**PROFESSIONAL SUMMARY:**

* Full Stack Java Developer with **7+ years of experience** designing, developing, and deploying **enterprise-grade applications** across healthcare, life sciences, education, and supply chain domains.
* Strong expertise in **Java (17/11/8), Spring Boot, Microservices, and Node.js** for developing **scalable and modular back-end systems**.
* Proficient in **React.js, Next.js, Angular (6–16), TypeScript, Redux, NgRx, RxJS**, delivering **Single Page Applications (SPAs)** and **Server-Side Rendered (SSR)** applications.
* Skilled in building **responsive UIs** using **HTML5, CSS3, Bootstrap, Material-UI, and Tailwind CSS**, ensuring cross-browser compatibility and accessibility.
* Experienced in **API design and integration** with **REST, GraphQL, SOAP, and WSDL**, ensuring secure and high-performing service communication.
* Hands-on expertise with **SQL and NoSQL databases** including **Oracle, PostgreSQL, MySQL, MongoDB, DynamoDB, Cassandra, Redis**, with a focus on **query optimization, indexing, and caching strategies**.
* Skilled in **performance optimization** through **Redis Cache, Memcached**, and **load-balancing techniques** for high-speed data retrieval and scalable applications.
* Proficient in **event-driven and real-time processing systems** using **Apache Kafka, RabbitMQ, IBM MQ, Google Pub/Sub, and WebSockets**.
* Experience with **cloud-native development and deployment** across **AWS, Azure, and GCP**, using services such as **EC2, S3, RDS, Lambda, ECS, App Services, Functions, Blob Storage, Key Vault, DevOps, GCS, Big Query, Pub/Sub, Cloud Functions, Firestore.**
* Hands-on with **containerization and orchestration** using **Docker, Kubernetes, OpenShift, and Helm**, ensuring **high availability and fault tolerance**.
* Implemented **infrastructure as code** using **Terraform, CloudFormation**, and automated **CI/CD pipelines** with **Jenkins, GitHub Actions, GitLab CI/CD, and Argo CD (GitOps)**.
* Strong background in **security and compliance**, implementing **OAuth2, JWT, Spring Security, Okta, LDAP**, ensuring **HIPAA and PCI compliance** for healthcare and payment systems.
* Built and optimized **end-to-end payment processing solutions** integrating **Stripe, PayPal, and banking APIs**, with secure transaction handling.
* Skilled in **Databricks and Spark (Scala/Python)** for **large-scale data processing and analytics**, enabling data-driven insights.
* Proficient in **monitoring and observability** with **ELK Stack, Prometheus, Grafana, Splunk, and Spring Boot Actuator** for proactive system health checks.
* Experienced in **Test-Driven Development (TDD) and BDD**, using **JUnit, Mockito, Selenium, Cypress, Jest, Jasmine, Karma, Mocha, Chai, JMeter, and Postman** for unit, integration, and performance testing.
* Strong problem-solving and debugging skills with ability to **troubleshoot production issues**, optimize performance, and ensure **minimal downtime**.
* Proven track record of **leading Agile teams, conducting code reviews, mentoring junior developers**, and collaborating with **cross-functional stakeholders** to align technology with business goals.
* Enthusiastic about **emerging technologies**, continuous learning, and driving **innovation in cloud, AI/ML integration, and modern full-stack frameworks**.

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Languages & Frameworks** | **Java (17/11/8), Spring Boot, Spring Cloud, Microservices, Hibernate, JPA, Kafka, RabbitMQ, JavaScript (ES6+), TypeScript, Node.js, React.js, Next.js, Angular (6–16), Redux, GraphQL, Python, Scala, Kotlin, Spark** |
| **Web Technologies** | **HTML5, CSS3, Bootstrap, Tailwind CSS, XML, Material-UI, JSON, AJAX, JSP, Servlets, WCAG Accessibility** |
| **Application/Web Servers** | **Apache Tomcat, IBM WebSphere (v8+), JBoss, Jetty, Jenkins, Elastic Compute Cloud (EC2), Bitbucket, Hudson, Antilipo, Bamboo, WebSphere MQ** |
| **J2EE Technologies** | **Core Java, JMS, JSF, RabbitMQ, MQ Series, Java Beans, Java Multithreading, Generics and Collections, EJB, Tiles JSP, Servlet, Spring 2.x/3.x, Struts/Struts 2, Hibernate 3.x, Expression Language, JPA, JDBC, Java Mail, AMQP, Apache Camel, Oauth1/2, Swagger (OpenAPI v3)** |
| **Databases** | **Oracle, PostgreSQL, MySQL, IBM DB2, MongoDB, Cassandra, DynamoDB, Redis Cache** |
| **Cloud & DevOps** | |  | | --- | | **AWS(EC2, ECS, RDS, S3, DynamoDB, Lambda, API Gateway, CloudWatch, Code Pipeline), Azure( App Services, Functions, Blob Storage, Key Vault, DevOps), GCP(GCS, BigQuery, Pub/Sub, Cloud Functions, Firestore), Docker, Kubernetes, OpenShift, Terraform, Helm, CloudFormation, Jenkins, GitHub Actions (GitOps), GitLab CI/CD** | |
| **Messaging & Streaming** | **Apache Kafka, RabbitMQ, JMS, ActiveMQ** |
| **Operating Systems** | **MacOS, Windows, Linux Mint, Unix, Ubuntu** |
| **Testing & QA** | **JUnit, Mockito, Selenium, Cypress, Postman, Jest, Jasmine, Karma, Mocha, Chai, JMeter** |
| **Monitoring & Logging** | **ELK Stack, Prometheus, Grafana, Azure Monitor, GCP Stackdriver, Spring Boot Actuator, Jasper Reports, Crystal Reports XI, SSRS** |
| **Methodologies** | **Agile (Scrum, Kanban), TDD, BDD, CI/CD, Enterprise Integration Patterns** |
| **Build & IDE Tools** | **Maven, Ant, Gradle, Eclipse, IntelliJ, VS Code, Spring Tool Suite (STS), Git** |
| **Web Services** | **RESTful APIs, GraphQL, SOAP, Swagger (OpenAPI v3), WSDL** |

**PROFESSIONAL EXPERIENCE**

**Client: Thermo Fisher Scientific, MA**  **Sep 2023 – Present**

**Role: Full Stack Java Developer**

**Project Overview:** Led the development of a **cloud-native, PCI/HIPAA-compliant payment system** for Thermo Fisher’s lab platform, improving reconciliation speed by 30% and enabling real-time transaction monitoring. Built a **Java 17/Spring Boot microservices architecture** on **AWS ECS** with Terraform/CloudFormation, integrated Stripe and PayPal, and implemented event-driven processing with Kafka and RabbitMQ. Leveraged AWS services to achieve **99.9% uptime** and reduce costs through auto-scaling and resource optimization.

**Responsibilities:**

* Designed and built microservices using **Java 17, Spring Boot, and RESTful APIs,** ensuring modularity, scalability, and maintainability of payment processing applications.
* Applied sealed classes to enforce domain-driven constraints in transaction workflows (e.g., Payment - CreditCardPayment | PayPalPayment | InternalTransfer), making business logic more secure and predictable.
* Leveraged switch expressions and pattern matching for instance, to simplify conditional branching in payment processing pipelines, improving code readability and maintainability.
* Defined and implemented **REST APIs with Swagger (OpenAPI v3)**, improving usability and integration across cross-functional teams.
* Architected real-time payment workflows using **Apache Kafka, RabbitMQ**, and **IBM WebSphere MQ** for asynchronous communication, ensuring resilience and high throughput.
* Seamlessly integrated Stripe, PayPal, and internal systems to support multi-channel digital payments while adhering to **PCI/HIPAA** compliance.
* Optimized **Spring Batch** with **Kafka** for reconciliation processes, improving processing throughput by 30%.
* Designed and consumed **GraphQL** schemas with **Apollo Client,** reducing over-fetching and improving front-end query efficiency.
* Developed **Node.js** **microservices** for lightweight **APIs** and rapid prototyping, complementing the core Java backend.
* Built **React.js + TypeScript UI** with reusable components, integrated with backend services, and enhanced dashboards using **Next.js** for server-side rendering.
* Delivered responsive layouts using **Bootstrap, Tailwind**, and media queries, ensuring accessibility (ARIA roles) across devices and browsers.
* Developed **React.js** dashboards for live payment monitoring, refunds, and transaction histories, providing real-time visibility to business stakeholders.
* Modeled relational schemas in **MySQL/Oracle** with indexing, partitioning, and performance tuning.
* Leveraged **MongoDB** for flexible storage of unstructured lab and experimental data.
* Used **DynamoDB** for idempotency, high-velocity lookups, and TTL-based lifecycle rules.
* Implemented **Redis** and **Memcached** caching strategies for high-speed data retrieval, reducing **DB** load and improving **API** response times.
* Containerized microservices on **AWS ECS** (**EC2** launch type) with **ALB**-based routing and auto-scaling.
* Modeled payment ledgers in **Amazon RDS (MySQL)** with Multi-AZ + replicas for HA.
* Used **AWS S3** with SSE-KMS and lifecycle policies for secure artifact storage and archival.
* Automated asynchronous workflows with **AWS Lambda, EventBridge** schedules, and S3 triggers.
* Codified infra using **Terraform & CloudFormation,** automated multi-env deployments via Jenkins & **GitHub Actions**, and streamlined blue/green deployments with **Argo CD (GitOps).**
* Used **Docker** for containerization and **Kubernetes** for orchestrating services, ensuring reliable scaling and self-healing.
* Enforced **Spring Security, OAuth2**, and **JWT** for authentication/authorization.
* Applied least-privilege **IAM roles, VPC** segmentation, and **KMS** encryption across data stores.
* Ensured **PCI/HIPAA** compliance for healthcare payment data handling.
* Implemented **ELK Stack (Elasticsearch, Logstash, Kibana)** for centralized logging and **Spring Boot Actuator + AWS CloudWatch** for health checks and telemetry.
* Automated **UI testing** with **Selenium** for cross-browser compatibility.
* Wrote unit and integration tests using **JUnit, Mockito, Jasmine,** and **Karma,** achieving high coverage and reducing defects.
* Conducted performance/load testing with **JMeter**, identifying bottlenecks under peak loads.
* Built pipelines integrating **Databricks** for big data processing and analytics, enabling real-time insights into transaction trends.
* Reduced deployment time by 40% with **CI/CD** automation in **Jenkins & GitHub Actions**, integrating linting, build, test, and deployment stages.
* Worked in **Agile Scrum**, contributing to sprint planning, stand-ups, retros, and cross-team collaboration, ensuring timely delivery of features.
* Collaborated with data scientists to integrate **ML**-driven **fraud detection** models into payment pipelines, ensuring real-time anomaly detection.
* Actively collaborated with product owners, QA teams, and DevOps engineers; mentored junior developers on **Spring** Boot, **React.js**, and **AWS** best practices; and consistently aligned technical solutions with business goals.

**Environment:** Java 17/11, Spring Boot, Spring Cloud, Spring Security, Spring Batch, Node.js, React.js, Next.js, TypeScript, HTML5, CSS3 AJAX. Bootstrap, Tailwind, GraphQL, Visual Basic (Scripting), Python, RESTful API, Hibernate ORM, Spring JPA, Redis, Databricks, Apollo Client, AWS (ECS, EC2, RDS, S3, DynamoDB, Lambda, API Gateway, CloudWatch, IAM, VPC, KMS), Terraform, CloudFormation, Docker, Kubernetes, Jenkins, GitHub Actions, Argo CD, GitOps, Apache Kafka, RabbitMQ, IBM WebSphere MQ, AWS EventBridge, MySQL, Oracle, MongoDB, ELK Stack (Elasticsearch, Logstash, Kibana), Spring Boot Actuator, Spring Security, OAuth2, JWT, Maven, Jenkins, GitHub Actions, JUnit, Mockito, Selenium, Jasmine, Karma, JMeter, Spring Cloud Config Server, Eclipse, IntelliJ, NetBeans, Spring Cloud Gateway, Swagger/OpenAPI v3, IBM WebSphere MQ, IBM DB2, Media Queries, ARIA roles for accessibility.

**Client: McKesson, Texas**  **n 2022- Aug 2023**

**Role: Full Stack Java Developer**

**Project Overview:** At McKesson, I played a key role in modernizing the company’s healthcare distribution and supply chain platform by migrating legacy systems to a **scalable microservices architecture** on **AWS**. Delivered RESTful APIs, GraphQL endpoints, and real-time tracking features while implementing **serverless workflows**, optimizing **DynamoDB/RDS**, and automating provisioning with **Terraform**. Containerized services on Kubernetes, enhanced observability with CloudWatch, Grafana, and Prometheus, and established **GitOps-driven CI/CD pipelines**, improving efficiency, reliability, and compliance with **HIPAA and FDA standards**.

**Responsibilities:**

* Designed and implemented **RESTful APIs** using **Spring Boot** and **Java 8,** ensuring seamless integration between microservices and client applications.
* Leveraged **Java Streams, Lambda expressions**, and **Optional (Java 8)** to simplify data processing and reduce boilerplate code.
* Adopted Records **(Java 17)** for immutable data carriers, reducing verbosity in DTO classes.  
  Utilized **Pattern Matching** instance of **(Java 16/17)** to simplify conditional logic and improve readability in backend services.
* Applied **Sealed Classes (Java 17)** in the domain model to restrict inheritance and enforce stricter type hierarchies for healthcare data models.
* Built scalable **Spring Boot microservices** with modular boundaries, improving fault isolation and system resilience.
* Developed **Spring Batch** jobs for large-scale healthcare data processing, ensuring scalability and fault tolerance.
* Implemented **API** versioning to ensure backward compatibility while evolving services.
* Modernized front-end by migrating legacy **SPA** modules to **Next.js,** improving **SSR (Server-Side Rendering) and SEO.**
* Built **Angular 12/13** applications with **TypeScript** for real-time pharmaceutical tracking, ensuring strong type safety and maintainable code.
* Created reusable **React.js** components with **Hooks and Redux** for **state management**, improving **UI** responsiveness and reducing load times.
* Integrated **React Hook** Form with **GraphQL** schemas for advanced form workflows and real-time validation.
* Designed **GraphQL** schemas and resolvers, enabling clients to fetch tailored datasets and reducing network payloads.
* Documented **APIs** using **Swagger/OpenAPI v3,** improving developer onboarding and **API** discoverability.
* Optimized IBM DB2, Oracle, MySQL, and PostgreSQL queries with indexing and partitioning strategies to improve response times.
* Implemented **Redis Cache** for low-latency retrieval in frequently accessed healthcare data.
* Built **DynamoDB Streams + GSIs** for real-time pharmaceutical tracking, ensuring millisecond response times under peak loads.
* Integrated **MongoDB** for unstructured healthcare data like patient documents and imaging records.
* Designed **ETL** pipelines in **Databricks** for advanced analytics, enabling healthcare insights from multi-source datasets.
* Used **Apache Kafka** to decouple **microservices** with event-driven architectures, enabling asynchronous communication.
* Built **Kafka** c**onsumers and producers** for reliable real-time data synchronization across supply chain systems.
* Migrated services to **AWS Elastic Beanstalk** with rolling/immutable deployments and integrated **ALB** for traffic routing.
* Used **Terraform** modules to provision **AWS** resources **(S3, DynamoDB, IAM, alarms),** ensuring repeatable infrastructure deployments.
* Deployed containerized services to **Kubernetes clusters,** leveraging auto-scaling and self-healing for high availability.
* Adopted **Argo CD** with **GitHub Actions** for **GitOps-**driven **CI/CD** pipelines, ensuring traceable and version-controlled deployments.
* Implemented **Lambda + S3** events for automated document processing workflows in supply chain pipelines.
* Applied **Spring Security** with **Okta SDK** for **SSO**, role-based access control, and secure **OAuth2/JWT authentication.**
* Enforced least-privilege **IAM** policies and deployed services within private **VPC** subnets for data protection.
* Centralized configurations in **AWS Parameter Store** (encrypted) to maintain compliance and consistency across environments.
* Configured **AWS** **CloudWatch** metrics and alarms to detect anomalies and alert on **KPIs**.  
  Built custom **Grafana** dashboards and integrated with **Prometheus** metrics, improving incident detection and MTTR.
* Implemented structured logging across services for traceability and auditing under **HIPAA** compliance.
* Wrote unit tests using **JUnit + Mockito** for backend services, improving defect isolation.  
  Tested Angular/React components using J**asmine, Karma, Mocha, and Chai** to validate UI reliability.
* Built end-to-end test suites with **Selenium** for cross-browser and device compatibility.
* Conducted load testing with **JMeter**, identifying bottlenecks and improving performance under heavy traffic.
* Collaborated with cross-functional teams (product owners, QA, DevOps, architects) in an **Agile Scrum environment.**
* Conducted code reviews to ensure best practices, maintainability, and team knowledge-sharing.
* Mentored junior developers in adopting modern **Java 17** features and full-stack development practices.
* Actively participated in design discussions to align architectural choices with scalability, compliance, and business goals.
* Ensured clear communication with stakeholders to translate business requirements into technical deliverables.

**Environment:** Java 17/11/8, Spring Boot, Spring Cloud, Spring MVC, Node.js, Next.js, React.js, Angular 12/13, TypeScript, Redux, Hooks, Python, Visual Basic (Scripting), AWS (Elastic Beanstalk, S3, DynamoDB, Lambda, Route 53, CloudWatch), Terraform, Docker, Kubernetes, Jenkins, GitHub Actions, GitLab CI, Argo CD, GitOps practices, IBM DB2, MySQL, Oracle, PostgreSQL, Redis Cache, DynamoDB, MongoDB, NoSQL stores, Apache Kafka, IBM MQSeries, Databricks, ETL pipelines, Grafana, Prometheus, Okta SDK, Spring Security, OAuth2, JWT, AWS IAM, AWS VPC Security Groups, AWS Parameter Store, Unit, Mockito, Selenium, Jasmine, Karma, Mocha, Chai, JMeter, Swagger/OpenAPI v3, RESTful Web Services, GraphQL, Maven, HTML5, CSS3, AJAX, Agile methodology.

**Client: HSO, India**  **Dec 2020 – July 2021**

**Role: Java Developer**

**Project Overview:** At HSO, I contributed to the development of a **scalable microservices-based enterprise app** using **Java/Spring Boot** on **Azure App Services**, implementing serverless workflows, secure storage, and real-time business insights. Optimized SQL/NoSQL performance, built interactive UIs with An**gularJS and ReactJS** and automated deployments via **Azure DevOps**, improving processing speed by 40% and ensuring high system reliability.

**Responsibilities:**

* Designed and developed modular microservices using **Spring Boot** to support real-time business data insights for enterprise applications.
* Deployed Spring Boot microservices to **Azure App Service** with slot-based blue/green releases and auto scale, eliminating VM ops overhead.
* Implemented event-driven workflows with **Azure Functions** (timer/HTTP triggers) and durable patterns for reliable background processing.
* **Worked with IBM DB2** for database performance optimization, including creating efficient stored procedures and queries for large-scale transaction systems.
* Managed artifacts in **Azure Blob Storage** using Hot/Cool/Archive tiers and SAS-secured access with lifecycle policies to control costs
* Developed user interfaces using **AngularJS** and **ReactJS**, improving the end-user experience and enabling real-time data visualization.
* Leveraged **indexing and caching** techniques with **SQL** (Oracle, MySQL) and **NoSQL** (MongoDB) to reduce data processing time by 40%.
* Integrated **Spring MVC** to create **RESTful APIs** that facilitated seamless communication between front-end and back-end systems.
* Secured app secrets in **Azure Key Vault** with managed identities, removing secrets from code and enabling controlled rotation.
* Hardened environments using **VNets/NSGs** and **Private Endpoints** for private PaaS access, aligning with enterprise security baselines.
* Implemented **log analytics** with the **ELK Stack** (Elasticsearch, Logstash, Kibana) for real-time monitoring and quick error resolution.
* Instrumented services with **Application Insights**/**Azure Monitor** (OpenTelemetry) to enable distributed tracing and proactive alerting
* Collaborated in the complete **SDLC**, using Agile methodologies to ensure timely software delivery and consistent communication with stakeholders.
* Created and maintained **CI/CD pipelines** using **Jenkins**, automating the build and deployment process to reduce manual effort and errors.
* Deployed through **Azure DevOps Pipelines** (YAML) with gated promotions, security scans, and slot swaps for safe, auditable releases.
* Conducted **load testing** with **JMeter** to assess system performance and identify areas for optimization.
* Designed and optimized database schemas, reducing query execution times by implementing best practices for database indexing and partitioning.
* Developed data processing pipelines using **Kafka** for real-time event-driven architectures in microservices applications.
* Enhanced system security by implementing **Spring Security** for authentication and authorization in a distributed environment.
* Built data reports using **Jasper Report**s and integrated **SSRS** for executive dashboards and business analytics.

**Environment:** Java 11/8, Spring Boot, Spring Cloud, Spring MVC, Spring Security, AngularJS, ReactJS, SQL, NoSQL (MongoDB), Oracle, MySQL, Azure App Service, Azure Functions, Azure Blob Storage (Hot/Cool/Archive tiers), Azure Key Vault, Azure VNets & NSGs, Azure Private Endpoints, Azure DevOps Pipelines (YAML), Application Insights, Azure Monitor (Open Telemetry), Spring Security, OAuth2, JWT, Azure Key Vault, Azure VNets, Private Endpoints, IBM DB2, Oracle, MySQL, MongoDB, Apache Kafka, ELK Stack (Elasticsearch, Logstash, Kibana), Application Insights, Azure Monitor, Azure DevOps Pipelines, Jenkins, Maven, JUnit, JMeter, Jasper Reports, SSRS

**Client: Byju’s, India**  **April 2018 - Nov 2020**

**Role: Jr. Java Developer**

**Project Overview:** Contributed to developing high-traffic **educational platforms** by migrating monolithic systems to **microservices** and deploying on **GCP** and **AWS** for scalability. Built backend services in Java and Spring Boot, implemented **Kafka** and **GCP Pub/Sub** for real-time data streaming, and developed serverless workflows with **Cloud Functions**. Leveraged **GCS, BigQuery, and Firestore** to improve content delivery speed by 45% and enable real-time collaborative learning features. Containerized services with Docker, deployed to Kubernetes, and optimized system performance through database tuning, monitoring with Stackdriver, and performance testing with JMeter.

**Responsibilities:**

* Assisted in the design and implementation of **microservices** using **Spring Boot** to improve scalability and system performance.
* Contributed to migrating monolithic systems to **microservices architecture**, improving deployment efficiency and fault tolerance.
* Developed **backend services** using **Java 8**, **J2EE**, and integrated **JMS** messaging systems for asynchronous communication.
* Built real-time collaborative features on **Fire store**, enabling instant quiz updates and group learning sessions with offline access.
* Developed **Cloud Functions** triggered by GCS/Pub/Sub events for auto-grading, content processing, and email notifications, cutting processing costs by 30%.
* Managed educational media content in **Google Cloud Storage (GCS)** with signed URLs, lifecycle policies, and CDN integration, improving global video/content delivery performance by 45%.”
* Implemented **Kafka-based messaging systems** to enable real-time data streaming and processing in the educational platform.
* Leveraged **BigQuery** for near real-time analytics of billions of learning activity events, enabling personalized content recommendations and data-driven platform improvements.
* Optimized database performance through **query tuning** and **schema design** in both **SQL (Oracle, MySQL)** and **NoSQL** databases.
* Built and maintained **RESTful APIs** for integration with third-party services and front-end applications.
* Assisted in developing a **real-time learning platform** using **AngularJS** and **ReactJS**, ensuring a responsive and engaging user experience.
* **Worked with IBM DB2** to enhance data retrieval and transaction processing through stored procedures and optimized queries.
* Implemented **GCP Pub/Sub** for event-driven communication across learning modules, decoupling services and improving system resilience.
* Implemented **containerization** using **Docker** to improve development workflows and simplify deployment on **Kubernetes** clusters.
* Configured centralized monitoring with **Stackdriver** for GCP workloads, creating proactive alerts to reduce MTTR by 40%
* Assisted with **performance testing** using **JMeter** to ensure the platform could handle high traffic volumes.
* Participated in Agile **Sprint Planning** and **Daily Stand-ups**, collaborating closely with cross-functional teams to ensure timely delivery of features.
* Created reusable unit tests using **Mockito** and **JUnit** to validate core business logic and ensure regression safety.

**Environment:** Java 8, J2EE, JMS, Spring Boot, Spring Cloud, Spring MVC, AngularJS, ReactJS, JMS, RESTful API’s, GCP (Google Cloud Storage, BigQuery, Pub/Sub, Cloud Functions, Firestore, Stackdriver), AWS (S3, Lambda, DynamoDB, EC2), Docker, Kubernetes, Jenkins, Apache Kafka, GCP Pub/Sub, Oracle, MySQL, IBM DB2, NoSQL, JUnit, Mockito, JMeter, Swagger, Git, GitHub, Eclipse, NetBeans, TDD methodology

**EDUCATION:**

**MASTERS IN INFORMATION SYSTEMS,**

**CENTRAL MICHIGAN UNIVERSITY.**