**SUCHITHA VEMURI**

**Sr. Java Full Stack Developer**

**Phone:** 4696079194

**Email:** suchithacv5@gmail.com

**LinkedIn:** www.linkedin.com/in/suchitha-vemuri-138aa9202

**PROFESSIONAL SUMMARY**

* Overall 10+ Years of experience and includes all phases of Software Development Life Cycle (SDLC) including planning, analysis, design, implementation, testing, and maintenance of multi-tier distributed, secure enterprise applications using Java/J2EE technologies.
* Have strong exposure and work experience on core Java concepts collection API, Stream API, Parallel Streams, exception handling, Multithreading, HashMap, concurrency, JDBC, modularity, executor service, and fork and join frameworks.
* Experience in developing web applications using MVC, or Web API, enabling the development of robust and scalable web applications. Experience in integrating Flink with other big data technologies such as Apache Kafka or Hadoop, enabling seamless data ingestion and processing.
* Experience Front-end part of the application using AngularJS, Backbone.js, jQuery, Node.JS, Ajax, Bootstrap, React JS, HTML5, and CSS3.  Used Java 8.0 features like Lambda Expressions with Lambda, and AWS Lambda.
* Experience in developing distributed systems using Scala and Apache Kafka, improving system scalability and resiliency.
* Solid Experience in creating Kubernetes Clusters in AWS and deploying a Docker Image on top of Kubernetes Cluster.
* Strong experience in data structures, algorithms, and design patterns.
* Experience in developing and deploying Java web applications on WebLogic servers. Knowing (JMS) MQs like Apache Kafka.
* Worked in using React.js Router, Forms, Animations, Keys, components, Events, and Flux concepts.
* Experienced in writing front-end Unit Testing on both client and JS server side using Jasmine, Karma, and the Integration testing tool Protractor for AngularJS. Good knowledge of Angular 8/9 directives to create reusable elements and attributes, along with easy-to-test and maintain client-side interactivity in web applications.
* Expertise in implementing Java 8 features like Lambda expressions, Functional interfaces, Stream API, and Time API for Bulk data operations on Collections, concurrency, and I/O which would increase the performance of the Application.
* Expertise in designing, developing, and deploying applications using J2EE technologies including Servlets, JSP, EJB, JSTL, Struts, Spring, Spring Boot, JSF, JMS, AJAX, JDBC, JNDI, Hibernate, and Web Services.
* Experience in designing and implementing microservices using AWS services like Amazon ECS and Amazon EKS.
* Deployed Docker-contained spring boot-microservices into Spring EC2 container service using AWS admin console.
* Knowledge of JavaScript frameworks such as React and Angular, and experience in using TypeScript with these frameworks. Hands-on experience in working with Kubernetes for managing related, distributed cluster components.
* Experience in publishing and subscribing streams of records in enterprise message systems and in designing and building real-time streaming data pipelines using Apache Kafka.
* Strong Knowledge in MERN/MEAN stack with vast experience in building Web Applications, using React.js/AngularJS for the client side, Node.js/Express for the server side, and Mongo DB, SQL Server for the database.
* Experience in integrating Scala applications with other technologies such as Java, Python, or SQL, enabling seamless integration with other systems. JUnit, Mockito for unit, integration testing. Extensive experience in creating dynamic web applications using Java Servlets.
* Developed API for using AWS Lambda to manage the Serverless architecture and run the code in AWS.
* Proficient in version control tools such as Git, GitLab, and CVS, enabling effective collaboration and code management within projects. Extensive experience in front-end development using JSP, JavaScript, HTML5, CSS3, jQuery, AJAX, JSON, XSL, XSLT, and Angular JS, showcasing expertise in web application design.
* Expertise in API testing and documentation using tools like Postman, Insomnia, and Swagger, ensuring the reliability and comprehensibility of APIs. Excellent Technical, Analytical, and Problem-Solving skills, strict attention to detail, and ability to work independently, and work within a team environment.
* Proficient in implementing security protocols like OAuth2 and SSL/TLS, ensuring robust authentication and secure communication in web applications.
* In-depth expertise in Jenkins Continuous Integration Server, including installation, configuration, and administration, with a focus on seamless integration with other systems for automated build and deployment processes.

**EDUCATION**

* Bachelors in Computer Science from KL University, India.

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Programming****Languages** | **C, Java 8/11/15/16/17, Javascript, PHP, J2EE, PL/SQL, Scala, Python** |
| **Web Technologies** | **HTML, HTML5, CSS, XML, JavaScript, Bootstrap, JSP, JPA, JDBC, JSTL, JQuery, AJAX, Angular JS, Angular 2/4/6/8/10/15, TypeScript, JSON, Backbone.js, Servlets, Applets, ReactJS** |
| **Frame Works** | **Struts 2.5, JSF, Spring 5.0, Spring MVC, Spring Security, Spring Rest Hibernate, and Junit** |
| **Web****Services/Logging** | **REST and SOAP Web Services, JAX WS, Apache axis, XSD, GraphQL, log4j.** |
| **Development Tools** | **Eclipse, Net Beans, IntelliJ Idea, STS, Microsoft Visual Studio Code** |
| **Testing** | **Jasmine, Karma, JUnit, Mockito, Selenium, Cypress, Integration Testing** |
| **Web/Application****Servers** | **Apache Tomcat, IBM Web Sphere, Weblogic, and JBoss.** |
| **Methodologies** | **Agile, Scrum, Waterfall** |
| **Messaging** | **Apache Kafka, JMS(Java Message Service)** |
| **Databases**  | **Oracle 10g/11g/12c, SQL Server, Oracle DB2, MYSQL, NoSQL( MongoDB, Cassandra)** |
| **API Testing and****Documentation** | **Postman, Insomnia, Swagger** |
| **Security** | **OAuth, OAuth2, SSL/TLS** |
| **Tools & Utilities** | **Apache Maven, Gradle, Jenkins, ANT, and SonarQube.** |
| **Project management****and communication** | **Jira, Asana, Slack, Microsoft Teams** |
| **Cloud Technologies and****Containerizations** | **AWS, Jenkins, Dockers, Kubernetes, Azure, GCP** |
| **Cloud and database****tools** | **AWS EC2, AWS ECS, AWS S3, AWS Lambda and MongoDB compass, MYSQL Workbench** |
| **Big Data Technologies** | **Apache Kafka, Apache Nifi** |
| **Operating Systems** | **Windows 7/8/10, Unix and Linux** |
| **Version control** | **GitHub, Git, GitLab, and CVS.** |

**PROJECT EXPERIENCE**

***Client: Global Payments, Las Vegas, CA(Remote). Aug 2023 to till now***

***Role: Senior Java Full Stack Developer
Project Tittle: GlobalPay – A Unified Payments Platform
Project Description:*** Developed a unified, cloud-native payments platform focused on streamlining digital transactions across diverse channels. The system was built using a microservices architecture with Java 21/17, Spring Boot, and Angular 15, enabling flexible integration and high performance. Applications were containerized with Docker and orchestrated using Kubernetes, then deployed on cloud platforms like AWS using EC2, ECS, Lambda, and RDS. Real-time communication was established via Kafka, while robust CI/CD pipelines, TDD practices, and automated testing using Cypress, JUnit, Jasmine, and SonarQube ensured system reliability, security (OAuth2), and development efficiency***.***

***Responsibilities***

* Followed Agile Software Development Methodology to build the application iteratively and incrementally. Participated in scrum-related activities and daily scrum meetings.
* Developed and maintained microservices using Java 21, Core Java, and Java 17, emphasizing effective design and collaboration for seamless integration with other system components.
* Created dynamic and responsive user interfaces using Angular 15 and TypeScript, ensuring a positive user experience. Integrated the front end seamlessly with microservices through the implementation of Restful API for efficient communication.
* Established and managed Continuous Integration/Continuous Deployment (CI/CD) pipelines using Jenkins, Maven, and GitLab. Automated the build, test, and deployment processes to achieve efficiency and reliability, and continuously monitored and optimized the CI/CD pipelines.
* Utilized MongoDB for data storage and management, employing MongoDB Compass to ensure the integrity and optimal performance of databases within the system.
* Implemented a comprehensive automated testing strategy, incorporating tools such as JUnit, Jasmine, Karma, Cypress, and Postman. Followed Test-Driven Development (TDD) practices and used Mockito for dependency mocking to achieve high test coverage.
* Deployed and managed applications on the AWS cloud platform, leveraging services like EC2, ECS, S3, and RDS. Utilized Docker for containerization and Kubernetes for orchestration, ensuring scalability and efficient resource utilization. Implemented Swagger for thorough API documentation.
* Implemented RESTful API and JAX-WS for effective communication between microservices. Utilized Apache Kafka for event-driven communication and messaging. Enforced standardized API practices to facilitate integration.
* Prioritized security by implementing OAuth and OAuth 2 for secure authentication and authorization. Applied best security practices and regularly updated dependencies to address potential vulnerabilities.
* Implemented Log4j for centralized logging, enabling efficient monitoring and troubleshooting. Set up monitoring tools, such as SonarQube, to analyze code quality and integrated monitoring solutions for overall system health and performance.
* Implemented Kubernetes Helm charts for defining, installing, and upgrading even the most complex Kubernetes applications. Utilized IntelliJ for Java development and Git for version control, ensuring a streamlined and organized development process.
* Fostered effective team communication through collaboration tools like Slack, and adhered to Agile methodologies for project management. Documented and shared knowledge through appropriate channels to facilitate a well-informed and collaborative development environment.
* Developed and deployed serverless functions on AWS Lambda to execute code without the need for provisioning or managing servers.
* Integrated PHP components as needed, ensuring compatibility and interoperability within the project architecture.
* Addressed compatibility with Windows environments, ensuring that the developed solutions worked seamlessly in diverse operating system setups.
* Implemented Kubernetes Ingress controllers to manage external access to services and applications within the cluster.
* Utilized SOAP UI for testing and validating SOAP-based web services, ensuring interoperability with existing systems.
* Implemented JSTL and JSP for efficient server-side rendering and dynamic content generation in web applications.
* Deployed and managed applications on Apache Tomcat, a widely used application server for Java-based projects.
* Implemented Cypress for end-to-end testing, providing a comprehensive testing approach that covered the entire application workflow.
* Configured AWS Lambda functions to respond to various triggers, including API Gateway events, S3 events, and CloudWatch Events. Utilized Mockito extensively for mocking dependencies during testing, ensuring the isolation and effectiveness of individual components.
* Followed Test-Driven Development (TDD) practices rigorously to maintain code quality and achieve a robust and reliable codebase. Ensured compatibility with various GitLab features for version control, collaboration, and continuous integration processes.
* Integrated and utilized AWS services such as AWS EC2, ECS, and S3, ensuring optimal utilization of cloud resources.
* Implemented OAuth and OAuth 2 protocols for secure and standardized authentication and authorization processes.
* Utilized Node.js for server-side JavaScript development, enhancing the versatility of the technology stack.
* Employed IntelliJ as the primary integrated development environment (IDE) for Java development, ensuring a consistent and efficient coding environment.
* Leveraged Git for version control, ensuring proper code versioning, collaboration, and change tracking.
* Integrated and utilized Apache Kafka for efficient and scalable event-driven communication between microservices.
* Implemented Swagger for detailed and interactive API documentation, enhancing the clarity and usability of the developed APIs. Utilized Log4j for centralized logging, aiding in monitoring and troubleshooting activities throughout the development and deployment lifecycle.
* Implemented Cypress for end-to-end testing, providing a comprehensive testing approach that covered the entire application workflow.
* Developed dynamic web applications using JSTL and JSP for server-side rendering. Implemented Angular 15 and TypeScript for interactive and dynamic user interfaces.
* Conducted extensive testing using Jasmine and Karma for front-end testing. Utilized SOAP UI for comprehensive testing of SOAP-based web services.
* Implemented Cypress for end-to-end testing, covering the entire application workflow. Created and documented RESTful APIs using Swagger, ensuring clear communication and understanding.
* Implemented JAX-WS for building web services using Java. Developed XML Schema Definitions (XSD) for structured data representation.
* Utilized Hibernate and JPA for efficient data modelling and persistence. Implemented Maven for project build management, ensuring consistency and efficiency.
* Utilized Jenkins for continuous integration and continuous deployment (CI/CD) pipelines. Utilized Docker for containerization, ensuring consistent deployment across various environments.
* Deployed applications on AWS services, including EC2, ECS, S3, and RDS, for scalable and reliable cloud infrastructure.
* Leveraged Kubernetes for efficient container orchestration, ensuring seamless scaling and resource management.
* Utilized Slack for effective team communication and collaboration. Documented and shared knowledge through appropriate channels for seamless collaboration.
* Integrated Log4j for centralized logging, aiding in monitoring and troubleshooting activities. Implemented Sonar Qube for continuous code quality analysis and improvement. Utilized Git for version control, ensuring proper code versioning and collaboration.

**Environment**: Java 17, Core Java, JavaScript, Java 21, Spring, Spring Boot, JPA, Hibernate, Microservices, Angular 15, Typescript, CI/CD, Scala, SOAP UI, JSTL, JSP, Agile, NoSQL(MongoDB), Mongo DB compass, Jasmine, Karma, Apache Tomcat, Apache Kafka, Restful API, JAX WS, XSD, Maven, Jenkins, AWS, AWS EC2, AWS ECS, AWS S3, AWS RDS, AWS Lambda, Dockers, Kubernetes, Swagger, Log4j, Slack, Postman, Sonar Qube, PHP, OAuth, OAuth 2, Windows, IntelliJ, node.js, JUnit, Git, GitLab, Cypress, Mockito, TDD

***Client: Centene, St. Louis, MO. Jan 2022 to July 2023***

***Role: Java Full Stack Developer***

***Project: Care Connect – Integrated Member Health Management System
Project Description***: Developed a cloud-native health management platform aimed at streamlining member services, claims processing, and provider engagement. The solution employed a microservices architecture using Java 16/17, Spring Boot, and React, with real-time communication powered by RESTful APIs, Kafka, and GraphQL. Applications were deployed on AWS using Docker and Kubernetes, supported by CI/CD pipelines built with Jenkins and Maven. Security, scalability, and code reliability were prioritized through OAuth2, Log4j, SonarQube, and adherence to Agile practices using Jira and Asana***.***

***Responsibilities***

* Designed, developed, and maintained microservices using Java 16, Core Java, and Java 17, ensuring a modular and scalable architecture.
* Implemented microservices using Spring, Spring Boot, JPA, Hibernate, and JDBC for efficient data access and management.
* Created Restful APIs for seamless communication between microservices, promoting interoperability and flexibility in the system. Developed user interfaces using React and React JS, ensuring a responsive and intuitive user experience.
* Implemented JSP, JSTL, and Handlebars.js for efficient server-side rendering and dynamic content generation.
* Integrated front-end components with microservices, facilitating a cohesive and integrated application.
* Established and maintained CI/CD pipelines using Maven, Jenkins, and Docker for automated build, test, and deployment processes. Ensured smooth integration with AWS services (EC2, ECS, S3, RDS) and Kubernetes for efficient container orchestration.
* Monitored and optimized CI/CD pipelines to achieve reliable and rapid releases. Utilized NoSQL database (MongoDB) for efficient and scalable data storage.
* Managed MongoDB databases using Mongo DB Compass, ensuring data integrity and performance. Implemented JDBC for traditional relational database interaction, when necessary.
* Implemented automated tests using JUnit, Cypress, and Mockito, following Test-Driven Development (TDD) practices.
* Used Postman for API testing and validation, ensuring robust and reliable functionality. Integrated Sonar Qube for continuous code quality analysis and improvement.
* Deployed and managed applications on AWS, leveraging EC2, ECS, S3, and RDS for scalable and reliable cloud infrastructure.
* Proficient in WebLogic Server administration, including installation, configuration, and maintenance tasks.
* Utilized Docker for containerization and Kubernetes for efficient container orchestration.
* Integrated with Apache Tomcat for web application deployment. Implemented OAuth and OAuth 2 for secure authentication and authorization.
* Utilized Jira and Asana for project management, fostering effective collaboration and communication within the Agile development framework.
* Implemented event-driven architectures using AWS Lambda, ensuring seamless integration with different components within the AWS ecosystem.
* Implemented rolling updates and rollback strategies in Kubernetes to minimize downtime during application updates.
* Deployed Kafka clusters for real-time streaming applications, implementing scaling strategies to handle increased data loads.
* Integrated Log4j for centralized logging, aiding in monitoring and troubleshooting activities.
* Incorporated Linux environments into the development and deployment processes, ensuring compatibility and efficiency.
* Implemented GraphQL for efficient and flexible data querying. Maintained comprehensive documentation for Kafka configurations, deployments, and operational procedures. Utilized Git for version control, ensuring proper code versioning and collaboration.
* Integrated Node.js for server-side JavaScript development when required. Implemented Scala for additional language support in the microservices architecture.
* Utilized Jasmine and Karma for front-end testing in addition to React and Cypress.
* Utilized AWS Lambda layers to share code and dependencies across multiple serverless functions, improving code reuse and maintainability.
* Implemented PHP components as needed, ensuring compatibility within the project environment.
* Integrated and maintained Apache Kafka for efficient event-driven communication between microservices.
* Used Maven for project build management, ensuring consistent and streamlined build processes.
* Leveraged Docker for containerization, facilitating consistent deployment across various environments.
* Integrated Jenkins for continuous integration and continuous deployment (CI/CD), automating the development lifecycle.
* Deployed and managed applications on AWS services, including AWS EC2, ECS, S3, and RDS, for reliable and scalable cloud infrastructure.
* Utilized Kubernetes for efficient container orchestration, ensuring seamless scaling and resource management.
* Employed Handlebars.js for dynamic templating, enhancing server-side rendering capabilities.
* Conducted unit testing using JUnit, ensuring the reliability and stability of individual components.
* Utilized Cypress for end-to-end testing, providing comprehensive coverage of application workflows.
* Employed Mockito for effective mocking during testing, ensuring the isolation and accuracy of test cases.
* Implemented and maintained GraphQL for flexible and efficient data querying within the system.
* Utilized Node.js for server-side JavaScript development, enhancing the versatility of the technology stack.
* Integrated and utilized PHP components as needed, ensuring compatibility within the broader project environment.
* Followed Test-Driven Development (TDD) practices rigorously to maintain code quality and achieve a robust and reliable codebase. Ensured compatibility with various GitLab features for version control, collaboration, and continuous integration processes.
* Skilled in deploying J2EE applications on WebLogic Server, ensuring proper configuration for performance. Implemented OAuth, and OAuth 2 protocols for secure and standardized authentication and authorization processes.
* Utilized Log4j for centralized logging, aiding in monitoring and troubleshooting activities throughout the development and deployment lifecycle.
* Integrated Asana for project management, facilitating effective collaboration and communication within the Agile development framework. Configured and maintained Kubernetes clusters, ensuring high availability, fault tolerance, and efficient resource utilization.
* Implemented RESTful APIs, adhering to industry standards and best practices. Utilized GraphQL for efficient and flexible data querying within the system. Followed Agile methodologies for project management, fostering iterative development and continuous improvement. Leveraged Git for version control, ensuring proper code versioning, collaboration, and change tracking. Conducted unit testing using Jasmine and Karma, ensuring the reliability and stability of front-end components.
* Utilized Scala for additional language support within the microservices architecture.
* Employed PHP components as needed, ensuring seamless integration within the broader project environment.

**Environment**: Java 16, Core Java, JavaScript, Java 17, Spring, Spring Boot, JSP, JSTL, JPA, Hibernate, JDBC, Microservices, Restful API, React, React JS, IntelliJ, Agile, JUnit, WebLogic, Cypress, Mockito, NoSQL(MongoDB), Mongo DB compass, Apache Tomcat, Apache Kafka, Maven, Dockers, Jenkins, AWS, AWS EC2, AWS ECS, AWS S3, AWS RDS, AWS Lambda, Kubernetes, Handlebars.js, Postman, OAuth, Jira, Log4j, Asana, OAuth 2, Linux, Restful API, GraphQL, Git, node.js, CI/CD, Scala, Jasmine, Karma, Sonar Qube, PHP, TDD

***Client: Hanover Insurance Group, Worcester, MA Sep 2019 to Dec 2021***

***Role: Java Full Stack Developer
Project: Policy Management Platform***

***Project Description:*** Developed a cloud-native claims processing system using Java, Spring Boot, and Angular 8, structured around microservices for modular scalability. RESTful APIs and GraphQL enabled smooth communication between services, while Docker and Kubernetes supported efficient deployment across AWS and GCP. The application followed CI/CD best practices with Jenkins and Maven, reinforced by automated testing through Selenium, JUnit, and SonarQube for code quality assurance***.***

***Responsibilities***

* Responsible for gathering business requirements from the business and documentation of the enhancement of the project.
* Developed and implemented microservices using Java 8, Core Java, and Java 16 for a modular and scalable architecture.
* Utilized Spring, Spring Boot, and JPA for efficient data access and management in microservices.
* Implemented Restful APIs to ensure seamless communication between microservices, promoting flexibility and interoperability.
* Developed user interfaces using Angular 8 and Typescript, providing a responsive and engaging user experience.
* Utilized JSP for server-side rendering and dynamic content generation.
* Integrated front-end components with microservices through Restful API for a cohesive and integrated application.
* Implemented backend services using technologies such as Hibernate, JDBC, and RMI for data processing and transactions.
* Developed and maintained database schemas and data models for optimal data representation and retrieval.
* Conducted unit testing using JUnit and Mockito to ensure the reliability and stability of individual components.
* Implemented end-to-end testing using Selenium for comprehensive testing of the entire application workflow.
* Administered Oracle DB, including installation, configuration, and ongoing maintenance tasks to ensure optimal performance. Employed Jasmine and Karma for front-end testing to guarantee the functionality of user interfaces.
* Demonstrated ability in troubleshooting issues related to WebLogic Server, including diagnosing and resolving performance bottlenecks.
* Configured Kafka producers and consumers for optimized data flow and reliable message delivery. Implemented and configured WebLogic Server clusters to achieve high availability and load balancing.
* Documented AWS Lambda configurations, deployment procedures, and integration strategies, facilitating knowledge sharing and collaboration within the team. Implemented and managed Oracle DB schemas, tablespaces, and data files, optimizing storage and retrieval efficiency.
* Stayed updated with AWS Lambda updates and best practices, incorporating new features and improvements into existing projects. Utilized Docker for containerization, ensuring consistent deployment across various environments.
* Implemented Kubernetes for efficient container orchestration, allowing seamless scaling and resource management.
* Established and maintained CI/CD pipelines using Jenkins and Maven for automated build, test, and deployment processes.
* Ensured smooth integration with cloud platforms such as AWS, GCP, and IBM Websphere.
* Utilized AWS services like EC2, ECS, S3, RDS, and Lambda for scalable and reliable cloud infrastructure.
* Integrated Sonar Qube for continuous code quality analysis and improvement. Implemented and managed Oracle DB partitioning for large datasets, optimizing query performance.
* Designed and implemented event-driven architectures using Kafka for decoupling and scalability. Utilized Swagger for detailed API documentation, enhancing clarity and understanding.
* Implemented parameterized builds in Jenkins to enhance flexibility and support different deployment scenarios. Implemented Jenkins job notifications and alerts to keep the team informed about build and deployment status.
* Followed Test-Driven Development (TDD) practices to maintain code quality and ensure a robust codebase.
* Used Jira and Asana for project management, fostering effective collaboration within the Agile development framework.
* Collaborated and version-controlled using Eclipse, GitHub, and Git, ensuring proper code versioning and collaboration.
* Utilized GraphQL for efficient and flexible data querying within the microservices architecture. Deployed and managed applications on various environments, including Windows and Eclipse. Leveraged AWS Lambda for serverless computing, optimizing resource utilization. Implemented and managed microservices on IBM Websphere and GCP platforms.

**Environment**: Java 8, Core Java, Javascript, Java 16, Spring, Spring Boot, JPA, RMI, AWT, Applets, JDBC, Hibernate, JSP, Restful API, Oracle DB, Angular 8, Typescript, Weblogic, Dockers, Selenium, Agile, JUnit, Mockito, Jasmine, Karma, Apache Kafka, IBM Websphere, Swagger, Microservices, Jenkins, Kubernetes, GCP, Maven, AWS, AWS EC2, AWS ECS, AWS S3, AWS RDS, AWS Lamda, Sonar Qube, Jira, Asana, Eclipse, Windows, GitHub, GraphQL, CI/CD, TDD

***Client: CVS, Pittsburgh, PA Feb 2018 to Aug 2019***

***Role: Java Developer***

***Responsibilities***

* Responsible for gathering business requirements from the business and documentation of the enhancement of the project.
* Developed and implemented microservices using Java 8, Core Java, and Java 16, ensuring a modular and scalable architecture. Utilized Hibernate for efficient data access and management within the microservices.
* Integrated SOAP UI, JAX WS, and XSD for the development and testing of web services, ensuring interoperability and compliance with standards. Implemented Restful APIs in microservices using Spring and Spring Boot, fostering seamless communication and integration.
* Utilized Selenium for automated testing of microservices, ensuring robust functionality and performance. Integrated React and React JS for dynamic and responsive user interfaces within the microservices architecture.
* Followed Agile methodologies for project management, ensuring adaptability and responsiveness to changing requirements. Documented best practices, performance optimization techniques, and troubleshooting guidelines for Oracle DB usage within the organization.
* Collaborated with cross-functional teams to design and implement robust, scalable, and highly available solutions on the Azure cloud. Implemented and configured Azure Storage solutions, including Blob Storage and Azure SQL Database, for efficient data storage and retrieval.
* Configured and optimized Azure resources to ensure scalability, performance, and cost-effectiveness for diverse projects.
* Conducted unit testing using JUnit and Mockito, guaranteeing the reliability and stability of individual microservices.
* Employed Jasmine and Karma for front-end testing, ensuring the proper functioning of user interfaces.
* Managed and maintained MySQL, Oracle DB, and Postgres SQL databases within the microservices environment.
* Utilized JMS for efficient messaging between microservices, ensuring reliable communication and event-driven architecture. Implemented and maintained Oracle Database indexes to improve query performance and reduce data retrieval times.
* Deployed and managed microservices on IBM Websphere and Azure platforms, ensuring scalability and reliability. Utilized Docker for containerization, streamlining deployment, and managing dependencies. Developed and maintained microservices using Visual Studio and Unix environments.
* Configured and managed Azure Networking services, such as Azure Load Balancer and Azure Application Gateway, for optimal application delivery. Implemented Gradle for build automation in the microservices project, ensuring consistent and efficient build processes. Utilized Git for version control, facilitating collaboration, and code management within the project.
* Implemented CI/CD pipelines to automate the build, test, and deployment processes, ensuring a streamlined development lifecycle. Followed Test-Driven Development (TDD) practices, ensuring comprehensive test coverage and code quality.

**Environment**: Java 8, Core Java, Servlets, Javascript, Java 16, Hibernate, SOAP UI, JAX WS, XSD, RMI, Applets, AWT, Spring, Spring Boot, Selenium, Restful API, React, React JS, Azure, Agile, JUnit, Mockito, Jasmine, Karma, MySQL, JMS, PgAdmin, IBM Websphere, Insomnia, Oracle DB, Azure, Dockers, Visual Studio, Unix, Gradle, Git, Postgres SQL, JDBC, CI/CD, TDD

***Client*** ***Datamatics Global Services, Hyderabad, India. July 2013- Aug 2016***

***Role: Java Developer***

***Responsibilities***

* Developed and maintained Java-based applications, leveraging Core Java and Servlets for server-side logic.
* Implemented front-end functionalities using JavaScript to create dynamic and responsive user interfaces.
* Utilized C for specific system-level programming requirements within the project environment.
* Followed Test-Driven Development (TDD) practices, ensuring comprehensive test coverage and code quality.
* Implemented JDBC for efficient interaction with databases, ensuring seamless data access and management.
* Conducted unit testing using JUnit and Mockito, guaranteeing the reliability and stability of individual components.
* Developed dynamic web pages using JSP and JSTL for efficient server-side rendering and content generation.
* Implemented Java Message Service (JMS) for asynchronous communication between components.
* Designed and implemented Restful APIs, fostering interoperability and efficient communication between services.
* Managed and maintained MySQL and Oracle DB databases, ensuring optimal data storage and retrieval.
* Utilized Selenium for automated testing, validating the functionality and performance of web applications.
* Utilized ANT for build automation, streamlining the build and deployment processes. Developed and deployed applications in a Linux environment, ensuring compatibility and efficiency.
* Utilized NetBeans as the integrated development environment (IDE) for Java development. Integrated Jenkins for continuous integration, automating the build and testing processes.
* Followed the Waterfall methodology for project management, ensuring a structured and sequential development approach. Managed version control using CVS, facilitating collaborative code management within the project.

**Environment**: Java, Core Java, Javascript, C, Servlets, TDD, JDBC, JUnit, Mockito, JSP, JSTL, JMS, Restful API, MySQL, Oracle DB, Selenium, ANT, Linux, SSL/TLS, Netbeans, Jenkins, Waterfall, CVS