

Raj Technologies

FULL STACK JAVA WITH ARTIFICIAL INTELLIGENCE

Complete Career-Ready Program · 2026 Edition

*From absolute beginner to job-ready full stack developer —
with the power of AI built right in.*

8 Modules

Core Java → AI Tools → Web → DB → Spring Boot → Microservices → DevOps → Live Project

Prerequisites

Basic idea of computers · Programming knowledge is optional

01 Core Java

Build a rock-solid foundation — the language that powers enterprise applications worldwide

■ Getting Started with Java

- ◆ Introduction to Programming Concepts
- ◆ Download & Install Java (JDK setup)
- ◆ Writing your first Java app at command prompt
- ◆ Download & Install Eclipse IDE
- ◆ Developing Java apps inside Eclipse IDE
- ◆ Understanding how Java runs (JVM, JRE, JDK)

■ Java Fundamentals

- ◆ Comments, Keywords & Variables
- ◆ Data Types — Primitive & Reference Types
- ◆ Type Casting — Implicit & Explicit
- ◆ JShell Interactive Tool
- ◆ Operators (all types)
- ◆ Reading input via Scanner class
- ◆ if, switch, while, do-while, for loop
- ◆ break & continue
- ◆ One-Dimensional Arrays
- ◆ Example programs using Arrays

■ Object-Oriented Programming (OOP)

- ◆ Classes & Objects
- ◆ Constructors
- ◆ static keyword
- ◆ Packages
- ◆ Inheritance
- ◆ Polymorphism — Overloading & Overriding
- ◆ super reference & Constructors in Inheritance
- ◆ final keyword & Inner Classes
- ◆ Access Modifiers — private, default, protected, public
- ◆ Encapsulation
- ◆ Abstract Classes & Interfaces
- ◆ Java Coding Standards
- ◆ java.lang package — Object, String, Math, Wrapper classes
- ◆ StringBuffer & StringBuilder

■ Exception Handling & Multithreading

- ◆ Types of Errors
- ◆ try, catch, finally, throw, throws
- ◆ Exception Hierarchy
- ◆ Checked & Unchecked Exceptions
- ◆ Multiple catch blocks & User-Defined Exceptions
- ◆ Thread Lifecycle & State Transitions
- ◆ Thread class & Runnable interface
- ◆ Thread Priorities & Synchronization
- ◆ Inter-thread communication
- ◆ Methods of Throwable class

■ Collections, Java 8/9 Features & I/O

- ◆ Collections Hierarchy — List, Set, Map
- ◆ ArrayList, LinkedList, HashSet, TreeSet, LinkedHashSet
- ◆ HashMap, TreeMap, LinkedHashMap
- ◆ Generics, Enhanced for-loop, Comparator
- ◆ Functional Interfaces & Lambda Expressions
- ◆ default, static & private methods in Interfaces
- ◆ forEach() method & Stream API
- ◆ Byte Streams & Character Streams
- ◆ FileInputStream/OutputStream, BufferedReader/Writer
- ◆ Serialization — ObjectOutputStream / ObjectOutputStream

02 Artificial Intelligence Tools

Learn to work smarter — use AI as your co-pilot throughout your entire development career

AI is no longer optional for developers. This module gives you a practical, hands-on command of the tools shaping how software is built today.

■ Foundations of AI

- ◆ What is Artificial Intelligence? — a clear, jargon-free introduction
- ◆ Generative AI — text, code, and content generation explained
- ◆ Large Language Models (LLMs) — how they work and why they matter to developers
- ◆ The difference between traditional programming and AI-assisted development

■ Prompt Engineering — Communicate Effectively with AI

- ◆ What is a prompt and why crafting it well changes everything
- ◆ Role prompting — making AI behave like a senior developer, reviewer, or architect
- ◆ Iterative refinement — getting better answers through follow-up questions
- ◆ Zero-shot, one-shot, and few-shot prompting techniques
- ◆ Chain-of-thought prompting for step-by-step problem solving
- ◆ Common mistakes in prompting and how to avoid them

■ AI Tools You Will Master

ChatGPT (OpenAI)	The world's most widely used AI — writing code, explaining concepts, debugging, summarising documentation in seconds
Claude AI (Anthropic)	Exceptional for long-context tasks, analysing large codebases, drafting technical documents and thoughtful code review
Gemini (Google)	Google's multimodal AI — understand images, integrate with Google Workspace, and leverage Google Search context
NotebookLM (Google)	Upload your course notes, textbooks, or project docs and have a personal AI tutor answer questions from your own material
Perplexity AI	AI-powered search engine — get cited, up-to-date answers for research, tech comparisons, and industry trends

■ AI for Coding & Debugging

- ◆ Using AI to write boilerplate code — save hours every week
- ◆ Code explanation — make any unfamiliar code readable in plain English
- ◆ AI-assisted code review — spot issues before your team does
- ◆ Using AI inside VS Code, IntelliJ, and Eclipse with extensions
- ◆ Debugging with AI — paste your error, understand the root cause immediately
- ◆ Writing unit tests with AI assistance
- ◆ Generating SQL queries, API documentation, and config files using AI

By the end of this module, AI becomes a tool you reach for every day — not something you read about.

03 Web Technologies

Build beautiful, responsive, interactive front-ends — the face of every application

- ◆ HTML — structure and semantics of web pages
- ◆ CSS — styling, layouts, flexbox and grid
- ◆ Bootstrap — responsive design with ready-made components
- ◆ XML — data exchange format used in enterprise systems
- ◆ JSON — the universal language of web APIs
- ◆ JavaScript — bring pages to life with dynamic behaviour
- ◆ TypeScript — JavaScript with types, cleaner & scalable
- ◆ Angular — Google's full-featured frontend framework
- ◆ React — Facebook's component-based UI library
- ◆ Build real-world responsive UIs combining all of the above

04 Database Concepts using MySQL

Store, retrieve and manage data — the backbone of every application

- ◆ Introduction to Databases & SQL
- ◆ Install & configure MySQL Server
- ◆ DDL — CREATE, ALTER, DROP, TRUNCATE
- ◆ DML — INSERT, UPDATE, DELETE
- ◆ DQL — SELECT with filters, sorting & grouping
- ◆ TCL — COMMIT, ROLLBACK, SAVEPOINT
- ◆ Constraints — PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK
- ◆ SQL Joins — INNER, LEFT, RIGHT, FULL
- ◆ SQL Command Types — DDL, DML, DQL, TCL overview
- ◆ Connecting MySQL to Java applications

05 Advance Java / Java EE

Connect Java to the web — build server-side applications that power real products

■ JDBC — Connecting Java to Databases

- ◆ Introduction to JDBC
- ◆ Steps to connect to a database using JDBC
- ◆ Statement Interface
- ◆ PreparedStatement Interface

■ Java EE & Servlets

- ◆ Java EE Introduction & Components
- ◆ Types of Servers — Web Servers vs Application Servers
- ◆ Introduction to Servlets & Servlet Hierarchy
- ◆ Servlet Lifecycle
- ◆ Creating a Servlet using Apache Tomcat in Eclipse
- ◆ Reading HTML form data & building a Login Application

■ JSP — JavaServer Pages

- ◆ Introduction to JSP & JSP Lifecycle
- ◆ JSP Tags — Directives, Scripting elements, Action tags
- ◆ JSP Implicit Objects
- ◆ Building dynamic pages with JSP

06 Frameworks

Industry-standard tools — Hibernate, Spring Boot, Spring AI & Microservices

■ Hibernate — ORM Framework

- ◆ ORM Overview & Hibernate Architecture
- ◆ Developing a Hibernate Application
- ◆ Annotations in Hibernate
- ◆ CRUD Operations in Hibernate
- ◆ Hibernate Query Language (HQL)

■ Spring Framework & Spring Boot

- ◆ Introduction to Spring Framework
- ◆ Dependency Injection & Inversion of Control (IoC)
- ◆ Spring Core & IoC Modules
- ◆ Spring Boot Introduction & Spring Initializer
- ◆ Creating Spring Boot apps in STS (Spring Tool Suite)
- ◆ Spring Boot Starters
- ◆ Spring DAO Module (Spring JDBC)
- ◆ Spring Data JPA Module
- ◆ Spring Web MVC Module
- ◆ Spring REST Module — building REST APIs
- ◆ Spring Security Module — securing applications
- ◆ Spring AI Module
- ◆ Generative AI using Spring AI
- ◆ Spring Framework vs Spring Boot — when to use what

■ Microservices Architecture

- ◆ Introduction to Microservices
- ◆ Monolithic vs Microservices — pros, cons and when to choose
- ◆ Introduction to Netflix Eureka Server
- ◆ Service Registry
- ◆ API Gateway
- ◆ Building a Microservice Application using Netflix Eureka Server

07 Real-Time Tools & DevOps Tools

Work like a professional — the tools every development team uses every day

- ◆ Maven — build automation and dependency management
- ◆ Lombok — eliminate boilerplate with annotations
- ◆ GitHub — version control, branching and collaboration
- ◆ Postman — test and document your REST APIs
- ◆ JUnit — writing and running automated tests
- ◆ Log4J — logging for production applications
- ◆ Docker — containerise and deploy applications anywhere

08 Full Stack Java Project

Put everything together — build and deploy a real application from scratch

You will design, develop and deploy a complete full-stack application combining Spring Boot + Microservices on the backend with Angular or React on the frontend — using all the tools, frameworks, and AI-powered workflows covered throughout the programme. This is the project you carry into interviews and your portfolio.

Ready to start your Full Stack Java with AI journey?