**Sanjana Shanigaram**

**Email**: sanjanashanigaram1@gmail.com

**Phone**: +1 4693337461 **LinkedIn**: www.linkedin.com/in/ssanjana06

**PROFESSIONAL SUMMARY:**

* Full Stack Java Developer with over 10+ years of experience, leveraging a robust proficiency in **Java 8/11**, **J2EE**, and utilizing frameworks like **Spring, Spring Boot, Spring MVC, Spring IOC, Spring Security**, and **Hibernate** for robust backend solutions.
* Expertise in front-end technologies including **React**, **Angular 2/4/6/8/9/11/13/15, Node.js**, and proficient in **HTML5, CSS3, Bootstrap, jQuery, AJAX, SASS, and SCSS.** Skilled in crafting dynamic and user-centric **SPAs**, and proficient in responsive web design.
* Experienced in client-side scripting and **JavaScript** frameworks, including extensive work with **AngularJS, React.js, Redux, Backbone.js, Require.js**, and server-side JavaScript with **Node.js, Vue.js and Express.js**.
* Proficient in **Agile** and **Waterfall** methodologies, excelling across the **SDLC** with strong collaboration, communication, and time management skills, ensuring efficient and scalable code delivery aligned with business goals.
* Proficient in developing micro services and managing servers like **Apache Tomcat, WebLogic, WebSphere, JBoss, Nginx,** and **Jetty.**
* Skilled in database design and management with **Oracle, MS SQL, MySQL**, leveraging **NoSQL** databases like **MongoDB, DynamoDB, Cassandra, and CouchDB**. Expert in **PL/SQL, PostgreSQL**, and data processing with tools like **BigQuery and Spark**.
* Expert in **Windows**, **UNIX**, **Linux**, and **Mac OS** with proficiency in managing web servers.
* Proficient in **Python** scripting, programming with **Scala**. Also experienced in **TypeScript** for enhancing front-end development. Proficient in project management and documentation using **JIRA, Swagger**, and **Grafana**.
* Proficient in **AWS** (S3, ECS, EC2, Lambda, RDS, CloudWatch, SQS, Simple DB, Elastic Load Balancing, EBS, CloudFront) for serverless architectures, integrating **AWS Lambda** with API Gateway and **DynamoDB**.
* Skilled in cloud infrastructure provisioning with **Terraform** on **Google Cloud Platform** and **Azure**, focusing on virtual networks, compute engines, and auto-scalers in varied subnets.
* Experienced in migrating **MongoDB** workloads to **AWS DocumentDB**, ensuring minimal downtime, data consistency, and compatibility with existing applications.
* Proficient in managing and streaming messaging systems with a focus on event-driven architecture, using tools like **Apache Kafka, IBM MQ, RabbitMQ,** and **Spring JMS**. Experienced in enhancing asynchronous data flow and communication in distributed systems, with hands-on expertise in **Apache Kafka, Zookeeper**, and **JMS** messaging services.
* Implemented robust security protocols and features like **LDAP**, **OAuth 2.0**, **Spring Security**, SSL, and **JWT** for token-based authentication, ensuring the safeguarding of applications and sensitive data.
* Hands-on experience with big data technologies such as **Apache Spark**, skilled in data visualization with **Kibana**, providing actionable insights for strategic decision-making.
* Skilled in scripting test scenarios in **JMeter** using various protocols such as **HTTP, HTTPS, JDBC, FTP, SOAP**, and **REST**, and executing them to simulate realistic user behavior and workload.
* Developed frontend applications using **React JS** and backend services deployed on **Kubernetes**, ensuring seamless integration and scalability.
* Managed **Kubernetes** for containerized environments, optimizing deployment for high availability and automating scalable applications.
* Pleasant Experience in React.js for creating interactive UI's using One-way data flow, **Virtual DOM, JSX, React** Native concepts.
* Strong expertise in J2EE testing, proficient in developing and executing test cases with **JUnit, Mockito, Jasmine, Karma, Jest,** and **Cucumber**, emphasizing mock testing and **TDD** principles.
* Experienced in **Agile** methodologies, including **SCRUM**, **Kanban**, and XP, with advanced skills in logging using **Log4J** and **Splunk** for quality assurance in dynamic development environments.
* Specialized in Continuous Integration and Continuous Deployment (**CI/CD**) using Jenkins with **GitHub**, adept at constructing streamlined pipelines and effective application delivery through **Docker** container services.

**Educational Details Bachelor’s**: Jawaharlal Nehru Technological University, Hyderabad, INDIA. | **Passed year**: May 2011.

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| Languages | C, C++, Java 11/8, J2EE, JavaScript, Python 3.6.0, SQL |
| Web Technologies | jQuery, HTML, HTML5, CSS, CSS3, Bootstrap, TypeScript, AJAX, DOM, RWD, DHTML, XHTML, JSON, XML, SASS, SCSS, Microservices |
| JavaScript Technologies | AngularJS, Angular 2/ 4/ 6/ 8/ 9/ 11/ 13, React.js, NgRx, EXT.js, Backbone.js, Require.js, Node.js, Vue.js, Express.js |
| Java Technologies | Spring, Spring Boot, Spring Cloud, Spring MVC, Spring IOC, Spring Security, Hibernate, Servlets, JDBC, JSP, JSTL, Swing, EJB, Scala, JAXB, Struts, JDK1.8 |
| Database/RDBMS and ORM | Oracle, MS SQL Server, MySQL, MongoDB, Redis, Cassandra, CouchDB, PL/ SQL, PostgreSQL, Graph DB Dynamo DB, Boto3, NumPy, Panda’s library, Postgres, BigQuery, Spark |
| Web Services | RESTful, SOAP, WSDL, UDDI, AWS (Amazon Web Services), Axis2, Graph QL, OAuth, Elasticsearch |
| Operating Systems | Windows, UNIX, LINUX, Mac OS. |
| Testing Frameworks | JUnit, Jasmine, Karma, Mockito, Jest, Log4j, Cucumber, Splunk, TDD, Postman, Selenium |
| Web Servers | Apache Tomcat, Glassfish, WebSphere, WebLogic, JBoss, Nginx, Jetty, RBAC |
| Tools and Packages | GIT, IntelliJ IDE, Eclipse IDE, GitHub, Maven, Gradle, ANT, Jenkins, JIRA, Swagger, Grafana, Bitbucket, Docker, Kubernetes, Apache Camel, Kibana, JFrog Artifactory, YAML and Helm charts. |
| Cloud | AWS, AWS S3, AWS ECS, AWS EC2, AWS Lambda, AWS RDS, Cloud watch, GCP, Azure, OpenShift |
| Messaging | JMS MQ, IBM MQ, IBM MQ 8.0, Rabbit MQ, Apache Kafka, Amazon SQS |

**PROFESSIONAL EXPERIENCE:**

**Client:** **TD Bank, New Jersey Oct 2022 – Till Date**

# Role: Sr. Java Full Stack Developer

# Responsibilities:

* Develop dynamic, responsive user interfaces with **Angular 15**, TypeScript, HTML and CSS focusing on high-quality, maintainable code in main application.
* Responsible for the development of Single Page Applications (SPAs) using **Angular** frameworks to improve the productivity of the internal team.
* Designed web pages adhering to **WCAG** standards using **Angular's** accessibility features.
* Enhanced data stream processing in microservices using **Java 8** features like Streams, Collectors, and lambda expressions, streamlining code, and improving backend efficiency.
* Leveraged **Java 8**'s new Date and Time API for precise and efficient handling of time-related operations in application logic.
* Enhanced performance with **Java 11**'s improved garbage collection and HTTP Client API; apply new lambda and switch expression syntax for readability.
* Streamline code using **Java 11** features like var in lambdas, enhanced string handling for efficiency and improved readability.
* Enhanced stability and memory management using **Core** Java's exception handling and garbage collection, ensuring efficient error handling and resource utilization.
* Designed robust **RESTful APIs** using **Spring MVC** and **JAX-RS** within a microservices architecture, ensuring seamless front-end communication and durable endpoints.
* Utilized **Spring Cloud** for distributed systems management, facilitating configuration management and service discovery in microservices.
* Utilized **Python** for specialized backend scripting tasks, adding depth and versatility to the backend technology stack.
* Employ **Scala**'s functional programming and Akka's actor model for efficient, concurrent backend processing, ensuring compatibility with Java systems.
* Designed and implemented complex forms using **Angular Reactive Forms**.
* Utilized **AJAX** with **JSON** and **XML** for efficient, asynchronous data exchange in web services.
* Optimized **Oracle databases** in Java applications using advanced **SQL, PL/SQL**, and efficient indexing for high performance and data integrity.
* Integrated **Oracle** databases with Java back-ends using **JDBC** and **JPA/Hibernate** for seamless data communication and CRUD operations, Integrated with API Gateway for scalable **RESTfu**l APIs.
* Deployed **Cassandra** for scalable, high-throughput databases with optimized, consistent data handling.
* Leveraged **Cassandra's** decentralized architecture for global distribution and fault-tolerant resiliency.
* Tracking and managing project issues using **JIRA**, leveraging its **agile** board and custom workflows.
* Containerized **microservices** with **Docker**, focusing on scalability and consistent deployment across diverse environments.
* Efficiently handle high-volume data with **Spring Batch** in microservices and **CI/CD** pipelines, using chunk-based processing and automated retry/skip logic for resilience.
* Knowledgeable in integrating Java applications with **IBM** WebSphere MQ for reliable messaging and asynchronous communication.
* Used **JMeter** to perform API testing by sending **HTTP** requests, validating **JSON/XML** responses, and verifying API endpoints' functionality, performance, and compliance with specifications such as **RESTful API** standards or **SOAP** protocols.
* Integrated external data sources and **APIs** into Java-based trading platforms to provide real-time market data and enhance decision-making capabilities for traders.
* Designed **Kubernetes** strategies for microservices, ensuring scalability and high availability.
* Utilized **Kubernetes** and **Docker** for efficient deployment and resilience.
* Designed **Kafka** clusters for messaging and implement **Kafka** Streams for real-time processing in microservices.
* Utilized **Kafka** Connect for seamless integration with external databases, services, and applications, automating the movement of data into and out of Kafka.
* Oversee **CI/CD** pipelines using Jenkins for automation, **Maven** for dependency management, and **Git** for version control.
* Implemented **Groovy scripts** in Jenkins pipelines to automate and enhance **CI/CD** processes, particularly for managing and deploying Java-based applications.
* Conducted system monitoring and performance tuning using **Elasticsearch** and **Kibana**, focusing on high application performance and availability.
* Collaborated effectively with cross-functional teams to design, develop, and deploy Java applications on the **IBM** WebSphere platform.
* Implemented **Graph DB** for complex data relationships and optimized graph-based queries; enhance data modeling with its schema-less, scalable design.
* Implemented robust security measures with **Spring Security** using **OAuth** for authorization and **LDAP** for authentication.
* Managed **AWS EC2** for scalable web applications and **S3** for secure storage, integrating **AWS Lambda** with API Gateway for serverless **Python** and **Node.js** scripts.
* Utilized **AWS IAM** for robust access control and user management; implement **AWS RDS** for database management.
* Leveraged **AWS Glue** for serverless data integration and ETL, streamlining data transformation and analytics processes.
* Utilized **GraphQL** with **Node.js** for efficient, structured API queries, dynamic data retrieval, and real-time updates, minimizing over-fetching and enhancing interactivity.
* Optimized application performance and response times by implementing **Redis** for caching frequently accessed data, leveraging its in-memory data storage capabilities.
* Tested APIs with **Postman** and ensured code quality with **Sonar Lint**, Documented APIs using **Swagger**, ensuring clear and concise API documentation.
* Developed robust unit and integration tests using **JUnit**, **Mockito**, and **Selenium**, adhering to Test-Driven Development (**TDD**) principles for comprehensive end-to-end testing.
* Developed and maintained unit tests for Angular applications using **Jasmine** and **Karma**, focusing on validating front-end components.
* Managed application server environments with **WebLogic**, optimizing for performance and reliability.

**Environment**: Angular 15, TypeScript, HTML, CSS, JSON, NgRx, RxJs, Redux, Java 8, Java 11, JDK 1.8, Spring Boot, Spring batch, Spring MVC , Spring Cloud, REST, Python, Scala, ,Graph QL, Graph DB, Oracle, SOLR, Cassandra, Hibernate, Docker, Kubernetes, Terraform, Kafka, Jenkins, Groovy scripts, Maven,Git, Elasticsearch, Kibana, OAuth, LDAP, Spring Security, AWS (EC2,EKS,IAM, Lambda, S3,Glue), Agile, TDD, Junit, Mockito, Selenium, Jasmine, Karma, WebLogic, SQL, PL/SQL**,** Jira, Swagger, Postman, SDLC.

**Client: Fresenius Health Care, Illinois**.*April 2021 to Sep 2022*

**Role: Java Full Stack Developer**

**Responsibilities:**

* Created dynamic user interfaces using **ReactJS** for its **virtual DOM** feature, **JavaScript** for client-side scripting, and **HTML/XML and CSS** for webpage structure and style.
* Developed Ul using **ReactJS** and **Redux** and **NextJS** and used **Redux**-**Promise** in application to retrieve data from Back-End.
* Implemented **Redux** for state management and **Redux Saga** for asynchronous **API** calls, including middleware integration for **RESTful** services and side effects management in React applications.
* Managed data exchange and **API** interactions using **AJAX, JSON, XML**, ensuring robust communication between frontend and backend.
* Managed data serialization and deserialization using **JAXB** for **XML** binding, and **JSON** for data representation.
* Involved writing modules using **React** as a view engine to build component-based application.
* Created reusable React components and managed application state with **Redux.**
* Applied **redux middleware** using **redux-promise, redux-thunk and redux-saga**.
* Used **redux-form** to manage form state in **redux.**
* Applied Oriented Programming techniques to track UI components being added to **DOM** and to change the behavior of framework methods.
* Experienced in using **React** **JS** for web development and used **axios** and **redux** **saga** for making calls (GET, POST, PUT, DELETE) for triggering or fetching data from **Restful** **API’s** in reporting tool application.
* Involved in the development of server side Rendered progressive web applications built with **react**.**js**, **Next**.**js** , Fire base and material – UI using **HTTPS**, Web app manifests , service works and browser API’S in reporting tool application.
* Utilized **NgRx** and **RxJs** for state management, aligning with **Agile** and **Kanban** methodologies for rapid, responsive development.
* Utilized **Java 11**'s HTTP Client API to build robust and responsive applications.
* Optimized multi-threading for **Core Java** applications and minimize memory leaks in Core Java code.
* Architect and deploy **Spring Boot** microservices; enhance event processing with **Kafka** in distributed systems.
* Build efficient **Kafka** producers and consumers for reliable messaging in distributed systems.
* Applied **Spring Framework** in **MVC** for modular and loosely coupled backend development.
* Architect and deploy Spring Boot microservices, integrating **Spring IOC** for dependency injection and **Spring API** for creating loosely coupled, modular applications.
* Optimized database interactions and management with **Hibernate ORM** and **JPA** for efficient data access and transaction handling.
* Utilized **React Hooks** for functional component development, optimizing code readability and maintaining scalability.
* Leveraged **Spring Boot** **4.x** for microservices, emphasizing auto-configuration and rapid development.
  + Implemented **JWT** with **Spring Security** for secure authentication and integrate Spring Boot Actuator for enhanced security and management of microservices.
* Developed and maintained **RESTful** APIs, ensuring seamless integration with external systems.
* Designed and implemented complex user interfaces (UIs) with **React Router** for navigation and conditional rendering based on application state.
  + Boost application performance and scalability using **NodeJS** for server-side scripting and Core Java for backend optimization, ensuring robustness and efficiency.
  + Implemented reusable **React** components, reducing code redundancy and improving development efficiency across multiple applications.
* Created advanced web applications with **JSP** and **JSF**, incorporating **AJAX** for real-time user interactions and data updates.
* Deployed applications with **Docker** and **Kubernetes,** automate **CI/CD** using **Jenkins** and **Maven**
* Utilized Enterprise JavaBeans (**EJB**) for transactional services and implement **Log4j** for comprehensive logging.
* Developed and execute unit tests using **JUnit**, **Mockito**, and **Cucumber**, adhering to **TDD** and **BDD** methodologies for high-quality software.
* Simplify complex **JavaScript** with **jQuery** to enhance frontend development efficiency and user interface.
* Designed and implement **MongoDB** schemas, use **MongoDB** aggregation for analysis, apply replication and sharding for **NoSQL** scalability and efficiency.
* Applied advanced build automation with **Maven** and manage **MongoDB** versioning with **Mongock4** and **5**.
* Architect and deploy Java-based applications on **GCP**, utilizing services like Google App Engine and Google Kubernetes Engine for scalable, managed environments.
* Integrated **GCP**’s Cloud SQL and Firestore for robust, scalable database solutions in Java applications, ensuring efficient data management and storage.
* Implemented **GCP**'s Pub/Sub for real-time messaging and event-driven architectures, enhancing communication in distributed systems.
* Utilized **GCP**'s Cloud Functions for serverless Java and Python functions, facilitating scalable, event-driven backend processes.
* Employ **GCP's Apigee** API Management for designing, securing, and monitoring APIs, ensuring efficient interaction between microservices and external applications.
* Utilized **Kibana** for log analysis, real-time monitoring, and data visualization.
* Integrated **Redis**/**Memcached** for caching to enhance application performance.
* Implemented ELK stack by integrating **Kibana** with **Elasticsearch** and **Logstash** for comprehensive log management.
* Developed advanced front-end features with **React.js**, focusing on responsive design and interactive engagement.
* Utilized **Spark** for large-scale data processing and analytics, employing **BigQuery** for efficient handling of extensive datasets.
* Processed large-scale data with **Scala** and **Spark**, utilizing **Spark SQL** for structured data querying and **GraphX** for graph data analysis. Integrate data between **BigQuery** and other sources.
* Deployed and managed applications on **Apache** **Tomcat**, optimizing for performance and scalability.

**Environment**:  Java 8, Java 11,Core Java, Spring, Spring Boot, Spring IOC, Spring API, Kafka, Hibernate ORM, HTML5, CSS3, JavaScript, React.js, Agile, Redux, JSP, JSF, AJAX, JSON, XML, JMS, Docker, Kubernetes, Solr, Chef, Jenkins, Maven**,** GIT, Rest Services, Apache Tomcat, MongoDB, GCP, Log4j, JUnit, Mockito, Cucumber, jQuery, Scala, Spark, RBAC, Kibana, TDD, BDD, BigQuery, Redis/Memcached, JWT, Spring Security, YAML and Helm charts.

**Client: State of Colorodo, CO.** *July 2019 to Mar 2021*

**Role: Java Full Stack Developer**

**Responsibilities:**

* Created responsive front-end components using **AngularJS**, prioritizing user-friendly design and interactive experiences.

• Designed modern web pages using **HTML5** and **CSS**, supplemented by **Bootstrap** for responsive layouts.

• Write clean, maintainable **JavaScript** code for front-end development, ensuring cross-browser compatibility.

* Developed enterprise-level applications with **Core** **Java** and **J2EE**, ensuring robust backend functionality and system efficiency**.**
* Optimized code using **Java 8's** concurrency features such as Completable Future for asynchronous programming, enhancing application responsiveness and scalability.

• Implemented scalable **microservices** using **Spring Boot**, focusing on independent deployment and system flexibility.

* Upgraded to **Spring Boot 3.x** for new features and build microservices with Spring Cloud.

• Build rich user interfaces with the **JSF** Framework, enhancing user engagement and interactivity in enterprise applications.

• Employ **Spring Security** for data protection, user authentication, and safe guarding against security threats.

• Tested web services using **SoapUI**, validating SOAP based web services and their responses.

* Implemented **SOAP** web services using **JAX-WS** for interoperable machine-to-machine communication,

• Optimized data retrieval with **MongoDB** indexing and configure cloud clusters with **MongoDB Atlas.**

* + Written and maintained unit tests with **Junit** using Test-Driven Development **(TDD)** principles, ensuring code reliability and robustness.
  + Used **Azure** DevOps for automated **CI/CD**, Implement event-driven microservices with Azure Functions.
  + Harness **Azure** cloud services for scalable, reliable hosting using Azure App Service and event-driven computing with Azure Functions.
  + Deployed and managed containerized applications efficiently using **Azure Kubernetes Service** (AKS), integrated with Azure monitoring tools.
  + Utilized **Azure's** support for **PostgreSQL** to manage and scale relational databases seamlessly, ensuring efficient data storage and high availability.
  + Utilized **Docker** for containerization, and **Kubernetes** and **Terraform** for orchestrating and managing containerized applications.
  + Utilized **RabbitMQ** for efficient message brokering across distributed microservices, ensuring scalability and reliability

• Implemented comprehensive logging with **Log4j** for effective application monitoring and debugging.

• Managed project workflows and issue tracking with **JIRA**, streamlining development processes.

• Implement front-end optimizations with advanced **JavaScript** and **AngularJS**, enhancing user experience and performance.

• Designed and optimized **MongoDB** database schemas and queries for efficient data storage and retrieval.

* + Implemented robust security measures with **Spring Security** to protect against vulnerabilities and ensure code quality with **Spring Boot testing**.

• Implemented caching strategies and optimize server-side performance with **NodeJS** for efficient data operations.

* Apply **Selenium**, **Jasmine**, and **Karma** for comprehensive front-end and back-end automated testing, ensuring application reliability.
* Leveraged **Apache Tomcat** for server management, focusing on deployment and environment handling.

**Environment**: Java,Java8, J2EE, JDK 1.8, Spring Boot, Microservices, JSF Framework, Spring Security, AngularJS, NodeJS, RabbitMQ, JavaScript, Apache Tomcat,Waterfall, HTML5, CSS, Bootstrap, SOAP, MongoDB, PostgreSQL, Graph DB, JUnit, Jenkins, Log4j, JIRA, Git, AZURE, Docker, Kubernetes, Terraform, TDD, Selenium, Jasmine, Karma

**Client:** *Lowe’s, NC***.** *June 2018 to June 2019*

**Role: Java Full Stack Developer**

**Responsibilities:**

* + Developed scalable **Java** **8** and **11** applications focusing on **Lambda** Expressions, **Streams**, and Spring Boot microservices, employing Spring Framework in MVC design for backend operations, and adhering to Test-Driven Development (**TDD**) principles.
  + Managed microservices architecture using **Spring** **Boot** for scalable solutions, deploying services with Docker for containerization, and configuring spring-config.xml.
  + Experience in using **React JS** code for making calls (GET, POST, PUT, DELETE) for triggering or fetching data from Restful API's
  + Worked with **Redux** reducers to capture the essence of flux stores and gradually migrated from existing Flux towards **Redux**.
  + Integrated custom Docker images with Spring Boot **microservices** for containerization, manage database interactions with **Hibernate** **ORM**, and apply Dependency Injection for development efficiency.
  + Oversee **Kubernetes**-managed **Docker** clusters, utilizing Jenkins for **CI/CD**, JFrog Artifactory for artifact management, GIT for version control and implement **RBAC** and administer **RHEL** servers, configure network policies in Kubernetes.
  + Implemented Spring Security with Hibernate for secure applications, develop **Kafka** clusters for messaging, and handle real-time data analytics with **Kafka** Streams.
  + Leveraged **OpenShift's** integration with Jenkins and other **CI/CD** tools to automate build, testing, and deployment workflows.
  + Implemented the multi datacentre cluster consisting of **Cassandra** rings, leverage **Apache Tomcat** for server management.
  + Developed queries in UI for receiving live polling data from **GraphQL** and map the data into the table.
  + Designed ETL processes with **AWS** **Glue**, manage **AWS** **Redshift** data migration, automate cloud processes with **AWS** Cloud
  + Formation, manage Docker deployments using **AWS Elastic Beanstalk** and **AWS CloudWatch** for monitoring.
  + Developed REST microservices with Spring Boot, handle CI/CD with Kubernetes, data caching with **Redis**, and asynchronous processing with **Amazon SQS**; leverage **Node.js** for server-side development.
  + Integrated **Drools** with Java for rule-based logic, implement database access for **MySQL** on relational data storage.

**Environment**: Java 8, Java 11, Spring, Spring Boot, Hibernate, Reat.js, Redux, HTML5, CSS3, JavaScript, TypeScript, JSF, SQL, Cassandra DB, Hibernate ORM, Docker, Kubernetes, Jenkins, JFrog Artifactory, GIT, Gradle, AWS (Glue, Redshift, Lambda, EC2, S3, Kafka, Amazon SQS, Elasticsearch, Spring Security, OAuth, RBAC, Eclipse, Tomcat, Red Hat Enterprise Linux (RHEL), Drools, Redis, ECM workflows, YAML, and Helm charts.

**Client:** *Kairos Technologies - Hyderabad, India From Oct 2011 - Dec 2016*

**Role: Java Developer**

**Responsibilities:**

* Developed **Java/J2EE** applications focusing on backend logic, data processing, and system integration, utilizing **Core Java** for efficiency, and integrating scalable **microservices** with **Spring Boot**.
* Utilized **Servlets** for effective web application request-response handling, enhancing server-side processing and frontend integration.
* Used the **Spring** framework to create scalable backend applications, focusing on dependency injection and aspect-oriented programming.
  + Implemented **XML**-based configurations for structured data management, easy integration, and enhanced maintainability.
  + Managed and optimize **Oracle** databases with complex **SQL** queries and procedures, focusing on performance, data integrity, and applying best practices in design and indexing.
  + Developed web services using **JAX-R** and **JAX-WS**, emphasizing system interoperability, API design, and **SOAP**-based services, alongside writing, and optimizing **PL/SQL** scripts and **Hibernate ORM** for Oracle performance.
  + Configured and maintained **Apache Server** for web hosting, focusing on performance, security, and scalability, and use Struts framework for MVC-based web application development.
* Conducted performance testing and follow Test-Driven Development (**TDD**) principles for web applications using **Apache JMeter.**
* Developed and consume **RESTful** web services for secure and scalable data exchange between systems.
* Managed data with **DynamoDB**, focusing on performance and scalability for handling unstructured data.
* Implemented **Git** for version control in enterprise-level Java projects, ensuring effective code management and collaboration.
  + Developed web applications using **MVC Spring**, ensuring separation of concerns, testability, and flexible integration; write and maintain unit tests with JUnit adhering to **TDD** principles.
* Integrated **Hibernate** for streamlined database interactions and employ **Maven** for scalable and maintainable service development.
* Utilized **JIRA** for project management and issue tracking, ensuring organized workflows and timely software delivery.

**Environment**: Java, JSP, JDK 1.8, HTML, CSS, Servlets, XML,Agile, SOAP, WSDL, Oracle, Eclipse, JAX-R, JAX-WS, PL/SQL, Apache Server, Struts, DAO, AOP Spring, MVC Spring, Hibernate, JUnit, Maven, RabbitMQ, J2EE, Git, WaterFall, Spring Framework, Core Java, REST Web Services, JavaScript, HTML5, CSS, SASS, JIRA, Selenium, DynamoDB (NoSQL), AngularJS, AJAX, Apache JMeter.

**Certifications:**

* Certified Introduction to Microsoft Azure Cloud Services.
* Certified Microsoft Azure Services and Lifecycles.
* Certified Object-Oriented programming with Java.
* Certified Java Programming: Arrays, Lists, and Structured Data.