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 95400 07917
Course Code: JMSB • **Updated:** 05 Nov 2026
https://mockattempt.com/courses/java-spring-boot-microservices-system-design

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

Java Spring Boot Microservices & System Design

Start from Day-1 Core Java, OOP, Collections, SQL & JDBC and progress step-by-step to Java 21, Spring Boot 3, Microservices, Kafka, Docker, Kubernetes & System Design.

Duration: 20–22 Weeks • Live + Hands-On Projects (238 Hrs) • **Modules:** 16 • **Live Topics:** 50 • **Subtopics:** 400+ • **Labs & Tasks:** 100 • **Mocks / AI Vivas:** 100 / 100 • **Projects:** 16 Module + 1 Live Client Prod

All-Inclusive Direct Fee (Incl. 18% GST • Zero Hidden Charges): Total Payable: ₹6,499 (Net Tuition ₹5,508 + Included 18% GST ₹991) • **Split Option:** ₹1,999 Lock + 2 × ₹2,250 • **Effective ROI:** ~₹27/Guided Hr (~₹130/Live Topic)

Faculty: 100% Active Working IT Professionals • **Live Batch Tracks (7:00 AM – 8:00 PM IST Flexible Slots):** Weekday Live Online Batch (Mon – Fri • 1 Hr/Day • 7:00 AM – 8:00 PM) (Monday to Friday (1 Hour Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 1-Hour Slot Based on Trainer Availability) • 20–22 Weeks Fast-Track) | Weekend Live Online Batch (Sat – Sun • 2 Hrs/Day • 7:00 AM – 8:00 PM) (Saturday & Sunday (2 Hours Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 2-Hour Slot Based on Trainer Availability) • 22–24 Weeks Weekend Track)

✓ **Trainer LeetCode & HackerRank Mentorship (Core Java to Java 21 LTS & SQL): Live trainer support to crack LeetCode & HackerRank problems, optimize complexity, and handle edge cases.**

Core Technologies & Engineering Stack: Core Java Foundations • OOP & Collections • Java 21 LTS • Spring Boot 3 • Hibernate JPA • PostgreSQL • Apache Kafka • Docker & Kubernetes

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Master Core Java from scratch: syntax, control flow, arrays, strings, OOP, SOLID, exceptions, generics & Collections
2. Build production Spring Boot 3, Hibernate JPA, Spring Security 6 & OAuth2 resource servers
3. Design fault-tolerant Microservices, Saga distributed transactions & Apache Kafka consumers
4. Pass Core-to-Architect Java Mock Tests & Spoken MockAttempt AI System Design Interviews
5. Domain-Driven bounded contexts across Spring Boot 3 microservices with API Gateways and centralized configuration.
6. Crack LeetCode and HackerRank problems in Core Java to Java 21 LTS & SQL with live trainer mentorship on algorithmic/query patterns, edge cases, and optimal complexity.
7. Event-driven asynchronous choreography & Saga orchestration over Apache Kafka with idempotent consumers and Outbox pattern.
8. Resilience4j circuit breakers, bulkheads, rate limiters, retries, and Spring Security 6 OAuth2/JWT propagation.

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.

2. DETAILED ZERO-TO-EXPERT MODULE, TOPIC & SUBTOPIC SYLLABUS (16 MODULES • 50 LIVE TOPICS • 400+ SUBTOPICS)

Progressive Zero-to-Expert Mastery Stages: Modules 1–4: Stage 1: Day-1 Foundations & Core Mental Models (13 Topics / 104 Subtopics) • Modules 5–8: Stage 2: Applied Engineering, Data & API Workflows (13 Topics / 104 Subtopics) • Modules 9–12: Stage 3: Internals, Concurrency, Security & Performance (12 Topics / 96 Subtopics) • Modules 13–16: Stage 4: Production Architecture, Testing, CI/CD & Expert Mastery (12 Topics / 96 Subtopics)

MODULE 1 OF 16 • STAGE 1 • 20 LIVE + LAB HOURS • 1 TOPICS (8 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Core Java Foundations: JDK/JVM Setup, Syntax, Control Flow, Arrays & Strings

Bytecode compilation, JDK 21 setup, primitive data types, operators, loops, arrays, String pool, and modular methods from scratch.

Objectives: ✓ JDK / JRE / JVM Architecture, Primitive Data Types, Operators & Control Flow — How Java compiles source code to platform-independent bytecode and executes variables, operators, and loops. • ✓ Build, verify, and ship Module Capstone: CLI Banking Account & Transaction Validator in Pure Core Java.

1.1 JDK / JRE / JVM Architecture, Primitive Data Types, Operators & Control Flow

[60 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Starting Java From Scratch • Java compiles human • Security Hygiene & Data Safety Rules for JDK / JRE / JVM Architecture, Primitive Data Types, Operators & Control Fl... • Performance Optimization, Complexity & Latency Impact of JDK / JRE / JVM Architecture, Primitive Data Types, Operat... • Runnable Code Implementation (JAVA) • Verify java • Inspect compiled bytecode with javap • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j

Purpose: To understand and apply JDK / JRE / JVM Architecture, Primitive Data Types, Operators & Control Flow in real Java Spring Boot Microservices & System Design systems — spec...

What Topic Is Next → Module 1 Capstone (CLI Banking Account & Transaction Validator in Pure Core Java) → Next Topic 2.1: 1D / 2D Arrays, String Constant Pool, StringBuilder, Regex & Stack vs Heap Memory

Module 1 Hands-On Capstone Project (100 Marks): CLI Banking Account & Transaction Validator in Pure Core Java — Build a menu-driven Core Java CLI banking utility practicing primitive types, exact integer paise math, arrays, StringBuilder formatting, and modular methods.

MODULE 2 OF 16 • STAGE 1 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Arrays, Strings, Methods, Complete Object-Oriented Programming (OOP) & SOLID

Production-grade Java Spring Boot Microservices & System Design engineering module covering Arrays, Strings, Methods, Complete Object-Oriented Programming (OOP) & SOLID with hands-on labs and capstone build.

Objectives: ✓ 1D / 2D Arrays, String Constant Pool, StringBuilder, Regex & Stack vs Heap Memory — Purpose: Master Java array indexing, immutable Strings vs mutable StringBuilder, and reference memory lifecycles. • ✓ Classes, Constructors, Encapsulation, Access Modifiers & Packages — Purpose: Model real-world entities cleanly while protecting internal invariants from invalid state changes. • ✓ Inheritance, super, Method Overriding, Runtime Polymorphism & Abstract Classes — Purpose: Share common behavior responsibly and dispatch polymorphic business rules cleanly. • ✓ Interfaces, Default / Static Methods & All 5 SOLID Design Principles in Java — Purpose: Design loosely coupled, unit-testable service architectures ready for Spring Dependency Injection.

2.1 1D / 2D Arrays, String Constant Pool, StringBuilder, Regex & Stack vs Heap Memory

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: 1D & 2D Array Traversals, Binary Search & Matrix Operations • String Constant Pool, Immutability & StringBuilder Loop Efficiency • Stack Frames vs Heap Objects & Pass-by-Value Reference Semantics • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for 1D / 2D Arrays

Purpose: To understand and apply 1D / 2D Arrays, String Constant Pool, StringBuilder, Regex & Stack vs Heap Memory in real Java Spring Boot Microservices & System Design systems —...

What Topic Is Next → Next Topic 2.2: Classes, Constructors, Encapsulation, Access Modifiers & Packages

2.2 Classes, Constructors, Encapsulation, Access Modifiers & Packages

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Constructor Overloading, Chaining (this()) & Static Members • Encapsulation & Package-Private / Private Visibility Guards • Defensive Copying of Mutable Date / Collection Fields • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Classes

Purpose: To understand and apply Classes, Constructors, Encapsulation, Access Modifiers & Packages in real Java Spring Boot Microservices & System Design systems — specifically: M...

What Topic Is Next → Next Topic 2.3: Inheritance, super, Method Overriding, Runtime Polymorphism & Abstract Classes

2.3 Inheritance, super, Method Overriding, Runtime Polymorphism & Abstract Classes

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Method Overriding vs Overloading & Dynamic Method Dispatch • Abstract Classes & Template Method Workflows • Preventing Fragile Base-Class Bugs via Composition Over Inheritance • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Inheritance

Purpose: To understand and apply Inheritance, super, Method Overriding, Runtime Polymorphism & Abstract Classes in real Java Spring Boot Microservices & System Design systems — sp...

What Topic Is Next → Next Topic 2.4: Interfaces, Default / Static Methods & All 5 SOLID Design Principles in Java

2.4 Interfaces, Default / Static Methods & All 5 SOLID Design Principles in Java

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Single Responsibility & Open/Closed Principles • Liskov Substitution & Interface Segregation Principles • Dependency Inversion Principle (Constructor-Injected Interfaces) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary

Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Interfaces

Purpose: To understand and apply Interfaces, Default / Static Methods & All 5 SOLID Design Principles in Java in real Java Spring Boot Microservices & System Design systems — spec...

What Topic Is Next → Module 2 Capstone (Object-Oriented Multi-Account Banking Domain Engine in Core Java) → Next Topic 3.1: Checked vs Unchecked Exceptions, try-with-resources & Custom Business Exceptions

Module 2 Hands-On Capstone Project (100 Marks): Object-Oriented Multi-Account Banking Domain Engine in Core Java — Design, implement, test, and document 'Object-Oriented Multi-Account Banking Domain Engine in Core Java' applying every production concept from Module 2 (Arrays, Strings, Methods, Complete O...

MODULE 3 OF 16 • STAGE 1 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Exceptions, File I/O, Generics, Complete Collections Framework, SQL & JDBC Foundations

Production-grade Java Spring Boot Microservices & System Design engineering module covering Exceptions, File I/O, Generics, Complete Collections Framework, SQL & JDBC Foundations with hands-on labs and capstone build.

Objectives: ✓ Checked vs Unchecked Exceptions, try-with-resources & Custom Business Exceptions — Purpose: Handle I/O and validation failures cleanly with zero swallowed stack traces or leaked handles. • ✓ Java Generics & Complete Collections Framework (List, Set, Queue, Deque, Map) — Purpose: Master ArrayList, LinkedList, HashSet, TreeSet, PriorityQueue, ArrayDeque, and HashMap internals. • ✓ Relational Databases & SQL From Scratch: Normalization, JOINs, Aggregations & Indexes — Purpose: Write rock-solid SQL DDL, DML, multi-table JOINs, GROUP BY, CTEs, and ACID transactions. • ✓ JDBC From Scratch, HikariCP Connection Pooling, Maven Lifecycle & JUnit 5 TDD — Purpose: Connect Java directly to PostgreSQL safely via PreparedStatement and verify logic with JUnit 5.

3.1 Checked vs Unchecked Exceptions, try-with-resources & Custom Business Exceptions

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Throwable Hierarchy: Error vs Checked Exception vs RuntimeException • Deterministic Cleanup with AutoCloseable & try-with-resources • Rich Domain Exceptions with Error Codes & Context Metadata • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Checked vs Unchecked Exceptions

Purpose: To understand and apply Checked vs Unchecked Exceptions, try-with-resources & Custom Business Exceptions in real Java Spring Boot Microservices & System Design systems —...

What Topic Is Next → Next Topic 3.2: Java Generics & Complete Collections Framework (List, Set, Queue, Deque, Map)

3.2 Java Generics & Complete Collections Framework (List, Set, Queue, Deque, Map)

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: ArrayList vs LinkedList vs ArrayDeque Memory & CPU Cache Locality • HashMap Bucket Array, hashCode/equals Contract & Red-Black Treeify Threshold • Comparable vs Comparator, TreeMap/TreeSet & PriorityQueue Top-K • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Java Generics

Purpose: To understand and apply Java Generics & Complete Collections Framework (List, Set, Queue, Deque, Map) in real Java Spring Boot Microservices & System Design systems — spe...

What Topic Is Next → Next Topic 3.3: Relational Databases & SQL From Scratch: Normalization, JOINs, Aggregations & Indexes

3.3 Relational Databases & SQL From Scratch: Normalization, JOINs, Aggregations & Indexes

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: 1NF–3NF Schema Design, Primary/Foreign Keys & Check Constraints • INNER / LEFT JOINs, GROUP BY / HAVING & Window Functions • ACID Transactions & B-Tree Indexing Fundamentals • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Relational Databases

Purpose: To understand and apply Relational Databases & SQL From Scratch: Normalization, JOINs, Aggregations & Indexes in real Java Spring Boot Microservices & System Design syste...

What Topic Is Next → Next Topic 3.4: JDBC From Scratch, HikariCP Connection Pooling, Maven Lifecycle & JUnit 5 TDD

3.4 JDBC From Scratch, HikariCP Connection Pooling, Maven Lifecycle & JUnit 5 TDD

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: JDBC Connection, PreparedStatement Parameter Binding & Batch Execution • HikariCP Connection Pooling & Explicit Transaction Commit/Rollback • Maven Dependencies, Build Lifecycles & JUnit 5 + AssertJ Assertions • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for JDBC From Scratch

Purpose: To understand and apply JDBC From Scratch, HikariCP Connection Pooling, Maven Lifecycle & JUnit 5 TDD in real Java Spring Boot Microservices & System Design systems — spe...

What Topic Is Next → Module 3 Capstone (JDBC + PostgreSQL Reconciliation & Collections Indexing Engine) → Next Topic 4.1: Records, Sealed Interfaces & Pattern Matching for Switch

Module 3 Hands-On Capstone Project (100 Marks): JDBC + PostgreSQL Reconciliation & Collections Indexing Engine — Design, implement, test, and document 'JDBC + PostgreSQL Reconciliation & Collections Indexing Engine' applying every production concept from Module 3 (Exceptions, File I/O, Generics, Comple...

MODULE 4 OF 16 • STAGE 1 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 1 CAPSTONE PROJECT (100 MARKS)

Modern Java 8 to Java 21 LTS: Lambdas, Streams, Records & Sealed Hierarchies

Production-grade Java Spring Boot Microservices & System Design engineering module covering Modern Java 8 to Java 21 LTS: Lambdas, Streams, Records & Sealed Hierarchies 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.

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

Subtopics Covered: Java Records for Concise Immutable Value Objects • Sealed Classes/Interfaces & Exhaustive Pattern Matching • Optional<T> Pipelines vs NullPointerException • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Records

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

What Topic Is Next → Next Topic 4.2: Collections Internals, Bounded Generics & Custom Comparators

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

Subtopics Covered: HashMap Collision Handling, hashCode/equals Contracts & Load Factor • PECS Generics (? extends T / ? super T) • Immutable Collections (List.of, Map.copyOf) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Collections Internals

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

What Topic Is Next → Next Topic 4.3: Functional Interfaces, Stream Collectors & BigDecimal Monetary Math

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

Subtopics Covered: Stream groupingBy, partitioningBy & Downstream Collectors • Exact Monetary Arithmetic with BigDecimal & RoundingMode.HALF_EVEN • java.time (Instant, ZonedDateTime, Duration) Best Practices • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Functional Interfaces

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

What Topic Is Next → Next Topic 4.4: Domain Exception Hierarchies, NIO.2 File Streams & Clean SOLID Boundaries

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

Subtopics Covered: Unchecked Domain Exceptions vs Checked I/O Recovery • try-with-resources AutoCloseable Discipline & Files.lines() Streaming • Dependency Inversion & Constructor-Driven Testability • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Domain Exception Hierarchies

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...

What Topic Is Next → Module 4 Capstone (Immutable Financial Ledger & Functional Analytics Engine in Java 21) → Next Topic 5.1: JVM Heap Generations, Metaspace, G1GC/ZGC & Heap Dump Forensics

Module 4 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 4 (Modern Java 8 to Java 21 LTS: Lam...

MODULE 5 OF 16 • STAGE 2 • 16 LIVE + LAB HOURS • 4 TOPICS (32 SUBTOPICS) • 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.

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

Subtopics Covered: Eden, Survivor, Old Gen & Object Promotion Lifecycle • Choosing G1GC vs Low-Pause ZGC & Reading GC Logs • Capturing & Analyzing Heap/Thread Dumps (jcmd, Eclipse MAT) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for JVM Heap Generations

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...

What Topic Is Next → Next Topic 5.2: Java Memory Model, Locks, Atomic Operations & ConcurrentHashMap

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

Subtopics Covered: volatile Visibility, Happens-Before & AtomicLong/LongAdder • ConcurrentHashMap Compute Operations vs Synchronized Maps • ReentrantLock, ReadWriteLock & Deadlock Prevention • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Java Memory Model

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

What Topic Is Next → Next Topic 5.3: CompletableFuture Pipelines, ExecutorPools & Structured Concurrency

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

Subtopics Covered: Dedicated Bounded ThreadPoolExecutor vs Common ForkJoinPool • CompletableFuture.allOf, orTimeout & exceptionally • Java 21 StructuredTaskScope (Cancel-on-Failure) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for CompletableFuture Pipelines

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

What Topic Is Next → Next Topic 5.4: JMH Microbenchmarking, Lock-Free Queues & Thread-Pinning Diagnostics

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

Subtopics Covered: JMH Warmup/Fork Benchmarks vs Naive System.nanoTime() Myths • Replacing synchronized Blocks with ReentrantLock for Virtual Threads • Bounded BlockingQueue Backpressure & Load Shedding • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for JMH Microbenchmarking

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

What Topic Is Next → Module 5 Capstone (High-Throughput Concurrent Price Aggregator with Virtual Threads) → Next Topic 6.1: Spring IoC Container, Constructor Injection, Profiles & @ConfigurationProperties

Module 5 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 5 (JVM Memory Internals, Garbage Collec...

MODULE 6 OF 16 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Wire clean, testable Spring Boot 3 services with strongly typed externalized config. • ✓ RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail — Expose clean, self-documenting REST APIs with consistent validation error contracts. • ✓ Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown — Add non-invasive audit logging and Kubernetes-ready lifecycle hooks. • ✓ Build, verify, and ship Module Capstone: Enterprise Customer Onboarding REST Service with ProblemDetails.

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

Subtopics Covered: Constructor Injection & Bean Lifecycle Scopes • Strongly Typed @ConfigurationProperties with @Validated • Conditional Auto-Configuration & Multi-Environment Profiles • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Spring IoC Container

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

What Topic Is Next → Next Topic 6.2: RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail

6.2 RESTful Resource Design, Jakarta Bean Validation & RFC 9457 ProblemDetail

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Request/Response Record DTOs & Custom Jakarta Constraint Validators • Global @RestControllerAdvice & RFC 9457 ProblemDetail • Content Negotiation, ETag Conditional Caching & OpenAPI 3 Docs • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for RESTful Resource Design

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

What Topic Is Next → Next Topic 6.3: Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown

6.3 Spring AOP Cross-Cutting Auditing, Actuator Health Probes & Graceful Shutdown

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Spring AOP @Around Aspects for Execution Timing & Audit Trails • Spring Boot Actuator Liveness & Readiness Groups • Graceful Embedded Tomcat/Netty Shutdown • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Spring AOP Cross-Cutting Auditing

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...

What Topic Is Next → Module 6 Capstone (Enterprise Customer Onboarding REST Service with ProblemDetails) → Next Topic 7.1: Hibernate Persistence Context, Dirty Checking & Entity Lifecycle States

Module 6 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 6 (Spring Boot 3 Core IoC, REST API Desi...

MODULE 7 OF 16 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Master Transient, Managed, Detached, and Removed states so Hibernate never surprises you. • ✓ Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections — Reduce 201 SQL queries per API call down to 1–2 indexed queries. • ✓ Optimistic (@Version) vs Pessimistic Locking & JDBC Batch Inserts — Prevent double-spending wallet balances and bulk-insert thousands of rows rapidly. • ✓ Build, verify, and ship Module Capstone: High-Performance Order & Catalog Persistence Layer.

7.1 Hibernate Persistence Context, Dirty Checking & Entity Lifecycle States

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: First-Level Session Cache, Flush Modes & Automatic Dirty Checking • FetchType.LAZY Everywhere by Default • MappedSuperclass Audit Columns & UUIDv7 Time-Sorted Keys • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Hibernate Persistence Context

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

What Topic Is Next → Next Topic 7.2: Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections

7.2 Solving N+1 Select Storms: @EntityGraph, JOIN FETCH & DTO Projections

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Detecting N+1 Queries in Integration Tests & SQL Logs • Interface/Record DTO Projections for Read-Only Endpoints • Dynamic Filtering with JPA Specifications & Keyset Pagination • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Solving N+1 Select Storms

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...

What Topic Is Next → Next Topic 7.3: Optimistic (@Version) vs Pessimistic Locking & JDBC Batch Inserts

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

Subtopics Covered: Optimistic Concurrency Control with @Version & Retry Policies • SELECT ... FOR UPDATE SKIP LOCKED Job Claiming • Hibernate JDBC Batching & HikariCP Pool Sizing Formula • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Optimistic (@Version) vs Pessimistic Locking

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

What Topic Is Next → Module 7 Capstone (High-Performance Order & Catalog Persistence Layer) → Next Topic 8.1: MVCC Internals, Transaction Isolation Levels & Deadlock Resolution

Module 7 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 7 (Spring Data JPA, Hibernate Internals, N+1 Eliminat...

MODULE 8 OF 16 • STAGE 2 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Choose Read Committed vs Repeatable Read vs Serializable for financial correctness. • ✓ B-Tree, GIN, Partial Indexes & EXPLAIN (ANALYZE, BUFFERS) Tuning — Read PostgreSQL query execution plans and optimize multi-million-row tables. • ✓ Versioned Flyway Migrations & Expand-Contract Zero-Downtime Rollouts — Evolve live production schemas while old and new microservice pods run side-by-side. • ✓ Build, verify, and ship Module Capstone: Zero-Downtime Schema Evolution & Ledger Database Architecture.

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

Subtopics Covered: PostgreSQL MVCC Snapshots, Write Skew & Phantom Reads • Consistent Lock Ordering to Eliminate Deadlocks • Spring @Transactional Propagation (REQUIRED vs REQUIRES_NEW) & Rollback Rules • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for MVCC Internals

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

What Topic Is Next → Next Topic 8.2: B-Tree, GIN, Partial Indexes & EXPLAIN (ANALYZE, BUFFERS) Tuning

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

Subtopics Covered: Composite B-Tree Column Ordering & Index-Only Scans (INCLUDE) • GIN Indexes on JSONB Attributes & Partitioned Audit Tables • Interpreting Seq Scan vs Bitmap Heap Scan & Shared Buffers • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for B-Tree

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...

What Topic Is Next → Next Topic 8.3: Versioned Flyway Migrations & Expand-Contract Zero-Downtime Rollouts

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

Subtopics Covered: Immutable Versioned SQL Migrations (V1__, V2__) & Checksums • Expand–Migrate–Contract Pattern for Column Renames/Drops • Read Replicas & Read/Write Routing DataSources • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Versioned Flyway Migrations

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

What Topic Is Next → Module 8 Capstone (Zero-Downtime Schema Evolution & Ledger Database Architecture) → Next Topic 9.1: Spring Security 6 FilterChain Architecture & Password Hashing

Module 8 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 8 (PostgreSQL Production Engineering, ACID...

MODULE 9 OF 16 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Configure modern lambda DSL security filter chains with zero deprecated WebSecurityConfigurerAdapter code. • ✓ OAuth2 Resource Servers, RSA/JWKS JWT Verification & Token Rotation — Validate asymmetric JWTs across 15+ microservices without hitting a central DB per request. • ✓ Fine-Grained @PreAuthorize Method Security & Service-to-Service Auth — Enforce role/ownership permissions and mTLS/client-credentials between internal services. • ✓ Build, verify, and ship Module Capstone: Centralized Identity & OAuth2 Resource Server Security Mesh.

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

Subtopics Covered: SecurityFilterChain Order, Stateless SessionPolicy & CSRF Strategy • Argon2id / BCrypt Password Encoding & Account Lockout Policies • Custom AuthenticationEntryPoint & AccessDeniedHandler • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Spring Security 6 FilterChain Architecture

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

What Topic Is Next → Next Topic 9.2: OAuth2 Resource Servers, RSA/JWKS JWT Verification & Token Rotation

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

Subtopics Covered: RS256/ES256 Asymmetric Signing & Public JWKS Endpoint Caching • Refresh Token Rotation & Reuse Detection • Propagating Tenant & Scope Claims into SecurityContext • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for OAuth2 Resource Servers

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

What Topic Is Next → Next Topic 9.3: Fine-Grained @PreAuthorize Method Security & Service-to-Service Auth

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

Subtopics Covered: Method Security Expressions (@PreAuthorize, @PostAuthorize) • OAuth2 Client Credentials Grant for Internal Microservices • Audit Logging Security Principal & Sensitive Data Masking • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Fine-Grained @PreAuthorize Method Security

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

What Topic Is Next → Module 9 Capstone (Centralized Identity & OAuth2 Resource Server Security Mesh) → Next Topic 10.1: Domain-Driven Bounded Contexts & Database-Per-Service Decomposition

Module 9 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 9 (Spring Security 6, OAuth2/OIDC, Stateless...

MODULE 10 OF 16 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Split monoliths along true business capabilities instead of creating a distributed monolith. • ✓ Spring Cloud Gateway: Reactive Routing, Redis Rate Limiting & Auth Offload — Centralize TLS, JWT verification, and per-tenant throttling at the API edge. • ✓ Declarative HTTP Clients (@HttpExchange), gRPC & Timeout Budgets — Communicate synchronously between services safely with strict connect/read deadlines. • ✓ Build, verify, and ship Module Capstone: Edge API Gateway with Rate Limiting, Routing & BFF Aggregation.

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

Subtopics Covered: Bounded Contexts, Ubiquitous Language & Context Mapping • Anti-Corruption Layers (ACL) Around Legacy External Systems • Contract-First API Versioning & Consumer-Driven Contracts • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Domain-Driven Bounded Contexts

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

What Topic Is Next → Next Topic 10.2: Spring Cloud Gateway: Reactive Routing, Redis Rate Limiting & Auth Offload

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

Subtopics Covered: Non-Blocking Netty Route Predicates & Header Rewrite Filters • Distributed Token-Bucket Rate Limiting with Redis • BFF (Backend-for-Frontend) Response Composition • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring

Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Spring Cloud Gateway

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...

What Topic Is Next → Next Topic 10.3: Declarative HTTP Clients (@HttpExchange), gRPC & Timeout Budgets

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

Subtopics Covered: Spring 6 Declarative @HttpExchange & RestClient • High-Performance Internal RPC with Protocol Buffers & gRPC • Deadline Propagation & Preventing Retry Amplification Storms • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Declarative HTTP Clients (@HttpExchange)

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

What Topic Is Next → Module 10 Capstone (Edge API Gateway with Rate Limiting, Routing & BFF Aggregation) → Next Topic 11.1: Circuit Breakers, Bulkhead Isolation & Exponential Jitter Retries

Module 10 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 10 (Microservices Boundaries, Spring Clou...

MODULE 11 OF 16 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Prevent one degraded downstream gateway from cascading across your entire microservice fleet. • ✓ Multi-Tier Caching (Caffeine L1 + Redis L2) & Graceful Degradation — Serve catalog and FX-rate reads from memory even during partial database maintenance. • ✓ Distributed Idempotency, ShedLock Scheduled Jobs & Chaos Fault Injection — Ensure scheduled settlements and retried requests execute safely across N replicas. • ✓ Build, verify, and ship Module Capstone: Self-Healing Checkout Orchestrator with Circuit Breakers & Fallbacks.

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

Subtopics Covered: Resilience4j CircuitBreaker State Machine (CLOSED -> OPEN -> HALF_OPEN) • Semaphore & ThreadPool Bulkheads • Bounded Retries with Exponential Backoff & Random Jitter • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Circuit Breakers

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

What Topic Is Next → Next Topic 11.2: Multi-Tier Caching (Caffeine L1 + Redis L2) & Graceful Degradation

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

Subtopics Covered: Near-Cache Caffeine L1 + Distributed Redis L2 Topology • Stale-If-Error Fallback Responses • Deterministic Cache Key Versioning & Pub/Sub Invalidation • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Multi-Tier Caching (Caffeine L1 + Redis L2)

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

What Topic Is Next → Next Topic 11.3: Distributed Idempotency, ShedLock Scheduled Jobs & Chaos Fault Injection

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

Subtopics Covered: Database/Redis Backed Idempotency Filter for Mutating Endpoints • Cluster-Safe Cron Coordination with ShedLock • Injecting Latency & Faults (Toxiproxy / Chaos Monkey) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Distributed Idempotency

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

What Topic Is Next → Module 11 Capstone (Self-Healing Checkout Orchestrator with Circuit Breakers & Fallbacks) → Next Topic 12.1: Kafka Partitions, Consumer Groups, Acks=all & Schema Registry

Module 11 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 11 (Fault-Tolerant Distributed Syst...

MODULE 12 OF 16 • STAGE 3 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Stream millions of ordered domain events reliably between decoupled Spring Boot services. • ✓ Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers — Eliminate dual-write bugs between PostgreSQL commits and Kafka publishes. • ✓ Distributed Sagas: Choreography vs Orchestration & Compensating Actions — Maintain cross-service data consistency without brittle two-phase XA distributed locks. • ✓ Build, verify, and ship Module Capstone: Distributed Order–Inventory–Payment Saga over Apache Kafka.

12.1 Kafka Partitions, Consumer Groups, Acks=all & Schema Registry

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Partition Keys (orderId / accountId) for Strict Per-Entity Ordering • Producer Durability (acks=all, enable.idempotence=true, min.insync.replicas) • Avro / JSON Schema Evolution & Backward Compatibility • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Kafka Partitions

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

What Topic Is Next → Next Topic 12.2: Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers

12.2 Transactional Outbox Pattern, Debezium CDC & Idempotent Consumers

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Why Saving to DB + Calling kafkaTemplate.send() Directly Loses Data • Transactional Outbox Table + Relay / Debezium CDC • Processed-Event Deduplication Table & Dead-Letter Topics (DLT) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Transactional Outbox Pattern

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

What Topic Is Next → Next Topic 12.3: Distributed Sagas: Choreography vs Orchestration & Compensating Actions

12.3 Distributed Sagas: Choreography vs Orchestration & Compensating Actions

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: Orchestrated Saga State Machine for Multi-Step Checkout • Compensating Transactions (RefundPayment, ReleaseStock) • Semantic Locks (PENDING_APPROVAL) & Pivot Transactions • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Distributed Sagas

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

What Topic Is Next → Module 12 Capstone (Distributed Order–Inventory–Payment Saga over Apache Kafka) → Next Topic 13.1: Distributed Tracing with OpenTelemetry across HTTP & Kafka Hops

Module 12 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 12 (Event-Driven Microservices with Apache Ka...

MODULE 13 OF 16 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Follow a single slow request across Gateway -> Order -> Kafka -> Payment in one trace flamegraph. • ✓ Micrometer Dimensional Metrics, Prometheus Histograms & Alerting SLOs — Measure latency percentiles (p50/p95/p99), error rates, and JVM saturation accurately. • ✓ Structured JSON Log Aggregation (Loki/ELK) & Incident Runbooks — Query terabytes of multi-pod logs by traceId and tenantId in seconds. • ✓ Build, verify, and ship Module Capstone: Full-Stack Golden-Signals Observability & Distributed Trace Suite.

13.1 Distributed Tracing with OpenTelemetry across HTTP & Kafka Hops

[65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Subtopics Covered: W3C traceparent Context Propagation over HTTP & Kafka Headers • Custom Business Spans & Database Query Span Tagging • Tail-Based vs Head-Based Trace Sampling • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva

Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Distributed Tracing with OpenTelemetry across HTTP

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

What Topic Is Next → Next Topic 13.2: Micrometer Dimensional Metrics, Prometheus Histograms & Alerting SLOs

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

Subtopics Covered: Counters, Gauges, Timers & Histogram Buckets • Avoiding High-Cardinality Tag Explosions (Never Tag userId!) • Defining Actionable SLO Burn-Rate Alerts • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Micrometer Dimensional Metrics

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

What Topic Is Next → Next Topic 13.3: Structured JSON Log Aggregation (Loki/ELK) & Incident Runbooks

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

Subtopics Covered: Logback JSON Encoders with Automatic traceId/spanId Injection • Masking PAN, Aadhaar, Tokens & Secrets at Logger Level • Post-Incident RCA Templates & Load-Shedding Playbooks • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Structured JSON Log Aggregation (Loki/ELK)

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

What Topic Is Next → Module 13 Capstone (Full-Stack Golden-Signals Observability & Distributed Trace Suite) → Next Topic 14.1: JUnit 5, Mockito, @WebMvcTest Slices & Testcontainers Real DB/Kafka Suites

Module 13 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 13 (Observability: OpenTelemetry Traci...

MODULE 14 OF 16 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 — Test Spring Boot microservices against real PostgreSQL, Redis, and Kafka containers in CI. • ✓ Spring Boot Layered JARS, Distroless Containers & JVM Container Ergonomics — Build fast-booting, security-hardened container images tailored for cloud orchestration. • ✓ Kubernetes Deployments, HPA Autoscaling, Secrets & Zero-Downtime Rollouts — Deploy resilient Spring Boot microservice fleets onto Kubernetes clusters. • ✓ Build, verify, and ship Module Capstone: Production Kubernetes Deployment with Testcontainers CI Gate.

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

Subtopics Covered: Fast Domain Unit Tests vs Narrow @WebMvcTest Controller Slices • @ServiceConnection Testcontainers for PostgreSQL & Apache Kafka • WireMock Stubs for Deterministic Third-Party Gateway Testing • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for JUnit 5

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...

What Topic Is Next → Next Topic 14.2: Spring Boot Layered JARS, Distroless Containers & JVM Container Ergonomics

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

Subtopics Covered: Spring Boot Layered Jartools Extraction in Multi-Stage Dockerfiles • CDS / AOT Class Data Sharing & Virtual Thread Readiness • Container-Aware Heap Sizing (-XX:MaxRAMPercentage) • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Spring Boot Layered JARS

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

What Topic Is Next → Next Topic 14.3: Kubernetes Deployments, HPA Autoscaling, Secrets & Zero-Downtime Rollouts

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

Subtopics Covered: Deployment Manifests, preStop Lifecycle Hooks & PodDisruptionBudgets • Startup, Liveness & Readiness Probes Mapped to Actuator • HorizontalPodAutoscaler (CPU/Custom Lag Metrics) & Canary Releases • Under-the-Hood Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle • Boundary Validation, Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10 & Failure Recovery • Automated JUnit 5, MockMvc & PostgreSQL Testcontainers Verification & Senior Viva Trade-Offs • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Kubernetes Deployments

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

What Topic Is Next → Module 14 Capstone (Production Kubernetes Deployment with Testcontainers CI Gate) → Next Topic 15.1: Transactional Outbox Pattern with Spring Data JPA & Debezium/Polling Relay

Module 14 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 14 (Testcontainers Integration Suites, Laye...

MODULE 15 OF 16 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 MockAttempt 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 + MockAttempt AI Technical Interviews.

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

Subtopics Covered: Security Hygiene & Data Safety Rules for Transactional Outbox Pattern with Spring Data JPA & Debezium/Polling Relay • Performance Optimization, Complexity & Latency Impact of Transactional Outbox Pattern with Spring Data JPA & Debezi... • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Transactional Outbox Pattern with Spring Data JPA • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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...

What Topic Is Next → Next Topic 15.2: Idempotent Kafka Consumers, Dead-Letter Topics & Poison-Pill Recovery

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

Subtopics Covered: Security Hygiene & Data Safety Rules for Idempotent Kafka Consumers, Dead • Performance Optimization, Complexity & Latency Impact of Idempotent Kafka Consumers, Dead • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Idempotent Kafka Consumers • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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

What Topic Is Next → Next Topic 15.3: Optimistic/Pessimistic JPA Locking & Redis Distributed Mutexes

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

Subtopics Covered: Security Hygiene & Data Safety Rules for Optimistic/Pessimistic JPA Locking & Redis Distributed Mutexes • Performance Optimization, Complexity & Latency Impact of Optimistic/Pessimistic JPA Locking & Redis Distributed Mut... • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Optimistic/Pessimistic JPA Locking • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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

What Topic Is Next → Module 15 Capstone (Zero-Data-Loss Outbox Relay & Redis Distributed Settlement Guard) → Next Topic 16.1: JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause Tuning in Java 21

Module 15 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 15 (Transactional Outbo...

MODULE 16 OF 16 • STAGE 4 • 14 LIVE + LAB HOURS • 3 TOPICS (24 SUBTOPICS) • 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 MockAttempt 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 + MockAttempt AI Technical Interviews.

16.1 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]

Subtopics Covered: Security Hygiene & Data Safety Rules for JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause Tuning in Java 21 • Performance Optimization, Complexity & Latency Impact of JDK Flight Recorder (JFR), Heap Dump Analysis & GC Pause T... • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for JDK Flight Recorder (JFR) • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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...

What Topic Is Next → Next Topic 16.2: Graceful Kubernetes Shutdown, Readiness/Liveness Probes & Testcontainers CI

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

Subtopics Covered: Security Hygiene & Data Safety Rules for Graceful Kubernetes Shutdown, Readiness/Liveness Probes & Testcontainers C... • Performance Optimization, Complexity & Latency Impact of Graceful Kubernetes Shutdown, Readiness/Liveness Probes &.... • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Graceful Kubernetes Shutdown • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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

What Topic Is Next → Next Topic 16.3: Enterprise Java Microservices System Design & Failure-Mode Walkthroughs

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

Subtopics Covered: Security Hygiene & Data Safety Rules for Enterprise Java Microservices System Design & Failure • Performance Optimization, Complexity & Latency Impact of Enterprise Java Microservices System Design & Failure • Live Debugging, Stack • Runnable Code Implementation (JAVA) • Hands • Idiomatic JMSB Implementation: constructor DI, Record DTOs, @Transactional boundaries & Resilience4j • Java 21 + Spring Boot 3 IoC & Hibernate 6 Persistence Context Execution & Memory Lifecycle for Enterprise Java Microservices System Design • Production Hardening & Security: Spring Security 6 JWT/OAuth2 RBAC & OWASP API Top-10

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

What Topic Is Next → Module 16 Capstone → End-of-Course Live Client Project: PayLedger Distributed FinTech Settlement & Event-Driven Order Engine

Module 16 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 16 (JVM Production...

3. LIVE CLIENT PRODUCTION PROJECT & 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.

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