**Srikar Cherukuri**

[**srikarch19@gmail.com**](mailto:srikarch19@gmail.com)

**+1-4694645448**

[**https://www.linkedin.com/in/srikarcherukuri/**](https://www.linkedin.com/in/srikarcherukuri/)

**Sr. Java Full Stack Developer**

# **Carrer Summary**

* **8+ Years** of experience in developing and deploying large-scale, **Object-Oriented**, web-based **Client-Server applications** using **Java** and **J2EE**, following **SDLC** processes.
* Expertise in **Agile**, **Scrum**, **Waterfall** methodologies to manage development processes and deliver high-quality products.
* Proficient in **Core Java**, including **object-oriented programming**, **Java SE APIs**, **concurrency**, **exception handling**, **I/O operations**, and **JVM internals**.
* Extensive experience with the **Spring Framework** (Core, Web, REST Services, Data, Batch, AOP, Security).
* Skilled in **UI development** with **HTML**, **CSS**, **JavaScript**, and frameworks like **Angular** and **React.js** for responsive, user-friendly interfaces.
* Experience utilizing **Java 8** features (**Lambda**, **Stream API**) for improved performance in data operations.
* Strong background in designing and deploying **Microservices** with **Spring Boot**, following the **12-factor app** methodology.
* Solid understanding of **RESTful** and **GraphQL based web services**, **Service-Oriented Architecture (SOA)** and **Microservices architectures**.
* Proficient with **SQL** and **NoSQL databases** (**MySQL**, **PostgreSQL**, **MongoDB**, **Redis**), including designing and optimizing stored procedures.
* Expertise in **design patterns** (e.g., **Singleton**, **Factory**, **Circuit Breaker**) to improve code maintainability and scalability.
* Skilled in **messaging systems** like **Apache Kafka** for reliable service communication in distributed systems.
* Proficient in **AWS services** (**EC2**, **S3**, **RDS**, **Lambda**, etc.) to build scalable, secure applications.
* Expertise in **authentication mechanisms** (**OAuth 2.0**, **JWT**) to secure web applications and APIs.
* Experienced in **CI/CD** using **Jenkins**, **Maven**, and **GitHub** for automated build, test, and deployment pipelines.
* Skilled in **Docker** and **Kubernetes** for containerization and managing applications in diverse environments.
* Experienced in **testing frameworks** (**JUnit**, **Mockito**, **Selenium**, **Postman**) for unit, integration, and end-to-end tests, with tools like **SonarQube** for code quality.

# **Technical Skills**

|  |  |
| --- | --- |
| Programming and Scripting Languages | Java 8/11/17, Python, SQL |
| Front-End Technologies | Angular, React, HTML5, CSS, Bootstrap, JavaScript, jQuery, TypeScript |
| Back-End Technologies | JDK, J2EE, Spring Framework (Core, Web, JDBC, Rest Services, Data JPA, Batch, AOP, Transaction, Security), Hibernate. |
| Database Technologies | Oracle, DB2, MYSQL, MSSQL, MongoDB, Redis. |
| Cloud | EC2, S3, ELB, RDS, VPC, Route53, Lambda, Cloud Watch, Cloud Formation, IAM, SNS, SQS. |
| Messaging Systems | Kafka, JMS. |
| Security | OAuth 2.0, JWT (JSON Web Token) |
| Log Monitoring | Log4j, Sl4j |
| Build and Automation Tools | Maven, Gradle, Jenkins |
| Methodologies | Agile (Scrum), TDD (Test-Driven Development) |
| IDE | Eclipse, STS (Spring Tool Suite), IntelliJ, VS Code |
| Testing Automation | Junit, Mockito, Selenium. |

# **PROFESSIONAL EXPERIENCE:**

## **Sr. Java Full Stack Developer | Retail Business Services, USA.**

**Jan 2023 to Till Date**

As a Java Full Stack Developer, my primary focus at Retail Business Services is to refine our online retail platform to enhance user experience and drive customer satisfaction. Together with my team, we're diligently integrating new features tailored to the needs of online shoppers. These enhancements include personalized product recommendations, streamlined checkout processes, and an intuitive customer support interface.

* Engaged in the full software development lifecycle with **Agile** methodology, including requirements gathering, analysis, design, development, testing, and maintenance phases.
* Designed, developed, and deployed **microservices** architecture using technologies like **Spring Boot**, fostering a scalable and modular system.
* Implemented J2EE components using **Spring MVC**, **Spring IOC**, **Spring transactions**, **AOP**, **Batch**, and **Spring security** modules for robust functionality.
* Integrated **Oracle** and **MongoDB** databases as the backend repositories for Spring Boot applications, leveraging **Spring Data JPA** for Oracle and Spring Data MongoDB for MongoDB to interact with the respective databases.
* Developed and maintained data access layer using **Hibernate** and **JPA** annotations, mapping Java objects to relational database tables.
* Crafted the user interface using **React JS, HTML5, CSS3, JavaScript, AJAX, and JSON**, providing a responsive design accessible from any device.
* Utilized **Bootstrap** for responsive design and employed **ReactJS** to create views as a **Single Page Application**.
* Developed reusable and composable React components, utilizing JSX syntax, **props**, **state**, and **lifecycle** **methods** to create interactive and dynamic UIs.
* Implemented **Redux** for **state** **management** in React applications, utilizing centralized stores, actions, reducers, and middleware to manage and synchronize application state across components.
* Designed and maintained **Redux** stores, defining normalized data structures, state slices, and selectors for efficient state manipulation and retrieval.
* Deployed and managed the project on AWS, leveraging services such as **EC2, S3 Buckets, EBS, Lambda.**
* Conﬁgured **AWS Lambda** event sources and automated responses to various events, including HTTP requests via **API** **Gateway**.
* Designed and developed **serverless applications using AWS Lambda**, triggering functions via API Gateway, S3 events, and DynamoDB streams.
* Built and deployed **microservices on AWS** using Lambda, API Gateway, Step Functions, and SQS for scalable event-driven architectures.
* Integrated **AWS Lambda functions with RDS, DynamoDB, and S3**, ensuring seamless data flow and minimal latency in real-time systems.
* Secured Lambda-based services using **IAM roles and policies**, ensuring least-privileged access and compliance with enterprise standards.
* Developed Python scripts for automating routine DevOps tasks such as log parsing, server health checks, and deployment validations.
* Wrote Python scripts to manage Jenkins jobs, trigger pipelines, and monitor build outcomes via REST and CLI integrations.
* Utilized Python for log file analysis, pattern recognition, and proactive issue detection across Java application logs.
* Created **log monitoring and alerting** for Lambda functions using CloudWatch, enabling fast troubleshooting and observability.
* Integrated **OAuth 2.0** into authentication and authorization processes, ensuring robust security measures for user identity and data protection.
* Secured REST APIs using **OAuth 2.0 authorization flows** like client credentials, authorization code, and refresh token for diverse client needs.
* Enabled **role-based access control (RBAC)** and scope-based permissions by parsing claims from OIDC tokens in Spring Boot services.
* Configured **API proxies** to handle cross-origin resource sharing (**CORS**), content negotiation, protocol translation, and traffic routing for distributed API architectures and hybrid cloud environments.
* Implemented **Redis** **Cache** to store frequently accessed data, reducing database hits and significantly improving application response times and overall performance in high-throughput environments.
* Integrated **Kafka** producer and consumer applications on **Kafka** cluster setup with **Zookeeper**.
* Utilized **GIT** for version control, and **JIRA** for reporting Bugs, ensuring efﬁcient collaboration and issue tracking.
* Implemented **CI/CD** pipelines using Jenkins to automate build, test, and deployment processes for Java-based applications.
* Utilized **Maven** for project management and build automation, ensuring consistent and reproducible builds across different environments.
* Integrated **Docker** into CI/CD pipelines to containerize Java applications, and Kubernetes for container orchestration, ensuring consistency and portability across development, testing, and production environments.
* Configured **Jenkins** jobs to execute Maven builds, run unit tests, and package Spring Boot applications for deployment.
* Implemented extensive unit test suites using **JUnit** to validate the functionality of critical backend service components, ensuring code stability and reliability.
* Employed **Mockito** to mock external dependencies and simulate various scenarios during unit testing, enabling thorough testing of edge cases and error conditions.

**Tech Stack: Java 11/17, J2EE, Spring 4.x, Spring Boot 3.x Hibernate 4.x, Restful Web Services, Micro-Services, HTML 5, CSS 3, Bootstrap, React, JSON, React, Maven 3.0, AWS EC2, AWS IAM, S3, Docker, Jenkins, Git, Junit 4.4, Mockito, Log4j, Apache Kafka, MongoDB, Oracle 11g, Agile Scrum.**

## **Sr. Java Full Stack Developer | Capital One, Plano, Texas.**

**Sep 2021 to Dec 2022**

Working closely with a talented team, we embarked on a mission to develop an innovative application aimed at ensuring that every digital offering within Capital One's extensive array of services not only met but exceeded the highest standards of digital excellence. With a vast customer base exceeding 100 million individuals, the significance of this endeavor was paramount.

* Employed **Agile** (**SCRUM**) methodologies for the development and testing of the infotainment module.
* Implemented **Micro-Services** using **Spring Boot**, achieving loose coupling between layers, and moving towards **Restful** Services.
* Developed and implemented Business Requirements using the **Spring MVC** framework and **Spring Boot**, addressing cross cutting concerns such as logging, authentication, and system performance with **Spring AOP**.
* Developed **Hibernate** DAO classes with **Spring ORM** for persistence operations on the database and developed **HQL**, and **SQL** queries for data retrieval.
* Integrated **PostgreSQL** with Spring Boot applications as the backend database for efficient data storage and retrieval and configured Spring Data JPA with **PostgreSQL** dialect to interact with the **PostgreSQL** database using **Hibernate** **ORM**.
* Implemented **Spring** **Batch** for lightweight and comprehensive batch application solutions.
* Utilized **Angular** to create a **Single Page Application (SPA)**, binding data to specific views and synchronizing data with the server.
* Refactored and enhanced an existing **Angular** application, adhering to strict **MVC** patterns for improved source code maintenance and scalability.
* Utilized **AWS** for high availability practices and deploying backup/restore infrastructure. Developed **serverless** Java functions using **AWS** **Lambda**, leveraging the AWS Lambda Java runtime and Lambda handler interface to process events and execute business logic.
* Deployed and maintained **Java Spring Boot microservices on AWS EC2** instances using custom AMIs, auto-scaling groups, and ALBs.
* Designed infrastructure using **Terraform** to provision EC2, ECS, VPC, IAM, and security groups declaratively.
* Integrated **Amazon S3** for storage of application assets, logs, and backups, enabling seamless interaction via AWS SDK for Java.
* Developed Python scripts to automate configuration management and environment setup across dev, QA, and UAT stages of Java microservices.
* Employed **Apache Kafka** and developed Kafka **producers** and **consumers** in Java applications to publish and consume messages from Kafka **topics**.
* Utilized **Git** commands for common version control tasks such as branching, merging, committing, and rebasing.
* Configured **Jenkins** jobs to trigger **Maven** builds, run unit tests, and package Java applications into deployable artifacts.
* Developed and maintained **Terraform** scripts to provision and configure resources across AWS.
* Integrated version control system **Git** with **Jenkins** pipelines to automate code integration and continuous integration workflows.
* Utilized **Docker** to containerize Spring Boot applications and Docker Compose for local multi-container setups. Deployed and managed these applications in **Kubernetes** clusters.
* Worked on **OAuth client registration, token validation, and introspection endpoints** for secure integration with external applications.
* Validated and decoded **ID tokens and access tokens** using JWK (JSON Web Key) validation libraries and custom authentication filters.
* Integrated OAuth 2.0 authentication in Spring Boot applications using **Spring Security and JWT**, enabling secure token-based access control.
* Implemented **Log4j** for logging purposes with defined debug levels and Integrated **Splunk** for aggregating logs from different systems and implemented custom monitoring systems through **Splunk** into deployment processes.
* Utilized **JUnit** annotations such as @Test, @Before, @After, and @RunWith to define and execute test scenarios.
* **Mocked** dependencies and external dependencies using **Mockito** to isolate units for testing.

**Tech Stack: Java, J2EE, JDK 11, Spring 4.x, Spring Boot 3.x, Xml, Angular, Bootstrap, TypeScript, CSS 3, Hibernate 3.x, Git, Aws, Jenkins, Web Services, SoapUI, JSON, Log 4j, PostgreSQL DB, Agile.**

## **Sr. Java Developer | Mercedes Benz, India**

**Jan 2020 to Jul 2021**

Played a key role in developing a scalable platform for Mercedes-Benz, collaborating throughout the SDLC. Leveraged **Java** and **Spring Boot** to optimize performance, supporting advanced automotive features and seamless integration. This contributed to a **25% increase in online sales** and enhanced **customer satisfaction**, driving innovation in the automotive industry.

* Played a key role as a Senior Java Developer, contributing to various SDLC phases, including requirement gathering, design, analysis, and code development.
* Designed, developed, and deployed **microservices** architecture using **Spring Boot** and utilized Spring Boot's **dependency** **injection**, **auto**-**configuration**, and embedded server features to simplify microservice development.
* Implemented **RESTful** **APIs** and microservices endpoints using **Spring** **MVC** for communication between microservices.
* Leveraged **Spring** **Cloud** components such as **Eureka** for service discovery, **Ribbon** for client-side load balancing, and **feign** for declarative REST client.
* Implemented distributed tracing and monitoring using Spring Cloud **Sleuth** and **Zipkin** for end-to-end visibility in microservices architecture.
* Implemented fault tolerance and resilience patterns such as circuit breakers with **Spring Cloud Circuit Breaker** (Hystrix) to handle failures gracefully.
* Leveraged **React** **JS** to create loop directives and components, facilitating dynamic rendering of views and object model updates.
* Utilized **Bootstrap** and **React JS** to develop Controllers handling client-triggered events and sending requests to the server.
* Conﬁgured **Hibernate** for data persistence to the **Microsoft SQL** Database, utilizing **Hibernate** **Session** **Factory** for database connections. Implemented persistence layer features using Hibernate framework, such as HQL queries and Criteria.
* Utilized **Docker** for fast deployment of products to the development environment and for application ﬁxing.
* Implemented **Kafka** for **Publish/Subscribe** pattern and employed multithreading to process data in chunks for enhanced performance.
* Followed coding best practices, including removing unnecessary casting, using generics in **Hash Map**, and employing Collections and Design patterns.
* Managed version control with **GIT** and used **JIRA** for issue tracking.
* Utilized **Maven** as a **build tool** to retrieve all dependencies required for the application.
* Utilized **Selenium** **WebDriver** for automated testing of web applications, ensuring functionality, compatibility, and regression testing.

**Tech Stack: Java, J2EE, JDK 11, HTML, CSS, Bootstrap, JavaScript, React JS, Spring Dependency Injection, Spring 3. x, Hibernate 3. x, Azure, Kafka, Docker, Maven, JIRA, Microsoft SQL, XML, Unix/Linux, PL/SQL, Oracle 10g.**

## **Java Developer | Unilever Home and Personal Care, India**

**Jan 2017 to Dec 2019**

* Actively participated in the full software development lifecycle, including Design, Development, Testing, and Implementation.
* Designed and implemented **custom** **made** **data** **structures** based on the existing data structures for customized services.
* **Overhauled** the back-end codebase, achieving a **30%** **reduction** in code complexity by introducing **reusable** **components** and applying industry-standard **design** **patterns**.
* Designed and developed **RESTful** APIs, managed **databases**, and implemented **event**-**driven** code to efficiently handle data ingestion and updates.
* Conducted research on **JMS** solutions to evaluate high availability **messaging** backends for the system, leading to the successful integration of **JMS** for **reliable** **communication**.
* Gained a deep understanding of messaging **APIs**, including **publish**/**subscribe** models and **queue** management.
* Utilized **Hibernate** **ORM** to map the application’s data from the **MVC** model to the **Oracle** **relational** **database** schema, improving data persistence.
* Contributed to coding, **debugging**, and **unit** **testing** across multiple system modules to ensure quality and performance.
* Designed and implemented a **regression** **test** suite, **automating** testing using JUnit test cases to ensure consistency and reliability.
* Developed robust backend services with **Spring** **Boot**, leveraging its capabilities for **rapid** **application** **development**, and integrated Spring Data JPA, Spring Security, and Spring Cloud for comprehensive functionality.
* Facilitated seamless data communication between the front end and server using **JSON** to transfer requests and responses.
* Enhanced **data** **persistence** by implementing Hibernate **ORM** and **JPA** for efficient object-relational mapping.
* Utilized Apache **Maven** for build configuration, packaging, and deployment of application projects.
* Designed of a **single**-**page**, Ajax-driven HTML application, focusing on **optimizing** user experience and implementing services for key modules such as Account Creation and Maintenance.

**Tech Stack:** **AJAX, HTML, JavaScript, CSS, Spring, Hibernate, JBoss, JUnit, DB2, DB Visualizer, Eclipse IDE, Web Services, XML Schema.**

# **Certifications**

[**AWS Certified Cloud Practitioner**](https://www.credly.com/badges/8d36f22e-7fbe-4b76-b877-cc03b24dc13b/linked_in_profile)

# **Skills**

Performance Optimization Team Leadership Agile Methodologies Problem Solving API Design Cloud Integration

Continuous Integration Java Frameworks RESTful Web Services Security Implementation

Java Proﬁciency System Architecture Full-stack Development Microservices Expertise Database Design

# **Education**

**Master of Science in Computer Science – UMKC**

**Bachelor of technology in Computer Science – SNIST**