# **Senior Java Full Stack Developer**

**Manish Chandra**

manish1c1h@gmail.com513-540-7040

**PROFESSIONAL SUMMARY:**

* **9+ Years** of experience as a **Full Stack Java Developer** in object oriented programming and development of Multi-Tier distributed and Enterprise applications in **Agile (SCRUM)**, waterfall environments. Strong background in Object Oriented Design **(OOD)**, Object Oriented Analysis **(OOA)** and drawing Unified Modeling Language (UML) diagrams.
* Extensive experience in front-end technologies **like HTML5, JQuery, TypeScript, Bootstrap, AJAX, JavaScript, CSS3/4, NodeJs, ReactJs, Redux, BackBoneJs, Angular 4/6/9/14.**
* Implemented the latest React features such as **React Context API, React Hooks, Stateless Functional Components, Error Boundaries,** and middleware such as **Redux Thunk, and Redux-Saga.**
* Experience in developing applications using **MEAN & MERN** stack technologies along with **SOAP** web services using **WSDL, SOAP, JAXB, CXF, AXIS, JAX-WS and RESTful web services using JAX-RS, CXF and JERSEY APIs.**
* Experience in implementing **Swagger** for microservices for documenting **Rest API.**
* Expertise in web development applications using **J2SE, Servlets, JSP, EJBs (Session Bean, Entity Beans, JMS/MQ Series), JDBC, RMI, JNDI, Java Mail, XML Parsers (DOM and SAX), JAXP, JAXB, Java Beans.**
* Expert level coding skills in web application development using **Struts, Spring, JSF, Custom Model view Controller (MVC) and Hibernate frameworks.**
* Experience in various **Spring modules** like **Spring Boot, Spring MVC, Spring Cloud, Spring Security.**
* Expertise on servers using **ApacheTomcat & HTTP Server, Web Sphere, WebLogic & JBOSS Application server.**
* Proficient in working with design patterns including **Singleton, Factory, Session Façade, MVC, Front Controller, Proxy, Service Locator, Builder Pattern, Business Delegate, DTO, DAO, VO** and **Command Patterns.**
* Built the code in **Project object models (POM)** by using **Maven, Ant, Gradle** and used **Jenkins** for CI/CD integration.
* Experience in utilizing Java 1.8 features like Lambda expressions and Stream API for Bulk data operations on Collections which would increase the performance of the Application.
* Secured API using **OAuth2.0** token-based authentication/authorization scheme using **Spring Security.**
* Experience in **Microservices** development using **Spring boot** and deployment in **Pivotal Cloud Foundry (PCF).**
* Strong understanding of **Microservices** patterns **- Circuit Breaker/Hystrix, DDD, Aggregator, Proxy Chaining**.
* Proficient experience in using java development IDEs like **Eclipse, My Eclipse, NetBeans and RAD.**
* Proficient in Asynchronous messaging using **JMS** to exchange information **and Active MQ, Rabbit MQ**, **and Kafka.**
* Worked on Database Design, data modeling, data transformation and data Migration in the databases includes **Oracle, DB2, MySQL, PostgreSQL and SQL Server.** Worked on **NoSQL** databases like **MongoDB, Cassandra, DynamoDB.**
* Performed Database operations like normalization and written **SQL queries and PL/SQL** – Stored Procedures, Functions, Triggers, Sequences, Indexes, cursors and Views.
* Experienced on various testing frameworks for front end/user interface like **Karma, Jasmine, Jest, Mocha, Chai etc.**
* Worked on **microservices** to build **Docker** containers and deploy **to Dev, iTest, Scale, UAT (SIT), PROD** environments**.**
* Used **Continuous Integration / Continuous delivery / (CI/CD)** tools Docker, Kubernetes, Jenkins, SonarQube and looper to deploy this application to **AWS along with GIT** for Version Control of code for incremental development.
* Experience in using various **Amazon Web Services (AWS)** Components like **EC2**for virtual servers, **S3, VPC, SQS, ELB, RDS, RedShift, AutoScaling, IAM, Cloud Watch, ELK, Kinesis, EMR.**
* Proficient on **Azure Cloud Services**, leveraging **Platform as a Service (PaaS)** offerings such as **Azure App Service**, **Azure Functions, Azure Logic Apps** along with **Azure Kubernetes services (AKS**) and used **Azure service bus** as messaging broker, **Azure API** as **API gateway** with **Azure Cosmos DB** for storing and accessing distributed data.
* Implemented **Dead-Letter Queues (DLQ)** to handle message failures and prevent data loss.
* Optimized message consumption using **Apache Kafka Consumer Groups** with backpressure handling.
* Integrated **Pact** for **Consumer-Driven** Contract Testing to ensure backward compatibility of **APIs.**
* Automated security testing with **OWASP ZAP** to identify vulnerabilities in **APIs** andfront-end applications.
* Secured **microservices** with **Kubernetes Network Policies** and **RBAC (Role-Based Access Control).**
* Implemented **Secrets Management** using **AWS Secrets Manager** and **HashiCorp Vault.**
* Optimized **indexing strategies (B-tree, Hash, Bitmap Indexes)** to enhance database performance.
* Experience Utilizing **Circuit Breaker Pattern** with **Resilience4J** to prevent cascading failures in microservices.
* Working skills on implementation of the version control tools like **CVS, VSS, SVN, GIT and Clear case.**
* Expertise in Unit testing using **Junit & Mockito,** Integration testing & smoke testing etc and **JIRA** for bug tracking.

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| **Languages** | JAVA 1.7/8/11/17, Python, SQL, PL/SQL,Kotlin |
| **Frameworks** | Spring, Spring Boot, Spring MVC, Hibernate, Struts |
| **Web Technologies** | Angular 4/6/9/14, Bootstrap, jQuery, JavaScript, CSS3, TypeScript, AJAX, Node JS, ReactJS, Redux, Backbone JS, Node JS |
| **Database** | Oracle, DB2, SQL Server, MySQL, PostgreSQL, NoSQL: MongoDB, Cassandra, DynamoDB |
| **Testing Frameworks** | JUnit, Mockito, TestNg, TDD, BDD |
| **Web Services** | JAX-WS, JAX-RPC, JAX-RS, ngRx,RxJs,SOAP, WSDL, REST, GraphQL Microservices |
| **Web/App-Servers** | JBoss, Tomcat, WebSphere, WebLogic |
| **Version Control Systems** | CVS, SVN, GitHub, Clear Case |
| **Build Tools** | Ant, Maven, Jenkins |
| **Cloud Technologies** | Azure, AWS(EC2, S3, EBS, ELK, CloudWatch, CFT, RDS, IAM, VPC, SQS, EMR) Docker, PCF |
| **Tools** | Rational Rose, SQL Developer, TOAD, Hudson, SOAPUI, Rest Client UI, Postman, Grunt |

**PROFESSIONAL EXPERIENCE:**

**T-Mobile, Issaquah, WA From Aug 2023\***

**Sr. Java Full Stack Developer**

* Built dynamic, interactive front-end applications using Angular 14, HTML5, JavaScript, JQuery, Bootstrap, and AJAX.
* Developed AI-driven service recommendations using Spring Boot microservices.
* Optimize UI performance with advanced techniques such as lazy loading, change detection strategies, and tree-shaking.
* Built adaptive UI modules by leveraging Angular 14’s ComponentFactoryResolver and standalone components for runtime component injection.
* Integrated RESTful, GraphQL APIs with Angular 14's HttpClient module, utilizing interceptors and standalone services.
* Implemented scalable, fault-tolerant, and high-performance systems using Java 17, Spring Boot, MVC, and Hibernate.
* Leveraged React Hooks (useState, useEffect, useReducer) to manage state and side effects in functional components.
* Implemented global state management using Redux, Redux Toolkit, or Context API for scalable React applications.
* Leveraged Java 17 features like sealed classes, pattern matching for switch expressions, and records to simplify and enhance code readability.
* Optimized garbage collection using G1GC and ZGC for improved application performance under heavy load.
* Enhanced performance with Foreign Function & Memory API (FFM) for efficient interaction with native libraries.
* Improved modularity and runtime performance by leveraging Java 17’s enhanced classloaders and ModuleLayer APIs.
* Enhanced rendering performance by using React.memo, useCallback, and useMemo to prevent unnecessary re-renders.
* Configured Cross-Origin Resource Sharing (CORS) policies and CSRF protection for security in Spring Boot applications.
* Developed microservices with service discovery, API gateways, circuit breakers, and event-driven architecture patterns.
* Optimized REST API performance through caching strategies (Redis, Memcached), pagination, and query optimization.
* Implemented secure REST API endpoints using OAuth 2.0, JWT, and role-based access control mechanisms.
* Optimized SOAP message sizes through MTOM/XOP for binary data transmission and Monitored SOAP APIs using enterprise monitoring solutions such as Dynatrace, New Relic to track performance and availability.
* Secured SOAP services with WS-Security standards, implementing encryption, digital signatures, token authentication.
* Optimized database schemas and queries for Oracle, MySQL, NoSQL, and MongoDB to handle complex transactions.
* Integrate and consume external services using SOAP and RESTful APIs with tools like Apache CXF.
* Establish contract testing for APIs and microservices to ensure compatibility across distributed systems.
* Developed robust RESTful APIs using Spring Boot with features like HATEOAS, exception handling, and request validation to ensure scalability and maintainability.
* Developed data persistence layers using Spring Data JPA, Hibernate, and advanced query capabilities with Criteria API and native SQL. Also caching with Spring Cache, Redis.
* Utilized Stream API enhancements (toList(), mapMulti()) to write cleaner and more efficient functional code.
* Performed detailed JVM diagnostics and fine-tuned JVM parameters specific to Java 17 using VisualVM, JConsole, and JFR to identify memory leaks, thread contention, and performance bottlenecks.
* Implement secure coding practices and protocols like OAuth2, JWT and ensure compliance with OWASP standards for application security.
* Implemented Camunda BPMN workflows to automate business processes such as document approvals, invoice processing, and customer onboarding.
* Designed and secured RESTful APIs using OAuth2, JWT, and HTTPS with TLS encryption.
* Used OAuth 2.0 Authorization Code Flow with Proof Key for Code Exchange (PKCE), ensuring secure authentication without exposing client secrets.
* Configured Prometheus to collect metrics from Spring Boot applications.
* Set up Grafana dashboards to visualize response times, error rates, and system performance.
* Implemented multi-tenancy in Spring Boot applications using Hibernate and custom database connection pools.
* Maintained CI/CD pipelines with Jenkins, Maven, and GIT to ensure efficient build, test, and deployment workflows.
* Configured Spring Boot Actuator and Implemented logging, monitoring, and alerting frameworks using tools like ELK stack, Log4j, and Prometheus/Grafana for production-grade observability.
* Dockerized Spring Boot application for deployment in container orchestration platforms like Kubernetes and OpenShift.
* Test automation using JUnit, Mockito, Jasmine, and Karma for comprehensive unit, integration, and regression testing.
* Harden microservices and APIs against vulnerabilities such as SQL injection, CSRF, and XSS.
* Prototype new technologies such as serverless architectures, event streaming with Kafka, or machine learning pipelines.
* Orchestrated application using Docker, Kubernetes, for multi-cluster management, scaling, service mesh integration.
* Built data pipelines using AWS Glue, Kinesis, and EMR, integrating with S3 and Redshift for storage and analytics.
* Deployed machine learning models using AWS SageMaker and integrating with Lambda and API Gateway for inference.
* Developed serverless solutions using AWS Lambda, API Gateway, and DynamoDB for highly scalable and cost-effective.
* Automated infrastructure provisioning, scaling, and monitoring using tools - Terraform, Jenkins, AWS CloudFormation.
* Implemented cloud-native applications on AWS and PCF, leveraging services such as EC2, S3, RDS, Lambda, and EKS.
* Utilized Kotlin alongside Java to develop microservices, improving code readability, maintenance, and integrating seamlessly with Spring Boot frameworks.
* Leveraged Kotlin coroutines for efficient asynchronous programming, enhancing the performance of applications by managing background tasks with less overhead.

**Bank of America, Charlotte, NC Sep 2021 to Jul 2023**

**Sr. Java Full Stack Developer**

**Responsibilities:**

* Used Angular 9 to parse data fetched from APIs and used Backbone JS to perform CRUD operations on the DB server.
* Designed advanced animations using Angular’s AnimationBuilder and AnimationTrigger APIs, incorporating Angular 9 enhancements for smoother transitions.
* Enhanced application performance by Angular 9 component based development for future angular transitions.
* Developed HTTP requests using RxJS observables to communicate between server and client with Angular 9.
* Developed various screens for UI using Angular 9, used predefined components from NPM (Node Package Manager).
* Integrated REST and GraphQL APIs into React applications using Axios, Fetch API, or Apollo Client.
* Wrote comprehensive tests for React components and APIs using Jest, React Testing Library, and Enzyme.
* Built type-safe React applications with TypeScript, ensuring improved maintainability and scalability.
* Developed micro-frontend applications using React with Webpack Module Federation for independent deployment.
* Implemented Spring for Dependency Injection (DI), support for Data Access Object (DAO), integrated with Hibernate.
* Used Hibernate as ORM to map Java classes to database tables. Involved in setting up configuration properties, ORM mapping and CRUD operations with databases.
* Designed idempotent APIs using @Transactional and optimistic locking to prevent duplicate requests.
* Implemented API Gateway with Spring Cloud Gateway for intelligent request routing and load balancing.
* Designed and secured RESTful APIs using OAuth2, JWT, and HTTPS with TLS encryption.
* Registered all microservices with Netflix Eureka, allowing dynamic service discovery without hardcoding service URLs.
* Migrated monolithic application to Microservice and Developed Microservices using Spring boot services on AWS.
* Implemented all the functionality using Spring IO / Spring Boot, Thymeleaf and Hibernate ORM.
* Executed JavaEE components using Spring MVC, Spring IOC, spring transactions and spring security modules.
* Used Node JS to interchange data between servers. Used NodeJS, Express JS to create MVC validations with MongoDB.
* Consumed Microservices with SOAP - JAX-WS, and RESTful web services - REST Templates, JAX-RS, CXF and JAXB.
* Integrated SOAP APIs with enterprise systems such as SAP ERP, Salesforce CRM, and Mulesoft ESB, Apache Camel.
* Used Microservices architecture with Spring Boot, Apache Kafka message broker, deployed to AWS.
* Configured API gateways (AWS API Gateway, Kong) to implement rate limiting and throttling for REST APIs
* Developed REST Microservices using the Spring Boot, Spring Data with JPA (Java Persistence API) and Used Swagger to interact with API endpoints of the Microservices.
* Implemented event-driven architectures using Spring Boot with Kafka, and ActiveMQ for asynchronous communication.
* Used Hibernate as ORM to map Java classes to DB tables. Configured properties, ORM mapping and CRUD operations.
* Implementing Oracle, SQL Queries & PL/SQL - stored procedures, functions, triggers, sequences and built-in functions to retrieve and update data from the databases.
* Secured applications using Spring Security with OAuth 2.0, JWT, and role-based access control (RBAC).
* Conducted comprehensive performance benchmarks on Java 11 using JMH (Java Microbenchmark Harness) to validate and improve runtime optimizations.
* Composed and executed MongoDB scripts to insert and update NoSQL database and reporting.
* Integrated Prometheus and Grafana dashboards for real-time performance monitoring.
* Leveraged AWS Lambda for serverless task execution, reducing operational costs.
* Built CI/CD pipelines using AWS CodePipeline, CodeBuild, and CodeDeploy to automate build, test, and deployment.
* Implemented centralized logging with Logback, ELK Stack, and distributed tracing with OpenTelemetry or Zipkin.
* Configured VPCs, subnets, NAT Gateways, and Security Groups to ensure secure and isolated network environments.
* Designed disaster recovery strategies using AWS services - CloudEndure DR, Route 53, and S3 Cross-Region Replication.
* Setting up databases in AWS using RDS, Storage using S3 buckets and configuring instance backups to S3 buckets.
* Build the services utilizing the Pivotal Cloud Foundry (PCF) as a platform as a service for Restful services.
* Implemented system logging solution using ElasticSearch, Kibana and Logstash (ELK).
* Created and managed cloud VMs with AWS EC2 Command line clients and AWS management console.
* Developed BDD tests using Cucumber by writing step definitions. Developed Selenium code in Java 11 for Cucumber.
* Project migration into cloud environment using Kubernetes, Docker container.
* Worked with unit testing of Angular applications using Jasmine and Karma and Spring Boot applications using JUnit, Mockito and performed end to end testing using Protractor.
* Designed and implemented distributed architecture using Spring Boot microservices, Docker, and Kubernetes for scalability.
* Utilized event-driven architecture with Kafka for asynchronous communication between distributed services.
* Implemented service discovery and API gateways for load balancing and fault tolerance in a distributed system.
* Utilized Kotlin to create scalable microservices that improved application performance and responsiveness in cloud environments.
* Enhanced existing systems by integrating Kotlin with Spring Boot, resulting in improved application architecture and faster development cycles.
* Participated in Agile ceremonies, contributing to team discussions on best practices for using Kotlin and improving overall code quality**.**

**Environment**: Java 1.8/11, Hibernate, kotlin,Angular 9, AWS, Spring Boot, MVC, Microservices, JavaScript, JQuery, Bootstrap, SOAP/REST, Jenkins, PCF, Kubernates, Docker, Mockito, JUnit, Jasmine, Karma, Apache CXF, Oracle, MySQL, NoSQL, Maven.

**Best Buy, Minneapolis, MN May 2019 to Aug 2021**

**Role: Full Stack Developer**

**Responsibilities:**

* Involved in Construction of UI using HTML5, CSS3, Ajax, TypeScript, ES6 (ECMAScript), JQuery, Angularjs, Ext JS, ReactJS, Redux, Bootstrap and JavaScript and worked on server-side web applications using Node.js.
* Worked in using React JS components, Forms, Events, Keys, Router, Animations and Flux concept. Used React-Router to turn application into a Single Page Application.
* Used Spring MVC to enable the interactions between JSP/View layer and implemented different design patterns with J2EE, XML technology and involved in handling transactions using Spring Declarative Transaction Management.
* Implemented core logic to build target view utilizing core framework API methods leveraging Java 1.8 features like Lambda Expressions, enhanced collection API etc.
* Integrated XState or Redux-Saga for advanced state orchestration in React applications.
* Integrated RxJS with React for managing event-driven and reactive state architectures.
* Utilized React Profiler and Chrome DevTools to identify and resolve performance bottlenecks in component rendering and state updates.
* Used Hibernate Query Caching and L2 Cache to improve performance for frequently accessed data.
* Implemented CDC (Change Data Capture) with Debezium for real-time data synchronization.
* Used Hibernate and MarkLogic, writing service methods and SQL queries to interact with the MySQL DBMS.
* Worked with NPM commands and using package.json for managing dependencies, and dec-dependencies in Node JS.
* Involved in ORM mappings for associations, Inheritance & fetching strategy, performing CRUD operations and managing second level cache using Hibernate.
* Involved in implementing the Microservices based on RESTful API utilizing Spring Boot with Spring MVC.
* Used Spring-boot-starter-parent as the parent POM and inheriting sensible default settings from Spring Boot and the spring-data- Mongodb dependency provides integration with the MongoDB.
* Used Spring Boot for cloud Microservices and to develop Spring application radically faster with very less configuration.
* Used Apache CXF to develop SOAP, RESTful services, Swagger API for auto-generated documentation for all REST calls.
* Used Spring Security for Authentication and Authorization of the application.
* Developed Restful services using Rest Template and Microservices using Spring Boot.
* Used SOA Integration service for Data Integration Service, Functional Integration Service and used SOAPUI for the functional testing, inspection and invocation of Spring based Web services.
* Implemented Oracle SQL Queries & PL/SQL - stored procedures, functions, triggers, sequences and built-in functions to retrieve and update data from the databases.
* Used Microservices to communicate using synchronous protocols HTTP and REST for implementing SOAP.
* Implemented Apache Kafka producer & consumer for asynchronous stream processing, search APIs using data indexing.
* Deployed and tested the application in UNIX on IBM Web Logic Application Server.
* Used Azure Functions to integrate applications and run backend processes on schedules without provisioning servers.
* Worked on configuration of internal load balancer, load balanced sets, and Azure Traffic manager, Azure Resource Manager (ARM) provides security, auditing, and tagging features to help you manage your resources after deployment.
* Managed Docker orchestration using Docker Swarm and containerization of applications using Docker.
* Monitored MongoDB (NoSQL) Replica Sets / Databases and prototyped MongoDB new features across multiple stacks.
* Working HA Environment, Configuring the Kubernetes pods and testing the pods.
* Enabled Continuous Integration, Continuous Deployment (CI/CD) and monitoring using Chef, Jenkins and Shell Scripts.
* Involving the code review and using the SonarQube to validate the violation.
* Configured Jenkins and GitHub Actions for CI/CD pipelines, automating build, test, and deployment workflows.
* Secured microservices with Kubernetes Network Policies and RBAC (Role-Based Access Control).
* Used Jenkins as Integration tool to deploy Spring Boot Microservices to Pivotal Cloud Foundry (PCF) using build pack.
* Integrated testing methods implemented by using Karma and Jasmine. Used Jira as bug reporting tool.
* Developed unit test cases using JUNIT, Easy Mock, Integration and User Acceptance Testing and developed test cases for bean components. Used Maven, Jenkins to build and deploy applications.
* Designed and implemented distributed architecture with microservices deployed on AWS and PCF.
* Leveraged distributed caching mechanisms (Redis, Memcached) for enhanced scalability and performance.
* Integrated event-driven architecture using Kafka and ActiveMQ for asynchronous, decoupled communication.

**Environment**: Java 1.8, MVC, Spring Boot, React JS, Redux, Flux, Node JS, Backbone JS, Hibernate ORM, MongoDB (NoSQL), PCF, REST/SOAP Web services, JUNIT, Microservices, Web Logic, Maven, JBOSS, JavaScript, Azure, SQL-Server, Log4j, Jenkins.

# **Centelon Solutions, Mumbai, India Sep 2016 to Nov 2018**

**Role: Application Developer**

**Responsibilities:**

* Used Java 8 new features like Lambda Expressions, Functional Interfaces, Stream API, Date API, Method reference, Default methods, Optional class.
* Implemented Microservices architecture using Spring Boot for making the application smaller & independent.
* Developed Rest Controller for mapping requests to generate Restful APIs.
* Developed the responsive UI web pages using Angular 6 & 8, TypeScript, Node JS and worked with angular filters in expressions and directives to filter data in the UI.
* Developed real-time video conferencing features in React applications using WebRTC and custom signaling servers.
* Implemented browser push notifications in React applications using the Notifications API and Service Workers.
* Utilized React Portals to render modals, tooltips, and overlays outside the DOM hierarchy for enhanced UI experiences.
* Deployed Spring Boot-based Microservices in Docker and AWS EC2 containers using Jenkins.
* Worked on creating login authentication modules using Angular and TypeScript.
* Built administrator page displaying profile information in React JS and Graph QL.
* Created RESTful based API utilizing Spring MVC Controllers, JSON to Java Object mapping is done using Jackson, JDBC JPA is used to connect to SQL Server.
* Developed Microservices using Spring MVC, Spring Boot, and Spring Cloud.
* Used Angular 8 to create Single, Multiple Page Applications to bind data to views, synchronize data with server.
* Implemented Cloud capabilities like Centralized Configuration, Load Balancing, Service-to-Service call using Feign, Service Registry, and Discovery using Eureka in Spring Cloud.
* Implement a JPA repository to access data as DAO. Implement Kafka for both Producer and Consumer.
* Tested REST APIs with GET, POST, PUT, and DEL by providing input in JSON parameters using the Postman.
* Configured Spring Security to grant permissions for different level resources.
* Perform troubleshooting and monitoring of the Linux server on AWS using Splunk.
* Worked with Spring scheduler for automatic scheduling of spring batch jobs.
* Used Spring IOC, configured application Context files, and used dependency injection using Spring annotations.
* Included security groups, network ACLs, Internet Gateways, and Elastic IPs in AWS public cloud.
* Developed Microservices (REST APIs) using Java Spring Boot, Express and used Maven as a build tool.
* Extensive experience in integrating services via Ajax, JavaScript, Node JS.
* Implemented functionality using Spring Boot and Hibernate ORM to map with Oracle Database.
* Used ReactJS to create components for data manipulations and to display data in company standard format.
* Implemented the Spring AOP Aspect-Oriented Programming for Logging, declarative Transaction Management, creating and applying transaction advice, and configuring the joint points.
* Troubleshooted the issues and tuned performance of JBoss Application Server and Apache Tomcat.
* Created Server instances on AWS. Tomcat, Glassfish, Swagger are installed and used for deploying Microservices.
* Developed User Interaction screens using HTML, CSS3, Bootstrap, JavaScript, Angular 9, and Node JS.
* Used Microservices, Spring Boot services interacting through a combination of REST, Kafka message brokers.
* Managed AWS Services CLI, EC2, VPC, S3, ELB Glacier, Route 53, CloudTrail, IAM, Trusted Advisor services.
* Created and configured Micro-Services architecture, integrated it with the React JS application.
* Used React & Redux Dev Tools plugins to observe component state and prop changes instantly.
* Worked with React JS Forms, Keys, Router, components, Events.
* Wrote SQL queries to get, update data from database, and connect databases in the DAO implementation classes.
* Created automated pipelines in AWS Code Pipeline to deploy Docker containers in AWS ECS using services like CloudFormation, Code Build, Code Deploy, S3, and puppet.
* Setup the CI/CD process for the application using Jenkins, Docker, Maven.
* Used Postman, Rest Client, Spring Mock MVC, Spring 4, Junit for testing the web services.
* Created Application using React JS, Node JS libraries NPM, gulp directories to generate view, flux to root URL.
* Select the appropriate AWS service based on computing, data, or security requirements.
* Integration of Amazon Web Services (AWS) with other applications infrastructure.
* Implemented automated local user provisioning instances created in AWS cloud.
* Used JIRA tool for tracking stories progress, following agile methodology and sprint planning.
* Used GitHub, Bitbucket to maintain the version of the files and took the responsibility to do the code merges.

**Environment:** Java 8, Spring Boot, Spring, Maven, Lambda, Hibernate, Angular, React, REST, SOAP, SQL, Oracle 12c, Log4j, Node JS, Microservices, JSP, JUnit, Spring Security, Karma, Jasmine, Mockito, MongoDB, AWS.

**Ides Labs, Mumbai, India Jun 2013 to Aug 2016**

**Java Full Stack Developer**

**Responsibilities:**

* Involved in software development life cycle (SDLC) by using agile methodology including project design, development, and implementation and testing.
* Developed Sequence, test cases, Use cases & class Diagrams and Process Flow diagrams using Rational Rose.
* Designed and implemented the User Interface using JavaScript, HTML5, XHTML, XML, CSS3, JSP, and AJAX.
* Migrated from Angular 2 to Angular 4 for making the code lesser, cleaner, improving the performance by supporting the new version of Typescript which increases Type security and speed of NGC-Compiler.
* Designed and developed Microservices using REST framework and Spring Boot and used Bootstrap along with AngularJS in creating the Application.
* Developed the administrative UI using Angularjs and Inserted External Style Sheets to various web pages using CSS.
* Worked on JavaScript framework (Angularjs) to augment browser based applications with MVC capability.
* Used Node.js as a proxy to interact with Web services and also interacting with Database and Node.js is also used to handle multiple concurrent connections with minimal overhead on a single process.
* Used Angular2 to develop the client side application in MVC manner. And used Bootstrap as a stylesheet.
* Microservices has been built using spring boot, spring security and deployed to AWS cloud.
* Designed and developed the REST based Microservices using the Spring Boot, Spring Data with JPA (Java Persistence API) and Used Swagger to interact with API endpoints of the Microservices.
* Utilized Spring framework and integrated with back-end database using Hibernate framework.
* Implemented application using Spring concepts – Dependency Injection/ IOC, AOP, Batch implementation and Spring Model View Controller (MVC).
* Extensively used Hibernate in the DAO layer to access and update information from the oracle database.
* Used Hibernate Query Language (HQL) for writing the custom queries.
* Used Hibernate to create a data layer to the services to do CRUD operations in Database.
* Implemented business tier using Spring AOP and spring IOC.
* Used micro-framework to implement with REST API and MongoDB (NO SQL) as back end database.
* Created JSON Data and stored as documents comprising documents, data types and conversions in MongoDB (NoSQL).
* Parsed the XML based responses using JAXB Parser and validated XML against the schema.
* Developing Web Service Client Component to make use of service APIs.
* Implemented and used Web services with the help of WSDL and SOAP to get updates from other APIs.
* Involved in CI/CD process using GIT, Nexus, Jenkins job creation, Maven build, Created Docker image, deployed in AWS.
* Deployed Spring Boot - Microservices Docker container using AWS EC2 container services and AWS admin console.
* Developed transformations using XSLT to simplify project specific large XML configuration files.
* Produced as well as consumed the SOAP/REST Web services using WSDL, SOAP and CXF.
* Worked on Rapid Application Development (RAD) as a developing environment.
* Consumed & developed Web services using JAX-WS/JAX-RS to communicate with different departments.
* Used RabbitMQ for messaging communication. Used Maven for building and deploying applications.
* Worked on SOAP & REST based Web services to create the web based services.
* Performed unit testing using JUnit framework and Jira for bug tracking.

**Environment**: Java 1.6, Spring IOC, Spring AOP, Spring Boot, Spring Batch, Hibernate, Angularjs, Bootstrap, Servlets, RAD, Microservices, SOAP-Web Services, DB2, MongoDB, IBM Clear Case, FileZilla, SFTP, JUnit, Mockito, JMS, ANT.

**EDUCATION:**

 **University of Wisconsin Milwaukee 2019 jan - dec 2020**

* Masters in Computer Science

**Andhra University**  2013

* Bachelors in Computer Science