

LIVE COURSE

Java AI Full Stack Development

Learn Java, Spring Boot, React and AI integration in live mentor-led sessions. Build 9 hands-on projects and become job-ready as a full stack developer who can ship AI-powered applications.

Industry-Led Course

Batch details updating soon

Technical



DURATION

6 Months



LEVEL

Beginner to Advanced



MODE

Live Online



LANGUAGE

English



BATCH STARTS

To be announced



ENROLMENT CLOSES

Open

COURSE FEE

₹7,999

₹30,000 - save 73% OFF



LIVE COURSE BROCHURE

Inside this brochure

Everything on the Java AI Full Stack Development page, in one document. Tap a section to jump to it.

01		Course Overview	03
02		How Our Courses Work	04
03		Eligibility & Prerequisites	05
04		Skills You Will Gain	05
05		How we designed our Product?	06
06		What You Will Learn	08
07		Meet Our SME Board Members	38
08		Key Tools and Techniques	40
09		Projects You Will Work On	41
10		Java AI Full Stack Development	43
11		World's First Ed-Tech with 8C Appraisal	44
12		Frequently Asked Questions	45
13		Ready to take the next step?	46

01 · AT A GLANCE



Course Overview

Learn Java, Spring Boot, React and AI integration in live mentor-led sessions. Build 9 hands-on projects and become job-ready as a full stack developer who can ship AI-powered applications.

This live, mentor-led course takes you from Java basics to building complete full stack applications with Spring Boot, React and databases, then adds AI features using LLM APIs. You practice through nine projects, code reviews and interview preparation, finishing with a deployable portfolio.



DURATION

6 Months

LEVEL

Beginner to Advanced

MODE

Live Online

LANGUAGE

English

BATCH STARTS

To be announced

ENROLMENT CLOSES

Open

LEARNERS

Trusted by ambitious learners nationwide

COURSE FEE

₹7,999

₹30,000 - save 73% OFF

Highlights

Industry-Led Upskilling Courses

Learn from industry professionals with years of real-world experience.

Personalized Mentorship

Get one-on-one guidance tailored to your career goals and aspirations.

Premium Resources

Access course materials, job portal, free events, challenges and learner 8C Appraisal reports.

Vibrant Community

Connect with like-minded peers and build lasting professional relationships.

**02 · SIMPLE 3-STEP JOURNEY**

How Our Courses Work

A structured, step-by-step path designed to take you from foundational learning to career success with industry guidance.

01**Step 1: Enroll & Get Started**

Choose your course and complete the enrollment process. Access your learning materials instantly and join our community of learners.

✓ Instant access to course materials

✓ Personalized learning dashboard

✓ Welcome orientation session

02**Step 2: Learn & Practice**

Engage with interactive lessons, live sessions, and hands-on projects. Learn from industry experts and apply your knowledge in real-world scenarios.

✓ Live classes with expert instructors

✓ Hands-on practical projects

✓ One-on-one mentorship support

03**Step 3: Complete & Get Certified**

Finish your coursework, pass assessments, and receive your professional certificate. Launch your new career with confidence and credentials.

✓ Industry-recognized certificate

✓ Portfolio of completed projects

✓ Career placement assistance

03 · BEFORE YOU ENROLL



Eligibility & Prerequisites

Check who this course is designed for and what you should know before your first class.

WHO CAN JOIN**Eligibility**

- ✓ Students from any degree or diploma stream
- ✓ Freshers looking for a developer job
- ✓ Working professionals switching to software roles
- ✓ Developers wanting to add AI skills

BEFORE YOU START**Prerequisites**

- ✓ Basic computer and internet skills
- ✓ Comfort with logical thinking
- ✓ A laptop with 8 GB RAM preferred
- ✓ Willingness to practice coding daily

04 · CAREER-READY SKILLS



Skills You Will Gain

Hands-on skills you will practise in live classes, assignments and projects throughout the course.

Core Java

OOP

Data Structures

Collections

Spring Boot

REST APIs

Spring Security

JPA Hibernate

MySQL

React

JavaScript

Git and GitHub

Docker

LLM API Integration

Prompt Engineering

05 · PRODUCT ECOSYSTEM ARCHITECTURE



How we designed our Product?

An all-in-one ed-tech ecosystem carefully engineered to bridge learning, practical skill assessment, career guidance, and employment.

WORLD'S FIRST FEATURE

8C Appraisal Report

Proprietary evaluation matrix measuring holistic performance across 8 career pillars for authentic proof-of-work.

Communication

Configuration

Chronicle

Completeness

Cognizance

Conception

Comprehension

Case Study

Industry Validated · 100% Data Score

SUBJECT MATTER EXPERT AI

Insyrio AI

Get instant 24/7 doubt solving and concept clarifications directly from our specialized AI trained on your enrolled courses.

- ✓ 24/7 Course-Specific SME Assistant
- ✓ Instant Step-by-Step Doubt Resolution

Instant Learning Support

GAMIFIED REWARDS

iSure Coins

Use iSure Coins for AI career tools, masterclasses and CertiQ - and part-pay your course fee with them.

- ✓ Earn Perks & Rewards · Pay with Coins

Redeem Rewards Store

AI ENGINE

AI Career Diagnosis

Algorithmic analysis mapping strengths, gaps, and tailored pathways for industry roles.

- ✓ Skill Gap Mapping

Personalized Roadmap

CURRICULUM

Industry-Led Courses

Handcrafted learning modules built directly alongside senior engineering, design, and management leads.

Practical Frameworks

LIVE

Live Classes

Interactive classrooms with live chat, code walk-throughs, and real-time engagement.

Real-time Q&A Support

AUTO-SCAN

AI Resume ATS

Instant ATS optimization scanner ensuring your resume passes screening software effortlessly.

High Parser Score

PRACTICAL**Assignments**

Industry-grade project briefs with detailed feedback and grading from experienced mentors.

Code & Work Reviews

WEEKLY**Challenges**

Weekly hackathons, coding sprints, and design contests to test your problem-solving skills.

Leaderboard Ranking

NETWORKING**Events**

Exclusive expert keynotes, career fairs, and interactive networking workshops.

Free Community Pass

1-ON-1**Live Mentorship**

Direct one-on-one video guidance sessions with experienced industry practitioners.

Personalized Growth

TRACKING**Attendance**

Class-by-class attendance tracking to ensure disciplined, consistent progression.

Attendance Reports

HIRING**Jobs**

Direct hiring pipeline connecting top-performing learners directly with recruiter managers.

Exclusive Placements

FREE ACCESS**Free Books**

Curated digital library filled with essential tech, management, and soft-skill books.

Instant Digital Library

ARTICLES**Blog**

In-depth technical tutorials, industry trend analysis, and career growth strategies.

Updated Weekly

FREE ACCESS**Free Study Material**

Downloadable cheat sheets, interview preparation guides, templates, and revision notes.

One-Click Downloads

06 · COURSE CURRICULUM



What You Will Learn

A carefully sequenced curriculum - every section breaks into modules and hands-on topics taught live by industry experts.

6

SECTIONS

36

MODULES

216

TOPICS

9

PROJECTS

01

SECTION 1

Core Java and Object Oriented Programming

Build a rock-solid foundation in Java syntax, object oriented design, collections, exceptions, and modern language features that every full stack developer relies on.

6 modules · 36 topics

M1.1 **Java Fundamentals and Environment Setup** · 6 topics

Get started with the Java platform, tooling, and the basic building blocks of every Java program.

1**Introduction to Java and the JVM**

Understand how Java code is compiled and executed through the JDK, JRE, and JVM. Write and run your first program from the command line.

JVM

JDK

COMPILATION

2**Setting Up IDE and Build Tools**

Install IntelliJ IDEA and configure a Java project with Maven basics. Learn to run, debug, and organize code efficiently.

INTELLIJ IDEA

MAVEN

DEBUGGING

3**Variables and Data Types**

Work with primitive and reference types, literals, type casting, and the var keyword. Learn how Java handles memory for different types.

PRIMITIVES

TYPE CASTING

MEMORY

4

Operators and Expressions

Use arithmetic, relational, logical, bitwise, and ternary operators to build expressions. Understand operator precedence and common pitfalls.

OPERATORS

EXPRESSIONS

PRECEDENCE

5

Input and Output in Java

Read user input with Scanner and format console output with printf and String.format. Build small interactive console programs.

SCANNER

CONSOLE IO

FORMATTING

6

Java Coding Conventions and Best Practices

Learn naming standards, code formatting, and commenting with Javadoc. Develop habits that make code readable and maintainable.

NAMING

JAVADOC

CLEAN CODE

M1.2

Control Flow, Arrays, and Strings · 6 topics

Master decision making, loops, arrays, and string handling to solve basic programming problems.

1

Conditional Statements

Apply if, else if, switch statements, and switch expressions to control program decisions. Practice with real-world branching scenarios.

IF ELSE

SWITCH

BRANCHING

2

Loops and Iteration

Use for, while, do while, and enhanced for loops to repeat tasks. Control loops with break and continue.

FOR LOOP

WHILE LOOP

ITERATION

3

Single and Multidimensional Arrays

Declare, initialize, and traverse one dimensional and two dimensional arrays. Solve problems such as searching and matrix operations.

ARRAYS

MATRICES

TRAVERSAL

4

String Class and Immutability

Explore the String class, the string pool, and common methods for manipulation. Understand why strings are immutable.

STRING

IMMUTABILITY

STRING POOL

5

StringBuilder and Regular Expressions

Build efficient strings with StringBuilder and validate text using regex patterns. Apply them to input validation tasks.

STRINGBUILDER

REGEX

VALIDATION

6 Problem Solving with Basic Algorithms

Implement sorting, searching, and pattern problems using loops and arrays. Learn to think through problems step by step.

SORTING

SEARCHING

LOGIC BUILDING

M1.3 Classes, Objects, and Encapsulation · 6 topics

Learn to model real-world entities using classes and protect data through encapsulation.

1 Classes and Objects

Define classes with fields and methods and create objects using the new keyword. Understand the difference between class and instance.

CLASSES

OBJECTS

INSTANCES

2 Constructors and Object Initialization

Use default, parameterized, and copy constructors along with constructor chaining. Explore initializer blocks and the this keyword.

CONSTRUCTORS

THIS KEYWORD

INITIALIZATION

3 Methods and Overloading

Write methods with parameters and return values, and apply method overloading. Understand pass by value in Java.

METHODS

OVERLOADING

PASS BY VALUE

4 Access Modifiers and Encapsulation

Control visibility with private, default, protected, and public modifiers. Build well encapsulated classes with getters and setters.

ENCAPSULATION

ACCESS MODIFIERS

GETTERS SETTERS

5 Static Members and Packages

Use static variables, methods, and blocks, and organize code into packages. Learn about imports and project structure.

STATIC

PACKAGES

IMPORTS

6 Object Class Methods and Immutable Classes

Override toString, equals, and hashCode properly and design immutable classes. Understand why these contracts matter.

EQUALS

HASHCODE

IMMUTABILITY

M1.4 Inheritance, Polymorphism, and Abstraction · 6 topics

Apply the core pillars of object oriented programming to design flexible and reusable code.

1 Inheritance and the super Keyword

Create class hierarchies using extends and access parent members with super. Understand constructor behavior in inheritance.

INHERITANCE

SUPER

HIERARCHY

2 Method Overriding and Runtime Polymorphism

Override parent methods and see dynamic method dispatch in action. Learn upcasting and downcasting with instanceof.

OVERRIDING

POLYMORPHISM

CASTING

3 Abstract Classes

Design abstract classes with abstract and concrete methods to share common behavior. Know when to choose abstraction.

ABSTRACT CLASS

ABSTRACTION

DESIGN

4 Interfaces and Default Methods

Define contracts with interfaces, including default, static, and private methods. Implement multiple interfaces in a class.

INTERFACES

DEFAULT METHODS

CONTRACTS

5 Composition versus Inheritance

Compare is-a and has-a relationships and learn to favor composition. Refactor a sample design to reduce tight coupling.

COMPOSITION

COUPLING

REFACTORING

6 SOLID Principles and Design Basics

Introduce the SOLID principles with simple Java examples. Apply them to improve class design.

SOLID

DESIGN PRINCIPLES

UML

M1.5 Exception Handling and Collections Framework · 6 topics

Handle errors gracefully and manage groups of data using the Java Collections Framework.

1 Exception Handling Basics

Use try, catch, finally, and multi catch to handle runtime errors. Understand the exception hierarchy and checked versus unchecked exceptions.

TRY CATCH

EXCEPTIONS

FINALLY

2 Custom Exceptions and Try with Resources

Create custom exception classes and manage resources safely with try with resources. Apply throw and throws correctly.

CUSTOM EXCEPTIONS

THROWS

AUTOCLOSEABLE

3 Lists and the ArrayList and LinkedList

Store ordered data using List implementations and compare their performance. Practice common operations and iteration.

LIST

ARRAYLIST

LINKEDLIST

4 Sets and Maps

Work with HashSet, TreeSet, HashMap, and TreeMap to handle unique and keyed data. Understand how hashing works internally.

HASHMAP

HASHSET

HASHING

5 Queues, Deques, and Iterators

Use Queue, Deque, and PriorityQueue for ordered processing and traverse collections with Iterator. Avoid concurrent modification errors.

QUEUE

PRIORITYQUEUE

ITERATOR

6 Sorting with Comparable and Comparator

Sort custom objects using natural and custom ordering. Use the Collections utility class for common operations.

COMPARABLE

COMPARATOR

COLLECTIONS

M1.6 Modern Java Features and Core APIs · 6 topics

Write concise and expressive code using generics, lambdas, streams, and essential Java APIs.

1 Generics

Create type safe classes and methods with generics and bounded types. Understand wildcards and type erasure.

GENERICS

WILDCARDS

TYPE SAFETY

2 Enums and Records

Model fixed sets of values with enums and create concise data carriers using records. Learn when to use each.

ENUMS

RECORDS

DATA MODELING

3 Lambda Expressions and Functional Interfaces

Write lambdas and use built in functional interfaces such as Predicate, Function, and Consumer. Apply method references.

LAMBDA

FUNCTIONAL INTERFACES

METHOD REFERENCES

4

Stream API

Process collections declaratively with map, filter, reduce, and collectors. Build data pipelines for realistic datasets.

STREAMS

COLLECTORS

DATA PROCESSING

5

Optional and Date Time API

Avoid null errors using Optional and work with LocalDate, LocalDateTime, and Duration. Format and parse dates correctly.

OPTIONAL

LOCALDATE

NULL SAFETY

6

File Handling and Introduction to Multithreading

Read and write files using NIO and get introduced to threads and the Runnable interface. Understand basic synchronization concepts.

FILE IO

THREADS

SYNCHRONIZATION

02

SECTION 2

Data Structures, Collections and Databases with SQL

Learners master core data structures, the Java Collections Framework, relational database design and SQL, and connect Java applications to databases using JDBC.

6 modules · 36 topics

M2.1

Foundations of Data Structures · 6 topics

Learners understand how data is organized in memory and how to reason about the efficiency of code.

1

Introduction to Data Structures and Algorithms

Explore why data structures matter and how they shape program performance. Compare linear and non-linear structures with real-world examples.

DATA STRUCTURES

PROBLEM SOLVING

2

Time and Space Complexity with Big O

Learn to measure algorithm efficiency using Big O notation. Analyze simple loops and nested loops to estimate cost.

BIG O

COMPLEXITY ANALYSIS

3

Arrays and Multidimensional Arrays

Work with fixed-size arrays and two-dimensional arrays in Java. Practice traversal, insertion and searching patterns.

ARRAYS

TRAVERSAL

INDEXING

4

Linked Lists from Scratch

Build singly and doubly linked lists using custom node classes. Implement insert, delete and reverse operations.

LINKED LIST

POINTERS

OOP

5

Stacks and Queues Implementation

Implement stacks and queues using arrays and linked nodes. Apply them to problems like bracket matching and task scheduling.

STACK

QUEUE

LIFO FIFO

6

Trees and Binary Search Trees

Understand hierarchical data using binary trees and binary search trees. Implement insertion, search and traversal methods.

BINARY TREE

BST

RECURSION

M2.2

Searching Sorting and Recursion · 6 topics

Learners apply classic algorithms and recursive thinking to solve common programming problems.

1

Recursion Fundamentals

Learn how recursive calls work using the call stack. Solve problems like factorial and Fibonacci with clear base cases.

RECURSION

CALL STACK

2

Linear and Binary Search

Implement both search techniques and compare their performance. Understand when sorted data enables faster lookups.

SEARCHING

BINARY SEARCH

3

Elementary Sorting Algorithms

Code bubble, selection and insertion sort step by step. Trace each algorithm and discuss its limitations.

BUBBLE SORT

INSERTION SORT

TRACING

4

Merge Sort and Quick Sort

Study divide and conquer sorting techniques. Implement both and compare their running times on large inputs.

MERGE SORT

QUICK SORT

DIVIDE AND CONQUER

5

Hashing and Hash Tables

Understand hash functions and collision handling strategies. Build a simple hash table to see how HashMap works internally.

HASHING

COLLISIONS

HASH TABLE

6

Graphs and Traversal Basics

Represent graphs with adjacency lists and matrices. Implement breadth first and depth first traversal.

GRAPHS

BFS

DFS

M2.3

Java Collections Framework · 6 topics

Learners use the built-in Java collection classes effectively in real applications.

1

Collections Framework Overview

Tour the interface hierarchy including Collection, List, Set and Map. Learn how to choose the right collection for a task.

COLLECTIONS

INTERFACES

2

List Implementations ArrayList and LinkedList

Compare ArrayList and LinkedList in usage and performance. Practice common list operations and iteration.

ARRAYLIST

LINKEDLIST

ITERATION

3

Sets with HashSet LinkedHashSet and TreeSet

Work with collections that enforce uniqueness. Understand ordering differences and the role of equals and hashCode.

SET

EQUALS HASHCODE

TREESET

4

Maps with HashMap and TreeMap

Store and retrieve key value pairs efficiently. Explore iteration over entries, keys and values.

HASHMAP

TREEMAP

KEY VALUE

5

Queues Deques and PriorityQueue

Use queue based collections for ordered processing. Build a simple task scheduler with PriorityQueue.

QUEUE

DEQUE

PRIORITYQUEUE

6

Comparable Comparator and Utility Classes

Define custom sorting using Comparable and Comparator. Apply the Collections and Arrays utility methods.

COMPARATOR

SORTING

UTILITIES

M2.4

Relational Database and SQL Fundamentals · 6 topics

Learners design relational databases and write core SQL queries to manage data.

1

Introduction to Databases and RDBMS

Understand database concepts, tables, rows and keys. Install MySQL and explore its client tools.

RDBMS

MYSQL

SETUP

2 Creating Databases and Tables with DDL

Use CREATE, ALTER and DROP to define schemas. Choose suitable data types and apply constraints.

DDL

CONSTRAINTS

DATA TYPES

3 Inserting Updating and Deleting Data with DML

Manipulate records using INSERT, UPDATE and DELETE. Learn safe practices including transactions for changes.

DML

CRUD

DATA SAFETY

4 Querying Data with SELECT and Filters

Retrieve data using SELECT, WHERE, ORDER BY and LIMIT. Use operators like LIKE, IN and BETWEEN.

SELECT

FILTERING

SORTING

5 Aggregate Functions and Grouping

Summarize data with COUNT, SUM, AVG, MIN and MAX. Group results using GROUP BY and HAVING.

AGGREGATION

GROUP BY

HAVING

6 Database Design and Normalization

Model entities and relationships using ER diagrams. Apply normal forms to reduce redundancy.

ER DIAGRAMS

NORMALIZATION

SCHEMA DESIGN

M2.5 Advanced SQL and Query Optimization · 6 topics

Learners write complex queries and improve database performance and reliability.

1 Joins Across Multiple Tables

Combine data with inner, left, right and full joins. Solve practical reporting problems using multi table queries.

JOINS

RELATIONSHIPS

REPORTING

2 Subqueries and Common Table Expressions

Write nested queries and readable CTEs. Decide when each approach improves clarity.

SUBQUERIES

CTE

QUERY DESIGN

3 Window Functions and Ranking

Use ROW NUMBER, RANK and running totals with OVER clauses. Analyze data without collapsing rows.

WINDOW FUNCTIONS

RANKING

ANALYTICS

4 Indexes and Query Performance

Learn how indexes speed up lookups and when they hurt. Read execution plans with EXPLAIN to tune queries.

INDEXING

EXPLAIN

OPTIMIZATION

5

Transactions and ACID Properties

Understand commit, rollback and isolation levels. See how transactions keep data consistent under concurrency.

TRANSACTIONS

ACID

ISOLATION

6

Views Stored Procedures and Triggers

Create reusable database logic with views, procedures and triggers. Discuss best practices and trade-offs.

VIEWS

STORED PROCEDURES

TRIGGERS

M2.6

Connecting Java with Databases using JDBC · 6 topics

Learners integrate Java applications with relational databases and build a data driven project.

1

JDBC Architecture and Drivers

Understand the JDBC API, driver types and connection lifecycle. Connect a Java program to MySQL.

JDBC

DRIVERS

CONNECTIVITY

2

Executing Queries with Statement and ResultSet

Run SELECT and update queries from Java code. Read and map results using ResultSet.

STATEMENT

RESULTSET

JAVA SQL

3

PreparedStatement and SQL Injection Prevention

Use parameterized queries for safety and speed. Learn how SQL injection attacks work and how to block them.

PREPAREDSTATEMENT

SQL INJECTION

SECURITY

4

Transactions and Batch Processing in JDBC

Control commits and rollbacks from Java. Use batch updates to process many records efficiently.

JDBC TRANSACTIONS

BATCH UPDATES

5

DAO Pattern and Connection Pooling

Separate data access logic using the DAO pattern. Configure connection pooling with HikariCP.

DAO PATTERN

HIKARICP

ARCHITECTURE

6

Mini Project Student Management System

Build a console application combining collections, SQL and JDBC. Implement full CRUD with a layered design.

PROJECT

CRUD

INTEGRATION

03

SECTION 3

Spring Boot Backend and REST APIs

Learners build secure, well-tested and production-ready backend services with Spring Boot, from first project setup to deployed REST APIs backed by relational and NoSQL databases.

6 modules · 36 topics

M3.1 Spring Boot Foundations · 6 topics

Learners understand the Spring ecosystem and create their first configurable Spring Boot application.

1 Introduction to Spring and Spring Boot

Explore why Spring Boot simplifies enterprise Java development and how it differs from the traditional Spring Framework. Review the architecture of a typical backend application.

[SPRING BOOT](#)[BACKEND BASICS](#)[ARCHITECTURE](#)

2 Project Setup with Maven and Spring Initializr

Generate a project, understand the folder structure and manage dependencies using Maven. Run the application from the IDE and the command line.

[MAVEN](#)[SPRING INITIALIZR](#)[IDE SETUP](#)

3 Inversion of Control and Dependency Injection

Learn how the Spring container creates and wires objects. Practice constructor injection and compare it with other injection styles.

[IOC](#)[DEPENDENCY INJECTION](#)[SPRING CONTAINER](#)

4 Beans, Scopes and Component Scanning

Define beans using annotations and Java configuration. Understand singleton and prototype scopes and the bean lifecycle.

[BEANS](#)[ANNOTATIONS](#)[LIFECYCLE](#)

5 Application Properties, Profiles and Configuration

Externalize configuration with properties and YAML files. Use profiles to switch between development and production settings.

[CONFIGURATION](#)[PROFILES](#)[YAML](#)

6 Logging, Actuator and Developer Tools

Add structured logging and monitor application health with Spring Boot Actuator. Use DevTools to speed up the development cycle.

[LOGGING](#)[ACTUATOR](#)[DEVTOOLS](#)

M3.2 Building RESTful APIs · 6 topics

Learners design and implement clean REST endpoints that follow industry conventions.

1 REST Principles and HTTP Fundamentals

Understand resources, HTTP methods, status codes and statelessness. Learn how to design meaningful URIs.

REST

HTTP

API DESIGN

2 Controllers and Request Mapping

Build controllers using RestController and mapping annotations. Handle path variables and query parameters.

RESTCONTROLLER

REQUEST MAPPING

PARAMETERS

3 Request and Response Bodies with JSON

Serialize and deserialize JSON using Jackson. Control response formats with ResponseEntity and status codes.

JSON

JACKSON

RESPONSEENTITY

4 DTOs and Object Mapping

Separate internal entities from API contracts using data transfer objects. Apply mapping with manual code and MapStruct.

DTO

MAPSTRUCT

CLEAN DESIGN

5 Validation and Global Exception Handling

Validate input using Bean Validation annotations. Return consistent error responses with ControllerAdvice.

VALIDATION

CONTROLLERADVICE

ERROR HANDLING

6 API Versioning, Pagination and Sorting

Evolve APIs safely with versioning strategies. Return large result sets efficiently with pagination and sorting.

VERSIONING

PAGINATION

SORTING

M3.3 Data Persistence with Spring Data JPA · 6 topics

Learners connect Spring Boot services to relational databases using JPA and Hibernate.

1 JPA and Hibernate Fundamentals

Learn object relational mapping concepts and how Hibernate translates entities into tables. Configure a database connection in Spring Boot.

JPA

HIBERNATE

ORM

2

Entities and Repositories

Model entities with annotations and use Spring Data repositories for CRUD operations. Build a complete persistence layer for a sample project.

ENTITIES

JPAREPOSITORY

CRUD

3

Entity Relationships and Fetch Strategies

Map one to one, one to many and many to many relationships. Understand lazy and eager loading and avoid the N plus 1 problem.

RELATIONSHIPS

LAZY LOADING

PERFORMANCE

4

Derived Queries, JPQL and Native Queries

Write queries using method naming conventions and the Query annotation. Choose the right approach for complex data needs.

JPQL

NATIVE SQL

QUERY METHODS

5

Transactions and Data Integrity

Manage transactions with the Transactional annotation. Understand propagation, isolation and rollback behavior.

TRANSACTIONS

ISOLATION

DATA INTEGRITY

6

Database Migrations with Flyway

Version control database schemas using Flyway migration scripts. Keep environments consistent across teams and deployments.

FLYWAY

MIGRATIONS

SCHEMA VERSIONING

M3.4

Security and Authentication · 6 topics

Learners protect APIs using Spring Security, JWT and role based access control.

1

Spring Security Architecture

Understand the filter chain, authentication and authorization flow. Configure default security and customize it for an API.

SPRING SECURITY

FILTER CHAIN

AUTHENTICATION

2

User Registration and Password Hashing

Build a registration flow and store passwords securely using BCrypt. Discuss common credential storage mistakes.

BCRYPT

REGISTRATION

SECURE STORAGE

3

JWT Based Authentication

Issue and validate JSON Web Tokens for stateless authentication. Implement a login endpoint and a token filter.

JWT

STATELESS AUTH

FILTERS

4

Role Based Access Control

Restrict endpoints and methods by user roles and authorities. Apply method level security annotations.

RBAC

AUTHORIZATION

METHOD SECURITY

5

CORS, CSRF and Common Vulnerabilities

Configure cross origin access for frontend applications. Review CSRF, injection and other threats from the OWASP list.

CORS

CSRF

OWASP

6

OAuth2 and Social Login

Understand the OAuth2 flow and integrate login with an external identity provider. Compare it with custom JWT authentication.

OAUTH2

SOCIAL LOGIN

IDENTITY

M3.5

Testing and Quality Assurance · 6 topics

Learners verify backend behavior with automated tests and quality practices.

1

Unit Testing with JUnit 5

Write clear unit tests using JUnit 5 assertions and lifecycle annotations. Apply test driven thinking to service logic.

JUNIT 5

UNIT TESTING

ASSERTIONS

2

Mocking with Mockito

Isolate classes under test by mocking dependencies. Verify interactions and stub responses.

MOCKITO

MOCKING

TEST ISOLATION

3

Testing Controllers with MockMvc

Test REST endpoints without starting a full server. Assert status codes, headers and JSON content.

MOCKMVC

WEBMVCTEST

API TESTING

4

Integration Testing with Testcontainers

Run tests against real databases in containers. Use SpringBootTest to verify the full application context.

TESTCONTAINERS

SPRINGBOOTTEST

INTEGRATION TESTS

5

API Documentation with OpenAPI and Swagger

Generate interactive documentation for endpoints using springdoc. Share contracts clearly with frontend teams.

OPENAPI

SWAGGER

DOCUMENTATION

6

Code Quality and Test Coverage

Measure coverage with JaCoCo and enforce standards with static analysis. Practice refactoring guided by tests.

JACOCO

STATIC ANALYSIS

REFACTORING

M3.6

Advanced Backend Concepts and Deployment · 6 topics

Learners extend their services with caching, messaging, microservice basics and cloud ready packaging.

1

Caching with Spring Cache and Redis

Improve response times by caching frequent queries. Configure cache eviction and use Redis as a distributed store.

CACHING

REDIS

PERFORMANCE

2

Working with MongoDB in Spring Boot

Use Spring Data MongoDB to store document based data. Compare when to choose NoSQL over relational storage.

MONGODB

NOSQL

SPRING DATA

3

Asynchronous Processing and Scheduling

Run background tasks with Async and scheduled jobs. Understand thread pools and task execution.

ASYNC

SCHEDULING

THREAD POOLS

4

Messaging with Kafka Basics

Understand event driven design and publish and consume messages with Apache Kafka. Build a simple producer and consumer service.

KAFKA

EVENT DRIVEN

MESSAGING

5

Microservices Fundamentals

Learn how to split a system into services and handle communication with REST clients. Explore API gateways and service discovery concepts.

MICROSERVICES

API GATEWAY

SERVICE DISCOVERY

6

Dockerizing and Deploying Spring Boot Applications

Package the application as a Docker image and run it with Docker Compose. Deploy to a cloud platform and configure environment variables.

DOCKER

DEPLOYMENT

CLOUD

04

SECTION 4

Frontend Development with React

Learners build modern responsive user interfaces with React and connect them to the Spring Boot REST APIs created in earlier sections.

6 modules · 36 topics

M4.1 Web Foundations for React Developers · 6 topics

Learners refresh the modern HTML, CSS and JavaScript skills that every React project depends on.

1 Semantic HTML and Accessible Page Structure

Learners write meaningful HTML using semantic elements and accessibility attributes. They see how good structure supports screen readers and search engines.

[HTML5](#)[ACCESSIBILITY](#)[SEMANTICS](#)

2 CSS Layouts with Flexbox and Grid

Learners build responsive page layouts using Flexbox and CSS Grid. They practice aligning, spacing and reordering content across screen sizes.

[CSS](#)[FLEXBOX](#)[GRID](#)

3 Responsive Design and Media Queries

Learners adapt layouts to mobile, tablet and desktop using media queries and relative units. They follow a mobile first workflow.

[RESPONSIVE DESIGN](#)[MEDIA QUERIES](#)[MOBILE FIRST](#)

4 Modern JavaScript ES6 Essentials

Learners use let, const, arrow functions, destructuring, spread and template literals. These features appear in nearly every React codebase.

[ES6](#)[DESTRUCTURING](#)[ARROW FUNCTIONS](#)

5 Asynchronous JavaScript with Promises and Async Await

Learners handle asynchronous work using promises and async await syntax. They fetch data from a public API and handle errors.

[PROMISES](#)[ASYNC AWAIT](#)[FETCH API](#)

6 DOM Manipulation and Developer Tooling

Learners manipulate the DOM directly and debug with browser developer tools. They also set up Node, npm and Vite for the coming modules.

[DOM](#)[DEVTOOLS](#)[NPM](#)

M4.2 React Fundamentals and Components · 6 topics

Learners understand how React thinks and build UIs from reusable components.

1 Introduction to React and Project Setup

Learners create a React project with Vite and tour its structure. They learn why component based UI libraries exist.

REACT

VITE

PROJECT SETUP

2 JSX Syntax and Rendering Expressions

Learners write JSX and embed JavaScript expressions in markup. They learn the differences between JSX and HTML.

JSX

EXPRESSIONS

RENDERING

3 Components and Props

Learners create function components and pass data through props. They practice composing small components into larger screens.

COMPONENTS

PROPS

COMPOSITION

4 Conditional Rendering and Lists with Keys

Learners render UI conditionally and map arrays to lists of elements. They learn why stable keys matter for performance and correctness.

CONDITIONAL RENDERING

LISTS

KEYS

5 Handling Events and Forms

Learners respond to clicks and input changes and build controlled forms. They validate input before submission.

EVENTS

CONTROLLED FORMS

VALIDATION

6 Styling React Apps with CSS Modules and Tailwind

Learners style components using CSS Modules and utility classes from Tailwind CSS. They compare the tradeoffs of each approach.

CSS MODULES

TAILWIND CSS

STYLING

M4.3 State Management and Hooks · 6 topics

Learners manage dynamic data in components using React hooks and well known patterns.

1 The useState Hook and Local State

Learners store and update component state with useState. They learn immutable update patterns for objects and arrays.

USESTATE

STATE

IMMUTABILITY

2 The useEffect Hook and Side Effects

Learners run side effects such as data loading and subscriptions with `useEffect`. They manage dependency arrays and cleanup functions.

USEFFECT

SIDE EFFECTS

LIFECYCLE

3 Lifting State Up and Component Communication

Learners share state between sibling components by lifting it to a common parent. They design clean data flow with callbacks.

LIFTING STATE

DATA FLOW

CALLBACKS

4 useRef and useMemo and useCallback

Learners access DOM nodes with `useRef` and optimize expensive work with memoization hooks. They learn when optimization is actually needed.

USEREF

USEMEMO

USECALLBACK

5 Context API for Shared State

Learners share global values such as theme and user session using Context. They avoid prop drilling in deeper component trees.

CONTEXT API

GLOBAL STATE

PROVIDERS

6 Custom Hooks and Reducers

Learners extract reusable logic into custom hooks and manage complex state with `useReducer`. They write cleaner and more testable components.

CUSTOM HOOKS

USEREDUCER

REUSABILITY

M4.4 Routing and Application Architecture · 6 topics

Learners structure multi page single page applications with routing and scalable folder organization.

1 Client Side Routing with React Router

Learners configure routes and navigation links with React Router. They understand how single page apps change views without reloads.

REACT ROUTER

ROUTES

SPA

2 Dynamic Routes and URL Parameters

Learners build detail pages using route parameters and query strings. They read and update URL state from components.

URL PARAMS

QUERY STRINGS

NAVIGATION

3 Nested Layouts and Protected Routes

Learners create shared layouts with nested routes and guard private pages based on login status. They redirect unauthorized users.

NESTED ROUTES

ROUTE GUARDS

LAYOUTS

4 Project Structure and Reusable UI Patterns

Learners organize features, components and utilities into a maintainable folder structure. They apply container and presentational patterns.

ARCHITECTURE

FOLDER STRUCTURE

REUSABILITY

5 Global State with Redux Toolkit

Learners manage large application state using Redux Toolkit slices and the store. They connect components with hooks.

REDUX TOOLKIT

STORE

SLICES

6 Error Boundaries and Code Splitting

Learners catch rendering errors with error boundaries and load pages lazily with Suspense. They improve resilience and startup speed.

ERROR BOUNDARIES

LAZY LOADING

SUSPENSE

M4.5 Connecting React to Spring Boot APIs · 6 topics

Learners integrate the React frontend with the REST backend built earlier in the course.

1 Calling REST APIs with Fetch and Axios

Learners send GET, POST, PUT and DELETE requests to the Spring Boot backend. They compare Fetch with Axios and centralize API calls.

AXIOS

FETCH

REST INTEGRATION

2 Handling CORS and Environment Configuration

Learners configure CORS on the Spring Boot server and manage API URLs with environment variables. They run frontend and backend together locally.

CORS

ENVIRONMENT VARIABLES

CONFIGURATION

3 Data Fetching with TanStack Query

Learners cache, refetch and synchronize server data using TanStack Query. They reduce boilerplate for loading and error states.

TANSTACK QUERY

CACHING

SERVER STATE

4

Loading and Error States and User Feedback

Learners design skeletons, spinners and toast messages for slow or failed requests. They build interfaces that stay clear under poor network conditions.

LOADING STATES

ERROR HANDLING

UX

5

JWT Authentication in the Frontend

Learners implement login and logout against the secured backend and store tokens safely. They attach tokens to requests with interceptors.

JWT

AUTHENTICATION

INTERCEPTORS

6

Building a CRUD Dashboard with Pagination and Search

Learners create a full dashboard that lists, filters, creates, edits and deletes records through the API. They add pagination and debounced search.

CRUD

PAGINATION

SEARCH

M4.6 Quality, Performance and Deployment · 6 topics

Learners test, optimize and ship their React applications to production.

1

Component Testing with Jest and React Testing Library

Learners write unit tests that render components and simulate user behavior. They learn to test what users see rather than implementation details.

JEST

REACT TESTING LIBRARY

UNIT TESTING

2

Mocking APIs and Integration Tests

Learners mock network calls with Mock Service Worker and test whole features. They verify loading, success and failure flows.

MOCK SERVICE WORKER

INTEGRATION TESTING

MOCKING

3

TypeScript Basics for React

Learners add types to props, state and API responses using TypeScript. They catch bugs early and improve editor support.

TYPESCRIPT

TYPED PROPS

TYPE SAFETY

4

Performance Profiling and Optimization

Learners profile renders with React DevTools and reduce unnecessary updates. They apply memoization and list virtualization where useful.

PROFILING

OPTIMIZATION

VIRTUALIZATION

5 Web Accessibility and SEO Best Practices

Learners audit apps with Lighthouse and fix accessibility and metadata issues. They improve usability for all users.

LIGHTHOUSE

ARIA

SEO

6 Building and Deploying the React App

Learners create production builds and deploy to a hosting platform with a CI pipeline. They connect the live frontend to the deployed backend.

PRODUCTION BUILD

DEPLOYMENT

CI/CD

05**SECTION 5****AI Integration with LLMs and Spring AI**

Learners progress from generative AI fundamentals to building production-ready AI features in Spring Boot applications using prompts, embeddings, retrieval and agents.

6 modules · 36 topics

M5.1 Foundations of Generative AI and LLMs · 6 topics

Introduces how large language models work and how to choose and access them responsibly.

1 Introduction to Generative AI

Understand what generative AI is and how it differs from traditional machine learning. Explore real-world use cases in full stack products.

GENERATIVE AI

AI CONCEPTS

2 How Large Language Models Work

Learn about tokens, context windows, transformers and next-token prediction at a practical level. See how these ideas affect cost and output quality.

LLM BASICS

TOKENS

TRANSFORMERS

3 Model Providers and Selection

Compare OpenAI, Anthropic, Google and open source models on cost, latency and capability. Learn how to pick a model for a given feature.

MODEL SELECTION

COST ANALYSIS

4 Working with LLM APIs using Java HTTP Clients

Call a hosted LLM REST API directly from Java using HttpClient and JSON parsing. Handle API keys, errors and timeouts safely.

REST CALLS

JAVA HTTPCLIENT

API KEYS

5 Model Parameters and Output Control

Tune temperature, top-p, max tokens and stop sequences to shape responses. Observe how each setting changes output behavior.

TEMPERATURE

MAX TOKENS

TUNING

6 Responsible AI, Safety and Privacy

Discuss bias, hallucinations, data privacy and compliance concerns when using LLMs. Apply practical guardrails to protect users and data.

AI ETHICS

DATA PRIVACY

GUARDRAILS

M5.2 Prompt Engineering Essentials · 6 topics

Teaches systematic techniques for writing prompts that produce reliable and useful output.

1 Anatomy of an Effective Prompt

Break prompts into role, context, task, constraints and format. Practice rewriting weak prompts into strong ones.

PROMPT DESIGN

CLARITY

2 Zero Shot and Few Shot Prompting

Use examples inside prompts to steer model behavior. Compare results with and without examples.

FEW SHOT

ZERO SHOT

EXAMPLES

3 Chain of Thought and Reasoning Prompts

Encourage step by step reasoning for complex tasks. Learn when reasoning prompts help and when they add unnecessary cost.

REASONING

CHAIN OF THOUGHT

4 System Prompts and Role Based Instructions

Define assistant personas and persistent rules using system messages. Understand message roles in chat style APIs.

SYSTEM PROMPT

CHAT ROLES

5 Structured Outputs and JSON Responses

Force models to return JSON that matches a schema and validate it in Java. Handle malformed responses gracefully.

JSON OUTPUT

VALIDATION

SCHEMA

6 Prompt Testing and Iteration

Build a small evaluation set to compare prompt versions. Track improvements with repeatable tests.

EVALUATION

PROMPT VERSIONING

TESTING

M5.3 Spring AI Fundamentals · 6 topics

Covers the core abstractions of Spring AI for building provider independent AI features in Spring Boot.

1 Spring AI Overview and Project Setup

Add Spring AI dependencies and configure a provider in a Spring Boot project. Understand the framework architecture and goals.

SPRING AI

MAVEN

CONFIGURATION

2 ChatClient and ChatModel APIs

Send prompts and receive responses using the fluent ChatClient. Learn how it abstracts differences between providers.

CHATCLIENT

SPRING BOOT

ABSTRACTION

3 Prompt Templates and Message Types

Create reusable parameterized prompts with templates. Use system, user and assistant messages in Spring AI.

PROMPT TEMPLATES

MESSAGES

4 Streaming Responses with Reactive Support

Stream tokens to clients as they are generated using Flux and server sent events. Improve perceived responsiveness in chat features.

STREAMING

SSE

REACTIVE

5 Structured Output Converters to Java Objects

Map model responses directly into Java records and beans. Reduce manual parsing and improve type safety.

OUTPUT CONVERTER

JAVA RECORDS

TYPE SAFETY

6 Chat Memory and Conversation Context

Maintain multi turn conversations using advisors and memory stores. Manage context length and conversation identifiers.

CHAT MEMORY

ADVISORS

CONTEXT

M5.4 Embeddings, Vector Databases and RAG · 6 topics

Shows how to ground model answers in private data using embeddings and retrieval augmented generation.

1 Understanding Embeddings and Semantic Search

Learn how text becomes vectors and how similarity search finds related content. Compare keyword search with semantic search.

EMBEDDINGS

SEMANTIC SEARCH

2

Vector Databases and Spring AI Vector Store

Store and query vectors using pgvector or a similar store through the VectorStore interface. Understand indexing and similarity metrics.

VECTOR STORE

PGVECTOR

SIMILARITY

3

Document Loading and Chunking Strategies

Ingest PDFs and text files using document readers and split them into effective chunks. Explore how chunk size affects retrieval quality.

ETL PIPELINE

CHUNKING

DOCUMENTS

4

Building a Retrieval Augmented Generation Pipeline

Combine retrieval and generation so the model answers using relevant passages. Implement the pipeline with Spring AI advisors.

RAG

RETRIEVAL

ADVISORS

5

Improving RAG Quality

Apply metadata filtering, query rewriting and reranking to improve answers. Reduce hallucinations with source citations.

RERANKING

METADATA FILTERS

CITATIONS

6

Evaluating and Monitoring RAG Systems

Measure relevance and faithfulness of answers using test questions. Track latency and token usage in production.

EVALUATION

MONITORING

OBSERVABILITY

M5.5

Function Calling, Agents and Advanced AI Features · 6 topics

Extends LLM applications with tools, agent workflows and multimodal capabilities.

1

Function Calling and Tool Use Concepts

Understand how models request tool execution and how results return to the conversation. Review common tool use patterns.

FUNCTION CALLING

TOOLS

2

Implementing Tools in Spring AI

Expose Java methods as tools that the model can call, such as database lookups or service calls. Validate arguments and handle failures.

TOOL CALLING

SPRING BEANS

VALIDATION

3

Model Context Protocol Basics

Learn how the Model Context Protocol standardizes connections between models and external tools. Connect a Spring application to an MCP server.

MCP

INTEGRATION

4

Building AI Agents and Multi Step Workflows

Design agents that plan, call tools and iterate toward a goal. Compare simple chains with routing and orchestration patterns.

AI AGENTS

ORCHESTRATION

WORKFLOWS

5

Multimodal AI with Images and Audio

Send images to vision models and generate or transcribe audio using Spring AI. Build features such as image description and speech to text.

MULTIMODAL

VISION

SPEECH

6

Running Local Models with Ollama

Run open source models locally and connect them to Spring AI. Weigh privacy and cost benefits against performance tradeoffs.

OLLAMA

LOCAL LLMS

OPEN SOURCE

M5.6

Productionizing AI Powered Full Stack Applications · 6 topics

Integrates AI services with the React frontend and prepares them for secure and reliable production use.

1

Designing AI Endpoints in Spring Boot

Create REST and streaming endpoints that wrap AI logic cleanly. Apply layered design and DTOs for AI features.

REST API

DTOS

ARCHITECTURE

2

Building a Chat Interface in React

Create a chat UI that consumes streaming responses from the backend. Manage message state, loading and error handling.

REACT

STREAMING UI

STATE

3

Securing AI APIs and Preventing Prompt Injection

Protect endpoints with authentication and rate limiting. Defend against prompt injection and data leakage.

SECURITY

PROMPT INJECTION

RATE LIMITING

4

Cost Control, Caching and Resilience

Reduce spend with caching, token limits and model routing. Add retries and fallbacks for provider failures.

CACHING

RESILIENCE

COST CONTROL

5 **Observability and Testing of AI Features**

Log prompts and responses safely and track metrics with Micrometer. Write tests using mocked models and evaluation datasets.

OBSERVABILITY

MICROMETER

TESTING

6 **Capstone Build of an AI Powered Assistant**

Combine RAG, tools and a React frontend into a complete document aware assistant. Present the design decisions and results to peers.

CAPSTONE

FULL STACK

INTEGRATION

06

SECTION 6

DevOps Deployment and Career Preparation

Learners package, test, containerize, automate and deploy their Java AI full stack applications to the cloud and then prepare for interviews and job placement with a polished portfolio.

6 modules · 36 topics

M6.1 **Version Control and Collaborative Workflows** · 6 topics

Learners master Git and GitHub practices used by professional engineering teams to manage code and collaborate.

1 **Git Fundamentals and Local Repositories**

Understand commits, staging, history and the Git object model. Initialize repositories and track changes in a Java project.

GIT

VERSION CONTROL

COMMITTS

2 **Branching and Merging Strategies**

Create feature branches and merge them using fast-forward and three-way merges. Compare Git Flow and trunk-based development.

BRANCHING

MERGING

GIT FLOW

3 **Resolving Conflicts and Rewriting History**

Resolve merge conflicts confidently and use rebase, stash and cherry-pick. Learn when rewriting history is safe.

REBASE

CONFLICT RESOLUTION

STASH

4 **GitHub Collaboration and Pull Requests**

Work with remotes, forks and pull requests. Practice structured code reviews and issue tracking.

GITHUB

PULL REQUESTS

CODE REVIEW

5 Semantic Versioning and Release Management

Tag releases, write changelogs and apply semantic versioning to a Spring Boot application.

[SEMANTIC VERSIONING](#)[TAGS](#)[RELEASES](#)**6 Build Tools with Maven and Gradle**

Manage dependencies, profiles and build lifecycles to produce runnable JAR files for deployment.

[MAVEN](#)[GRADLE](#)[BUILD LIFECYCLE](#)**M6.2 Testing and Code Quality Assurance** · 6 topics

Learners build automated test suites and quality gates that keep full stack applications reliable.

1 Unit Testing with JUnit 5

Write isolated unit tests for service and utility classes. Use assertions, parameterized tests and lifecycle annotations.

[JUNIT 5](#)[UNIT TESTING](#)[ASSERTIONS](#)**2 Mocking with Mockito**

Replace dependencies with mocks and spies to test business logic in isolation. Verify interactions and stub responses.

[MOCKITO](#)[MOCKING](#)[TEST DOUBLES](#)**3 Integration Testing Spring Boot APIs**

Test controllers and repositories using MockMvc and SpringBootTest. Validate request handling and database behavior.

[MOCKMVC](#)[SPRINGBOOTTEST](#)[API TESTING](#)**4 Testing with Testcontainers**

Run real databases in disposable containers during tests for reliable and repeatable integration checks.

[TESTCONTAINERS](#)[INTEGRATION TESTS](#)[DATABASES](#)**5 Frontend Testing with Jest and React Testing Library**

Test React components and user interactions. Mock API calls and assert rendered output.

[JEST](#)[REACT TESTING LIBRARY](#)[COMPONENT TESTS](#)**6 Code Quality with SonarQube and Coverage Reports**

Measure coverage with JaCoCo and analyze code smells and vulnerabilities. Define quality gates for the pipeline.

[SONARQUBE](#)[JACOCO](#)[CODE QUALITY](#)

M6.3 Containerization with Docker · 6 topics

Learners package the full stack application and its dependencies into portable containers.

1 Container Concepts and Docker Setup

Understand images, containers and registries and how they differ from virtual machines. Install Docker and run first containers.

DOCKER

CONTAINERS

IMAGES

2 Writing Dockerfiles for Spring Boot

Build optimized images for the backend using layered JARs and slim base images.

DOCKERFILE

IMAGE OPTIMIZATION

SPRING BOOT

3 Containerizing the React Frontend with Nginx

Use multi-stage builds to compile the React app and serve it through Nginx.

MULTI-STAGE BUILDS

NGINX

REACT

4 Docker Compose for Multi-Service Applications

Orchestrate frontend, backend and database services locally with networks, volumes and environment files.

DOCKER COMPOSE

NETWORKING

VOLUMES

5 Managing Secrets and Configuration

Handle environment variables, API keys and LLM credentials securely across environments.

SECRETS

ENVIRONMENT CONFIG

SECURITY

6 Publishing Images to Docker Hub and Registries

Tag, push and pull images from public and private registries. Apply image scanning basics.

DOCKER HUB

REGISTRIES

IMAGE SCANNING

M6.4 CI/CD Pipelines and Automation · 6 topics

Learners automate building, testing and releasing applications with continuous integration and delivery.

1 CI/CD Principles and Pipeline Design

Learn the stages of a modern delivery pipeline and the benefits of automation. Design a pipeline for the capstone project.

CI/CD

PIPELINE DESIGN

AUTOMATION

2 GitHub Actions Workflows

Write workflow files that trigger on pushes and pull requests to build and test the application.

GITHUB ACTIONS

YAML

WORKFLOWS

3

Automated Build and Test Stages

Run Maven builds, backend tests and frontend tests in the pipeline with caching and test reports.

AUTOMATED TESTING

CACHING

BUILD STAGE

4

Building and Pushing Docker Images in the Pipeline

Authenticate to registries and publish versioned images automatically after successful builds.

IMAGE PUBLISHING

PIPELINE SECRETS

VERSIONING

5

Continuous Deployment Strategies

Compare rolling, blue-green and canary deployments. Implement rollback plans for failed releases.

BLUE-GREEN

CANARY

ROLLBACK

6

Introduction to Jenkins and Pipeline as Code

Explore Jenkins and write a declarative Jenkinsfile to understand alternative CI servers.

JENKINS

JENKINSFILE

PIPELINE AS CODE

M6.5 Cloud Deployment and Production Operations · 6 topics

Learners deploy the application to the cloud and operate it with monitoring, security and scalability in mind.

1

Cloud Fundamentals with AWS

Understand regions, IAM, compute, storage and networking services. Set up a secure account with budget alerts.

AWS

IAM

CLOUD BASICS

2

Deploying the Backend on EC2 and Elastic Beanstalk

Launch the Spring Boot service on virtual machines and managed platforms. Configure security groups and environment variables.

EC2

ELASTIC BEANSTALK

DEPLOYMENT

3

Managed Databases with Amazon RDS

Provision a managed relational database and connect the application securely. Configure backups and access rules.

AMAZON RDS

DATABASE HOSTING

BACKUPS

4

Hosting the Frontend with S3 and CloudFront

Publish the React build to object storage and distribute it through a CDN with HTTPS and custom domains.

S3

CLOUDFRONT

HTTPS

5

Kubernetes Essentials for Java Applications

Learn pods, deployments and services. Deploy a containerized Spring Boot app to a Kubernetes cluster.

KUBERNETES

PODS

DEPLOYMENTS

6

Monitoring Logging and Observability

Expose Actuator metrics and collect logs. Visualize health with Prometheus and Grafana and set alerts.

ACTUATOR

PROMETHEUS

GRAFANA

M6.6

Career Preparation and Capstone Presentation · 6 topics

Learners convert their skills into a strong portfolio and the confidence to succeed in Java and AI engineering interviews.

1

Capstone Project Finalization and Demo Deployment

Polish the capstone AI full stack application, write documentation and deploy a live demo with a public URL.

CAPSTONE

DOCUMENTATION

LIVE DEMO

2

Building a Professional GitHub Portfolio and README

Structure repositories, write clear READMEs with architecture diagrams and showcase projects to recruiters.

PORTFOLIO

README

PERSONAL BRANDING

3

Resume and LinkedIn Optimization for Java Roles

Craft an ATS-friendly resume and LinkedIn profile that highlight projects, impact and technical keywords.

RESUME

LINKEDIN

ATS

4

Data Structures and Algorithms Interview Practice

Solve common coding interview problems in Java and explain time and space complexity clearly.

CODING INTERVIEWS

ALGORITHMS

COMPLEXITY

5

System Design and Technical Interview Preparation

Answer Java, Spring Boot and system design questions. Design scalable architectures including AI-powered features.

SYSTEM DESIGN

SPRING INTERVIEW

ARCHITECTURE

6

Mock Interviews and Job Search Strategy

Complete mock technical and behavioral interviews with feedback. Plan applications, networking and offer negotiation.

MOCK INTERVIEWS

JOB SEARCH

NEGOTIATION

07 · SMES



Meet Our SME Board Members

Expert professionals guiding your journey through real-world industry knowledge and practical experience.

**SME BOARD MEMBER · SENIOR SME**

Karteek Ch

Associate

Tata Consultancy Services · Subject Matter Expert

Technology professional, mentor, and engineer specializing in Oracle, AI, IoT, and emerging technologies.

Technology professional and TCS Engineer specializing in Oracle, Java, Python, AI, and IoT. An experienced Subject Matter Expert and Mentor at Intern Stump, passionate about practical learning, emerging technologies, and industry-ready student development.

**SME BOARD MEMBER · SENIOR SME**

Harshavardhan Kumar

Lead Consultant

Intern Stump Solutions · Subject Matter Expert

Computer Science professional specializing in AI, software development, education, and UI/UX consulting.

Harshavardhan Kumar Jajam is a technology professional, educator, and Lead UI/UX Consultant at Intern Stump Solutions, specializing in Artificial Intelligence, Python, Data Science, Full-Stack Development, cloud computing, and industry-oriented technical education.

**SME BOARD MEMBER · SENIOR SME**

Ruman Aktar

Founder

Rumandev.com · Subject Matter Expert

Guiding backend API integration, full-stack database architecture, and scalable Web application pipelines.

Technology professional specializing in backend API integration, full-stack database architecture, and scalable web application pipelines, with a focus on system performance, seamless data flow, application reliability, and efficient software development practices.

08 · TOOLS AND TECHNIQUES



Key Tools and Techniques

Master industry-standard methodologies used by top-tier professionals, gaining practical exposure to tools, frameworks, and strategies for results.

INDUSTRY STANDARD

01

Java JDK 21

The core language and runtime used to write robust, object oriented backend applications.

MOST DEMANDED

02

Spring Boot

The leading Java framework for building production-ready REST APIs and microservices quickly.

CORE BACKEND

03

Spring Data JPA and Hibernate

ORM tools that map Java objects to database tables and simplify data access.

WIDELY USED

04

MySQL

A widely used relational database for storing and querying application data with SQL.

FRONTEND LEADER

05

React

A component based JavaScript library for building fast and interactive user interfaces.

SECURITY

06

Spring Security and JWT

Techniques for authentication and authorization to protect APIs and user data.

AI READY

07

Spring AI and LLM APIs

Libraries and APIs for connecting Java applications to large language models for chat and generation.

ESSENTIAL

08

Git and GitHub

Version control and collaboration tools used in every professional development team.

DEVOPS SKILL

09

Docker

Container technology used to package and deploy applications consistently across environments.

Many More Tools Integrated in Curriculum

09 · COURSE PROJECTS



Projects You Will Work On

Build a strong portfolio by working on industry-relevant projects that simulate real job scenarios, helping you gain confidence and career-ready skills.

INDUSTRY-LED PROJECT

01

Console Banking System

Build a Java console app for accounts, deposits, withdrawals and transaction history using OOP principles. Apply exception handling and collections to keep the logic clean and reliable.

Duration: 6-10 hours

INDUSTRY-LED PROJECT

02

Student Management REST API

Create a Spring Boot REST API with full CRUD operations backed by MySQL and JPA. Add validation and global exception handling and test endpoints with Postman.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

03

Secure Login and Role Based Access

Implement registration and login using Spring Security and JWT tokens. Restrict endpoints by user role such as admin and user.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

04

Task Manager Web App

Develop a React frontend that consumes your Spring Boot API for creating and tracking tasks. Use hooks, routing and form handling for a smooth user experience.

Duration: 10-14 hours

INDUSTRY-LED PROJECT

05

E-Commerce Store

Build a full stack shopping app with product listings, cart, orders and admin management. Integrate a payment sandbox and manage state across the React application.

Duration: 14-20 hours

INDUSTRY-LED PROJECT

06

AI Chatbot Assistant

Create a chatbot that calls an LLM API through a Spring Boot service and shows responses in a React chat interface. Design prompts and manage conversation history.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

07

Document Q&A with RAG

Build a system that lets users upload documents and ask questions answered from their content. Use embeddings and a vector store to retrieve relevant context for the LLM.

Duration: 10-16 hours

INDUSTRY-LED PROJECT

08

AI Resume Analyzer

Develop an app that scores resumes against a job description and suggests improvements using an LLM. Store results in a database and display them on a dashboard.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

09

Dockerized Capstone Deployment

Containerize your full stack AI application with Docker and deploy it to a cloud platform. Set up a CI pipeline with GitHub Actions to automate builds and tests.

Duration: 8-12 hours



10 · COURSE ENROLLMENT

Java AI Full Stack Development

Learn Java, Spring Boot, React and AI integration in live mentor-led sessions. Build 9 hands-on projects and become job-ready as a full stack developer who can ship AI-powered applications.

Starts	To be announced	Enrolment closes	Batch details soon
Duration	6 Months	Timings	To be announced
Batch	Batch details updating soon	Batch status	Batch open · Batch dates announced soon

Your plan

One-time fee · inclusive of GST

Plan 1

Course + Live Work Experience

Build real projects and earn a live work experience certificate.

₹7,999~~₹30,000~~ · 73% OFF

Live classes & completion certificate	✓ Included
---------------------------------------	------------

Live projects	✓ Included
---------------	------------

Live work experience certificate	✓ Included
----------------------------------	------------

8C appraisal report	✓ Included
---------------------	------------

SME AI (Insyrio)	✓ Included
------------------	------------

Live chat with SME	✗ Not included
--------------------	----------------

Placement assistance	✗ Not included
----------------------	----------------

Plan perks	15% off across Intern Stump - coins, self-paced courses, masterclasses, mock interviews 20% off membership plans
------------	---

Also included

- Live classes with industry SMEs
- Course completion certificate
- Live projects reviewed by your SME

Good to know

Completion certificate

Verifiable ID - a sample certificate can be previewed on the course page.

8C Appraisal System

How the eight Cs work - see the 8C Appraisal section of this brochure.

Part-pay with iSure Coins

Use your coin balance at checkout.

Need help choosing?

See the FAQs or book a free demo.

11 · 8C APPRAISAL



World's First Ed-Tech with 8C Appraisal

Verified 8C Appraisal Certificate: every candidate receives an official corporate readiness report shared directly with hiring partners upon course completion.

The 8C Appraisal is a proprietary evaluation matrix measuring holistic performance across 8 career pillars for authentic proof-of-work:

Communication

Configuration

Chronicle

Completeness

Cognizance

Conception

Comprehension

Case Study



Frequently Asked Questions

Find answers to common questions about our learning tracks, placement assistance, and 8C evaluation model, or connect directly with an academic consultant.

Q1

Do I need prior coding experience to join Java AI Full Stack Development?

No. The course starts from Java fundamentals and gradually moves to advanced full stack and AI topics, so complete beginners can follow along.

Q2

How long is the course?

The program runs for 6 Months with live mentor-led sessions, practice assignments and project work throughout.

Q3

What is the course fee?

The current fee is ₹7,999 against the actual fee of ₹30,000, which gives you a discount of 73%.

Q4

When does the next batch start?

The next batch is the next batch and begins on soon. Sessions are held at to be announced.

Q5

Are the classes live or recorded?

Classes are taught live by mentors so you can ask questions and get feedback in real time. Session recordings are shared for revision.

Q6

Will I build real projects?

Yes. You build nine hands-on projects, including REST APIs, a React frontend, an e-commerce store, AI chatbots and a deployed capstone.

Q7

Do I need to know AI or machine learning beforehand?

No. You learn to integrate existing AI models through APIs and Spring AI, so no prior machine learning knowledge is needed.

Q8

Will this course help me get a job?

The Technical track includes portfolio building, mock interviews, resume guidance and coding practice to prepare you for developer roles. Placement outcomes depend on your effort and performance.

Ready to take the next step?

Java AI Full Stack Development · Live Course



Scan to view details & enrol

internstump.com/courses/java-ai-full-stack-development



WEBSITE
internstump.com



EMAIL
info@internstump.com



CALL / WHATSAPP
+91 7396168247



Intern Stump Solutions · Generated on 26 Oct 2026, 08:07 PM IST from the live details on internstump.com. Prices, batches and schedules can change.