**SAIKUMAR TAMMINENI**

**Sr. Java Full Stack Developer**

**Mobile**: ‬763-913-6029

**Email**: saitammin@gmail.com

**PROFESSIONAL SUMMARY**

* Having 11+ years of programming and enterprise-level application development experience including software information strategy planning, requirement analysis and prototyping, design, development, testing, debugging, performance tuning and production support.
* Strong knowledge in Object Oriented Concepts, Object Oriented Design (OOD), Object Oriented Analysis (OOA), Programming and its implementation.
* Expertise in designing and developing **web and enterprise-level applications** using **J2EE technologies** like **JSP, Servlets, JDBC, and Oracle**.
* Worked on **Java Dependency Analysis Tool (jeeps)** using **Java 17** and **JAR file attributes** on **Java 8**.
* Responsible for **building a framework** to support the **easy creation and consumption of REST Microservices** using **Java 8, JAX-RS, AngularJS (SPA), and Spring**.
* Experience in **Struts, Spring (Core, MVC, AOP, Security, Boot), and Hibernate**.
* Developed Struts **form beans, action classes, and JSPs** following Struts framework standards.
* Involved in the **development of model, library, Struts, and form classes (MVC)**.
* Used **SAX XML API** to parse XML and populate values for beans.
* Implemented **RESTful Web Services** for data transportation between multiple systems.
* Utilized **Java 8 features** like **Lambda expressions** and **Stream API** for bulk data operations on collections, improving application performance.
* Designed and developed **application architectures** based on **Struts Framework** using the **MVC design pattern**.
* Developed **Struts Action classes** using the **Struts controller component**.
* Used **PL/SQL stored procedures** for scheduled execution within applications.
* Experience in **Big Data ingestion** using streaming tools like **Flume, Sloop, Kafka, and Storm**.
* Hands-on experience implementing **data analytics and big data solutions** leveraging **Azure, AWS, Google Cloud, Hortonworks, Cloudera-CDH, Spark SQL, Spark Streaming, and Kafka**.
* Knowledgeable in **cloud computing** (**AWS EC2**) and **microservice container deployment in Docker**.
* Worked in **key supply chain domains** like **sourcing, procurement, and logistics**.
* Worked with **supply chain management** to enhance the **creation and delivery of product services**, including **logistics and customer service**.
* Involved in **underwriting process development**, which involved **communication with external systems** using **IBM MQ and JMS**.
* Experience designing and developing UIs with **Thymeleaf, JSP, HTML5, CSS3, AngularJS, and JavaScript**.
* The **CMS and server-side interaction** were developed using **Web Services** and exposed via **JSON and jQuery**.
* Involved in designing and estimating templates and components developed using **Day CMS (Communique)**.
* Used **display tag libraries** for UI decoration and **display tables** for reports and grid designs.
* Strong experience in **Shell scripting**, **SQL Server**, **Unix/Linux**, and **Python scripting** with a focus on **DevOps tools, CI/CD, and AWS Cloud Architecture**.
* Build highly available **content distribution sites** using **CI/CD tools** such as **Chef, Ansible, Docker, Maven, Jenkins, Jira, and Kubernetes**.
* Used **TDD methodologies**, writing failing tests first, making necessary corrections, and refactoring the code.
* Involved in **preparing high-level and detailed design** of enterprise applications using **J2EE**.
* Worked with **Solace messaging appliance administration and development**.
* Implemented **database connectivity** using **JDBC with Oracle 9i** as the backend.
* Extensive experience in **writing SQL queries, stored procedures, and triggers** for efficient data handling.
* Expert in **web services** (SOAP, RESTful), **ORM** (Hibernate, JPA), **XML, JSON, UML, ANT, JSTL**, and **Apache Log4J**.
* Worked on **core Java**, using file operations to read system files and present them on JSPs.
* Used **Jasper** to generate **rich-content reports**.
* Provided **troubleshooting and error handling support** across multiple projects.
* Experience in **using Java IDE tools** like **NetBeans** and **familiarity with Eclipse**.

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Programming Languages** | Java 21/17/11/8, Core Java, C, SQL. |
| **Software Methodologies** | Agile, Scrum, SDLC Waterfall. |
| **J2EE Technologies** | Java Beans XSD, JAX-RS, JAX-WS, SOAP, WSDL. |
| **Web Technologies** | HTML, JavaScript, AJAX, Bootstrap, JSON, Angular, React JS, Kubernetes. |
| **Web service Frameworks**  | Spring 4.3, Hibernate 4.3, Struts (2.5/2.3), Spring MVC, Spring IOC, Spring Boot, Spring AOP, Kafka |
| **Web Services** | SOAP,  |
| **Testing Tools**/**Others** | Junit, Mockito, Log4J, Postman.  |
| **Technologies/Tools**  | AWS, EC2, Azure, Docker, JIRA |
| **Build Tools** | Maven 3.3.9, Ant 1.10, Grade 3.4.1, Jenkins |
| **Version Control Tools** | GIT, GITHUB 2.12.0, SVN, Bit Bucket. |
| **Application**/**Web Servers** | Web Logic Server 12c, Apache Tomcat (8.5/8.0), Apache HTTP Server, Joss. |
| **IDE Tools** | Spring Tool Suite 3.8.3, Eclipse, IntelliJ IDEA 2017.1, Net Beans, Visual Studio. |
| **Operating Systems** | Windows, Linux, Unix, Mac OS |

**PROJECT EXPERIENCE**

**Client: Fiserv, New Jersey Duration: June 2024 – Current**

**Role: Sr. Full Stack Java Developer**

**Responsibilities**

* **Designed and developed** responsive web applications for **financial services** using **Angular 15**, **HTML5**, **CSS3**, and **SCSS**, ensuring optimized user experience across multiple devices.
* Implemented **client-side validation** and **form handling** in Angular applications for improved user interaction and data integrity in **financial transactions**.
* Developed **dynamic dashboards** for real-time financial reporting using Angular, ensuring seamless data presentation for end users.
*  Developed scalable front-end applications using **Angular 15**, **TypeScript**, **JavaScript**, and **RxJS**, ensuring efficient data handling and real-time updates with **RESTful APIs** integration.
*  Built dynamic user interfaces with **Angular 15**, **JavaScript**, and **TypeScript**, utilizing **Redux** for state management and **Reactive Forms** for handling complex forms and validations.
* **Developed RESTful APIs** for **financial applications** using **Java 17/21** and **Spring Boot 3.3.6**, ensuring secure, efficient, and scalable services for handling financial transactions.
* Developed robust backend systems using **Java 17** and **Java 21**, implementing **multi-threading**, **lambda expressions**, and **streams** for performance optimization in high-volume financial applications.
* Utilized **Java 17** and **Spring Boot 3.3.6** to design and implement **RESTful APIs**, enabling efficient communication between microservices in a distributed environment.
* Leveraged **Java 21** features such as **pattern matching**, **records**, and **enhanced switch statements** to simplify code and improve maintainability for large-scale financial systems.
* Designed and implemented **microservices** for **transaction processing** and **payment services** in the financial sector, leveraging **Spring Boot** and **Spring Data JPA** to ensure robust data handling.
* Integrated Swagger UI with Spring Boot microservices to auto-generate interactive API documentation, enabling real-time testing and faster integration for client and internal consumers.
* Developed and maintained OpenAPI 3.0-compliant specifications for REST APIs, facilitating secure and standardized communication across Fiserv’s distributed financial platforms.
* Implemented role-based security and JWT token authorization flows within Swagger documentation to support secure API access and simulate real-world request scenarios.
* Streamlined API-first development by leveraging Swagger Codegen to scaffold client SDKs and stubs, reducing onboarding time for frontend teams and third-party partners.
* Collaborated with backend and UI teams to ensure end-to-end consistency between API contracts, Swagger definitions, and Angular-based frontend data models.
* Implemented **OAuth 2.0** and **JWT** authentication mechanisms for secure user access and data integrity in the backend of financial services.
* Developed and implemented automated workflows using Python scripts, reducing manual processes and improving the efficiency of financial transactions and reporting in the automotive sector
* Leveraged Python libraries like Pandas, NumPy, and SQLAlchemy to automate data extraction, transformation, and loading (ETL) processes for seamless integration between automotive financial systems and external data sources
* Used **Spring Data JPA** for efficient database access and management with **MySQL** and **PostgreSQL**, implementing **CRUD** operations and ensuring data integrity.
* Implemented **Spring Batch** for processing large datasets and **Spring Boot Actuator** to monitor the health and performance of the application.
* **Integrated Spring Cloud Config** for centralized configuration and **Spring Cloud Netflix Eureka** for **service discovery** and load balancing across financial microservices.
* Enabled **Spring WebFlux** for reactive programming and used **Spring AMQP** with **RabbitMQ** for asynchronous, event-driven communication between microservices.
* Optimized data fetching with **Spring Data JPA** specifications and **Spring Cache** for improved performance in high-traffic financial applications.
* Used **Spring Session** for centralized session management and **Spring Integration** for handling message-driven architecture across services.
* **Migrated legacy financial applications** from monolithic architecture to **microservices** using **Spring Boot**, optimizing performance and scalability.
* Integrated **GraphQL** for efficient querying of financial data, allowing clients to retrieve precise transaction details and balances in real time.
* **Worked extensively with relational databases** like **PostgreSQL**, **MySQL**, and **Oracle** for managing large sets of financial transaction data, ensuring data integrity and compliance with regulatory standards.
* Implemented complex queries and optimized **SQL** procedures to handle high volumes of transactional data, improving database performance for financial services.
* Utilized **AWS DynamoDB** for NoSQL storage of transactional data, ensuring low-latency and scalable access to real-time financial records.
* **Managed database migrations** from on-premise systems to cloud-based databases (AWS RDS, DynamoDB), ensuring smooth transition of financial data with minimal downtime.
* Leveraged **AWS services** such as **EC2**, **Lambda**, **EKS**, **S3**, **DynamoDB**, and **CloudWatch** to build scalable, fault-tolerant applications, automating infrastructure provisioning, real-time monitoring, and optimizing storage and performance.
* Developed and executed **unit tests** and **integration tests** using **JUnit 5** and **Mockito** to ensure robust and secure backend services for financial applications.
* Utilized **Cucumber 6.x** for behavior-driven development (BDD) testing, ensuring that **financial services** meet business requirements and user expectations.
* **Automated API testing** using **Postman**, ensuring that financial REST APIs function correctly and securely before production deployment.
* **Used Jenkins** for **CI/CD pipeline management**, automating the build, test, and deployment process for financial applications to ensure timely and reliable releases.
* Employed **SonarQube** for continuous code quality inspection, ensuring that all financial code adheres to industry standards and secure coding practices.
* **Migrated legacy financial applications** to **cloud-native microservices** using **Spring Boot** and **Docker**, improving scalability and reducing infrastructure costs.
* **Optimized database performance** for financial services by implementing **indexing**, **query optimization**, and **database partitioning** for handling large datasets of financial transactions.
* **Implemented cloud migrations** to **AWS services** like **RDS**, **DynamoDB**, and **S3**, ensuring smooth transitions and enhanced performance for financial applications.
* Utilized **Swagger** for API documentation, allowing easy integration and testing of financial APIs by internal teams and clients.
* Leveraged **Redis** for caching frequently accessed financial data, reducing load on databases and improving the response time for transaction processing.
* Monitored the performance of financial applications using **Grafana** and **Prometheus**, enabling real-time tracking of system health and identifying bottlenecks.
* Deployed and managed containerized applications on Azure Kubernetes Service (AKS), improving scalability and reducing deployment time by leveraging Kubernetes orchestration.
* Provisioned AKS clusters using Azure CLI and Azure Portal, ensuring optimal configuration and high availability for production workloads.
* Managed containerized applications using **Kubernetes 1.22**, orchestrating deployments, scaling, and monitoring with integrated **Git** for version control, enabling seamless CI/CD workflows for microservices applications.
* Implemented **Jenkins** for continuous integration and delivery (CI/CD), automating build, test, and deployment pipelines; utilized **Docker** for containerizing applications, ensuring consistent environments across development, testing, and production.

**Environment**: Java 17, JDK 7, Spring 3.1, Hibernate 4.2, JSF 2.2, AngularJS, JavaScript 1.5, PHP 4/5, XHTML 1.0, Oracle 11g/12c, JDBC 4.2, REST, SOAP, Git, JUnit, Oracle SQL 4.0, ANT, Eclipse 4.3, WebLogic 11g, Apache Tomcat, AWS, Microservices, Docker, Kubernetes, Maven, React, TypeScript, MongoDB, MySQL, PostgreSQL, Redis, RabbitMQ, Nginx, Jenkins, Terraform, GraphQL, Elasticsearch, Kafka.

**Client: State of NewMexico, NM Duration: Nov 2022 to May 2024**

**Role: Sr. Full Stack Java Developer**

**Responsibilities**

* Participating in application solutions including assisting with planning and architectural design, development, resolution of technical issues, and application rationalization.
* **Developed dynamic, responsive user interfaces** using **JavaScript (ES6+)** and **TypeScript**, ensuring cross-browser compatibility with **HTML5**, **CSS3**, and **Bootstrap**.
* **Built and optimized Single Page Applications (SPA)** using **Angular (4/6/12)**, **RxJS**, and **NgRx**, improving performance and scalability.
* **Integrated backend services with frontend applications** via **Node.js** and **Express.js**, developing **RESTful APIs** to ensure seamless communication between client and server.
* **Implemented authentication and authorization** in **Angular** with **JWT** and **OAuth 2.0**, securing user access and data protection across microservices.
* **Optimized Angular applications** by lazy loading modules and implementing **AOT (Ahead-of-Time)** compilation for faster startup times.
* **Built real-time web applications** with **Node.js** and **Socket.IO**, enabling live data updates and seamless communication between clients and servers.
* **Integrated third-party APIs** into **Angular** apps using **Node.js**, including payment gateways, social media authentication, and geolocation services.
* **Developed server-side logic** in **Node.js**, creating middleware for error handling, request validation, and logging to improve system reliability and performance.
* **Implemented data caching** in **Node.js** using **Redis**, significantly improving application performance by reducing database calls.
* **Created responsive, mobile-first applications** using **Angular** and **TypeScript**, ensuring an optimal user experience across all devices.
* utilized Python to build machine learning models that predict patient outcomes, assisting healthcare providers in making data-driven decisions and optimizing treatment plan
* Integrated real-time health data from wearable devices into centralized systems using Python, enabling accurate tracking of vital signs and providing actionable insights for both patients and doctors
* **Developed scalable and high-performance backend services** using **Java 17/11**, leveraging new language features like **Pattern Matching**, **Sealed Classes**, and **Records** to improve code readability and reduce boilerplate.
* **Optimized application performance** by migrating legacy systems to **Java 17/11**, utilizing enhanced **JVM** capabilities and garbage collection improvements for faster execution and reduced memory footprint.
* **Designed and implemented microservices architectures** using **Java 17/11** and **Spring Boot**, incorporating **RESTful APIs** and **Spring Security** for secure, efficient, and maintainable systems.
* **Enhanced code quality and maintainability** by adopting **Java 17/11** features, including **var**, **Text Blocks**, and **Foreign Function API**, resulting in cleaner, more concise, and more reliable codebase.
* **Designed and developed microservices** using **Spring Boot 3.x**, **Spring Cloud**, and **Spring Security**, ensuring highly scalable, secure, and decoupled service architecture.
* **Implemented REST APIs** using **Spring Boot 3.x** and **Spring MVC**, creating a robust framework for backend communication and integrating with **Spring Data JPA** for database operations.
* **Secured microservices** with **JWT**, **OAuth 2.0**, and **Spring Security**, enforcing secure communication across services and ensuring safe access to endpoints.
* Implemented Swagger UI and OpenAPI specifications for RESTful services built with Spring Boot, ensuring standardized API documentation across various State of Michigan agencies and departments.
* Collaborated with cross-agency teams to create interactive API portals using Swagger, improving developer onboarding and integration time for applications in public health and unemployment systems.
* Secured Swagger-documented APIs with OAuth2.0/JWT authentication flows, aligning with State of Michigan’s data security and compliance mandates (HIPAA, CJIS).
* Automated generation of API contracts using Swagger annotations in Java, streamlining versioning and change tracking for services supporting citizen engagement and eligibility systems.
* Integrated Swagger with CI/CD pipelines (Jenkins/GitLab) to validate API contracts during build cycles and ensure seamless API updates across development, staging, and production environments.
* Used **Spring Boot Actuator** and **Spring Batch** for monitoring and managing application health, as well as implementing efficient batch processing for large datasets.
* **Integrated microservices** using **Kafka** and **Kafka Streams** for asynchronous message-driven architecture, improving data processing capabilities and scalability.
* **Implemented data persistence layer** using **Spring Data JPA**, **Hibernate ORM**, and **SQL** databases like **PostgreSQL**, **MySQL**, and **Oracle** to store and manage data.
* **Wrote complex queries** and **stored procedures** using **PL/SQL** and **T-SQL** for **Oracle** and **SQL Server**, improving backend efficiency and business logic processing.
* **Worked extensively with SQL databases** including **PostgreSQL**, **Oracle**, **MySQL**, and **SQL Server**, writing complex queries and optimizing database performance.
* **Designed and implemented NoSQL databases** using **AWS DynamoDB** and **Azure Cosmos DB**, ensuring high scalability and low-latency data access for modern applications.
* **Implemented data ingestion pipelines** using **Spark 3.0**, **Flink**, and **Kafka** for big data processing, transforming large datasets in real-time and storing them in **AWS DynamoDB** and **Cosmos DB**.
* **Migrated legacy applications** from monolithic architecture to **microservices-based architecture**, ensuring improved scalability and flexibility.
* **Integrated third-party services and APIs** in **Spring Boot** microservices, enhancing functionality while maintaining efficient, maintainable code.
* **Deployed and managed applications** on **AWS EC2**, **EKS**, **S3**, **Lambda**, and **DynamoDB**, utilizing AWS services for scalable and efficient cloud architecture.
* **Implemented Kubernetes for container orchestration** on **AWS EKS**, improving the deployment and scalability of microservices across multiple environments.
* **Automated infrastructure management** using **Terraform**, **CloudFormation**, and **Ansible**, enabling efficient provisioning and configuration management of cloud resources.
* **Built CI/CD pipelines** using **Jenkins**, **GitHub Actions**, and **Docker**, ensuring seamless integration, deployment, and version control management.
* **Utilized Docker for containerization**, packaging microservices in containers for seamless deployment across various environments.
* **Implemented Redis caching layer** to optimize response times for high-traffic applications, reducing database load and improving the overall performance of the system by leveraging **Redis** as an in-memory data store.
* **Designed and deployed Redis-backed message queues** to support real-time, asynchronous communication between microservices, enhancing the scalability and responsiveness of the application.
* **Designed and deployed multi-container pods to ensure high availability and fault tolerance of critical applications on AKS.**
* **Monitored cloud infrastructure** and services using **AWS CloudWatch** and **Grafana**, improving visibility and proactive troubleshooting of issues.
* **Configured logging and monitoring** for applications using **Prometheus**, **Grafana**, and **AWS CloudWatch**, enabling real-time monitoring and performance optimization.
* **Developed unit tests** using **JUnit**, **Mockito**, **Jasmine**, **Karma**, and **Mocha** for both **Angular** and **Node.js** applications, ensuring high code coverage and reliability.
* **Configured Swagger** for automatic API documentation and testing, improving development workflow and client integrations.

**Environment**: Java 17/11, Spring Boot 3.x, Spring Data JPA, Spring Security, Spring MVC, REST, GraphQL, Hibernate 5.x, JWT, OAuth 2.0, Node.js 18, Angular 12, AngularJS, TypeScript, JavaScript (ES6+), HTML5, CSS3, Bootstrap, React JS, Vue.js, Redis, Kafka, Docker, Kubernetes (v1.22+), AWS (EC2, CloudWatch, Lambda, S3, EKS, ECR, DynamoDB), Jenkins, Git, Maven, Gradle, SonarQube, Swagger, Prometheus, Grafana, PostgreSQL, MySQL, SQL Server, Oracle, IBM MQ, RabbitMQ, Apache Tomcat, IntelliJ IDEA, JUnit, Karma, Jasmine, Mocha, Mockito, Spock, Microservices, CI/CD, Agile, Apache Kafka, Redis Pub/Sub, Spring Boot Actuator, Spring Batch, CloudFormation, Terraform, Postman.

**Client:** **CVS, RI Duration: Sep 2020 to Oct 2022**

**Role: Full Stack Java Developer**

**Responsibilities**

* Involved in the complete SDLC including the design of System Architecture, development of System Use Cases based on the functional requirements.
* Participated in **Agile Scrum meetings**, Sprint planning, task estimation, retrospectives, and Sprint demos.
* Developed dynamic web applications using **HTML5**, **CSS3**, **JavaScript**, and **TypeScript** for rich front-end functionality and responsive UI design.
* Proficient in **ReactJS (v16)** and **React Native** for building scalable, high-performance web and mobile applications with **Redux**, **React Hooks**, and **Context API** for efficient state management.
* Built reusable, modular components in **ReactJS**, improving UI performance, scalability, and maintainability across large applications.
* Optimized **React** application performance through **Lazy Loading**, **Code Splitting**, and **React.memo** for faster loading and efficient rendering.
* Implemented **mobile-first design** in **React Native** with **TypeScript** and **Bootstrap**, ensuring seamless cross-platform mobile development.
* Led the **migration of legacy web applications** to **ReactJS** and **React Native**, enhancing performance, maintainability, and cross-platform support.
* Developed and maintained **Microservices** using **Spring Boot (v2.4)**, **Spring MVC**, and **GraphQL** to provide scalable back-end services and **RESTful APIs**.
* Integrated **Spring Data JPA** for efficient database interactions, and **Spring Security** for implementing secure authentication and authorization across microservices.
* Configured **JWT** and **OAuth 2.0** for implementing secure user authentication and protecting endpoints, improving security across services.
* **Migrated legacy authentication systems** to **OAuth 2.0** and **JWT**-based authentication to ensure more secure, scalable, and modernized application architecture.
* Implemented **unit testing** with **JUnit 5**, **Mockito**, and **Jest** for **ReactJS** components and backend services, ensuring high-quality code.
* Optimized **back-end performance** using **Spring Batch** for large-scale data processing tasks, and **Spring Boot Actuator** for monitoring and health checks.
* Built and deployed **cloud-native applications** leveraging **AWS EC2**, **Lambda**, **S3**, **CloudWatch**, **EKS**, and **ECR** for scalability, monitoring, and performance optimization.
* Leveraged **Kubernetes (v1.22)** for container orchestration, deployment, and scaling of containerized applications on **AWS EKS**, ensuring high availability and reliability.
* Managed infrastructure using **Terraform**, automating the provisioning and configuration of cloud resources across **AWS** environments.
* Designed and documented FHIR-compliant RESTful APIs using Swagger UI and Spring Boot to support secure, standards-based data exchange between CVS pharmacy systems and external healthcare providers.
* Integrated Swagger with CVS's clinical data platforms, enabling real-time interactive testing and documentation for APIs used in prescription processing, benefits eligibility, and patient engagement tools.
* Leveraged Swagger Codegen to generate Java client SDKs for consumption by mobile and kiosk apps, reducing manual effort and increasing consistency across CVS digital touchpoints.
* Implemented automated Swagger validation in CI/CD pipelines (Jenkins + SonarQube) to enforce contract accuracy and prevent deployment of undocumented or breaking API changes in healthcare systems.
* Extended Swagger UI with custom themes and metadata to align with CVS Health’s internal API governance standards and improve clarity for internal developer portals.
* Integrated **Redis** for caching, improving application performance by reducing data retrieval times.
* Utilized **Kafka** (v2.6) for real-time messaging and optimized **topic partitioning** for efficient message distribution and processing in microservices.
* Configured **Kafka** clusters for managing message queues and ensuring optimized message throughput between microservices.
* Integrated **IBM MQ** and **RabbitMQ** for reliable message queuing and asynchronous communication, enhancing the decoupling of services.
* Managed **application logging and monitoring** using **AWS CloudWatch**, **Grafana**, and **Prometheus**, enabling proactive issue detection and resolution.
* Deployed microservices on **Tomcat** for handling **HTTP requests**, optimizing response times, and ensuring efficient scalability.
* Built **Dockerized applications** for consistent environments across **development** and **production**, simplifying deployment processes.
* Configured **CI/CD pipelines** using **Jenkins** and **GitHub Actions**, automating build, deployment, and testing workflows to support **microservices** and facilitate rapid deployment cycles.
* Managed and optimized database interactions with **PostgreSQL**, **MySQL**, **SQL Server**, and **AWS DynamoDB**, ensuring reliable, high-performance data storage and retrieval.
* Optimized backend performance using **caching strategies (Redis), indexing, and query optimization**.
* Improved system reliability by implementing **circuit breakers & fallback strategies** using **Resilience4J**.
* Designed **real-time monitoring dashboards** for tracking API latency, error rates, and system health.

**Environment**: **Java (8/11), Spring Boot (v2.4), ReactJS (v16), React Native, TypeScript, GraphQL, Spring Data JPA, Spring MVC, Spring Security, JWT, OAuth 2.0, Spring Batch, Spring Boot Actuator, AWS EC2, AWS Lambda, AWS S3, AWS DynamoDB, AWS CloudWatch, AWS EKS, AWS ECR, Kubernetes (v1.22), Redis, Kafka, IBM MQ, RabbitMQ, Jenkins, GitHub Actions, Terraform, Docker, SonarQube, Grafana, Prometheus, Postman, Maven, Gradle, Git, MySQL, PostgreSQL, SQL Server, JDBC, IntelliJ IDE.**

**Client:** **TIAA,NY Duration: April 2018 to Aug 2020**

**Role: Full Stack Java Developer**

**Responsibilities**

* Involved in all phases of the portal changes – **analysis, design, development, testing, and deployment**.
* Used **Agile methodologies** and participated in **daily stand-up meetings, sprint planning, and retrospectives**.
* Used **JIRA** for defect tracking, bug resolution, and sprint management.
* Selected **AWS services** based on **compute, data, and security requirements** for application deployment.
* Integrated **Amazon Web Services (AWS)** with existing application infrastructures.
* Developed **interactive and responsive UI components** using **React.JS, Redux, and Flux architecture**.
* Created **Single Page Applications (SPAs)** using **Angular 6+ and React.js**.
* Used **Jasmine and Protractor** for unit testing Angular components.
* Worked on **REST API integration** with **React & Angular applications** for seamless data communication.
* Used **JSON and JSR353 API** for structured data exchange between frontend and backend services.
* Developed **UI enhancements and dynamic components** using **HTML5, CSS3, JavaScript (ES6+), and TypeScript**.
* Built **custom reusable UI components** using **Bootstrap and Material UI** for consistent user experience.
* Developed **Microservices in cloud environments** using **Spring Boot, Spring Cloud, and RESTful APIs**.
* Implemented **Spring Framework features** like **Spring IOC, Spring AOP, Spring Batch, and Spring Security**.
* Created **Spring configuration files** for **dependency injection and transaction management**.
* Designed and implemented **RESTful web services** for integration with frontend and third-party applications.
* Developed **web services using JAX-WS** to interact with **legacy backend applications**.
* Implemented **Asynchronous Messaging** with **Apache Kafka, RabbitMQ, and JMS**.
* Developed **Kafka Streams applications** for **real-time data processing, aggregation, and analytics**.
* Integrated **Kafka Streams** with **NoSQL databases like Apache Cassandra and Elasticsearch** for efficient data storage and retrieval.
* Used **Amazon DynamoDB (NoSQL)** for scalable **data storage and retrieval**.
* Updated **database scripts** to accommodate changes due to UI modifications.
* Developed **optimized SQL queries** for **Oracle, PostgreSQL, and MySQL** databases.
* Used **Hibernate ORM & JPA** for data persistence and transaction management.
* Implemented **batch processing** using **Spring Batch** for high-volume data processing.
* Used **Docker & Kubernetes** for **containerization and orchestration** of microservices.
* Built and deployed applications using **Maven, Gradle, and Jenkins** for **Continuous Integration (CI/CD)**.
* Used **JUnit & Mockito Frameworks** to write unit and integration test cases.
* Logged application events and errors using **Log4J and Splunk**.
* Automated testing workflows using **Selenium and Cypress**.
* Used **CloudFormation templates (CFT)** for **AWS infrastructure provisioning**.
* Monitored applications using **Amazon CloudWatch and CloudWatch Logs**.
* Integrated **AWS Lambda functions** with backend APIs for **event-driven workflows**.
* Deployed scalable applications on **AWS EC2, S3, RDS, and DynamoDB**.

**Environment**: Java 8 , React JS, Spring, Hibernate, Struts, EJB 3, WebSphere 8, JBoss EAP 5, Bootstrap, Java Script, Node Js, Oracle 11g, PostgreSQL9, UML, SVN, AJAX, JSR 353, JNDI, JAX-WS, Ant, AWS (Amazon Dynamo DB, Amazon SQS, Amazon Cloud Watch, Amazon Lambda), Agile Scrum.

**Client: Sun Life Financial - Kansas City, MO. Duration: July 2016 to Mar 2018**

**Role: Full Stack Java Developer**

**Responsibilities**

* Worked in big data ingestion using streaming tools like Flume, Sloop, Kafka, and Storm.
* **Big Data Ingestion & Streaming Tools**: Worked on big data ingestion using streaming tools like Flume, Sloop, Kafka, and Storm.
* **Development & Testing**: Worked on various use cases in development using Struts and tested functionalities.
* **System Design**: Involved in preparing High-Level and Detailed-Level system designs using J2EE.
* **Struts Framework Development**: Created Struts form beans, action classes, and JSPs following Struts framework standards.
* Developed model, library, Struts, and form classes (MVC).
* Developed Struts Action classes using Struts controller component.
* **Database Connectivity**: Implemented database connectivity using JDBC with Oracle 9i database as the backend.
* **UI & Reporting**: Used Display Tag libraries for decoration and Display Table for reports and grid designs.
* Used Jasper to generate rich content reports.
* **Testing & Refactoring**: Applied Test-Driven Development (TDD) methods to write failing tests, correct code, and refactor.
* **Core Java & File Operations**: Worked with core Java, using file operations to read system files and present them on JSP.
* **Supply Chain Domain**: Worked with the supply chain domain for the creation and delivery of product services, logistics, and customer service.
* **Underwriting Process**: Involved in the development of the underwriting process, including communication with external systems using IBM MQ and JMS.
* **CMS Development**: Involved in the design and estimation of templates and components using Day CMS (Communique).
* Developed CMS and server-side interaction using Web Services, JSON, and JQuery.
* **Java 8 Enhancements**: Used Java 8 features like lambda expressions and the Date and Time API for enhancements.
* **PL/SQL & Scheduling**: Used PL/SQL stored procedures for applications requiring scheduling mechanisms.
* **Error Handling & Troubleshooting**: Provided troubleshooting and error handling support across multiple projects.

**Environment**: Flex, EJB, spring, JSP1.2, Servlets2.1, Jasper reports, JMS, Scala, IBM MQ, XML, SOAP, Apache Tomcat, Clear Case, Log4J, ANT, Hadoop, Scala, Hive, Kafka, JUnit, Eclipse, UDDI, WSDL, JDBC, JNDI, Rational Rose 98, Oracle 10g, XSLT, XML, UML, HTML.

**Client: USDA - Kansas City, MO. Duration: Nov 2015 to June 2016**

**Role: Java Developer**

**Responsibilities**

* Designed and developed user interfaces that received the positive feedback from the warehouse operatives, leading to increased user satisfaction.
* Spring framework is used to implement Inversion of Control (IOC) and Model View Controller (MVC)
* Worked extensively on Web Services (SOAP & REST), XML, JMS and Spring Controller.
* Actively involved in designing and implementing Business Delegate, Data Access Object.
* Extensively used Java Collections framework for backend services and to display view on frontend.
* Developed reusable custom directives using Angular.JS using transude, restrict, event handlers and templating.
* Installation and configuration of Development Environment using Eclipse with JBoss Application server.
* Hands on experience in mapping tools like Hibernate (Hibernate Connection Pooling, HQL, Hibernate Caching, Transactions).
* Developed several modules using Spring MVC- Created Controllers, Services, used Dependency Injection feature for bean wiring.
* Developed complete independent microservices ton integrate with Identity management systems.
* Implemented microservices to fetch and display data from other apps.
* Worked on POC and develop prototypes for future enhancements of the current B2B application related to technology platform migration.
* Developed protype of current microservices to be ready for Eureka server
* Handling the activities such as user requirements analysis, design, construction, and implementation
* Good hands on setting up the environments and deploying ear’s and running database scripts on UAT environments through UNIX SSH Tectia and Putty tools.
* Developed Maven based project structure having data layer, ORM, and Web module.
* Used MongoDB to run the reports and query the database for low latency and fast accessing.
* Developed and consumed REST web services to retrieve data from different applications.
* Design of REST APIs that allow sophisticated, effective and low-cost application integrations.
* Used JUnit to test the application with Mockito and Power Mocks.

**Environment**: Spring Core, Spring MVC, Spring security, Java 8.0, Design Patterns, RESTful API, Web Services, JSON, Maven, Jenkins, WinSCP, Putty, Junit, Spring Web flow

**Client:** **Lowe’s,NC Duration: June 2014 to Sep 2015**

**Role: Java Developer**

**Responsibilities**

* Involved in Analysis, Design, and Implementation/translation of Business User requirements.
* Design and development of data enrichment business process objects using Rete based in-house rule engine framework
* Integrated the application with custom in-house GEMS and Netting frameworks
* Used Reference Data cache to load frequently used data into memory to avoid redundant database reads.
* Design and development of Template Remediation for regulatory reporting.
* Design and development of DAO's using Hibernate, JDBC, SQL and PL/SQL
* Design and development of operational data validation reports using JSP, Struts, HTML, JavaScript, AJAX
* Used design patterns like DAO, DTO, Singleton and Business Delegate, Chain of responsibility.
* Involving in client’s status calls to represent development progress and other key project discussions.
* Used Slot id-based table partitioning for partition strategy of high-volume data tables.
* Involved in writing SQL queries, stored procedures, and invoking them from Components.
* Design and Implemented development of ARTS balance sheet and MIS Drill Down Reports
* Involved in the unit testing the application using JUnit. Unit tested the changes and performed the code review.
* Performed joint system testing the application that involved other downstream systems.

**Environment**: Linux, Java 1.4, Oracle 10g, WebLogic 10.1, Eclipse, Apache POI, Log4j, Core Java, JDBC, JNDI, J2EE, Java Beans, Servlets, Multithreading, JUnit, DOM, SQL, PL/SQL, Hibernate 2.0, vi, SVN, Windows-XP, HP QC, Weblogic10.0.3, SVN, Git.