

Ramkrushna Madole

+91 7620877046  Mail  LinkedIn  Github  Leetcode

PROFILE SUMMARY

Results-driven Java Backend Developer with 2+ years of experience building scalable, secure banking applications. Proven track record in reducing transaction failures by 15% and delivering high-uptime microservices for HDFC Bank. Skilled in Java, Spring Boot, PL/SQL, and cloud-native deployments (Docker, Kubernetes, AWS).

EDUCATION

B.Tech in Electronics Engineering **2019 - 2023**
Walchand College of Engineering, Sangli. 8.33 CGPA

PROFESSIONAL EXPERIENCE

Associate Consultant | Oracle Financial Services Software Ltd. **Jun 2023 - Present**

- Designed, developed, and maintained high-volume Java microservices and RESTful APIs powering Flexcube banking screens and NEFT/IMPS/UPI payment modules, processing millions of transactions daily with improved reliability and reduced downtime.
- Implemented multiple **Spring Boot REST APIs** handling core banking operations such as locker opening, account onboarding, net banking services, and full transaction lifecycle actions (inquiry, create, amend, modify, delete, authorise), while meeting strict performance SLAs.
- Built JMS/Kafka-based messaging frameworks to handle asynchronous communication and high-volume transaction requests, improving throughput, resilience, and reducing API coupling.
- Built Angular UI modules to integrate with backend microservices, handling form validations, REST API integration, error handling, and role-based UI rendering.
- Applied Angular lifecycle hooks, RxJS observables, and dependency injection to create scalable, testable frontend components aligned with enterprise backend systems.
- Tuned PL/SQL-heavy Oracle integrations, refactoring stored procedures and indexing strategies to trim average query latency by **40 ms**.
- Implemented Spring microservices containerization on Tomcat as part of Oracle Banking Routing Hub project, using Docker/Kubernetes.
- Integrated CI/CD pipelines using Jenkins and GitHub Actions** for automated build, test, and deployment of microservices to Kubernetes clusters, reducing manual deployment effort by 70%.
- Containerized microservices with Docker and Kubernetes, enabling **blue-green deployments** and **99.9% up-time** in production clusters.
- Created a Java utility for auto-generation of REST/SOAP passthrough APIs on the Oracle Banking Platform kernel framework, leveraging **JavaFX, Velocity, and Swagger** for automation.
- Built comprehensive **JUnit 5** and **Mockito** test suites across service and API layers, improving code quality, stability and integration with CI pipelines.
- Integrated microservices with enterprise platforms like **Vision+**, **FCPB**, and **FinnOne**, widely used by HDFC Bank for critical business operations.
- Applied SOLID principles and clean code practices while developing enterprise-grade microservices, ensuring extensibility, maintainability, and ease of debugging.

TECHNICAL SKILLS

Programming Languages: Java (core + J2EE), C++, C, Type Script, Java Script

Technologies/Frameworks/Libraries: Spring Boot, Hibernate, Node.js, TypeScript, Angular, HTML5, CSS3

Databases: MySQL, MongoDB, PLSQL

Cloud Services: AWS (IAM, S3, EC2, VPC, Lambda,)

Version Control: Git, SVN

DevOps/Deployment Tools: Docker, Kubernetes, Jenkins, Maven, Git, JIRA

Course Work: Data Structures and Algorithms, Database Management System, Operating System, Computer Networking, System Design

PROJECTS

RAG-Based Document Chat System

Spring Boot, Java 17, JPA/Hibernate, MySQL, Vector DB (Qdrant), OpenAI, Docker, Spring Security

- Built a **Spring Boot 3–based RAG application** enabling users to upload documents and ask natural-language questions with high-accuracy, context-aware responses.
- Designed a **document ingestion pipeline** with text extraction, chunking, embedding generation, and vector storage to support scalable semantic search.
- Implemented **REST APIs** for document upload and querying, following clean layered architecture.
- Modeled relational data using **JPA/Hibernate**, maintaining one-to-many relationships between documents and document chunks.
- Integrated **Qdrant vector database** for similarity search during query processing.
- Secured APIs using **Spring Security**, with configurable endpoint access and JWT-ready design.

Inventory & Sales System

Spring Boot, Microservices, Docker, Kubernetes, MySQL, Kafka, Jenkins

- Built Spring Boot microservices (product, order, inventory, notification) with **JWT-secured REST APIs**.
- Orchestrated services using **Docker and Docker Compose**, enabling one-command local environment setup.
- Implemented event-driven communication via **Spring Cloud Stream and RabbitMQ**.
- Designed a **CI/CD pipeline** using Jenkins and Docker for automated deployments.
- Added centralized tracing and health metrics using **Zipkin and Spring Actuator**.