**Kranthi Kumar Alimelu**

**Sr. Java Developer**

**Phone:** 980-499-2804 **LinkedIn:** https://www.linkedin.com/in/kranthi-alimelu/

**Email:** [kranthitek@gmail.com](mailto:kranthitek@gmail.com)

**PROFESSIONAL SUMMARY**

* Over 10+ years of IT experience as a Java Backend Developer with a strong background in designing and building scalable web applications using Java/J2EE, Spring, and open-source technologies.
* Proven expertise in full Software Development Life Cycle (SDLC) including Analysis, Design, Development, Testing, and Deployment using Agile (SCRUM) and Test-Driven Development (TDD) methodologies, with hands-on experience in project management tools like JIRA and version control systems such as Git.
* Hands-on experience in designing, implementing, and optimizing Microservices architectures using Spring Boot, Spring Cloud, and RESTful APIs, incorporating messaging systems like Apache Kafka, JMS, and AWS SQS for efficient and scalable communication between services.
* Skilled in front-end development with technologies like React.js, Angular, and Node.js, utilizing frameworks like Redux and React Hooks for building interactive, dynamic user interfaces, and integrating them with backend systems via RESTful APIs.
* Proficient in integrating front-end components with back-end services, implementing Single Page Applications (SPA) using AngularJS, and leveraging modern JavaScript libraries and frameworks like jQuery, Bootstrap, and TypeScript to enhance user experiences.
* Extensive experience with databases, including designing schemas, writing complex SQL queries, working with relational databases like SQL Server, Oracle, and NoSQL databases such as MongoDB and Cassandra for high-performance, scalable data storage solutions.
* Experience in cloud-native development and deployment using Amazon Web Services (AWS), including EC2, S3, RDS, Lambda, and more, along with containerization tools like Docker and orchestration platforms such as Kubernetes for managing microservices.
* Proficient in utilizing CI/CD tools like Jenkins, Bamboo, and Maven to streamline build, deployment, and testing processes. Extensively used tools like SonarQube for static code analysis and ensuring adherence to best practices and security guidelines.
* Skilled in web service integration, including SOAP, RESTful services, and XML/JSON processing, using frameworks like JAX-RS, JAX-WS, and JAXB for efficient service-based architecture.
* Deep knowledge of Java core concepts, including Multi-threading, Collections, Exception Handling, and design patterns. Hands-on experience with ORM frameworks like Hibernate, JPA, and building applications using the MVC framework (Spring MVC).
* Strong focus on performance optimization, including JVM tuning, memory management, profiling, and query optimization for databases. Experienced in building scalable, high-performance systems in both traditional and cloud-based environments.
* Proficient in test automation frameworks like JUnit, TestNG, Mockito, and tools like Selenium for functional and regression testing to ensure high-quality deliverables.
* Demonstrated ability to lead technical teams, conduct training sessions, and mentor junior developers in adopting best practices in software development and Agile methodologies.

**TECHNICAL SKILLS**

* **Languages:** Java,**Python, JavaScript, SQL, Scala**
* **Frameworks:** **Spring Boot, Spring MVC, Spring Cloud, Spring Security, React.js, AngularJS, JSP, Hibernate, JAX-RS, JAX-WS**
* **Cloud Platforms:** **GCP, AWS (EC2, S3, RDS, Lambda, ECS, EKS, CloudWatch)**
* **Database Management:** **PostgreSQL, MongoDB, Cassandra, SQL Server**
* **Data Engineering:** **Apache Kafka, Apache Beam, Apache Spark, ETL Pipelines**
* **CI/CD & DevOps Tools:** **Jenkins, Git, Maven, Docker, Kubernetes, Chef, Bamboo**
* **Data Visualization & Reporting:** **Power BI, Tableau, Grafana, Elastic Search**
* **Testing Frameworks: JUnit, Mockito, Spock, Jasmine, Protractor, BDD frameworks**
* **Others:** **SOAP Web Services, RESTful APIs, GraphQL, AWS Lambda, Terraform, JIRA, Jenkins Pipelines, Kafka Connect, Log4j, JMS, Splunk**

**PROFESSIONAL EXPERIENCE**

**Client: Grant Solutions, VA**

**Role: Sr. Full Stack Developer Jul 2024 – present**

***Responsibilities***:

* Built and deployed **cloud-based microservices** using **Spring Boot** with RESTful APIs and integrated with **RabbitMQ**for asynchronous messaging and event-driven architecture.
* Implemented **Spring Batch framework** to develop custom readers, writers, and processors for batch processing, ensuring efficient data transformation.
* Enhanced **backend performance** by leveraging **Java 8 features** like Streams, Lambda expressions, and Functional Interfaces to reduce boilerplate code.
* Migrated code from **Angular 5 to Angular 6** using Angular CLI commands such as **ng update** and **ng add**, and optimized application performance using Ahead-of-Time (AOT) compilation.
* Built responsive and dynamic **UI components using React.js and Redux**, ensuring improved scalability, performance, and user experience.
* Utilized **RabbitMQ** as a message broker to decouple microservices communication, ensuring asynchronous communication and event-driven processing.
* Implemented **ElasticSearch** for advanced search capabilities, enabling near real-time searching and indexing of large datasets.
* Designed and deployed containerized applications using **Docker** and orchestrated them using **Kubernetes clusters** for seamless deployment and scalability.
* Worked with **OpenShift** to manage Kubernetes deployments, perform rolling updates, and scale applications with minimal downtime.
* Developed **CI/CD pipelines** using **Jenkins** integrated with **GitLab**, ensuring seamless deployment of microservices to **AWS EC2** instances with Nginx as a reverse proxy.
* Configured and managed **AWS S3 buckets** for hosting static content and enabled versioning for data storage.
* Utilized **Spring Security and OAuth2** to implement role-based access control (RBAC) and secure RESTful APIs.
* Performed **Unit Testing** and **Integration Testing** using **JUnit, Mockito, and Spring Test Framework**, ensuring high test coverage and stability of the application.
* Built RESTful APIs following **OpenAPI Specification (Swagger)** to streamline API documentation and reduce integration time with front-end services.
* Used **Hibernate ORM** with Spring Boot to handle database transactions, leveraging JPA annotations for data persistence and relational mappings.
* Leveraged **Redis Cache** to enhance application performance by reducing database queries and improving data retrieval time.
* Integrated **Kafka** as an event-driven messaging platform for decoupled microservices communication, ensuring scalability and fault tolerance.
* Automated build and deployment pipelines using **Jenkins** integrated with **Docker containers**, ensuring seamless deployment across environments.
* Deployed the application from **Jenkins to NGINX on AWS EC2** using **SSH protocol** and **rsync utility** for faster deployment and minimal downtime.
* Developed custom APIs using **Spring Boot, Java, and Microservices architecture**, integrating with third-party APIs using RESTful architecture.
* Hands-on experience with **OpenShift deployment strategies**, Kubernetes pods, and load balancing for high availability and scalability.
* Built centralized **error monitoring and logging pipelines** using **Splunk, ELK Stack, and Fluentd** for real-time alerting and monitoring.
* Worked with **Postman** and **Swagger (OpenAPI)** to test and document RESTful APIs before deployment.
* Enhanced performance of **ElasticSearch** clusters by optimizing queries, increasing indexing throughput, and reducing cluster latency.
* Worked with **Spring Boot Actuator** to monitor microservices health, custom metrics, and application behavior.
* Ensured robust microservices deployment using **Kubernetes, Docker Swarm, and OpenShift**, improving scalability and fault tolerance.
* Collaborated with cross-functional teams to deliver end-to-end functionality using **React.js, Spring Boot, RabbitMQ, and Kafka**.
* Implemented **API Gateway (Spring Cloud Gateway)** for microservices communication and secured APIs using JWT and OAuth2.
* Created **Infrastructure as Code (IaC)** using **Terraform** to provision EC2 instances, RDS databases, and S3 buckets. Developed **SQL/NoSQL hybrid solutions** using **MySQL, Redis, and Cassandra**, ensuring high availability and low-latency performance.
* Worked in **Agile/Scrum** development environments, collaborating with cross-functional teams and ensuring timely feature delivery.
* Implemented **Spring Boot DevTools** for hot deployment and rapid application development.

**Environment:** Java 8/11, Spring Boot, Spring Framework (Spring MVC, Spring Batch, Spring Security, Spring AOP), Microservices, RabbitMQ, Kafka, Elasticsearch, Redis, Kubernetes, Docker, OpenShift, Helm, Jenkins, GitLab, Terraform, AWS (EC2, S3), MySQL, Oracle DB, Cassandra, Hibernate, Maven, Gradle, JUnit, Mockito, Cucumber, Selenium, Spock, Jasmine, Angular 6, React.js, Node.js, Express.js, Swagger (OpenAPI), JSON, HTML5, CSS3, jQuery, Bootstrap, Nginx, Splunk, ELK Stack

**Client: Capital One, VA**

**Role: Sr. Full Stack Developer Feb 2024 – Jul 2024**

***Responsibilities***:

* Implemented client-side Interface using **React JS.** Designed and developed **RESTful APIs and Microservices** using **Spring Boot, Spring Data JPA, and Spring Security** for scalable backend services.
* Built **event-driven microservices** using **Kafka and RabbitMQ**, ensuring asynchronous communication and high system throughput.
* Utilized **Spring WebFlux (Reactive Programming)** to build non-blocking, high-performance REST APIs.
* Managed database operations with **Hibernate/JPA** for seamless ORM and optimized SQL queries for enhanced performance.
* Deployed microservices in **Kubernetes clusters** using **Docker containers** and automated deployment pipelines via **Jenkins and Bitbucket Pipelines**.
* Implemented **Spring Cloud Gateway** for API management and load balancing, ensuring smooth microservices communication.
* Enhanced system performance using **Redis caching**, reducing database calls and improving response time.
* Integrated third-party services using **Spring RESTTemplate and WebClient** for robust API communication.
* Built and managed infrastructure on **AWS (EC2, S3, RDS)** to host microservices and optimize deployment.
* Automated build, test, and deployment processes using **Bitbucket Pipelines and Jenkins**, minimizing manual intervention.
* Implemented **WebSocket-based real-time communication** between microservices using **RabbitMQ and Kafka**.
* Utilized **GitHub Copilot** to accelerate code generation, test writing, and refactoring, improving development efficiency.
* Enhanced user experience by developing and integrating **React.js, Redux, and TypeScript** for responsive UI components.
* Refactored React codebase using **Redux Thunk** for state management and optimized component rendering with **React Hooks**. Used **Spring Boot Actuator** for application monitoring, health checks, and log tracing.
* Configured centralized logging and monitoring using **Splunk** for real-time log aggregation and error detection.
* Improved performance by tuning **SQL queries, stored procedures, and triggers** in **Oracle DB** and **MongoDB**.
* Enabled seamless communication between microservices using **Kafka producers/consumers** and RabbitMQ queues.
* Built and secured APIs using **Spring Security with OAuth2 and JWT (JSON Web Token)** for secure authentication.
* Enhanced deployment scalability using **Kubernetes Horizontal Pod Autoscaling (HPA)** to handle increased traffic. Implemented unit and integration testing using **JUnit, Mockito, and Spock**, ensuring high test coverage.
* Managed cloud resources using **AWS EC2, S3, and RDS**, ensuring smooth deployment of microservices.
* Collaborated with front-end teams to integrate **React.js with Spring Boot APIs**, ensuring seamless end-to-end functionality.

**Environment:** Java 8/11, Spring Boot, Spring MVC, Spring Data JPA, Spring Security, Spring Cloud (Gateway, WebFlux), RabbitMQ, Kafka, Redis, Elasticsearch, Docker, Kubernetes, Jenkins, Bitbucket Pipelines, Terraform, AWS (EC2, S3, RDS), MongoDB, Oracle DB, React.js, Redux, TypeScript, GitHub Copilot, Nginx, Hystrix, REST APIs, Splunk

**Client: Altria, Richmond, VA Sep, 2021 – Nov 2023**

**Sr. Full Stack Developer**

***Responsibilities:***

* Actively participated in **Agile** (Scrum) methodology, contributing to the design, development, system testing, and user acceptance testing phases.
* **Designed and developed microservices** using **Spring Boot, Spring Data JPA, and Spring Security**, ensuring high scalability and modular architecture.
* Built **event-driven microservices** using **Apache Kafka and RabbitMQ** for asynchronous communication between distributed services.
* Implemented **GraphQL API** to aggregate data from multiple microservices, reducing round trips and improving API response times.
* Developed and deployed **Spring Boot microservices in Docker containers** on **AWS EC2 using ECS (Elastic Container Service)** for scalable deployment.
* Utilized **Spring WebFlux (Reactive Programming)** to build non-blocking, high-performance RESTful APIs for faster response times.
* Integrated **Drools Rule Engine** with Spring Boot to implement dynamic business rules and decision-making processes.
* Designed and optimized **database operations** using **Hibernate/JPA, PostgreSQL, and Cassandra**, ensuring high data throughput and minimal latency.
* Deployed microservices on **AWS EC2 instances** and managed containers using **AWS ECS and Docker Compose**.
* Improved system performance by implementing **JVM optimization, memory tuning, and garbage collection (GC) strategies**.
* Built a message-driven architecture using **Apache Kafka and RabbitMQ**, enabling fault-tolerant and high-throughput data processing.
* Utilized **Spring Cloud Config and Spring Cloud Gateway** for microservices configuration management and routing traffic.
* Automated build, test, and deployment processes using **Jenkins pipelines and Bamboo**, ensuring faster delivery of services.
* Enhanced microservices scalability by implementing **Horizontal Pod Autoscaling (HPA)** using **Docker and Kubernetes**.
* Created **RESTful APIs** to integrate **Drools, GraphQL, and Kafka**, enabling seamless communication across microservices.
* Implemented **Spring Security with OAuth2 and JWT** for user authentication and API security.
* Utilized **Redis caching** to minimize database calls and reduce response time for frequently accessed data.
* Integrated **Power BI dashboards** with SQL Server, providing real-time analytics and insights for business stakeholders.
* Worked on **PostgreSQL database**, optimizing complex SQL queries, indexing, and reducing query response time.
* Managed cloud resources using **AWS EC2, S3, ECS, RDS, and Lambda** for deploying and running microservices.
* Implemented **Drools rules engine** for dynamic business logic, reducing manual changes in code.
* Enhanced system availability using **Hystrix Circuit Breaker and Resilience4j**, ensuring high fault-tolerance during failures.
* Built **message aggregators** using **Kafka consumer groups and RabbitMQ queues**, ensuring fast and reliable data processing.
* Utilized **Spring AOP (Aspect Oriented Programming)** for cross-cutting concerns like logging, security, and exception handling.
* Implemented **Jenkins pipelines** for continuous integration and deployment, reducing deployment time by 30%.
* Deployed **Spring Boot microservices** on **Docker containers** using **AWS ECS**, ensuring zero downtime during deployments. Wrote and optimized **unit and integration tests** using **JUnit, Mockito, and Spock**, achieving high test coverage.
* Developed **GraphQL resolvers** for data aggregation from multiple microservices, ensuring a single source of truth for front-end applications.
* Configured **Splunk for centralized log monitoring**, enabling real-time log aggregation and incident detection.
* Improved microservices performance by implementing **lazy loading, caching, and request batching** in GraphQL API.
* Collaborated with front-end teams to integrate **React.js, Redux, and TypeScript** with backend APIs, ensuring seamless communication. Configured and maintained **Jenkins** pipelines for implementing the **CI/CD** process.

**Environment:** Java 8/11, Spring Boot, Spring Data JPA, Spring Security, Spring Cloud (Config, Gateway, WebFlux), RabbitMQ, Kafka, GraphQL, Redis, Cassandra, PostgreSQL, Docker, Kubernetes, Jenkins, Bitbucket, Bamboo, Terraform, AWS (EC2, S3, ECS, RDS, Lambda), Power BI, Drools, Hibernate, React.js, Redux, TypeScript, Nginx, Splunk, Log4j.

**Client: Amazon, Falls Church, VA Mar,2019 - Sep,2021**

**Sr. Full Stack Java Developer**

***Responsibilities:***

* Developed and deployed **microservices-based applications** using **Spring Boot, Spring Cloud, Spring Data JPA, and Spring Security**, ensuring scalability and modularity.
* Implemented **Kafka Connect and Kafka Streams** for real-time data processing and seamless communication between microservices.
* Built **RESTful APIs** using **Spring Boot** for CRUD operations, integrated with **MongoDB and PostgreSQL** for data persistence.
* Utilized **Spring Cloud Config, Spring Sleuth, and Spring Actuator** for microservice configuration, logging, and monitoring.
* Deployed **microservices in Docker containers** using **Kubernetes (OpenShift)** for container orchestration, ensuring fault-tolerant and scalable deployments.
* Configured **Kubernetes ConfigMaps and Secrets** for secure application configuration without exposing sensitive data.
* Developed business logic using **Drools Rule Engine**, allowing dynamic rule management and flexible decision-making in microservices.
* Worked on **Spring WebFlux (Reactive Programming)** to develop high-performance, non-blocking APIs.
* Utilized **RabbitMQ** for asynchronous messaging between microservices, ensuring high throughput and data consistency.
* Integrated **Jenkins pipelines** for CI/CD, automating build, test, and deployment processes, reducing deployment time by 40%.
* Configured **Kafka Connect** with various data sources, enabling real-time data transfer and synchronization.
* Utilized **Spring AOP (Aspect-Oriented Programming)** for centralized logging, exception handling, and performance monitoring.
* Implemented API security using **Spring Security with JWT (JSON Web Token)** and OAuth2 authentication.
* Deployed applications on **AWS EC2, ECS, and S3** using Docker containers, ensuring high availability and scalability.
* Designed and built **gRPC-based APIs** for high-performance, low-latency microservice communication.
* Utilized **Hibernate/JPA** for database operations and optimized SQL queries for high-performance data retrieval.
* Worked with **MongoDB and PostgreSQL** for database persistence and implemented replication strategies for high availability.
* Developed unit and integration tests using **JUnit, Mockito, and Karate**, achieving over 90% code coverage.
* Utilized **RabbitMQ and Kafka** for event-driven architecture, ensuring real-time data processing.
* Automated deployments using **Jenkins pipelines** and managed infrastructure through **AWS CloudFormation**.
* Leveraged **Docker Compose** for local development, allowing smooth containerized application development.
* Improved database performance by optimizing SQL queries, indexing, and implementing caching using Redis.
* Integrated external APIs and services using **RESTful web services and SOAP APIs**, enabling seamless communication with third-party services.
* Utilized **Spring Batch** for handling large-scale data processing tasks in microservices. Configured **OpenShift clusters** for container orchestration and managed microservice scaling and deployment.
* Implemented **WebSockets** in Spring Boot for real-time data streaming in web applications.
* Created dynamic and reusable UI components using **React.js, Redux, and Material-UI**, ensuring smooth user experiences.
* Reduced deployment time by 30% through containerized microservices using **Docker, Kubernetes, and Jenkins pipelines**.
* Utilized **Spring Cloud Eureka (Service Registry)** and **Spring Cloud Gateway** for dynamic microservice discovery and routing.
* Designed and built **GraphQL APIs** for efficient data aggregation from multiple microservices.
* Implemented **Kubernetes** **ConfigMaps** and **Secrets** to manage application configuration and sensitive data, ensuring secure and flexible **configuration management** without exposing sensitive information in the **container images** or deployment configurations.

**Environment:** Java 8/11, Spring Boot, Spring Cloud (Sleuth, Config, Actuator), Kafka, RabbitMQ, Drools, Hibernate, PostgreSQL, MongoDB, Docker, Kubernetes, OpenShift, gRPC, Jenkins, RESTful APIs, GraphQL, Redis, AWS (EC2, ECS, S3, RDS, Lambda), Microservices, JWT, OAuth2, React.js, Redux, Material-UI, Apache Kafka Connect, Jenkins, RabbitMQ, ElasticSearch, Kubernetes ConfigMaps, WebFlux, Maven, Log4j, GitHub, Terraforms.

**Client: Nationwide Mutual Health Insurance Company, OHIO Oct,2017 – Mar,2019**

**Sr. Java Developer**

***Responsibilities:***

* Followed **Agile Software Development** practices, including paired programming, TDD, and Scrum status meetings, ensuring timely and high-quality delivery.
* Developed **RESTful web services** using **JAX-RS** and **Spring MVC**, ensuring seamless integration with front-end applications and external systems.
* Implemented **CI/CD pipelines** using **Jenkins** and **Bamboo**, integrating **BDD testing** to automate tests, enabling early issue detection and ensuring continuous delivery.
* Utilized **Spring MVC** for developing the web tier, creating **Controller** and **Business layers** to ensure separation of concerns in the application design.
* Integrated and consumed **SOAP and RESTful web services** with **XML** and **JSON** payloads, enabling smooth communication between internal and external systems.
* Applied **Spring Security** for role-based access control, securing the application by managing user roles and access to sensitive resources.
* Worked with **MongoDB** (NoSQL) for integrating dynamic data models and **PostgreSQL** for relational data storage, ensuring efficient data access and management.
* Developed **Spring IoC (Inversion of Control)** and **Spring Security** for backend integration and authentication, enhancing application security.
* Leveraged **Hibernate ORM** for managing relational data and implementing **JPA** for efficient database interactions, ensuring optimal performance and data integrity.
* Configured **AWS Cloud** environments using **Jenkins** and **Chef** for automated provisioning and efficient infrastructure management.
* Utilized **React.js** for integrating frontend components with **RESTful APIs**, ensuring efficient communication between the backend and dynamic UI components.
* Employed **JUnit** and **BDD** frameworks for test-driven development (TDD), ensuring application robustness and quality through automated testing.
* Designed **UML diagrams**, including **Use Case Diagrams, Class Diagrams**, and **Object Mapping Diagrams**, for clear project documentation and effective team collaboration.
* Refactored legacy code to adopt **best design patterns**, improving maintainability and performance.
* Used **JIRA** for project management, tracking progress, and writing test cases, ensuring smooth development workflows.
* Experienced in **JVM memory management**, using profiling tools to identify and resolve memory issues, improving application performance.
* Used **Bindy, Stax, and JAXB** for XML data processing in integration flows, ensuring reliable data exchange between systems.

**Environment:** Java 8, Spring MVC, Spring Security, Spring IoC, Hibernate, JAX-RS, JAX-WS, MongoDB, PostgreSQL, AWS (EC2, S3), Jenkins, Chef, JUnit, BDD, SOAP, RESTful APIs, JIRA, Angular 5.0, React.js, Node.js, Express.js, Maven, Apache Tomcat, GIT, CSS3, HTML5, SQL, JSON, XML.

**Client: Cardinal Health, OHIO Dec,2015 – Oct,2017**

**Java Developer**

***Responsibilities:***

* Extensively used **Spring Framework** for **dependency injection (IoC)** and integrated **Controller, Service**, and **DAO**layers for efficient application architecture.
* Developed and consumed **RESTful web services** using **JAX-RS** and **Spring MVC**, ensuring seamless communication between frontend and backend components.
* Leveraged **Spring Boot** for developing microservices, ensuring lightweight and rapid deployment of business components with minimal configuration.
* Integrated **SOAP Web Services** for external communication, enabling interoperability with third-party applications and legacy systems.
* Utilized **MongoDB** (NoSQL) for handling unstructured data, ensuring scalable and flexible storage for application data.
* Employed **Apache Kafka** for real-time data streaming, enabling efficient messaging and communication across distributed systems.
* Migrated legacy systems to the **AWS Cloud**, optimizing architecture by transitioning from **EC2 Classic** to a **private network** with a 3-tier web application setup.
* Integrated **AWS APIs** to automate network provisioning, enhancing cloud management and application scalability.
* Worked with **Cassandra** for distributed database management, ensuring high availability and fault tolerance across multiple nodes. Used **log4j** for detailed logging and debugging, improving system monitoring and troubleshooting.
* Managed version control with **Git**, ensuring seamless collaboration and code management across teams.
* Implemented **CI/CD** pipelines using **Jenkins** and **Maven**, automating build and deployment processes for faster development cycles.
* Experienced in **mocking frameworks** such as **Karma**, **Jasmine**, and **Protractor** for testing frontend components in AngularJS-based applications.
* Developed internal applications with **Apache HTTP Server**, providing a scalable and efficient platform for web application hosting.

**Environment:** J2EE, Spring Framework, Spring Boot, Core Java (Multithreading, Collections), Spring MVC, JAX-RS, MongoDB, Apache Kafka, AWS (EC2, Cloud), Cassandra, SOAP Web Services, Jenkins, Maven, Git, log4j, JAX-WS, SVN, AngularJS, JavaScript, JQuery, Bootstrap, CSS, Karma, Jasmine, Protractor, Apache HTTP Server, AWS APIs, EC2, Private Network, SPA, Maven, CI/CD.

**Client: Digital Nirvana, Hyderabad May,2014 – Jul,2015**

**Java Developer**

***Responsibilities:***

* Developed **Core Java** components to develop the transaction report. Stored the dynamic data on to Collection Objects and used the predefined methods to perform all DB related operations efficiently.
* Developed the application using **spring framework** that leverages model view layer architecture, also configured **Dependency Injection**. Used **Agile methodology** process in the development project.
* Worked on developing backend components and services using **Hibernate** and **spring**.
* Used **JMS** for the asynchronous exchange of critical business data and events among J2EEcomponents and legacy system. Involved in installing and configuring **Maven** for application builds and deployment.
* Deployed Applications on **Apache Tomcat Server** and used **CVS** as the version control manager.
* Wrote **SQL queries**, stored procedures, modifications to existing database structure as required for addition of new features using **Oracle database**.
* Supported and provided important feedback to various development teams with regards to grid computing and caching technology.
* Used **GIT** to check-in and check-out and co-ordinate among team members for **Version Controlling**. Designed and developed **REST web service** for validating address. Deployment of web, **enterprise java components**, **messaging components** and **multi-threading**.

**Environment:** Java, Spring, Spring JDBC, Spring Batch, SNS, Oracle, Maven, GitHub, REST, Apache Tomcat, Agile, Windows