Enterprise Full Stack Development with Java, Spring Boot and React

Duration: 10 Days | 80 Hours

Course Overview

This intensive 10-day, 80-hour program equips developers with the end-to-end skills needed to design, build, secure, and deploy enterprise-grade full stack applications using Java, Spring Boot, and React. Participants progress from Java and web fundamentals through React with TypeScript, Spring Boot REST services, PostgreSQL data engineering, and secure frontend-backend integration. They then move on to scalable architecture, observability, and DevOps practices including CI/CD, Docker, and Kubernetes. Throughout the course, hands-on labs build toward a capstone Enterprise Customer Onboarding Platform, which is delivered with automated tests, monitoring, documentation, and a deployable pipeline.

Prerequisites

- Basic programming knowledge in any language, including variables, loops, conditionals, and functions.
- Familiarity with fundamental web concepts such as browsers, URLs, and client-server interaction.
- Basic understanding of databases and SQL concepts (tables, rows, simple queries) is helpful but not mandatory.
- Comfort using a computer's command line and installing development tools such as an IDE and Git.
- A laptop with admin rights to install the JDK, Node.js, Docker, and related tools, plus a willingness to complete hands-on labs and the capstone project.

Learning Outcomes

- Build robust Java applications using OOP, Collections, Streams, and clean code principles, and develop responsive, accessible frontends with HTML, CSS, JavaScript, TypeScript, and React (Hooks, Router, TanStack Query).
- Design and implement RESTful backend services with Spring Boot, including validation, exception handling, OpenAPI documentation, and externalized configuration.
- Model and manage relational data with PostgreSQL, Spring Data JPA/Hibernate, and Liquibase migrations, applying indexing, transactions, and query optimization.
- Secure full stack applications using Spring Security, JWT, role-based access control, and OWASP best practices while integrating React with Spring Boot end to end.
- Apply enterprise architecture, caching (Redis), and observability (Prometheus, Grafana), then automate build, test, containerization (Docker/Kubernetes), and CI/CD deployment with rollback strategies.

Course Agenda

Module 1: Java Programming Fundamentals and Development Essentials

- Object-oriented programming: Classes, objects, inheritance, polymorphism and abstraction
- Interfaces, packages and access modifiers
- Exception handling and debugging techniques
- Collections Framework and Generics
- Lambda expressions and Streams API
- Maven fundamentals and dependency management
- file I/O, clean code, SOLID and basic data structures
- Java development environment: JDK, VSCode, terminal, JShell, debugging breakpoints and Git basics.
- Memory model, stack/heap, equals/hashCode, immutability, Optional, date/time, exceptions, records and concurrency basics.
- Hands-on: Develop Java-based business applications

Module 2: Frontend Development with HTML, CSS, JavaScript and TypeScript

- Web application architecture and client-server communication
- HTML5 elements, forms and page structures
- CSS3, responsive design, Flexbox and layouts
- JavaScript fundamentals, functions, objects and arrays
- DOM manipulation and event handling
- TypeScript fundamentals, types, interfaces and classes
- Modules, asynchronous programming and Promises
- HTTP/HTTPS, DNS, browser network tools, status codes, headers, cookies, CORS and JSON.
- JavaScript: closures, scope, event loop, async/await, error handling, ES modules and npm tooling.
- Accessibility, semantic HTML, CSS Grid, responsive breakpoints and browser compatibility.
- Hands-on: Develop a responsive frontend interface using TypeScript

Module 3: Frontend Application Development with React

- React fundamentals and component-based architecture
- JSX, components, props and state
- React Hooks and state management
- Event handling, forms and validations
- React Router and application navigation
- API integration using Fetch / Axios
- Authentication flows and protected routes
- Frontend debugging and development best practices
- React: useEffect dependencies, custom hooks, context, controlled forms and component lifecycle.
- React Query/TanStack Query, loading/error/empty states, pagination, reusable components and API error handling.
- Vite, ESLint, React Testing Library, Vitest, accessibility and performance fundamentals.
- Hands-on: Build a React-based customer onboarding interface

Module 4: Backend Development with Java and Spring Boot

- Introduction to Spring Framework and Spring Boot architecture
- Creating Spring Boot projects using Spring Initializr
- Understanding controllers, services and repositories
- Dependency Injection and Inversion of Control
- Building REST APIs using Spring Web
- Request mapping, HTTP methods and response handling
- Data validation and exception handling
- Configuration management and application properties
- Logging and debugging Spring Boot applications
- Spring Boot: DTOs, layered design, Bean Validation, global exception handling, profiles and externalized configuration.
- REST design: OpenAPI/Swagger, pagination, filtering, idempotency, versioning, status codes and API contracts.

- Spring Boot Actuator, structured logging, resilience/timeouts/retries and HTTP client integrations.
- Hands-on: Build backend services for the customer onboarding application

Module 5: Database Development and Data Engineering Fundamentals

- Relational database concepts and PostgreSQL architecture
- Database design, schemas, tables and relationships
- SQL queries, joins, subqueries and aggregate functions
- Indexing, constraints and normalization
- Transactions and ACID properties
- Spring Data JPA and Hibernate ORM
- Entity relationships and database migrations
- Database connection pooling and query optimization
- Data validation, consistency and integrity
- Introduction to data pipelines, ETL/ELT and data integration
- SQL: explain plans, N+1 queries, lazy/eager loading, optimistic locking and transaction boundaries.
- Liquibase migrations, Testcontainers, local database setup, seed data and backup/restore basics.
- Hands-on: Design and implement the application database and data layer

Module 6: Full Stack Integration and Application Security

- Integrating React frontend with Spring Boot backend
- Designing and consuming REST APIs
- Implementing CRUD operations
- JSON serialization and deserialization
- Authentication and authorization fundamentals
- Spring Security and role-based access control
- JWT-based authentication
- CORS, secure API communication and input validation

- Environment-based configuration
- Security: OAuth 2.0/OIDC concepts, password hashing, session vs token tradeoffs, CSRF, XSS, SQL injection and OWASP Top 10.
- End-to-end authentication, authorization enforcement on backend, secure secrets, HTTPS and secure error handling.
- API contract testing (optional), frontend/backend environment configuration and browser devtools network troubleshooting.
- Hands-on: Complete end-to-end customer onboarding workflow

Module 7: Enterprise Architecture, Scalability and Observability

- Monolithic versus microservices architecture
- Layered architecture and separation of concerns
- Designing scalable and maintainable applications
- Stateless applications and session management
- Application performance and database optimization
- Caching fundamentals and Redis
- Load balancing and horizontal scaling concepts
- Application logging, metrics and distributed tracing
- Observability using Prometheus and Grafana
- Managing application and dependency version upgrades
- System design fundamentals: requirements, tradeoffs, API/data modeling, latency, throughput and bottleneck identification.
- Async messaging and queues, eventual consistency, rate limiting, resilience, caching invalidation and basic threat modeling.
- Hands-on: Enhance application architecture for scalability and monitoring

Module 8: DevOps, CI/CD and Enterprise Deployment

- Git fundamentals and collaborative development workflows
- Branching strategies: Main, feature, release and hotfix branches
- Pull requests, code reviews and merge strategies
- Build automation using Maven

- CI/CD pipeline development using Jenkins or GitHub Actions
- Managing development, QA, UAT and production environments
- Environment-specific configuration management
- Secrets management and secure credential handling
- Docker images, containers and Docker Compose
- Kubernetes fundamentals: Pods, deployments, services and namespaces
- Resource requests, limits and horizontal pod autoscaling
- Deployment strategies: Rolling updates and rollback
- Linux shell, environment variables, process/network troubleshooting, Git conflict resolution and pull-request collaboration.
- Cloud deployment fundamentals (AWS or Azure): compute, managed PostgreSQL, object storage, IAM, networking and TLS.
- CI/CD: lint, tests, dependency scanning, build artifacts, image scanning, deployments, health checks and rollback.
- Hands-on: Build an automated CI/CD pipeline and deploy the application

Module 9: Capstone Project – Enterprise Customer Onboarding Platform

- Requirement analysis and solution architecture
- Designing application components and data models
- Building React and TypeScript frontend interfaces
- Developing backend services using Spring Boot
- Integrating PostgreSQL and REST APIs
- Implementing customer data validation and business rules
- Implementing authentication and role-based access
- Writing unit, integration and API tests
- Containerizing the application using Docker
- Creating development, QA, UAT and production-style environments
- Configuring CI/CD pipelines and deployment workflows
- Implementing application monitoring and logging
- Demonstrating deployment, troubleshooting and rollback

- Deliver working code in Git with README, architecture diagram, API specification, database migrations and setup instructions.
- Demonstrate sign-up/login, customer onboarding CRUD, validation, search, pagination, roles, audit logging and error handling.
- Show automated tests, CI pipeline, deployed frontend/backend/database, observability, troubleshooting and rollback.
- Acceptance criteria: another developer can clone, run, test and deploy the application from documentation.