Restful API Development with Spring Boot and Docker

Duration: 5 days / 40 hrs

Prerequisites: Knowledge of Backend Development and Databases

Day 1: Web Frameworks & Spring Boot Basics

Topics:

- Introduction to RESTful APIs
  - HTTP Methods (GET, POST, PUT, DELETE)
  - Request/Response format (JSON)
- Overview of Web Frameworks
  - Why Spring?
  - Spring vs Spring Boot
- Spring Boot Architecture
  - Auto Configuration
  - Spring Initializr, Maven Structure
  - application.properties setup
- Introduction to Inversion of Control (IoC) & Dependency Injection (DI)

🧪 **Lab 1:**

- Use Spring Initializr to create your first Spring Boot project.
- Create a HelloController that returns a simple JSON message via a REST endpoint.

Day 2: Spring Boot – Beans, MVC & CRUD API

Topics:

- Understanding Spring Beans & Autowiring
- Spring MVC Fundamentals
  - Layers: Controller, Service, Repository
- CRUD Operations using Spring Boot
  - Create, Read, Update, Delete endpoints
  - Path & Query Parameters
  - Error and Exception Handling

🧪 **Lab 2:**

- Build a basic CRUD API for a Student entity (in-memory H2 DB).
- Add endpoints: GET /students, POST /students, PUT /students/{id}, DELETE /students/{id}.

Day 3: Databases – SQL, JPA, MySQL, and MongoDB

Topics:

- Introduction to Relational Databases
  - Tables, Relationships, Primary & Foreign Keys
- Spring Data JPA with MySQL
  - @Entity, @Repository, @Id, @GeneratedValue
  - Basic JPQL
  - Pagination & Sorting with Spring Data
- NoSQL Basics
  - SQL vs NoSQL overview
- Connecting Spring Boot with MongoDB using Spring Data Mongo

🧪 **Lab 3:**

- Modify the Student CRUD app to store data in **MySQL**.
- Implement pagination (/students?page=0&size=5) and sorting (/students?sort=name).
- BONUS: Build a separate simple MongoDB-based Product API with POST and GET.

Day 4: Git for Beginners & Testing with JUnit

Topics:

- **Version Control with Git**
  - git init, clone, add, commit, push, pull
  - Creating and switching branches
  - Merge and resolving simple conflicts
- **Testing Basics**
  - Unit vs Integration vs System Testing
  - JUnit 5 Overview
  - Introduction to Mockito for mocking

🧪 **Lab 4:**

- Initialize a Git repository for your CRUD project.

- Practice basic Git commands: stage, commit, push to GitHub.
- Write JUnit tests for the Service layer (e.g., StudentService).
- Mock the Repository using Mockito.

Day 5: Docker Essentials & Final Project

Topics:

- Containerization Overview
  - What is Docker & why it's useful
  - Images vs Containers
- Writing a Dockerfile for Spring Boot app
- Running Spring Boot in Docker
- Overview of Docker Registry

🧪 **Lab 5:**

- Write a Dockerfile to containerize your CRUD API.
- Build and run the Docker image locally.
- Tag and push the image to Docker Hub (if time permits).

Final Project (Cumulative Lab):

- Build a complete Spring Boot application for managing Books:
  - RESTful endpoints with CRUD
  - Connect to MySQL database
  - Add pagination & sorting
  - Add JUnit tests for Services
  - Containerize the app with Docker
  - Push project to GitHub