**A picture containing text, sign, outdoor

Description automatically generatedSanketh G**

**Sr. Full Stack Developer**

**Phone: +1 737-304-0779**

**Professional Summary**

* 8+ years of comprehensive experience as a Full Stack Developer, proficient in all phases of the Software Development Lifecycle (SDLC), encompassing Analysis, Design, Development, Testing, and Maintenance of Java UI applications tailored for large Enterprise Customers.
* Strong programming skills in **Java**, **J2EE**, **J2SE** with extensive ability to develop scalable applications.
* Proficient in developing Web UI Applications using **HTML5**, **CSS3**, **Bootstrap**, **JavaScript**, **TypeScript**, **Ajax**, **jQuery**, **Angular.js**, **React.js**, **Express.js**, **Node.js**, **jQuery UI**, and **Spring MVC**, with hands-on experience in creating robust and dynamic interfaces.
* Experienced in **XML** related technologies like **XSL**, **XSD**, **XPATH**, **DTD**, and **XML** **Schemas**, **XML** Parsers like **DOM** & **SAX**.
* Proficient in **Java 8** features including **Lambda Expressions**, **Functional Interfaces**, **Stream APIs**, **Time APIs**, and enhancements in **Collection**, **Concurrency**, and **IO operations**.
* Extensive hands-on experience in **Amazon Web Services (AWS)** provisioning, with proficiency in key services such as **Elastic Compute Cloud (EC2)**, **Simple Storage Service (S3)**, **Glacier**, **Auto Scaling**, **IAM**, **DynamoDB**, **Virtual Private Cloud (VPC)**, **Route 53**, **CloudWatch**, **AWS CLI**, **CloudFormation**, **CloudFront**, **Elastic Beanstalk**, **Elastic Load Balancers (ELB)**, **RDS**, **SNS**, **SWF**, **SES**, **SQS**, and **EBS**,
* Experienced in using **AWS DynamoDB** for database tracking, with significant proficiency in various **Amazon Web Services** like **Elastic Beanstalk**, **EC2**, and **Linux-based AMIs**.
* Implemented permissions for **AWS SQS** to manage Queue interactions and contributed to crafting custom policies.
* Proficient in deploying applications on **EC2** instances using **Eureka Servers**.
* Extensive hands-on experience in Microsoft Azure provisioning, with proficiency in key services such as **Virtual Machines (VMs), Azure Blob Storage, Azure Archive Storage, Azure Active Directory (AAD), Azure Cosmos DB, Azure DNS, Azure Monitor, Azure CLI, Azure Content Delivery Network (CDN), Azure App Service, Azure Load Balancer, Azure SQL Database, Azure Service Bus, Azure Workflow Service, Azure Email Delivery, Azure Queue Storage**.
* Comprehensive understanding of **microservice** architecture, coupled with hands-on experience implementing various **design patterns**.
* Designed and developed **microservice** business components utilizing **Java/J2EE** and **Spring Boot**, implementing **RESTful web services** based on **RESTful APIs**.
* Expertise in design and development of enterprise level **spring**-based web applications using **Spring REST**, **Hibernate**, **EJB**, **Struts**, **Servlets**, **JSP** for User Interface Applications.
* Proficient in **Spring framework** applications including **Spring Core**, **Spring AOP**, **Spring IOC**, **Spring MVC**, **Spring Batch**, **Spring JPA**, **Spring Security**, and **Spring Boot**. Successfully integrated **microservices** into existing system architecture.
* Developed **DAOs (Data Access Objects)** employing **Hibernate** as **Object-Relational Mapping (ORM)** for seamless interaction with **MySQL**, **PostgreSQL**, **Oracle10g** (**SQL/PL-SQL**), and **NoSQL** databases like **MongoDB**.
* Extensive experience in developing **Web Services** and **RESTful APIs** with **Jersey framework**, and **SOAP** services with **Spring-WS**.
* Proficient in **web services (SOAP/RESTful)** and skilled in scaling applications to improve Business Operational Agility (BOA).
* Well-versed in developing and consuming **SOA services**, with experience in **Apache Axis** web services utilizing **SOAP**, **REST**, and **WSDL**, along with exposure to **JAXB**, **JAX-RPC**, **JAX-RS**, and **JAX-WS**.
* Engineered hypermedia-driven connectors to seamlessly integrate **HATEOAS** with **Spring REST services**.
* Enhanced overall performance through effective utilization of **Multithreading**, **collections**, **synchronization**, and **exception handling** strategies.
* Utilized **Apache** for integration and data processing with **Spring**, including **multi-threaded** message consumption, and proficient in **Zookeeper** for service maintenance.
* Extensive experience in using **NPM**, **BOWER**, **Grunt** and **Gulp** for build process.
* Experienced in **Docker images** and **Containers**, **Spring Boot** for push/pull operations, **Jenkins** for CI/CD pipelines, and **SonarQube** for Java code quality assessment.
* Proficient in message streaming with **Apache Kafka** and experienced in large-scale distributed systems utilizing **JMS**, **Message Queues**, and **Spring JMS** Integration.
* Hands-on experience in using message brokers like **ActiveMQ** and **RabbitMQ**, as well as practical expertise in **Netflix cloud** and fallback technologies such as **Eureka**, **Hystrix**, and **Ribbon**.
* Experienced in utilizing **Selenium**, **Jenkins**, **JUnit**, and **Curl** for effective **Unit and System Testing**, alongside proficiency in **Build/CICD/SCM** tools such as **Ant**, **Maven**, **Gradle**, **Jenkins**, **CVS**, **Git**, and **GitHub**.
* Experienced with web and application servers including **WebSphere**, **WebLogic**, **Jetty**, **JBoss**, and **Tomcat Server**.
* Experienced in performing **Unit testing**, **Integration Testing**, and generating test cases using **JUnit**.
* Experienced in working with **version control** tools like **SVN**, **CVS**, and **Git**, as well as utilizing **JIRA** for bug tracking.

**Technical Skills**

|  |  |
| --- | --- |
| **Programming Languages** | Java, C, C++, PL/SQL, Python |
| **Software Methodologies** | Agile, Scrum, SDLC |
| **J2EETechnologies& frameworks** | JSP, JSTL, Servlets, JDBC, EJB, Java Beans XSD, JAX-RS, JAX-WS, SOAP, WSDL, Axis, Apache CXF, Java Security, JBPM |
| **Web Technologies** | HTML, JavaScript, AJAX, Bootstrap, jQuery, XML, XSLT, XSL, CSS, JSON, Chase Standards, Node JS, EXTJS, AngularJS, Angular, React |
| **Web service Frameworks** | Spring, Hibernate, Struts, Spring MVC, Spring IOC, Spring Boot, Spring AOP, Spring Web Flow, JSF, Log 4J, Junit |
| **Web Services** | SOAP, RESTful, Oracle SOA, UDDI, WSDL |
| **Design Patterns** | Singleton, Business Delegate, Data Transfer Object, Façade, Data Access Object, MVC, Filter |
| **IDEs** | Eclipse, Net Beans, IntelliJ, Spring Tool Suite |
| **Application/Web Servers** | Web Logic Server 12c, Web Sphere Application Server 8.5, Apache Tomcat, Apache HTTP Server, IBM HTTP Server, JBoss |
| **Databases** | Oracle, SQL Server, DB2, MySQL, PostgreSQL, MongoDB, Cassandra |
| **Build Tool** | Maven, Ant, Gradle and MSBuild |
| **Testing Tools** | JUnit, Log4J, Mock Framework, Mockito, Cucumber, Selenium |
| **Version Control Tools** | GIT, SVN, Clear Case |
| **Operating Systems** | Windows, Unix, Linux, Mac OS |
| **Cloud Technologies, DevOps Tools & Other** | AWS, Azure, Docker, JIRA |
| **O-R Mapping Tools** | Hibernate, JPA, IBatis. |
| **Testing Tools/Others** | Junit, SoapUI, Putty, Mockito |
| **CICD** | Jenkins, Harness, TeamCity |
| **Code Analysis & SAST** | Sonarqube, Veracode, Checkmarx, Fortify |

**Work Experience**

**Client: AT&T, Dallas, Texas May 2024 – Present**

**Role: Full Stack Java Developer**

**Responsibilities:**

* Developed dynamic and responsive web applications using **HTML5**, **CSS3**, and **JavaScript**, enhancing user experience for customer-facing applications such as account management, service tracking, and notifications.
* Developed a robust and user-friendly subscription management system using **Angular**, enhancing the user experience by providing a seamless interface for managing subscriptions, preferences, and service plans.
* Designed and implemented a real-time customer support system in **Angular**, enabling users to track service issues, manage support tickets, and receive live updates, improving overall customer satisfaction and operational efficiency.
* Designed and implemented a **microservices** architecture using **Spring Boot** to modularize critical functionalities such as user account management, service provisioning, and billing, enabling faster development cycles and scalability for backend systems.
* Developed **RESTful APIs** using **Spring Boot** to enable seamless integration of core services, including customer management, service configuration, and billing, ensuring smooth data flow and communication across the platform.
* Utilized **Spring Data JPA** to interact with databases, optimizing query performance and ensuring robust data management for large-scale telecommunication operations.
* Designed and implemented **NoSQL** database solutions using **MongoDB** to handle large volumes of unstructured data, ensuring efficient storage and retrieval for telecom data, including call logs, user preferences, and service usage statistics.
* Optimized **MongoDB** queries and indexed collections to improve performance, ensuring low-latency data access for high-throughput telecom applications, especially in scenarios involving high-frequency transactions.
* Developed custom **MongoDB** schema designs to store diverse and complex telecom data structures, ensuring flexible and efficient handling of customer billing, subscription data, and service quality metrics.
* Developed a real-time notification system using **Kafka** to manage event-driven architecture, enabling real-time alerts for account changes, service disruptions, and promotions, providing users with timely updates.
* Containerized **microservices** using **Docker**, ensuring consistent deployment across development, testing, and production environments, and improving portability and scalability of Java applications.
* Deployed and managed **microservices** on **Kubernetes**, automating container orchestration, scaling, and self-healing, ensuring high availability and reliability of applications in production environments.
* Implemented **CI/CD pipelines** using **Azure DevOps** to automate the build, test, and deployment of **microservices**, ensuring efficient and consistent delivery of Java applications.
* Integrated **Maven** with Azure DevOps pipelines to automate the build, packaging, and deployment of Java-based service modules, improving delivery speed and consistency.
* Integrated **SonarQube** into **Azure DevOps** pipelines to enforce static code analysis on **microservices**, ensuring high code quality and reducing technical debt.
* Automated **SonarQube** scans during pull request validation and build stages to maintain clean, consistent code across multiple development teams.
* Integrated **Black Duck** scans into **Azure DevOps CI/CD** pipelines to automatically detect security vulnerabilities and license compliance issues in third-party Java dependencies, safeguarding critical business services from potential threats.
* Configured automated policies in **Black Duck** to block builds with high-severity Common Vulnerabilities and Exposures (CVEs) or prohibited licenses, ensuring secure and compliant delivery of Java applications.
* Implemented structured logging using **SLF4J** with **Logback** in **Spring Boot** applications, enabling integration with ELK stack for efficient log aggregation and analysis.
* Integrated distributed tracing with **Spring Cloud Sleuth**, capturing trace and span IDs in logs to enable request tracking and improve observability across **microservices**.
* Leveraged **Azure Functions** to implement serverless compute solutions for event-driven processes, reducing infrastructure overhead and optimizing resource usage for Java-based microservices.
* Integrated **Azure Active Directory** (**Azure AD**) for secure authentication and user management in Java-based web applications, improving security and access control for enterprise applications.
* Worked with **Azure Key Vault** to manage sensitive data and secrets in a secure, compliant manner, integrating it seamlessly into Java applications.
* Practiced **Test-Driven Development (TDD)** using **JUnit** for developing high-quality, reliable code for service management and billing systems, ensuring robust application performance and security in production environments.
* Utilized **Git** for version control, facilitating efficient code collaboration and reviews across teams. Maintained multiple Git repositories to ensure modular development and easy maintenance of various components.
* Integrated **Datadog APM** with **microservices** to monitor real-time metrics, request traces, and latency bottlenecks across distributed systems, improving visibility and performance of Java applications.

**Environment: HTML5, CSS3, JavaScript, Angular, Microservices, Spring boot, RESTful APIs, Spring Data JPA**, **NoSQL, MongoDB, Kafka, Docker, Kubernetes, CI/CD pipelines**, **Azure DevOps, Maven, SonarQube, Black Duck, SLF4J, Logback, Spring Cloud Sleuth, Azure Functions, Azure Active Directory**(**Azure AD**), **Azure Key Vault, Junit, Test-Driven Development, Git, Datadog APM.**

**Client: Do It Best, Fort Wayne, Indiana Jan 2023 – April 2024**

**Role: Full Stack Java Developer**

**Responsibilities:**

* Developed a responsive, single-page eCommerce application using **React.js, HTML5, CSS3,** and **JavaScript**, ensuring cross-device compatibility and a seamless user experience.
* Implemented dynamic data rendering with **React** and **JSX**, utilizing **react-data-grid** for efficient display of product data fetched via **RESTful APIs** from Java backend services.
* Implemented **React** **state** to manage the dynamic cart content during the checkout process and utilized **props** to pass essential order information between components, ensuring a seamless flow of data.
* Utilized **React Router** for seamless client-side routing across product pages, cart, and checkout, and integrated **RESTful APIs** within **React components** to dynamically fetch and display product details, inventory updates, and order information.
* Implemented **RESTful microservices** using **Spring Boot**, incorporating method-level metrics collection and persistence through **Spring AOP** and **Spring Actuator** to monitor service performance and reliability.
* Designed and developed scalable **REST APIs** for managing products, categories, and inventory using **Spring MVC**, and leveraged **Spring Data JPA** for efficient CRUD operations and seamless database integration.
* Developed asynchronous **Spring JMS** components for reliable messaging between **microservices** and legacy systems and utilized **Spring REST controllers** to expose **JSON**-based **RESTful APIs** for inter-service communication.
* Designed and implemented the **PostgreSQL** database schema for an ecommerce application, ensuring efficient storage and retrieval of data related to products, orders, customers, and transactions.
* Implemented effective concurrency control mechanisms in **PostgreSQL** to manage concurrent user transactions and maintain data consistency in high-traffic e-commerce environments.
* Implemented **CI/CD** pipelines in **CloudBees Jenkins** for automated build and deployment of **Dockerized** microservices on **Azure**, supporting scalable e-commerce applications.
* Implemented **Maven**-based automation within **CI/CD** pipelines for building, testing, and packaging microservices to ensure reliable delivery in a high-traffic e-commerce platform.
* Integrated **CloudBees Jenkins CI/CD** pipelines with **Kubernetes** for automated build, testing, and deployment of **Dockerized** **microservices**, enabling rapid and reliable delivery of scalable e-commerce applications.
* Configured **SonarQube** on the dev server for continuous static code analysis and integrated it with **Jenkins** to enable automated code quality inspection within the **CI/CD** pipeline.
* Ensured adherence to security and performance standards by performing **Checkmarx** scans alongside **SonarQube** analysis, maintaining high-quality and secure Java code.
* Utilized **Spring Sleuth** and **Zipkin** for distributed tracing to monitor and debug complex request flows across **microservices** and leveraged **Splunk** for centralized log management and analysis.
* Implemented **test-driven development (TDD)** methodologies using **JUnit** and **Mockito**, developed comprehensive unit tests, and resolved bugs and defects using **Firebug** and **JIRA**.
* Managed and organized project tasks, user stories, bugs and feature requests utilizing **Azure** **DevOps** and collaborated with cross-functional teams to define user stories, sprint planning and backlog grooming.
* Developed **Kafka Stream** applications in Java to process data across **Kafka topics** using **Kafka Streams DSL** with **Avro** and **JSON** for efficient real-time streaming.
* Leveraged **Git** branching, pull requests, and code reviews to maintain code integrity and facilitate smooth collaboration within the development team.
* Utilized **Kubernetes** to deploy and manage scalable **microservices** supporting critical e-commerce functionalities such as product catalog, user authentication, and order processing, ensuring high availability and fault tolerance.
* Utilized **Rancher Desktop** tools—including **kubectl**, **Helm**, **Skaffold**, and **K9s**—to effectively monitor and manage **Kubernetes clusters**, ensuring optimal performance and rapid issue resolution.
* Developed web solutions and **REST API’s** using **Azure App Service** web apps, **Azure Logic Apps**, **API** apps and implemented **Azure functions** and **API** management for web apps, **Azure Service Fabric** and **Microservices**.
* Implemented the SSO authentication and authorization using **Azure Active Directory** and automated the deployment of **Azure cloud services**, **Azure Key Vault,** and **Azure Redis** cache cluster.
* Designed and installed distributed **ELK** log monitoring system using **Filebeat**, **Redis**, **Logstash** and **Kibana** and **ELK** **stack** for monitoring the logs and metrics.
* Utilized **Postman** to send **HTTP requests** to **Kafka Streams** applications and verify the response data and deployed and managed **stream processing** applications in a distributed environment, such as **Kubernetes**.

**Environment: React.js, HTML5, CSS3,** **JavaScript, JSX**, **react-data-grid**, **React** **state, React Router, React components, RESTful** **Microservices,** **Spring boot, Spring AOP, Spring Actuator, Spring MVC, Spring JMS, Spring Data JPA, Spring REST controllers, JSON, RESTful APIs, PostgreSQL, CI/CD,** **CloudBees Jenkins, Docker,** **Kubernetes, Checkmarx**, **SonarQube,** **Spring Sleuth**, **Zipkin,** **Splunk,** **Microservices, Maven, Git, JUnit, Mockito, Firebug, JIRA, Azure DevOps, RESTful APIs,** **Azure App Service**, **Azure Logic Apps, Azure functions**, **Azure Active Directory,** **Azure Service Fabric**, **Azure cloud services**, **Azure Key Vault,** **Azure Redis,** **Kafka Streams, Avro, JSON, Kafka Streams DSL, kubectl, Helm, Skaffold, Rancher Desktop, K9s, Kubernetes cluster, Filebeat**, **Redis**, **Logstash**, **Kibana**, **ELK** **stack, Postman, HTTP requests, stream processing.**

**Client: BNY Mellon - New York, NY Feb 2022 – Dec 2022**

**Role: Full Stack Java Developer**

**Responsibilities:**

* Developed user interface modules using **HTML5, XHTML, XML, CSS3, JavaScript, jQuery, JSON,** and **AJAX** to deliver dynamic and responsive web applications.
* Developed dynamic, high-performance application features using **Java 8** (including **lambda expressions**) and **Angular**, leveraging dependency injection, data binding, and controllers to optimize UI component management and ensure seamless user interactions.
* Worked on migrating legacy **AngularJS** applications to modern **Angular**, enhancing performance, maintainability, and scalability while ensuring a smooth transition with minimal business impact.
* Implemented **Angular Router** to enable seamless navigation across views as agents performed application tasks, and developed features to consume and process **JSON** and **XML** data from **RESTful web services** using **Angular**.
* Developed reusable **JavaScript** and **Angular** functions to implement robust client-side validations, performed data parsing with **JSON**, and leveraged **AJAX** to dynamically update the UI.
* Implemented transaction management using **Spring MVC** and leveraged **Spring/Hibernate** batch processing for efficient handling of large-scale financial data, while optimizing data retrieval through **Hibernate caching** with **Oracle 12c** database.
* Utilized **Spring MVC**, **Spring Boot**, **Spring Data**, and **AOP** modules to design and develop well-structured, scalable banking applications aligned with the Spring MVC architecture.
* Developed data access layers using **Hibernate ORM** for seamless interaction with the database and created resilient **microservices** with **Spring Eureka** for dynamic **API** route discovery and load balancing across banking service clusters.
* Enhanced security of **RESTful APIs** by implementing **OAuth 2.0** token-based authentication and authorization using **Spring Security**, and safeguarded applications against session fixation, clickjacking, and CSRF attacks to protect sensitive financial transactions.
* Designed and optimized complex **Oracle PL/SQL** stored procedures, functions, and packages to efficiently support critical banking operations such as transaction processing, account management, and regulatory reporting.
* Implemented **Hibernate** second-level caching and query optimization to significantly enhance data retrieval speed and reduce database load in high-transaction banking applications.
* Integrated **Oracle** database with **Spring Data JPA** and **Spring Transaction Management** to ensure seamless and secure interaction between application layers and persistent storage.
* Automated batch data processing jobs leveraging **Spring Batch** and **Oracle PL/SQL** for efficient handling of end-of-day financial reconciliations and reporting.
* Consumed third-party **web services** to retrieve credit card information and utilized **Java 8 Streams** and **Lambda expressions** to efficiently process and store transactional data.
* Developed **Kafka** consumer **APIs** using **Scala** to ingest data from **Kafka topics**, processing **XML** messages through **Spark** Streaming to capture real-time UI updates for banking applications.
* Developed **RESTful web services** using **JAX-RS API** for network communication and implemented **SOAP**-based **web services** with **WSDL** to facilitate secure information exchange across banking systems.
* Installed and configured **IBM MQ** components including listener channels, remote queues, and transmit queues to enable reliable, secure message transportation between banking applications within a mainframe environment.
* Applied design patterns such as **Singleton, Factory, DAO, DTO, Prototype,** and **Composite View** to create modular, maintainable code, and leveraged **Java SE 8** features like **Lambdas, Streams,** and **Predicates** to enhance functionality and reduce code complexity in the service layer.
* Uploaded and managed financial datasets in **AWS S3** buckets and implemented event-driven workflows using **AWS Lambda** to automate banking process orchestration and data pipeline execution.
* Developed and deployed banking applications utilizing **AWS** services including **EC2, S3, Data Pipeline, Elastic Beanstalk, CloudWatch, Server Migration Service, SQS,** and **Kinesis,** ensuring scalability and reliability of data processing workflows.
* Deployed and monitored **microservices** on Cloud Foundry using **Spring Cloud**, managing service discovery, domain configurations, and routing via **Spring Cloud Service Registry** to support scalable and resilient banking applications.
* Utilized **Maven** for build automation, dependency management, reporting, and documentation, and executed nightly and weekly builds and releases using Ant and Maven build scripts.
* Implemented **CI/CD** pipelines with **Jenkins** in a **Kubernetes** environment, leveraging **Docker** containers to build, test, and deploy Java-based banking applications efficiently.
* Developed **RESTful** endpoints to **cache** application-specific financial data in **Redis** in-memory clusters, improving performance and enabling rapid data retrieval through **REST APIs**.
* Created unit tests by mocking external dependencies with **Mockito**, accelerating test development, and leveraged monitoring tools like **Dynatrace, Splunk,** and **New Relic** to track application and infrastructure performance.
* Achieved over 80% code coverage through comprehensive unit and integration tests using **JUnit** and **Mockito**, with coverage analysis performed using **JaCoCo**.

**Environment: HTML5**, **XHTML, XML, CSS3, JavaScript, JQUERY, JSON, AJAX, Angular, Angular.js, Angular, Spring Boot, Spring MVC**, **Spring Security, OAuth 2.0, Spring Data JPA**, **Spring Transaction Management, Spring Batch, AOP, Spring Eureka**, **Hibernate, Hibernate ORM, Hibernate caching, Microservices**, **Singleton, Factory, DAO, DTO, Prototype, Composite View,** **AWS, AWS Lambda**, **Amazon S3, EC2, Lambda expressions, Streams, Webservices, Java 8, JUnit, Oracle, Oracle PL/SQL, Spring Cloud Foundry, Spring Cloud Service, AWS EC2, S3, Data Pipeline, Elastic Beanstalk, CloudWatch, Server Migration Service, SQS, Kinesis, ANT, Maven, JSON, Jenkins, Kafka, Spark, Scala, RESTful, Redis, Junit, Mockito, Redis, Cache, Docker, Kubernetes, IBM MQ, SOAP, WSDL, JAX-RS, Splunk, Dynatrace, New Relic.**

**Client: College Board, New York, NY Aug 2021 – Jan 2022**

**Role: Full Stack Java Developer**

**Responsibilities:**

* Developed UI components for email and link sharing of documents and files for a content management system and added new features using **ReactJS.**
* Integrated real-time collaboration features using **React** to facilitate seamless interaction and collaboration among students, promoting an interactive and participative learning environment.
* Created Controllers using **ReactJS** to handle events triggered by clients and send request to server and to create views to hook up models to the DOM and synchronize data with server as a single page Application.
* Utilized **Node.js** for rapid development of cloud **microservices**, leveraging frameworks like **Express.js** and developed **Node.js** applications with minimal manual configuration, emphasizing agility and speed.
* Integrated **Node.js** with suitable **Object-Relational Mapping (ORM)** libraries like **Sequelize** for seamless interaction with **Oracle** databases, fostering efficient data management in applications.
* Implemented the **MVC** architecture using **Express.js**, facilitating the smooth flow of data between the front-end and business layers in **Node.js** applications.
* Implemented real-time analytics using **Azure Cosmos DB** change feed and **Node.js** and leveraged change feed to capture and process data changes in real-time, enabling instant insights and updates.
* Integrated **Azure Cosmos DB** with **Node.js** framework **Express.js** for building robust and scalable web applications and ensured secure and efficient communication between the **Node.js** application and **Cosmos DB**.
* Integrated **Black Duck** and **SonarQube** into the continuous integration/continuous deployment **(CI/CD)** pipeline for **Node.js** applications and automated scans to ensure the early detection of vulnerabilities and maintain code quality.
* Utilized **ELK stack** (Elasticsearch, Logstash, Kibana) to perform log analysis, identify patterns, and troubleshoot bugs in the **Node.js** codebase.
* Configured **Docker** container for automated testing purposes, utilized **JUnit** for comprehensive **unit testing** and proficiently worked on **JUnit** test cases, employing **Mockito** for data mocking to effectively test application classes.
* Developed test scripts to implement test cases, test scenarios, and features for BDD (Behavior Driven Development), using **Jbehave** and implemented automation using **Selenium** **WebDriver**, **JAVA**, **BDD**, **Jbehave**, **Serenity**, **SVN** and **Maven.**
* Developed and deployed **serverless** Java functions using **Azure Functions** for educational use cases, such as automating administrative tasks or handling real-time interactions.
* Integrated **Azure Active Directory** with Java educational applications to provide secure authentication and authorization for students and faculty and implemented **Single Sign-On (SSO)** for a seamless and secure user experience in educational platforms.
* Optimized Java code for efficient interaction with **Azure Blob Storage** and **Azure Table Storage**, supporting the storage and retrieval of educational content.
* Developed a **Node.js**-based integration service to receive EDI messages from the N4 terminal operating system on **Azure**, incorporating **Azure Service Bus**, **Azure Functions**, **Azure Storage Account**, and **Azure Cosmos DB** for seamless processing and storage.
* Designed and implemented **SOAP** and **Restful** Web services using **WSDL**, **SOAP**, **JAX-WS**, **CXF**, and **JAX-RS**, and successfully consumed **SOAP** web services and **RESTful** **JSON** web services to contribute to the creation of an automated reporting system.
* Developed modules using **Core Java** **APIs**, **Java collection**, **Java 8**(Streams), **multi-threading**, and **object-oriented designs** and involved in writing **Maven** scripts for automated deployment of the application and managed **GitHub** Projects.
* Integrated existing NodeJS and Java micro-services with **Kafka** connector to act as publisher or subscriber to a particular **Kafka** topic and designed and developed Integration solutions based on **IBM MQ** and Message Broker technologies.
* Utilized **Splunk** dashboards to highlight key business metrics such as transaction volume and average processing time, as well as to measure the performance of other third-party systems.
* Configured and deployed **tomcat** application and web **apache** servers, **Redis** Cache and **Solr** Servers and built environments for test, development, and production from the ground up.

**Environment: React, ReactJS, Node.js, Express.js, Object-Relational Mapping (ORM)**,**Sequelize, microservices, Azure, Azure Cosmos DB, Black Duck**, **SonarQube, CI/CD,** **ELK stack**, **Docker, Junit, Unit testing, Mockito, Selenium** **WebDriver**, **JAVA**, **BDD**, **Jbehave**, **Serenity**, **SVN**, **Maven, serverless,** **Azure Functions, Azure Active Directory, Single Sign-On (SSO), Azure Blob Storage**, **Azure Table Storage**, **Azure Service Bus**, **Azure Functions**, **Azure Storage Account**, **SOAP**, **Restful**, **Web services,** **WSDL**, **SOAP**, **JAX-WS**, **CXF**, **JAX-RS, RESTful** **JSON, Core Java** **APIs**, **Java collection**, **Java 8**, **multi-threading**, **object-oriented designs, GitHub, Kafka, IBM MQ, Splunk, tomcat, apache, Redis, Cache**, **Solr.**

**Client: Bank of the West, San Francisco, CA Nov 2019 – June 2021**

**Role: Application Software Developer**

**Responsibilities:**

* Developed responsive, user-friendly interfaces using **HTML, CSS, Bootstrap,** and **JavaScript**, ensuring cross-platform compatibility and a seamless user experience across devices.
* Implemented dynamic, interactive features with **JavaScript** and **jQuery**, such as real-time updates, transaction history, and **AJAX**-based form submissions, enhancing interactivity and performance.
* Customized and optimized UI components with **Bootstrap, JavaScript,** and **CSS**, ensuring functionality alignment and delivering a consistent, intuitive user experience.
* Designed and implemented **MVC**-based web applications using **Spring MVC** to create scalable and maintainable banking services, ensuring clean separation of concerns between the model, view, and controller.
* Developed and integrated **RESTful APIs** with **Spring MVC** to facilitate seamless data exchange between front-end user interfaces and back-end services, optimizing the performance of customer-facing banking features.
* Utilized **Spring MVC** features such as form validation, data binding, and exception handling to streamline user interactions, enhance data accuracy, and ensure a smooth user experience across banking modules.
* Integrated **Spring MVC** with **Spring Security** to implement role-based access control (RBAC) and ensure secure authentication and authorization for users accessing sensitive financial data.
* Optimized **Spring MVC** application performance by implementing caching strategies, improving response times for frequently accessed financial data, and reducing server load.
* Developed and optimized complex **SQL** **queries** and **PL/SQL stored procedures** to improve the efficiency of data retrieval and transaction processing in high-performance **Oracle** Database environments.
* Managed data integrity and database transactions within **Oracle** Database, implementing indexing, partitioning, and performance tuning techniques to handle large-scale datasets and ensure system reliability.
* Implemented **CI/CD** pipelines using **Jenkins** for automated integration and delivery, following Agile methodology to ensure iterative development and fast, reliable deployment of Java applications.
* Utilized **AWS CloudWatch** to monitor and log Java application performance, configure alarms for critical issues, and ensure proactive troubleshooting and optimal application health.
* Developed web services using **REST** interfaces using **MuleSoft's** Mule **Enterprise Service Bus (ESB)** platform to deliver fast, reliable, and scalable integration API code.
* Developed messaging solutions with **RabbitMQ** and **Spring AMQP** to support asynchronous communication between backend services, enhancing system performance and ensuring seamless data flow in a Spring MVC-based application.
* Utilized **Amazon S3** for storing application and database snapshots, and configured **S3** lifecycle policies to manage log retention, including automated deletion and archival of logs based on predefined retention policies.
* Developed Java-based solutions using the **AWS SDK** to interact with various **AWS services**, including **S3** for data storage and retrieval, and **SQS** for bulk email processing, ensuring efficient data backup and archiving with **Glacier**.
* Implemented **Test-Driven Development (TDD)** using **JUnit** and **Spring Integration Test** to validate application functionality and **AWS integrations** **(S3, SQS)**, leveraging Advanced **REST Client (ARC)** for **API testing**.
* Managed version control with **Git** and automated build, test, and deployment processes using **Maven**, integrating with **AWS** services for efficient application deployment.

**Environment: HTML, CSS, Bootstrap, JavaScript, jQuery, AJAX, Java, Spring MVC, RESTful APIs**, **Spring Security, SQL** **queries**, **PL/SQL stored procedures**, **Oracle, CI/CD**, **Jenkins**, **AWS, AWS CloudWatch,** **REST,** **MuleSoft,** **Enterprise Service Bus (ESB), RabbitMQ, Spring AMQP, Amazon S3, AWS SDK, AWS services , SQS, Test-Driven Development (TDD)**, **Junit**, **Spring Integration Test, AWS integrations** **(S3, SQS)**, **REST Client (ARC)**, **API testing, GIT,** **MAVEN.**

**Client: REPLICON, Bangalore, Karnataka, India Feb 2017 – Nov 2019**

**Role: Java Developer**

**Responsibilities:**

* Involved in all phases of project from analysis and requirement phase to delivering the project.
* Worked in **Agile** environment and participated throughout the SDLC process and implemented **MVC** design pattern for the application for the front-end appearance and log in Screens.
* Developed applications using application lifecycle management to deliver reliable applications.
* Developed the view-controller components using **Servlet/JSPs, JavaScript, CSS, and HTML** and involved in creating **JSP** pages for presentation of the data to the end user.
* Developed various modules using technologies like **AngularJS, JSON, and JavaScript** and involved in developing forms using **HTML** and performing **client-side** validations using **JavaScript**.
* Applied **J2EE** design patterns such as **MVC, Singleton, and Data Access Objects** in the architecture and used **JDBC** as **DAO** class to persist the data on to **Oracle** Database.
* Wrote **ANT** scripts to deploy the components on Jetty application Server and did regular code reviews and fixed major bugs without side effects.
* For unit testing of the program, the **JUnit** framework was used, and **Log4j** was used to record the log of runtime exceptions. Worked with Testers in resolving various defects in the application Prepared **JUnit** test cases and executed the test cases using **JUnit**.
* Application development using **Eclipse IDE**, building of applications with **Maven** scripts, testing of application with **RAD local Web Sphere Server7.0**
* Throughout the user acceptance testing, joint system testing, and system testing phases, used **Firebug** to keep track of issues and participate in bug repair.

**Environment: Java, HTML, JSP, Servlet, EJB, Agile**, **MVC, AngularJS, JSON, JDBC, Oracle, ANT, Maven, JUnit, Java Script, CSS, Ant, Log4j, RAD local Web Sphere Server7.0**, **Firebug, Eclipse.**