Angular Full Stack Development with Java Spring Boot

Prerequisites: Knowledge of Java and Web Programming

Day 1: Introduction and Basics

Objective:

Lay the foundation for understanding Angular and Spring Boot, covering their architectures, setup, and essential concepts.

Topics:

1. **Introduction to Full-Stack Development**
    - What is Full-Stack Development?
    - Overview of Angular and Spring Boot.
    - Benefits of combining Angular with Spring Boot.
2. **Setting Up the Development Environment**
    - Installing Node.js, Angular CLI, Java, and Maven.
    - Setting up an IDE (IntelliJ, VS Code, or Eclipse).
    - Configuring Postman for API testing.
3. **Basics of Angular**
    - TypeScript overview.
    - Creating an Angular project.
    - Angular components, templates, and modules.
4. **Basics of Spring Boot**
    - Spring Boot project setup.
    - REST APIs: Introduction and basic implementation.
    - Using Spring Initializr.

Lab:

- **Angular:** Create a "Hello World" application and explore its folder structure.
- **Spring Boot:** Build and test a simple REST API that returns a JSON response.

Day 2: Angular Deep Dive

Objective:

Explore the core concepts of Angular, including data binding, directives, and services.

Topics:

1. **Data Binding in Angular**
    - Property binding.
    - Event binding.
    - Two-way binding.
2. **Directives**
    - Built-in directives (ngIf, ngFor, etc.).
    - Creating custom directives.
3. **Services and Dependency Injection**
    - Creating and using services.
    - Dependency injection mechanism.
4. **Routing and Navigation**
    - Setting up routing in Angular.
    - Lazy loading and route guards.

Lab:

- Create a multi-page Angular application with navigation and data-binding.
- Add a service to fetch mock data and display it in a component.

Day 3: Spring Boot Deep Dive

Objective:

Dive deeper into Spring Boot, focusing on REST APIs, database integration, and exception handling.

Topics:

1. **REST API Development**
    - HTTP methods: GET, POST, PUT, DELETE.
    - Mapping endpoints using annotations.
2. **Database Integration**
    - Configuring Spring Data JPA.
    - Connecting to a database (e.g., MySQL or PostgreSQL).
    - CRUD operations with JPA.
3. **Exception Handling**
    - Custom exceptions.

    - Global exception handling using @ControllerAdvice.

4. **Spring Boot Security Basics**
    - Introduction to securing REST APIs.
    - JWT overview and implementation.

Lab:

- Build a Spring Boot REST API connected to a MySQL database for managing a simple entity (e.g., "Employee").
- Add CRUD endpoints with basic exception handling.

Day 4: Angular-Spring Boot Integration

Objective:

Learn how to connect the Angular frontend with the Spring Boot backend.

Topics:

1. **Integrating Angular with Spring Boot**
    - Making HTTP calls using Angular's HttpClient module.
    - Consuming REST APIs.
2. **Handling Data in Angular**
    - Using services to manage data flow.
    - Displaying backend data in components.
3. **File Uploads**
    - Implementing file uploads in Angular.
    - Handling files in Spring Boot.
4. **Error Handling**
    - Angular HTTP error handling.
    - Backend error propagation.

Lab:

- Develop an Angular frontend to consume the Spring Boot REST API built on Day 3.
- Display and manage database records through Angular.

Day 5: Advanced Features and Deployment

Objective:

Cover advanced features like authentication, testing, and deployment of the full-stack application.

Topics:

1. **Authentication and Authorization**
    - Implementing JWT-based authentication in Spring Boot.
    - Creating a login system in Angular.
2. **Testing**
    - Writing unit tests in Angular using Jasmine and Karma.
    - Writing JUnit tests for Spring Boot APIs.
3. **Deployment**
    - Building and deploying the Angular app.
    - Packaging the Spring Boot app as a JAR/WAR.
    - Deploying to a server .
4. **Best Practices**
    - Code organization.
    - Error logging and monitoring.
    - Performance optimization.

Lab:

- Secure the Angular-Spring Boot application with JWT.
- Deploy the full-stack application