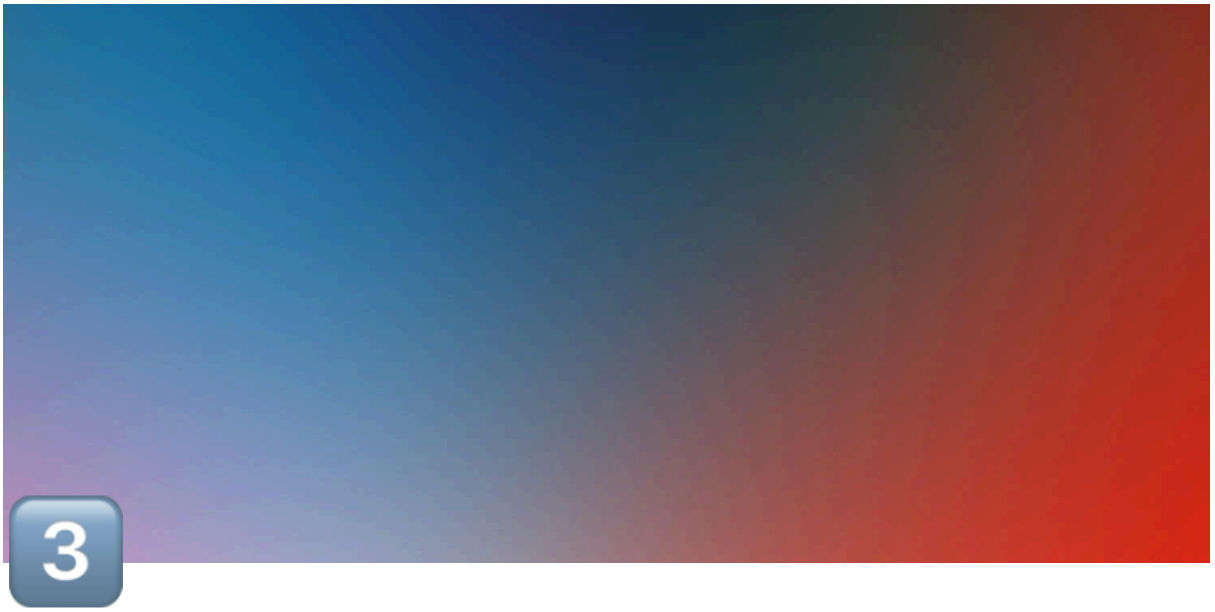

# ☕ Phase 3 — Backend with Java (Spring Boot)

**Duration:** 4 Months (±16 Weeks)

### 🎯 Goal

Master **enterprise-grade backend development** using **Java + Spring Boot**, clean architecture, JPA/Hibernate, security, testing, and build a **production-ready E-commerce Backend**.

This phase focuses on **scalability, structure, and robustness** — how real backend systems are built in companies.

---

## 🗓️ Roadmap Phase 3 — Weekly Breakdown

## 📌 Month 1 — Java & Spring Boot Foundations

### 🗓️ Week 1: Java for Backend Engineers 🧠

1. Review **core Java concepts**:
    - OOP principles (encapsulation, inheritance, polymorphism)
    - Interfaces vs abstract classes

- Exception handling (`try/catch`)

2. Learn **Java collections**:
   - `List`, `Set`, `Map`
3. Understand **immutability & best practices**.
4. Learn **Java project structure** for backend systems.
5. Write clean, readable Java code.

---

## 🗓️ Week 2: Spring Boot Fundamentals 🌱

1. Understand **Spring ecosystem**:
   - Spring Boot vs Spring Framework
2. Learn Spring Boot project structure.
3. Learn:
   - `@RestController`
   - `@RequestMapping`
   - `@GetMapping`, `@PostMapping`, etc.
4. Understand:
   - Dependency Injection (DI)
   - Inversion of Control (IoC)
5. Build a simple REST API endpoint.

---

## 🗓️ Week 3: Layered Architecture & Design 🏗️

1. Learn **backend layering**:
   - Controller
   - Service
   - Repository
2. Understand why business logic **must not** be in controllers.
3. Learn:
   - DTO (Data Transfer Object)

- Mapper (Entity ↔ DTO)

4. Structure code for **maintainability**.
5. Refactor API using layered architecture.

---

## 🗓️ Week 4: JPA & Hibernate ORM 🗄️

1. Understand ORM concepts deeply.
2. Learn JPA annotations:
   - `@Entity`
   - `@Id`
   - `@GeneratedValue`
3. Relationships:
   - One-to-Many
   - Many-to-Many
4. Learn:
   - Lazy vs eager loading
   - N+1 query problem
5. Connect Spring Boot to PostgreSQL.

---

# 📌 Month 2 — Database, Validation & Error Handling

## 🗓️ Week 5: Advanced JPA & Database Design ⚙️

1. Write complex queries:
   - JPQL
   - Native queries
2. Pagination & sorting with Spring Data.
3. Learn indexing strategy.
4. Handle transactions:
   - `@Transactional`
5. Optimize query performance.

## 🗓️ Week 6: Validation & Error Handling 🧯

1. Learn request validation:
   - `@NotNull`, `@Size`, `@Email`
2. Handle validation errors cleanly.
3. Create **global exception handler**:
   - `@ControllerAdvice`
4. Design **consistent error response format**.
5. Improve API reliability.

---

## 🗓️ Week 7: Authentication & Authorization 🔐

1. Implement **JWT authentication** in Spring Security.
2. Understand:
   - Authentication filter
   - Authorization filter
3. Role-based access control:
   - `ADMIN`
   - `USER`
4. Protect routes properly.
5. Secure password storage.

---

## 🗓️ Week 8: Security Best Practices 🔒

1. Learn common backend attacks:
   - SQL Injection
   - XSS
   - CSRF
2. Configure:
   - CORS
   - CSRF protection

3. Implement rate limiting (basic).
4. Secure application configuration.
5. Review security checklist.

---

# 📌 Month 3 — Project: E-commerce Backend 🛒

## 🗓️ Week 9: Project Setup & Design 📐

1. Create Spring Boot project repository.
2. Design ERD for:
    - User
    - Product
    - Category
    - Cart
    - Order
    - Order Item
3. Define API contracts.
4. Setup database migrations.
5. Initialize documentation.

---

## 🗓️ Week 10: Core Business Logic 💼

1. Implement User CRUD.
2. Implement Product & Category CRUD.
3. Implement cart logic:
    - Add item
    - Remove item
4. Implement checkout logic.
5. Handle business rules carefully.

---

## 🗓️ Week 11: Order & History System 📦

1. Create order processing flow.
2. Save order history.
3. Ensure data consistency using transactions.
4. Handle edge cases:
    - Empty cart
    - Invalid product
5. Test business workflows.

---

### 🗓️ Week 12: API Quality & Performance 🚀

1. Add pagination to product listing.
2. Implement filtering & sorting.
3. Improve API response design.
4. Optimize slow queries.
5. Refactor code for clarity.

---

## 📌 Month 4 — Testing, Documentation & Readiness

### 🗓️ Week 13: Backend Testing 🧪

1. Unit testing:
    - Service layer
2. Integration testing:
    - Repository & controller
3. Mock dependencies using Mockito.
4. Test authentication & authorization.
5. Ensure test reliability.

---

### 🗓️ Week 14: Documentation & Standards 📘

1. Write API documentation:
    - Swagger / OpenAPI

2. Improve README:
    - Setup
    - Architecture
3. Document database schema.
4. Prepare demo scenarios.

---

## 🗓️ Week 15: Refactoring & Best Practices ✨

1. Refactor duplicated code.
2. Improve naming & readability.
3. Enforce clean architecture rules.
4. Review logs & error messages.
5. Polish project for presentation.

---

## 🗓️ Week 16: Final Review & Evaluation 🏁

1. Full system testing.
2. Review security & performance.
3. Validate business logic.
4. Prepare project explanation:
    - Architecture
    - Design decisions
5. Final submission.

---

# 🧠 Phase 3 Summary ✨

- **Month 1:** Java + Spring Boot + JPA foundations
- **Month 2:** Security, validation, transactions, auth
- **Month 3:** Full E-commerce backend implementation
- **Month 4:** Testing, documentation, production readiness

---

## ✅ Phase 3 Exit Criteria (Very Important)

You are **Phase-3 complete** if you can:

- Design enterprise backend architecture
- Write clean Spring Boot APIs
- Implement secure JWT authentication
- Handle real business logic
- Optimize database performance
- Explain *why* your design works

---

## 🔥 After Phase 3, You Are:

- **Strong Backend Engineer (Mid-level)**
- Ready for:
  - Advanced backend concepts (Phase 4)
  - DevOps & deployment
  - Large-scale systems