Java Full Stack + Generative AI Engineering

Build full-stack Java applications with Spring Boot and React, then go deep on Generative AI: Spring AI, RAG, vector search, tool calling and AI agents.

DURATION	SESSIONS	TRAINING HOURS
14 weeks	**56 sessions of 90 minutes**	**84 hours**
MODE	LEVEL	INCLUDES
Live online, instructor-led	**Intermediate**	**3 projects incl. capstone**

What you will learn

- Modern Java: OOP, collections, streams, records and virtual threads
- SQL, PostgreSQL and pgvector
- Spring Boot REST APIs, Spring Data JPA and Spring Security with JWT
- Testing with JUnit and Mockito
- React with TypeScript, including a streaming chat interface
- How large language models work: tokens, context and cost
- Prompt engineering and calling LLM APIs from Java
- Spring AI: chat clients, prompt templates, structured output and chat memory
- Embeddings, vector search and document ingestion
- Retrieval-Augmented Generation (RAG): building, improving and evaluating it
- Tool calling, AI agents and the Model Context Protocol (MCP)
- Guardrails, prompt-injection defence and handling personal data
- Cost, latency, caching, observability and testing for AI applications
- Docker, deployment and CI/CD

Curriculum

Twelve modules across 56 live sessions of 90 minutes. Part A (sessions 1–24) builds the Java full stack: modern Java, PostgreSQL, Spring Boot and React. Part B (sessions 25–56) is Generative AI engineering: LLMs, Spring AI, vector search, RAG, tool calling and AI agents, production practices and a deployed capstone.

Module 1 Java Essentials

Core Java, fast and practical, as the base for everything that follows.

1. Java ecosystem, JDK and JVM, IDE setup and your first program
2. Variables, data types, operators and control flow
3. Object-oriented programming: classes, objects and encapsulation
4. Interfaces, inheritance, polymorphism and abstract classes
5. Collections and generics: List, Set and Map
6. Lambdas, streams and exception handling

Module 2 Modern Java for AI Applications

The modern Java features AI applications rely on.

7 Records, Optional, text blocks and modern Java syntax
8 Concurrency, CompletableFuture and virtual threads for parallel AI calls
9 Java HTTP client, JSON with Jackson and calling web APIs

Module 3 SQL and PostgreSQL

Relational data and the vector extension used later for AI search.

10 SQL fundamentals: tables, keys, CRUD and filtering
11 Joins, aggregation, indexes and transactions
12 Spring Data JPA: entities, repositories and relationships
13 PostgreSQL with pgvector: setup and storing vectors

Module 4 Spring Boot and REST APIs

Production-style backend services.

14 Spring Boot, dependency injection, configuration and profiles
15 Building REST controllers, DTOs and request validation
16 Spring Data JPA in a layered service
17 Global error handling and testing with JUnit and Mockito
18 Spring Security and JWT authentication
19 API documentation with OpenAPI and testing with Postman

Module 5 React Frontend

A modern UI for AI-powered applications.

20 JavaScript and TypeScript essentials for React
21 React components, props, state and hooks
22 Forms, routing and calling REST APIs
23 Building a streaming chat interface
24 State management, loading states and UX polish

Module 6 LLM Fundamentals

How large language models work and how to use them from Java.

25 How LLMs work: tokens, context windows, models and cost
26 Prompt engineering: system prompts, few-shot and prompt patterns
27 Calling LLM APIs from Java
28 Streaming responses to the browser

Module 7 Spring AI

The Spring way to build AI features.

29 Spring AI ChatClient and prompt templates
30 Structured output: mapping model responses to Java records
31 Chat memory and conversation history
32 Images and multimodal input

Module 8 Embeddings and Vector Search

Semantic search over your own data.

33 Embeddings and similarity search
34 pgvector as a Spring AI vector store
35 Ingesting PDFs and documents: parsing and chunking

Module 9 Retrieval-Augmented Generation (RAG)

Answers grounded in your documents and data.

36 RAG architecture and when to use it
37 Building a RAG API with Spring AI
38 Improving retrieval: chunking, metadata filters and re-ranking
39 Citations and reducing hallucinations
40 Evaluating RAG answers

Module 10 Tool Calling and AI Agents

Models that act: call your code, data and services.

41 Function and tool calling
42 Building an agent that uses your database and APIs
43 Model Context Protocol (MCP): servers and clients
44 Multi-step workflows and planning
45 Guardrails, human approval and safe actions

Module 11 Production Generative AI

Running AI features reliably, safely and affordably.

46 Cost, latency, caching and rate limits
47 Security: prompt injection, PII and output validation
48 Testing and observability for LLM applications
49 Docker, deployment and CI/CD for AI applications

Module 12 Capstone Project

Build, deploy and present your full-stack AI application.

50 Capstone design: requirements, architecture and sprint plan
51 Capstone: backend services and data model
52 Capstone: RAG and agent features
53 Capstone: React interface
54 Capstone: testing, security review and documentation
55 Capstone: deployment
56 Capstone demo and review

How you learn

- Live coding in every session
- Weekly assignments
- AI features built step by step on one growing application
- Code reviews
- A capstone project you deploy and present

Projects

- **AI Document Assistant**: A RAG application that answers questions over PDFs and documents, with citations.
- **AI Agent with Tools**: An agent that uses your own database and APIs to complete multi-step tasks safely.
- **Capstone: Full-Stack AI Application**: A deployed Spring Boot + React application with AI features of your choice, presented in a final demo.

Who should enrol

- BE / B.Tech, BCA, MCA and BSc Computer Science / IT students
- Final-year students and recent graduates
- Java developers who want to add Generative AI skills
- Anyone aiming for a backend, full-stack or AI engineering role

Prerequisites

Basic programming knowledge in any language is required. Java is taught from the essentials, at a fast pace.

Course completion

Learners who complete the training, assignments and capstone project receive a course completion certificate through GeeksArray, and finish with a deployed full-stack AI application for their portfolio.

Questions about batch dates, timings, fees or anything else?

Contact the course coordinator, **Kaumodini Kedar** · WhatsApp **+91 91726 23305**

Or see the course page on geeksarray.com.