise in designing and developing **backend-heavy, high-performance microservices** using **Java (8/11/17), Kotlin, Spring Boot, Spring MVC, Spring Cloud, Hibernate/JPA, JDBC, JMS**, and **REST/SOAP APIs**.
- Experienced in building **high-throughput, low-latency systems (10K+ TPS)** with strict SLA, resiliency, and fault-tolerance requirements.
- Strong knowledge of the Spring Framework and its ecosystem, including Spring Boot, Spring Cloud, Spring Web Flux, Spring Batch, Spring JMS, like Hibernate and JPA, as well as integrating database connectivity using Spring JDBC to build scalable, reliable, and maintainable applications.
- Led modernization efforts replacing AngularJS and Struts-based legacy systems with modern stacks using **Spring Boot**, and **API-first design**.
- Collaborated with cross-functional teams including QA, DevOps, business analysts, and product owners across onshore/offshore environments.
- Mentored junior developers and new team members, conducted code reviews, and championed clean code and SOLID principles.
- Worked with web/application servers including **Apache Tomcat**, **JBoss**, **WebLogic**, and **WebSphere**, handling configurations, deployments, and troubleshooting.
- Hands-on experience with version control systems including **Git**, **GitHub**, **GitLab**, **SVN**, and branching strategies like **Git Flow** and feature toggles.

Technical Skills:

Programming Languages	Java (8–17), Python, JavaScript (ES6+), TypeScript, PL/SQL, T-SQL
Web Technologies	HTML5, CSS3, Bootstrap, SASS, jQuery, Node.js, AJAX
Frontend Frameworks	React.js (Hooks, Context API, Redux), Angular (AngularJS, Angular 12+), Vue.js
Design Methodologies	Agile, Scrum, Waterfall, TDD, RUP, BDD.
Application Frameworks	Spring (Core, MVC, Boot, Security, Batch, Web Flux, AOP, Data JPA, JDBC, Cache, Actuator), Spring Cloud (Eureka, Config Server, Gateway, Feign, Sleuth, Hystrix, Ribbon, Zuul, Hibernate ORM, Struts, Express.js
Databases	SQL: Oracle, SQL Server (2012–2016), MS Access, MySQL, PostgreSQL, IBM DB2 NoSQL: MongoDB, Cassandra, DynamoDB
Testing Tools	JUnit, Mockito, TestNG, Spring Test Context, Selenium WebDriver, Cucumber (BDD), Spring Cloud Contract, SonarQube, JMeter, Jasmine, Jest
Monitoring & Analytics	Micrometer, Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Open Telemetry, Zipkin, JProfiler, custom dashboards
Cloud	AWS (Lambda, EC2, S3, RDS, DynamoDB, Aurora, ELB, EBS, CloudFormation, Code Pipeline, Code Build, Code Deploy), Google Cloud Platform (GCP - App Engine, Cloud SQL, GKE, Cloud Storage), Azure (Azure App Service, Azure Virtual Machines, Azure Blob Storage, Azure Functions, Azure SQL Database, Azure Managed Disks, Azure Load Balancer, Azure API Management, Azure DevOps, Azure Monitor, Application Insights, Azure RBAC, Azure Data Factory, Azure Logic Apps), Cloud Foundry
DevOps Tools	Docker, Kubernetes, ECS, EKS, Jenkins, GitHub Actions, Maven, Gradle, Terraform, Ansible, Chef, Puppet, Bamboo, Isac
IDEs	IntelliJ IDEA, Eclipse, NetBeans, MyEclipse, RAD
Version Control	Git, GitHub, Bitbucket, SVN, CVS
Operating System	Windows, UNIX, LINUX.

Client: USAA-San Antinio, TX **Jan 2024 – Till Date**
Role: Sr. Java Full-Stack Developer
Responsibilities:

- Participated in the complete SDLC, including requirements gathering, design, development, testing, deployment, and maintenance, using **Agile (SCRUM)** methodology.
- Implemented robust backend APIs using **Java 17**, **Spring Boot**, **Spring MVC**, and **Spring Data JPA** for modules such as account summary, fund transfers, and investment profiles.
- Secured APIs using **Spring Security** and **JWT**, enabling role-based access across Admin, Teller, and Customer portals.
- Designed microservices architecture using **Spring Boot** for modules like Payments, Customer Profiles, Loan Applications, and Card Management.
- Configured and used **AWS EC2 instances** to host backend services and **AWS S3** for storing customer documents and profile images.
- Built **serverless AWS Lambda functions** to trigger fraud alerts, notify account activities, and automate internal compliance workflows.
- Refactored batch operations using **Spring Batch**, reducing monthly job runtime by 40% and improving modularization.
- Implemented in-memory caching using **Redis**, resulting in a 50% improvement in API response times.
- Integrated services with **AWS API Gateway** to securely expose backend endpoints and enforce rate limiting.

  Developed and managed **Spring Batch jobs** to automate monthly statement generation and scheduled financial reports.
- Built and deployed containerized applications using **Docker** and orchestrated multi-environment configurations.
- Built custom **JMS listeners** with **Spring Boot** for asynchronous processing of transactions from **AWS SQS**.
- Orchestrated multi-region deployments using **Docker**, **Kubernetes**, and **AWS Code Deploy**, improving availability and scalability.
- Configured role-based access using **Spring Security** and fine-grained permissions via **OAuth 2.0 scopes**.
- Implemented scalable data ingestion pipelines using **Apache Kafka** and **Apache Spark Streaming**, transforming and storing data into **HBase** and **Azure Data Lake**.
- Monitored services using **AWS CloudWatch**, and used **ELK Stack (Elasticsearch, Logstash, Kibana)** and **Splunk** for log aggregation and diagnostics.
- Developed SOAP and RESTful APIs using **JAX-RS**, **JAX-WS**, **WSDL**, and integrated them with internal legacy systems.
- Used **Swagger/Open API** to document REST endpoints for frontend and QA collaboration.
- Performed backend testing using **JUnit** and **Mockito**, achieving over 90% test coverage.
- Followed Git Flow for version control using **Git** and **GitHub**, ensuring smooth branching, merging, and release management.
- Used **Spring AOP** to apply cross-cutting concerns such as logging, auditing, and exception handling across microservices.
- Participated in sprint planning, story point estimation, daily standups, and retrospectives to align with Agile development goals.
- Mentored junior developers and reviewed code for consistency, readability, and adherence to architecture standards.
- Used **Postman** and **Swagger UI** to validate and debug APIs during development and QA testing.
- Automated build and deployment processes using **Maven**, integrated with **Azure DevOps Pipelines** for end-to-end release workflows.

Environment: Java 17, Spring Boot, Spring MVC, Spring Security, Spring Batch, Spring Framework 5.x, Hibernate, Microservices Architecture, REST, SOAP, JAX-RS, JAX-WS, WSDL, PostgreSQL, Oracle, MongoDB, DynamoDB, Apache Kafka, Apache Spark Streaming, HBase, AWS (EC2, S3, RDS, Lambda, CloudWatch, SNS, SQS, CloudFormation, API Gateway), Docker, Maven, Git, GitHub, Azure DevOps Pipelines, JUnit, Mockito, Postman, Swagger, ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Linux, IntelliJ IDEA, Eclipse, JIRA

Client: Hilton– Little Rock, AR. **Jun 2022 - Dec 2023**
Role: Java Developer
Responsibilities:

- Responsible for all stages of design, development, and deployment of enterprise-grade applications using Agile methodology and **Test-Driven Development (TDD)**.
- Automated CI/CD pipelines using **Jenkins**, integrated with **Git**, **Maven**, and **AWS Code Deploy** for seamless versioned releases.
- Designed and implemented **RESTful APIs** and microservices using **Spring Boot**, **Spring MVC**, and **Java 11**, supporting high-throughput business functions such as onboarding, transactions, and reporting.
- Secured APIs using **Spring Security**, **OAuth 2.0**, and **JWT**, implementing **role-based access control (RBAC)** to meet security, compliance, and governance policies.
- Developed serverless APIs and backend processes using **Azure Functions**, focusing on scalability, event-driven logic, and cost efficiency.
- Set up and monitored distributed queues using **AWS SQS** to decouple services and handle asynchronous transactions securely.
- Leveraged **AWS SNS** to send customer notifications for payment confirmations, low balance alerts, and scheduled statements.
- Used **Azure App Services** to host full-stack applications, handling routing, domain mapping, and monitoring within a cloud-native environment.
- Utilized **Spring Data JPA**, **Hibernate**, and **JPA Criteria API** to dynamically construct queries for CRUD operations and report generation across Oracle and PostgreSQL databases.
- **Designed and optimized NoSQL data models in MongoDB**, leveraging the aggregation framework, embedded documents, and index strategies to improve query performance and storage efficiency.
- Wrote **UNIX shell scripts** and batch automation jobs for scheduled cleanups, data syncing, and log archival between services.
- Deployed and maintained Java microservices on **JBoss**, managing endpoints, transaction logs, and scaling configurations in production.
- Used **AWS RDS (PostgreSQL)** for transactional operations and integrated **MongoDB** for semi-structured product and offer catalogs.
- Configured **JMS queues/topics** in JBoss for asynchronous message handling across internal financial workflows and regulatory systems.
- Applied design patterns including **Factory**, **Builder**, and **Singleton** for maintainable service logic, applying **SOLID principles** across services and interfaces.
- Used **Swagger/Open API** to generate API documentation and validate REST services during integration and QA.
- Wrote unit and integration tests using **JUnit**, **Mockito** achieving high test coverage and reducing regression.
- Managed project tracking using **Jira** and maintained repositories, issues, and release branches using **GitHub** and **GitLab**.

Environment: Java, Core Java, Spring Boot, Spring Security, Spring MVC, Hibernate, JPA, JMS, Apache Camel, RESTful, SOAP, Microservices Architecture, Oracle, PostgreSQL, MongoDB, HBase, AWS (RDS, SQS, SNS, Code Deploy), Azure (API Management, Functions, App Services, DevOps, ARM Templates), Docker, Kubernetes, Jenkins, Maven, Git, GitHub, GitLab, GitLab CI/CD, Cloud Bees, Chef, JUnit, Mockito, Postman, Swagger, ELK Stack (Elasticsearch, Logstash, Kibana), Splunk, Log4j, JIRA, IntelliJ IDEA, Eclipse, Agile, TDD, Shell Scripting, YAML.

Client: Sentara – Nashville, TN **Mar 2020 – May 2022**
Role: Java Developer
Responsibilities:

- Involved in most phases of the Software Development Life Cycle (SDLC) using **Agile methodology**, participating in daily **Scrum meetings** for progress tracking and sprint planning.
- Designed and built **microservices** using **Spring Boot** and **Spring MVC**, enabling modular, scalable, and reusable business logic.
- Deployed applications on **Jetty** and **Apache Tomcat** for lightweight runtime environments and tested using local containers.
- Deployed containerized microservices using **Docker** into **AWS EC2 Container Service (ECS)**, and managed cloud infrastructure through the **AWS Admin Console**.
- Configured and managed **JMS Queues**, Data Sources, and connection factories for asynchronous communication in **IBM WebSphere** clustered servers for load balancing and failover.
- Implemented business logic using **Spring Boot**, **Spring Security**, **Spring AOP**, **Spring DAO**, and **Spring Transactions**, securing endpoints and handling user roles.
- Developed real-time anomaly detection pipelines using **Apache Spark ML lib** and **Scala** for patient data quality checks.
- Externalized environment-specific configuration using **Spring Cloud Config**, simplifying DevOps maintenance.
- Automated database migrations using **Flyway**, ensuring consistent schema versioning across **PostgreSQL** services.
- Used **Spring Data JPA** and **Hibernate ORM** for object-relational mapping and dynamic query building for CRUD operations.
- Designed and queried **SQL**, **MySQL**, and **MongoDB** databases, handling complex joins and schema definitions for structured and semi-structured data.
- Worked on **Apache Kafka** and **Spark Streaming** to build on premise data pipelines for ingesting real-time data from API Gateway services.
- Implemented **Spark ML lib** components in **Scala** to support machine learning models and streaming analytics.
- Developed scalable data ingestion processes from external sources to **HBase**, using **Kafka** and **Spark Streaming** with scheduled batch jobs.
- Created and triggered **AWS Lambda** functions for event-driven processing, including background tasks and real-time notification services.
- Implemented **AWS IAM** roles and policies to manage secure access between microservices, EC2, and S3 buckets.
- Automated microservice deployments to **Pivotal Cloud Foundry (PCF)** using **Jenkins** with build packs and CI/CD integrations.
- Created Jenkins pipeline templates using **Groovy**, automating builds, test runs, and deployments to Azure using container registries and AKS clusters.
- Developed and exposed **RESTful web services** using **Spring Boot**, integrated with Azure APIs, and implemented security using **OAuth 2.0** and **JWT**. Configured **Spring Quartz Scheduler** for automated job execution and time-based task management.
- Utilized **Spring AOP** for centralized logging and implemented monitoring using **Splunk** and **Log4j** for system diagnostics.
- Integrated backend services with **JSP**, **Spring MVC**, and **JPA mappings**, delivering full-stack applications with layered architecture.
- Developed backend modules using the full Spring stack including **IOC**, **Bean Wiring**, **Auto Wiring**, and **JDBC Templates**.
- Developed RESTful services using **Node.js** for lightweight backend support.
- Used **Maven** for dependency management and project builds, integrated with **Jenkins** pipelines for automated deployments.

- Worked with **Apache Tomcat** and **Node.js** to deploy and configure Java APIs.
- Managed version control using **GIT**, **GitHub**, and **SVN**, following branching strategies to support collaborative development.
- Used **JIRA** for issue tracking, sprint planning, and Agile board management during the software lifecycle.

Environment: Java, Spring Boot, Spring MVC, Spring Security, Spring IOC, Spring AOP, Spring DAO, Spring Transactions, Hibernate, Microservices, Kafka, Apache Spark Streaming, Spark ML lib (Scala), RESTful Services, Jersey, Spring Quartz Scheduler, AWS (EC2, S3, Lambda, RDS, IAM, SNS, SQS), Docker, Jenkins, Maven, Git, GitHub, SVN, Pivotal Cloud Foundry (PCF), Azure (AKS, Container Registries, APIs), HBase, MySQL, MongoDB, Cassandra, Log4j, Splunk, Postman, Swagger, SoapUI, WebLogic, Apache Tomcat, IBM WebSphere, Linux, Solaris, IntelliJ IDEA, Eclipse, JIRA.

Client: Gemini Communications Limited – Chennai, India **Oct 2016 – Nov 2019**
Role: Java Developer
Responsibilities:

- Involved in the full **Software Development Life Cycle (SDLC)**, including requirement analysis, design, development, testing, deployment, and ongoing maintenance.
- Used **Spring Boot**, **Spring Cloud**, **Spring Batch**, **Spring Security**, **Spring AOP**, and **Spring Transactions** to design scalable, secure backend services and microservices.
- Designed application modules, utility classes, and reusable backend components using **Core Java**, applying design patterns like **Factory**, **Singleton**, and **Builder**.
- Created and consumed **SOAP web services** using **JAX-WS** standards with **WSDL**, supporting integration with external and legacy systems.
- Utilized **JMS** for asynchronous communication between microservices, supporting event-driven architecture across modules.
- Created reusable query modules with **JPA Criteria API**, enabling flexible, type-safe dynamic queries.
- Migrated distributed configuration to **Spring Cloud Config**, supporting multi-tenant microservices.
- Wrote **unit and integration tests** using **JUnit**, **Mockito**, **Jasmine**, and **Karma**, following **Test-Driven Development (TDD)** principles to ensure high code quality.
- Used **Maven** for build automation and dependency management across multiple services.
- Managed code repositories and branching strategies using **Git** and **GitHub**, collaborating via pull requests and peer code reviews.
- Tracked development progress and managed backlogs using **JIRA**, contributing to sprint estimations and meeting release deadlines.
- Developed reusable and modular backend services using **Spring Boot**, **Spring Security**, and **Spring AOP**, enabling secure API-level access and centralized logging.
- Designed dynamic queries using **JPA Criteria API**, reducing boilerplate code and improving maintainability across DAO layers.
- Migrated configuration management to centralized repositories using **Spring Cloud Config**, supporting multi-environment and multi-tenant deployments.
- Created message-driven processing modules using **JMS** for asynchronous communication between distributed services.
- Implemented CI/CD automation for backend services using **Jenkins**, **Maven**, and **Docker**, ensuring reliable and repeatable deployments.

Environment: Java, J2EE, Spring Boot, Spring Security, Spring Batch, Spring Cloud, Spring AOP, Spring Transactions, Hibernate, Web Services, RESTful, SOAP, JMS, JAX-WS, JAX-RS, WSDL, JAXB, JUnit, Log4j, PostgreSQL, MySQL, MongoDB, Jenkins, Docker, Kubernetes, Git, GitHub, Maven, SOAPUI, IntelliJ IDEA, Eclipse, Windows, Red Hat Linux, ARM Templates, Agile, Scrum, UML, JIRA, Test-Driven Development (TDD)

Client: ADP– Hyderabad, India. **Aug 2014 - Sep 2016**
Role: Java Developer
Responsibilities:

- Participated in the entire **Software Development Life Cycle (SDLC)** from requirement gathering and analysis to development, testing, deployment, and post-production support.
- Developed enterprise-grade modules using **Java** and **J2EE**, leveraging technologies such as **JSP**, **Servlets**, **JDBC**, and **JMS** for business workflows and messaging services.
- Designed and implemented **Session Beans** using **EJB 3.x** to encapsulate business logic and promote modularity.
- Created scalable service components using **Spring Boot**, **Spring MVC**, and **Spring Security** for authentication, user management, and secure data flow.
- Used **Spring AOP** to implement cross-cutting concerns like logging, security auditing, and exception handling.
- Integrated **Hibernate ORM** with **Spring ORM**, building robust DAO layers with **JPA annotations**, **HQL**, and **SQL** for efficient data access across **PostgreSQL** and **Oracle** databases.
- Consumed and exposed **RESTful web services** using **JAX-RS**, and integrated third-party SOAP services using **Apache Axis**, **WSDL**, and **JAX-WS**.
- Led legacy application transformation from **EJB** monolith to modular **Spring Boot microservices**, improving cold start performance.
- Developed RESTful service routing logic using **Apache Camel**, facilitating orchestration of third-party and legacy services.
- Followed best practices in **Object-Oriented Programming (OOP)** and implemented architectural design patterns including **DAO**, **DTO**, **Singleton**, **Facade**, and **MVC**.
- Deployed and configured Java EE applications on **WebSphere Application Server 6.0/7.0**, managing cluster environments and load balancing setups.
- Developed unit tests using **JUnit** and **Mockito**, achieving over 85% coverage and ensuring functional integrity of APIs.
- Automated test scripts using **Selenium RC** and **Selenium Grid**, integrated with **Maven** for scheduled builds.
- Used **Log4j** to configure application logging, enabling efficient error tracking and troubleshooting across microservices.
- Created and maintained builds using **Maven**, managed source code using **SVN**, and later migrated repositories to **GitHub** and **GitLab**.
- Collaborated with cross-functional teams in an **Agile Scrum** setup, attending daily stand-ups, grooming sessions, retrospectives, and sprint planning meetings.
- Tracked feature development and bug fixing using **Jira**, maintaining traceability across sprints and improving team collaboration.

Environment: Java, J2EE, Spring v4.1.0, Spring MVC, Spring Security, Spring AOP, Hibernate, EJB 3.x, RESTful, SOAP, JMS, JAX-WS, JAX-RS, WSDL, JAXB, JUnit, Mockito, Log4j, Maven, SQL, MySQL, PostgreSQL, Oracle, TOAD, WebSphere Application Server, Apache, JBOSS, VMware, SVN, GitHub, Jenkins, SOAPUI, IntelliJ IDEA, MyEclipse, Windows, Unix, Agile Scrum, UML, JIRA