MockAttempt

TECH ACADEMY • OFFICIAL COURSE SYLLABUS BROCHURE

MockAttempt Academy Pvt. Ltd. • Live Architect Mentorship, AI Assessments & Client Production

Admissions & Support: support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: JMSB • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/java-spring-boot-microservices-system-design

JMSB • BACKEND ENGINEERING | INTERMEDIATE TO SENIOR | 100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

Java Spring Boot Microservices & System Design

Architect fault-tolerant Java 21 & Spring Boot microservices with circuit breakers, Kafka event streaming, topic-wise Mock Exams, and AI System Design Interviews.

Duration: 18–20 Weeks • Live + Hands-On Projects (214 Hrs) • **Modules:** 14 • **Live Topics:** 42 • **Labs & Tasks:** 84 • **Mocks / AI Vivas:** 84 / 84 • **Projects:** 14 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹8,999** + 18% GST (₹1,620) = **Total Incl. GST: ₹10,619**

Live Batch Tracks: Weekday Live Online Batch (Mon – Fri) (Monday to Friday (5 Live Interactive Classes / Week) • 1.5 Hours / Day (60 Mins Live Architecture & Coding + 30 Mins Hands-On Lab) • 18–20 Weeks Fast-Track) | Weekend Intensive Live Online Batch (Sat – Sun) (Saturday & Sunday (Designed for Working Professionals & College Students) • 3.0 Hours / Day (2 Hrs Deep-Dive Live Coding + 1 Hr Guided Project Build) • 20–22 Weeks Weekend Track)

Core Technologies & Engineering Stack: Java 21 • Spring Boot 3 • Hibernate • PostgreSQL • Apache Kafka • Docker

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Build production Spring Security 6 & OAuth2 resource servers
2. Design Saga distributed transactions & Kafka consumers
3. Pass Spring internals Mock Tests & Voice AI System Design Interviews
4. Domain-Driven bounded contexts across Spring Boot 3 microservices with API Gateways and centralized configuration.
5. Event-driven asynchronous choreography & Saga orchestration over Apache Kafka with idempotent consumers and Outbox pattern.
6. Resilience4j circuit breakers, bulkheads, rate limiters, retries, and Spring Security 6 OAuth2/JWT propagation.
7. Database-per-service isolation with PostgreSQL, Flyway migrations, Redis distributed caching, and OpenTelemetry tracing.
8. Java 21 Virtual Threads, Records & Spring Boot 3 Internals — Achieve massive I/O concurrency while keeping code clean, readable, and maintainable.

Eligibility & Prerequisites: No prior Java Spring Boot Microservices & System Design industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews. • Dedication to write original code in your GitHub portfolio and complete the mentor-guided Live Client Production Sprint. • No prior Java Spring Boot Microservices & System Design industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (14 MODULES • 42 LIVE TOPICS)

MODULE 1 OF 14 • 28 LIVE + LAB HOURS • 1 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Java 21 Virtual Threads & Spring Boot Core

JVM memory model, Project Loom virtual threads, IoC container, and bean lifecycles.

Objectives: ✓ Configure Java 21 virtual-thread executors in Spring Boot 3 for high-concurrency I/O • ✓ Inspect Spring IoC bean lifecycle hooks, JDK dynamic proxies, and @Transactional boundaries • ✓ Benchmark HikariCP connection pools and structured concurrency scopes under load

1.1 Spring IoC Container, Bean Scopes & Proxy Internals [50 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring IoC Container, Bean Scopes & Proxy Internals in real Java Spring Boot Microservices & System Design systems — specifically: How JDK dynamic...

Module 1 Hands-On Capstone Project (100 Marks): High-Throughput Spring Boot 3 Order Settlement Service — Build a Java 21 + Spring Boot 3 microservice using Virtual Threads, optimistic locking, and transactional outbox events.

MODULE 2 OF 14 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Modern Java 21 Core: Records, Sealed Hierarchies, Generics & Streams

Production-grade Java Spring Boot Microservices & System Design engineering module covering Modern Java 21 Core: Records, Sealed Hierarchies, Generics & Streams with hands-on labs and capstone build.

Objectives: ✓ Records, Sealed Interfaces & Pattern Matching for Switch — Purpose: Model immutable domain DTOs and exhaustive payment/order states in modern Java 21. • ✓ Collections Internals, Bounded Generics & Custom Comparators — Purpose: Choose optimal data structures (HashMap, TreeMap, ArrayDeque) for low-latency services. • ✓ Functional Interfaces, Stream Collectors & BigDecimal Monetary Math — Purpose: Transform complex datasets declaratively and compute exact financial balances. • ✓ Domain Exception Hierarchies, NIO.2 File Streams & Clean SOLID Boundaries — Purpose: Handle recoverable vs fatal business errors and process large files safely in Java.

2.1 Records, Sealed Interfaces & Pattern Matching for Switch [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Records, Sealed Interfaces & Pattern Matching for Switch in real Java Spring Boot Microservices & System Design systems — specifically: Model immu...

2.2 Collections Internals, Bounded Generics & Custom Comparators [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Collections Internals, Bounded Generics & Custom Comparators in real Java Spring Boot Microservices & System Design systems — specifically: Choose...

2.3 Functional Interfaces, Stream Collectors & BigDecimal Monetary Math [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Functional Interfaces, Stream Collectors & BigDecimal Monetary Math in real Java Spring Boot Microservices & System Design systems — specifically:...

2.4 Domain Exception Hierarchies, NIO.2 File Streams & Clean SOLID Boundaries [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Domain Exception Hierarchies, NIO.2 File Streams & Clean SOLID Boundaries in real Java Spring Boot Microservices & System Design systems — specifi...

Module 2 Hands-On Capstone Project (100 Marks): Immutable Financial Ledger & Functional Analytics Engine in Java 21 — Design, implement, test, and document 'Immutable Financial Ledger & Functional Analytics Engine in Java 21' applying every production concept from Module 2 (Modern Java 21 Core: Records, Sea...

MODULE 3 OF 14 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

JVM Memory Internals, Garbage Collection Tuning & Concurrency

Production-grade Java Spring Boot Microservices & System Design engineering module covering JVM Memory Internals, Garbage Collection Tuning & Concurrency with hands-on labs and capstone build.

Objectives: ✓ JVM Heap Generations, Metaspace, G1GC/ZGC & Heap Dump Forensics — Purpose: Diagnose OutOfMemoryError spikes and tune GC pause times for SLA-critical APIs. • ✓ Java Memory Model, Locks, Atomic Operations & ConcurrentHashMap — Purpose: Write thread-safe shared state without race conditions or lock contention. • ✓ CompletableFuture Pipelines, ExecutorPools & Structured Concurrency — Purpose: Fan out parallel microservice calls with strict timeouts and error fallbacks. • ✓ JMH Microbenchmarking, Lock-Free Queues & Thread-Pinning Diagnostics — Purpose: Measure real JVM throughput scientifically and prevent virtual-thread pinning.

3.1 JVM Heap Generations, Metaspace, G1GC/ZGC & Heap Dump Forensics [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply JVM Heap Generations, Metaspace, G1GC/ZGC & Heap Dump Forensics in real Java Spring Boot Microservices & System Design systems — specifically: Dia...

3.2 Java Memory Model, Locks, Atomic Operations & ConcurrentHashMap [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Java Memory Model, Locks, Atomic Operations & ConcurrentHashMap in real Java Spring Boot Microservices & System Design systems — specifically: Wri...

3.3 CompletableFuture Pipelines, ExecutorPools & Structured Concurrency [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply CompletableFuture Pipelines, ExecutorPools & Structured Concurrency in real Java Spring Boot Microservices & System Design systems — specifically:...

3.4 JMH Microbenchmarking, Lock-Free Queues & Thread-Pinning Diagnostics [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply JMH Microbenchmarking, Lock-Free Queues & Thread-Pinning Diagnostics in real Java Spring Boot Microservices & System Design systems — specifically...

Module 3 Hands-On Capstone Project (100 Marks): High-Throughput Concurrent Price Aggregator with Virtual Threads — Design, implement, test, and document 'High-Throughput Concurrent Price Aggregator with Virtual Threads' applying every production concept from Module 3 (JVM Memory Internals, Garbage Collec...

MODULE 4 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Spring Boot 3 Core IoC, REST API Design, Validation & OpenAPI

Production-grade Java Spring Boot Microservices & System Design engineering module covering Spring Boot 3 Core IoC, REST API Design, Validation & OpenAPI with hands-on labs and capstone build.

Objectives: ✓ Spring IoC Container, Constructor Injection, Profiles & @ConfigurationProperties — Purpose: Wire clean, testable Spring Boot 3 services with strongly typed externalized config. • ✓ RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail — Purpose: Expose clean, self-documenting REST APIs with consistent validation error contracts. • ✓ Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown — Purpose: Add non-invasive audit logging and Kubernetes-ready lifecycle hooks.

4.1 Spring IoC Container, Constructor Injection, Profiles & @ConfigurationProperties [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring IoC Container, Constructor Injection, Profiles & @ConfigurationProperties in real Java Spring Boot Microservices & System Design systems —...

4.2 RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail in real Java Spring Boot Microservices & System Design systems — specifi...

4.3 Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown in real Java Spring Boot Microservices & System Design systems — spe...

Module 4 Hands-On Capstone Project (100 Marks): Enterprise Customer Onboarding REST Service with ProblemDetails — Design, implement, test, and document 'Enterprise Customer Onboarding REST Service with ProblemDetails' applying every production concept from Module 4 (Spring Boot 3 Core IoC, REST API Desi...

MODULE 5 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Spring Data JPA, Hibernate Internals, N+1 Elimination & Projections

Production-grade Java Spring Boot Microservices & System Design engineering module covering Spring Data JPA, Hibernate Internals, N+1 Elimination & Projections with hands-on labs and capstone build.

Objectives: ✓ Hibernate Persistence Context, Dirty Checking & Entity Lifecycle States — Purpose: Master Transient, Managed, Detached, and Removed states so Hibernate never surprises you. • ✓ Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections — Purpose: Reduce 201 SQL queries per API call down to 1–2 indexed queries. • ✓ Optimistic (@Version) vs Pessimistic Locking & JDBC Batch Inserts — Purpose: Prevent double-spending wallet balances and bulk-insert thousands of rows rapidly.

5.1 Hibernate Persistence Context, Dirty Checking & Entity Lifecycle States [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Hibernate Persistence Context, Dirty Checking & Entity Lifecycle States in real Java Spring Boot Microservices & System Design systems — specifica...

5.2 Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections in real Java Spring Boot Microservices & System Design systems — specificall...

5.3 Optimistic (@Version) vs Pessimistic Locking & JDBC Batch Inserts [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Optimistic (@Version) vs Pessimistic Locking & JDBC Batch Inserts in real Java Spring Boot Microservices & System Design systems — specifically: P...

Module 5 Hands-On Capstone Project (100 Marks): High-Performance Order & Catalog Persistence Layer — Design, implement, test, and document 'High-Performance Order & Catalog Persistence Layer' applying every production concept from Module 5 (Spring Data JPA, Hibernate Internals, N+1 Eliminat...

MODULE 6 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

PostgreSQL Production Engineering, ACID Isolation & Flyway Migrations

Production-grade Java Spring Boot Microservices & System Design engineering module covering PostgreSQL Production Engineering, ACID Isolation & Flyway Migrations with hands-on labs and capstone build.

Objectives: ✓ MVCC Internals, Transaction Isolation Levels & Deadlock Resolution — Purpose: Choose Read Committed vs Repeatable Read vs Serializable for financial correctness. • ✓ B-Tree, GIN, Partial Indexes & EXPLAIN (ANALYZE, BUFFERS) Tuning — Purpose: Read PostgreSQL query execution plans and optimize multi-million-row tables. • ✓ Versioned Flyway Migrations & Expand-Contract Zero-Downtime Rollouts — Purpose: Evolve live production schemas while old and new microservice pods run side-by-side.

6.1 MVCC Internals, Transaction Isolation Levels & Deadlock Resolution [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply MVCC Internals, Transaction Isolation Levels & Deadlock Resolution in real Java Spring Boot Microservices & System Design systems — specifically:...

6.2 B-Tree, GIN, Partial Indexes & EXPLAIN (ANALYZE, BUFFERS) Tuning [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply B-Tree, GIN, Partial Indexes & EXPLAIN (ANALYZE, BUFFERS) Tuning in real Java Spring Boot Microservices & System Design systems — specifically: Re...

6.3 Versioned Flyway Migrations & Expand-Contract Zero-Downtime Rollouts [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Versioned Flyway Migrations & Expand-Contract Zero-Downtime Rollouts in real Java Spring Boot Microservices & System Design systems — specifically...

Module 6 Hands-On Capstone Project (100 Marks): Zero-Downtime Schema Evolution & Ledger Database Architecture — Design, implement, test, and document 'Zero-Downtime Schema Evolution & Ledger Database Architecture' applying every production concept from Module 6 (PostgreSQL Production Engineering, ACID...

MODULE 7 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Spring Security 6, OAuth2/OIDC, Stateless JWT & Method-Level RBAC

Production-grade Java Spring Boot Microservices & System Design engineering module covering Spring Security 6, OAuth2/OIDC, Stateless JWT & Method-Level RBAC with hands-on labs and capstone build.

Objectives: ✓ Spring Security 6 FilterChain Architecture & Password Hashing — Purpose: Configure modern lambda DSL security filter chains with zero deprecated WebSecurityConfigurerAdapter code. • ✓ OAuth2 Resource Servers, RSA/JWKS JWT Verification & Token Rotation — Purpose: Validate asymmetric JWTs across 15+ microservices without hitting a central DB per request. • ✓ Fine-Grained @PreAuthorize Method Security & Service-to-Service Auth — Purpose: Enforce role/ownership permissions and mTLS/client-credentials between internal services.

7.1 Spring Security 6 FilterChain Architecture & Password Hashing [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring Security 6 FilterChain Architecture & Password Hashing in real Java Spring Boot Microservices & System Design systems — specifically: Confi...

7.2 OAuth2 Resource Servers, RSA/JWKS JWT Verification & Token Rotation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply OAuth2 Resource Servers, RSA/JWKS JWT Verification & Token Rotation in real Java Spring Boot Microservices & System Design systems — specifically:...

7.3 Fine-Grained @PreAuthorize Method Security & Service-to-Service Auth [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Fine-Grained @PreAuthorize Method Security & Service-to-Service Auth in real Java Spring Boot Microservices & System Design systems — specifically...

Module 7 Hands-On Capstone Project (100 Marks): Centralized Identity & OAuth2 Resource Server Security Mesh — Design, implement, test, and document 'Centralized Identity & OAuth2 Resource Server Security Mesh' applying every production concept from Module 7 (Spring Security 6, OAuth2/OIDC, Stateless...

MODULE 8 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Microservices Boundaries, Spring Cloud Gateway & Service Communication

Production-grade Java Spring Boot Microservices & System Design engineering module covering Microservices Boundaries, Spring Cloud Gateway & Service Communication with hands-on labs and capstone build.

Objectives: ✓ Domain-Driven Bounded Contexts & Database-Per-Service Decomposition — Purpose: Split monoliths along true business capabilities instead of creating a distributed monolith. • ✓ Spring Cloud Gateway: Reactive Routing, Redis Rate Limiting & Auth Offload — Purpose: Centralize TLS, JWT verification, and per-tenant throttling at the API edge. • ✓ Declarative HTTP Clients (@HttpExchange), gRPC & Timeout Budgets — Purpose: Communicate synchronously between services safely with strict connect/read deadlines.

8.1 Domain-Driven Bounded Contexts & Database-Per-Service Decomposition [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Domain-Driven Bounded Contexts & Database-Per-Service Decomposition in real Java Spring Boot Microservices & System Design systems — specifically:...

8.2 Spring Cloud Gateway: Reactive Routing, Redis Rate Limiting & Auth Offload [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring Cloud Gateway: Reactive Routing, Redis Rate Limiting & Auth Offload in real Java Spring Boot Microservices & System Design systems — specif...

8.3 Declarative HTTP Clients (@HttpExchange), gRPC & Timeout Budgets [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Declarative HTTP Clients (@HttpExchange), gRPC & Timeout Budgets in real Java Spring Boot Microservices & System Design systems — specifically: Co...

Module 8 Hands-On Capstone Project (100 Marks): Edge API Gateway with Rate Limiting, Routing & BFF Aggregation — Design, implement, test, and document 'Edge API Gateway with Rate Limiting, Routing & BFF Aggregation' applying every production concept from Module 8 (Microservices Boundaries, Spring Cloud...

MODULE 9 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Fault-Tolerant Distributed Systems with Resilience4j & Caching

Production-grade Java Spring Boot Microservices & System Design engineering module covering Fault-Tolerant Distributed Systems with Resilience4j & Caching with hands-on labs and capstone build.

Objectives: ✓ Circuit Breakers, Bulkhead Isolation & Exponential Jitter Retries — Purpose: Prevent one degraded downstream gateway from cascading across your entire microservice fleet. • ✓ Multi-Tier Caching (Caffeine L1 + Redis L2) & Graceful Degradation — Purpose: Serve catalog and FX-rate reads from memory even during partial database maintenance. • ✓ Distributed Idempotency, ShedLock Scheduled Jobs & Chaos Fault Injection — Purpose: Ensure scheduled settlements and retried requests execute safely across N replicas.

9.1 Circuit Breakers, Bulkhead Isolation & Exponential Jitter Retries [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Circuit Breakers, Bulkhead Isolation & Exponential Jitter Retries in real Java Spring Boot Microservices & System Design systems — specifically: P...

9.2 Multi-Tier Caching (Caffeine L1 + Redis L2) & Graceful Degradation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Multi-Tier Caching (Caffeine L1 + Redis L2) & Graceful Degradation in real Java Spring Boot Microservices & System Design systems — specifically:...

9.3 Distributed Idempotency, ShedLock Scheduled Jobs & Chaos Fault Injection [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Distributed Idempotency, ShedLock Scheduled Jobs & Chaos Fault Injection in real Java Spring Boot Microservices & System Design systems — specific...

Module 9 Hands-On Capstone Project (100 Marks): Self-Healing Checkout Orchestrator with Circuit Breakers & Fallbacks — Design, implement, test, and document 'Self-Healing Checkout Orchestrator with Circuit Breakers & Fallbacks' applying every production concept from Module 9 (Fault-Tolerant Distributed Syste...

MODULE 10 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Event-Driven Microservices with Apache Kafka, Outbox & Saga Patterns

Production-grade Java Spring Boot Microservices & System Design engineering module covering Event-Driven Microservices with Apache Kafka, Outbox & Saga Patterns with hands-on labs and capstone build.

Objectives: ✓ Kafka Partitions, Consumer Groups, Acks=all & Schema Registry — Purpose: Stream millions of ordered domain events reliably between decoupled Spring Boot services. • ✓ Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers — Purpose: Eliminate dual-write bugs between PostgreSQL commits and Kafka publishes. • ✓ Distributed Sagas: Choreography vs Orchestration & Compensating Actions — Purpose: Maintain cross-service data consistency without brittle two-phase XA distributed locks.

10.1 Kafka Partitions, Consumer Groups, Acks=all & Schema Registry [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Kafka Partitions, Consumer Groups, Acks=all & Schema Registry in real Java Spring Boot Microservices & System Design systems — specifically: Strea...

10.2 Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers in real Java Spring Boot Microservices & System Design systems — specifically: E...

10.3 Distributed Sagas: Choreography vs Orchestration & Compensating Actions [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Distributed Sagas: Choreography vs Orchestration & Compensating Actions in real Java Spring Boot Microservices & System Design systems — specifica...

Module 10 Hands-On Capstone Project (100 Marks): Distributed Order–Inventory–Payment Saga over Apache Kafka — Design, implement, test, and document 'Distributed Order–Inventory–Payment Saga over Apache Kafka' applying every production concept from Module 10 (Event-Driven Microservices with Apache Ka...

MODULE 11 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Observability: OpenTelemetry Tracing, Micrometer Prometheus & Grafana

Production-grade Java Spring Boot Microservices & System Design engineering module covering Observability: OpenTelemetry Tracing, Micrometer Prometheus & Grafana with hands-on labs and capstone build.

Objectives: ✓ Distributed Tracing with OpenTelemetry across HTTP & Kafka Hops — Purpose: Follow a single slow request across Gateway -> Order -> Kafka -> Payment in one trace flamegraph. • ✓ Micrometer Dimensional Metrics, Prometheus Histograms & Alerting SLOs — Purpose: Measure latency percentiles (p50/p95/p99), error rates, and JVM saturation accurately. • ✓ Structured JSON Log Aggregation (Loki/ELK) & Incident Runbooks — Purpose: Query terabytes of multi-pod logs by traceId and tenantId in seconds.

11.1 Distributed Tracing with OpenTelemetry across HTTP & Kafka Hops [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Distributed Tracing with OpenTelemetry across HTTP & Kafka Hops in real Java Spring Boot Microservices & System Design systems — specifically: Fol...

11.2 Micrometer Dimensional Metrics, Prometheus Histograms & Alerting SLOs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Micrometer Dimensional Metrics, Prometheus Histograms & Alerting SLOs in real Java Spring Boot Microservices & System Design systems — specificall...

11.3 Structured JSON Log Aggregation (Loki/ELK) & Incident Runbooks [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Structured JSON Log Aggregation (Loki/ELK) & Incident Runbooks in real Java Spring Boot Microservices & System Design systems — specifically: Quer...

Module 11 Hands-On Capstone Project (100 Marks): Full-Stack Golden-Signals Observability & Distributed Trace Suite — Design, implement, test, and document 'Full-Stack Golden-Signals Observability & Distributed Trace Suite' applying every production concept from Module 11 (Observability: OpenTelemetry Traci...

MODULE 12 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Testcontainers Integration Suites, Layered Docker JARS & Kubernetes Prod

Production-grade Java Spring Boot Microservices & System Design engineering module covering Testcontainers Integration Suites, Layered Docker JARS & Kubernetes Prod with hands-on labs and capstone build.

Objectives: ✓ JUnit 5, Mockito, @WebMvcTest Slices & Testcontainers Real DB/Kafka Suites — Purpose: Test Spring Boot microservices against real PostgreSQL, Redis, and Kafka containers in CI. • ✓ Spring Boot Layered JARS, Distroless Containers & JVM Container Ergonomics — Purpose: Build fast-booting, security-hardened container images tailored for cloud orchestration. • ✓ Kubernetes Deployments, HPA Autoscaling, Secrets & Zero-Downtime Rollouts — Purpose: Deploy resilient Spring Boot microservice fleets onto Kubernetes clusters.

12.1 JUnit 5, Mockito, @WebMvcTest Slices & Testcontainers Real DB/Kafka Suites [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply JUnit 5, Mockito, @WebMvcTest Slices & Testcontainers Real DB/Kafka Suites in real Java Spring Boot Microservices & System Design systems — specif...

12.2 Spring Boot Layered JARS, Distroless Containers & JVM Container Ergonomics [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Spring Boot Layered JARS, Distroless Containers & JVM Container Ergonomics in real Java Spring Boot Microservices & System Design systems — specif...

12.3 Kubernetes Deployments, HPA Autoscaling, Secrets & Zero-Downtime Rollouts [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Kubernetes Deployments, HPA Autoscaling, Secrets & Zero-Downtime Rollouts in real Java Spring Boot Microservices & System Design systems — specifi...

Module 12 Hands-On Capstone Project (100 Marks): Production Kubernetes Deployment with Testcontainers CI Gate — Design, implement, test, and document 'Production Kubernetes Deployment with Testcontainers CI Gate' applying every production concept from Module 12 (Testcontainers Integration Suites, Laye...

MODULE 13 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Transactional Outbox, Idempotent Consumers & Distributed Locking

Self-explanatory list-and-bullet production mastery module covering Transactional Outbox, Idempotent Consumers & Distributed Locking in Java Spring Boot Microservices & System Design, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Transactional Outbox Pattern with Spring Data JPA & Debezium/Polling Relay — Purpose: Eliminate dual-write inconsistencies between PostgreSQL commits and Apache Kafka event publishing. • ✓ Idempotent Kafka Consumers, Dead-Letter Topics & Poison-Pill Recovery — Purpose: Process duplicate payment or order messages safely without corrupting downstream microservice ledgers. • ✓ Optimistic/Pessimistic JPA Locking & Redis Distributed Mutexes — Purpose: Prevent race conditions and lost updates when concurrent threads modify shared account balances. • ✓ Build, verify, and document 'Zero-Data-Loss Outbox Relay & Redis Distributed Settlement Guard' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice +

Google Gemini AI Technical Interviews.

13.1 Transactional Outbox Pattern with Spring Data JPA & Debezium/Polling Relay [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Transactional Outbox Pattern with Spring Data JPA & Debezium/Polling Relay in real Java Spring Boot Microservices & System Design systems — specif...

13.2 Idempotent Kafka Consumers, Dead-Letter Topics & Poison-Pill Recovery [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Idempotent Kafka Consumers, Dead-Letter Topics & Poison-Pill Recovery in real Java Spring Boot Microservices & System Design systems — specificall...

13.3 Optimistic/Pessimistic JPA Locking & Redis Distributed Mutexes [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Optimistic/Pessimistic JPA Locking & Redis Distributed Mutexes in real Java Spring Boot Microservices & System Design systems — specifically: Prev...

Module 13 Hands-On Capstone Project (100 Marks): Zero-Data-Loss Outbox Relay & Redis Distributed Settlement Guard — Architect, implement, test, and document 'Zero-Data-Loss Outbox Relay & Redis Distributed Settlement Guard' applying every bullet-wise engineering concept from Module 13 (Transactional Outbo...

MODULE 14 OF 14 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

JVM Production Profiling, GraalVM Readiness & Microservices System Design

Self-explanatory list-and-bullet production mastery module covering JVM Production Profiling, GraalVM Readiness & Microservices System Design in Java Spring Boot Microservices & System Design, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause Tuning in Java 21 — Purpose: Diagnose memory leaks, thread contention, and tail-latency spikes in high-throughput Spring Boot services. • ✓ Graceful Kubernetes Shutdown, Readiness/Liveness Probes & Testcontainers CI — Purpose: Run real PostgreSQL + Kafka integration suites in CI and deploy zero-downtime microservice releases. • ✓ Enterprise Java Microservices System Design & Failure-Mode Walkthroughs — Purpose: Articulate CAP theorem, Saga vs 2PC, and backpressure trade-offs clearly in senior Java interviews. • ✓ Build, verify, and document 'JFR Latency Profiler, Actuator Readiness Suite & Architecture Defense' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

14.1 JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause Tuning in Java 21 [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause Tuning in Java 21 in real Java Spring Boot Microservices & System Design systems — specif...

14.2 Graceful Kubernetes Shutdown, Readiness/Liveness Probes & Testcontainers CI [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Graceful Kubernetes Shutdown, Readiness/Liveness Probes & Testcontainers CI in real Java Spring Boot Microservices & System Design systems — speci...

14.3 Enterprise Java Microservices System Design & Failure-Mode Walkthroughs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Enterprise Java Microservices System Design & Failure-Mode Walkthroughs in real Java Spring Boot Microservices & System Design systems — specifica...

Module 14 Hands-On Capstone Project (100 Marks): JFR Latency Profiler, Actuator Readiness Suite & Architecture Defense — Architect, implement, test, and document 'JFR Latency Profiler, Actuator Readiness Suite & Architecture Defense' applying every bullet-wise engineering concept from Module 14 (JVM Production...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: PayLedger Distributed FinTech Settlement & Event-Driven Order Engine • **Confirmed Role:** Java Spring Boot Microservices Engineering Intern

Domain & Production Infra: BFSI & High-Volume Payment Processing Client (Java 21 + Spring Boot 3 + Apache Kafka + Docker Production Cluster)

- Developed Java 21 virtual-thread Spring Boot 3 microservices with Spring Security 6 OAuth2 resource protection.
- Implemented Saga distributed transaction orchestration and idempotent Apache Kafka event consumers.
- Tuned PostgreSQL Hibernate JPA query plans, HikariCP connection pools, and Resilience4j circuit breakers.
- Deployed containerized microservices to live production with OpenTelemetry tracing and health probes.

Aligned Global Vendor Certifications: **2V0-72.22** (Spring Certified Professional (Spring Boot & Enterprise Java)) • **1Z0-830** (Oracle Certified Professional: Java SE 21 Developer) • **CCDAK** (Confluent Certified Developer for Apache Kafka (CCDAK)) • **DVA-C02** (AWS Certified Developer – Associate)

Target Job Roles: Java Microservices Backend Engineer, Software Development Engineer (Java System Design), Distributed Backend Systems Engineer, Enterprise Solutions Engineer

Realistic Annual CTC Scope: **Entry-Level Microservices SDE (0–1 Yr):** ₹6.5 LPA – ₹13.0 LPA **Mid-Level Java Microservices SDE-2 (1–3 Yrs):** ₹14.0 LPA – ₹28.0 LPA **Principal Java Architect / Staff Backend SDE (3–6+ Yrs):** ₹28.0 LPA – ₹55.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 98765 43210 • Web: https://mockattempt.com/courses/java-spring-boot-microservices-system-design

Try 2 Free Topic-Wise 50-MCQ Mock Tests & Spoken Gemini AI Interviews daily on every topic above before enrolling.