**Amit Maharjan**
+1-718-313-3053 | maharjan.amit007@gmail.com | LinkedIn | Personal Website

## PROFESSIONAL SUMMARY

Hands-on full-stack developer with over 6 years of enterprise experience in **Java**, **Angular**, **Spring Boot**, and cloud-native services on **AWS**. Specialized in microservices architecture, API design, event-driven systems, and building scalable, testable, and resilient platforms. Skilled in frontend development using **TypeScript**, Material UI, and **RxJS** for dynamic, responsive applications. Proficient in CI/CD processes, containerized deployments with **Docker**, and testing strategies using **JUnit**, Mockito, and performance tools like **JMeter**. Worked across SQL and NoSQL ecosystems with **PostgreSQL**, MongoDB, and **DynamoDB**. Strong communicator in **Agile** teams with cross-functional participation in sprint activities, architecture reviews, and production deployments. Deeply involved in system monitoring using **Splunk**, **OpenTelemetry**, and Spring Actuator metrics.

## TECHNICAL TOOLS

**Backend:** Java 8/11/17/21, Spring Boot, REST APIs, JPA, Hibernate, JDBC, Python, Microservices
**Frontend:** Angular 10/14/16, TypeScript, RxJS, HTML5, CSS3, SCSS, Material Design
**Cloud & DevOps:** AWS (ECS, EC2, S3, Lambda, RDS, CloudWatch, Parameter Store), Docker, Jenkins, GitHub, Cloudformation
**Databases:** MySQL, PostgreSQL, MongoDB, DynamoDB, Redis
**Messaging:** Kafka, Topic Retry, Schema Registry, Message Offsets, Event-Driven
**Security:** OAuth2, JWT, RBAC, Spring Security Filters, API Gateway
**Testing:** JUnit, Mockito, TestNG, Cucumber, JMeter, K6
**Deployment:** Blue-Green, Canary, Docker, CI/CD Pipelines, ECS, Kubernetes
**Monitoring:** Splunk, OpenTelemetry, CloudWatch, New Relic
**Tools & Process:** GitHub, Bitbucket, JIRA, Confluence, Agile, IntelliJ, Maven, NPM

## PROFESSIONAL EXPERIENCE

***Carta*** *Dec 2023 - Present*
*Software Engineer* *San Francisco, California*

***Cap Table Explorer***

*Created APIs using **Java** and **Spring Boot** to support a dynamic **cap table viewer**, integrating data from **PostgreSQL** and exposing ownership breakdowns by entity and share class. Built **Angular** components that allow users to drill into shareholder positions, transaction history, and dilution scenarios. This helped Carta's legal and **finance clients** visualize ownership structure during funding rounds.*

**Responsibilities:**

- Built **Java 21** microservices using **Spring Boot** and **Spring Profiles** to manage environment-specific configurations and streamline config refresh without downtime.
- Developed **Angular 16** modules using lazy loading and interceptors to inject tokens, handle auth failures, and reduce initial bundle size for dashboard UI.
- Implemented **Kafka** consumer with schema validation for order events using **Avro** and configured alerting dashboards to track consumer lag and partition offsets.
- Designed **RESTful APIs** in Java using **@RestController** and custom response wrappers to centralize response metadata across **microservices** for UI consumption.
- Implemented **circuit breaker** and fallback logic using **Resilience4j** to keep critical flows functional during failures from downstream services.

- Used **TypeScript** to create shared **Angular services** for error handling and reusable form logic with async validation and dynamic control rendering.
- Wrote unit tests for **Java** applications with **JUnit** and **Mockito** using custom stubs and verified internal service behavior without relying on external dependencies.
- Utilized **Java Streams** and **Lambda expressions** for collection transformations, parallel processing, and functional-style programming in business logic.
- Managed asynchronous flows using **Kafka** producers for background processing queues in audit generation and notification delivery systems.
- Created and maintained AWS **CloudFormation** templates to provision **ECS** services, Application Load Balancers (ALBs), **IAM roles** and RDS instance components.
- Handled **PostgreSQL** and **MongoDB** integrations for different modules using JPA for relational data and MongoRepository for flexible document queries.
- Used **Spring Security** with **JWT** and **RBAC** filters to validate user roles, integrate OAuth2, and restrict access by route and scope.
- Created dashboards with **Angular** Material and connected them via **REST** APIs, including custom pagination logic and filter pipelines in frontend.
- Developed **AWS Lambda** functions in **Python** triggered by **S3 events**, **SNS topics**, and **DynamoDB streams** to handle asynchronous processing tasks.
- Used **AWS ECS** with Docker containers to deploy services and fetched secrets using **AWS Parameter Store** for config injection.
- Scheduled file cleanup using **AWS Lambda** and S3 lifecycle policies to archive old records and keep the cost of storage minimal.
- Built **CI/CD pipelines** in **Jenkins** to validate builds and run JUnit and Mockito tests before Docker image creation for **ECS** deployments.
- Monitored services using **New Relic** and **Spring Boot Actuator** for thread pool usage, memory consumption, and API call performance.
- Delivered real-time **collaboration** across developers and testers by documenting flows in Confluence and discussing test integration strategies.
- Participated in **Agile** ceremonies to estimate tickets and regularly led backend-frontend handoffs for sprint deliverables.

**Technological Environment:**

Java 17/21, Angular 16, Spring Boot, Spring Profiles, Spring Actuator, RESTful APIs, TypeScript, Kafka, Schema Registry, MongoDB, PostgreSQL, RDS, JWT, OAuth2, RBAC, Spring Security, AWS ECS, EC2, Lambda, S3, Python, Parameter Store, Cloudformation, Docker, JUnit, Mockito, Redis, CI/CD, Jenkins, New Relic, Agile, Splunk, Confluence

---

***Cedar Gate Technologies*** *June 2021 - July 2023*
*Software Engineer* *Greenwich, CT*

***Member Account Summary Portal***

*Built a backend service using Java and Spring Boot to fetch and organize healthcare plan details by provider, tier, and cost structure for side-by-side comparison. Developed a dynamic Angular interface with filters, expandable rows, and paginated results to help business analysts and support teams quickly assess plan differences. The tool addressed a growing need for transparent benefit comparison across custom payer networks. Integrated with internal APIs using version control and feature toggles to allow phased updates without UI downtime.*

**Responsibilities:**

- Refactored **Java 11** legacy monolith to modular **microservices** using **Spring Boot** with DTO layers and centralized config server.

- Developed **Angular 14 UI** components using **RxJS** observables and debounce techniques to manage auto-suggestions and form submissions in real-time dashboards.
- Built **RESTful APIs** with **Spring Boot** and used request validation via **@Valid** and exception translation for consistent frontend error handling.
- Migrated legacy UI into **Angular 14** using Material Design, integrating with versioned **Java APIs** and lazy-loaded feature modules to optimize load times and routing.
- Implemented **Kafka** consumer retries with **dead-letter queues** and topic-level error counters for failed records to enhance data traceability.
- Configured **MySQL** for transactional consistency in **Java microservices** and used MongoDB for flexible logging and real-time insights via Spring Data integrations.
- Set up **Spring Cloud Gateway** for header-based routing, auth **token** validation, and forwarding rules across multiple downstream microservices.
- Created **Docker** images with custom healthcheck scripts and used **Jenkins** pipelines to deploy into staging and **prod** via blue-green strategy.
- Used **AWS** services, including **Lambda** and CloudWatch for automated job execution and log alerting on high-memory usage scenarios.
- Containerized microservices using **Docker** and deployed them to **Kubernetes** clusters with **Helm charts** to manage environment-specific configurations and secrets.
- Connected APIs to **DynamoDB** for session tracking with TTL values so that stale entries could be purged without explicit cleanup logic.
- Wrote unit tests using **JUnit** and used **Mockito** to mock inter-service dependencies and verify conditional flows inside service methods.
- Logged application traces in **Splunk** using **Java-based** structured **logging**, and configured alert pipelines for **Kafka** retry failures and service-specific error codes.
- Used **Python** with **Pandas and JSON libraries** to reformat legacy data into REST-friendly structures that were ingested by Spring Boot services for initial system migration.
- Enabled file upload via **AWS S3** using signed URLs with session expiration to prevent unauthorized overwrites from external clients.
- Participated in **Agile** grooming and sprint reviews, discussed **API readiness** with frontend teams, and clarified acceptance criteria with QA.
- Set up **GitHub** Actions for build validation and **PR** enforcement so that only successfully tested code could merge to the main branch.
- Used **TypeScript** in **Angular** for strict typing across services and forms and managed shared models in a central workspace library.
- Designed versioned **microservices** using backward-compatible payloads and feature flags to allow phased rollouts without UI disruption.

**Technological Environment:**

Java 11, Angular 14, Spring Boot, RESTful APIs, Python, Spring Cloud Gateway, Kafka, Mysql, MongoDB, DynamoDB, RxJS, TypeScript, Docker, Jenkins, Kubernetes, Lambda, S3, CloudWatch, CI/CD, JUnit, Mockito, GitHub, Agile, Splunk, Material UI, Microservices

---

***Leapfrog Technologies*** *Dec 2018 - May 2021*
*Java Developer* *Seattle, WA*

***Employee Benefits Enrollment Tool***

*Developed backend **Spring services** to manage employee benefit selections and retrieve eligibility rules tied to department codes and employment types. Built the **Angular UI** with dynamic forms and **custom validators** to support conditional questions based on selected plan types. This helped HR teams **track** benefit enrollment during open periods and provided summaries for payroll*

*coordination. Replaced older SOAP-based flows with modern REST endpoints to improve consistency across new reporting modules.*

**Responsibilities:**

- Built **Java 8** services using **Spring MVC** and DAO-based persistence layer to connect to **MySQL** for employee and department record management.
- Used **Angular 10** and **Reactive Forms** for dynamic form field rendering and created validation messages using a shared **TypeScript** service.
- Parsed and validated incoming **XML** files using **JAXB** and transformed them into domain models through custom converters.
- Consumed external **SOAP** services using **WSDL** definitions and performed schema validation before request processing using XSD.
- Developed JSP views and backend **controllers** in **Spring MVC pattern** to render tables, charts, and CRUD screens using **JSTL**.
- Used **jQuery** to make **AJAX** calls for auto-refresh widgets and in-page content updates within dashboard modules.
- Stored temporary objects and expired records in **MongoDB** and wrote cleanup shell scripts scheduled via **Linux cron jobs**.
- Deployed WAR files to **Apache Tomcat** with manual configuration edits and restart sequences controlled through systemd scripts.
- Logged errors using **Log4J** into custom file paths and built grok patterns to ingest them into older **Splunk** dashboards.
- Added **RESTful** endpoints in **Spring Boot** to serve newer modules and replaced SOAP integrations with modern **JSON**-based alternatives.
- Wrote JUnit and **Mockito** tests for **DAO** and service classes and built a regression suite for critical reporting flows.
- Handled legacy **MVC** setup using **JSP, Servlets,** and form-based authentication using session cookies and Servlet filters.
- Managed config-driven routing and **role-based** redirection using **JavaScript** logic in JSP templates for different user roles.
- Used **Agile** methodology with weekly sprints and performed sprint demo of **CRUD** workflows and reporting module updates.
- Built HTML templates with **Bootstrap** and **CSS** for legacy modules' responsive layouts and custom UI sections.
- Used **XML** transformations in payroll flows to convert internal records into government-compliant **XML** formats for **SOAP** submission.

**Technological Environment:**

Java 8, Angular 10, Spring, Spring Boot, RESTful APIs, MVC, JSP, JSTL, SOAP, XML, JAXB, WSDL, XSD, TypeScript, Reactive Forms, JavaScript, jQuery, MongoDB, MySQL, WAR, Tomcat, Log4J, JUnit, Mockito, AJAX, Agile, HTML, CSS, Linux

## EDUCATION

**East Tennessee State University** — *Johnson City, TN*
*Master of Science in Computer Science (Software Engineering)* — *GPA: 4.0*

***Tribhuvan University*** — *Kathmandu, Nepal*
*Bachelor of Engineering in Computer Engineering* — *Grade: Distinction*