KOENIG
step forward

Spring Boot with Full Stack Design and Development

Pre-requisites: Basic understandings on Java Programming, Spring Framework, Spring Boot

Duration: 7 days

Theory of Contents:

1. **Introduction to REST API Principles**

   Understanding REST Architecture

   Principles of REST: Statelessness, client-server separation, uniform interfaces.

   HTTP methods: GET, POST, PUT, DELETE, PATCH, and their appropriate use cases.

   API design best practices: Resource naming, versioning, and HATEOAS.

2. **Building REST APIs with Spring Boot**

   Setting up a Spring Boot project.

   Creating RESTful endpoints and mapping HTTP methods to Java methods. Handling request parameters, path variables, and request bodies.

   Error Handling in REST APIs

   Custom exception handling in Spring Boot.

   Designing standard error responses.

   Implementing global exception handling using @ControllerAdvice.

3. **Advanced REST API Development**

   Securing REST APIs

   Authentication and Authorization: OAuth2 and JWT basics. Implementing security in Spring Boot using Spring Security.

   Protecting APIs from common vulnerabilities (e.g., CSRF, XSS).

   Data Serialization and Validation

   Using Jackson for JSON serialization and deserialization.

   Validating API requests with @Valid and custom annotations.

4. **Optimizing REST APIs**

   Pagination and filtering for large datasets.

   Caching responses to improve performance.

   Using asynchronous processing for long-running requests.

   Testing and Documentation

   Writing unit tests for REST APIs using JUnit and Mockito.

   Automating API testing with Postman and REST Assured.

   Generating API documentation with Swagger/OpenAPI.

5. **Introduction to Relational Databases:**

   Overview of relational database systems

   Key concepts: tables, rows, columns, and relationships

   Data Integrity Principles:

   Understanding data consistency, accuracy, and reliability

   Primary keys, foreign keys, and their role in maintaining integrity

   Referential integrity and cascading rules

6. **Database Schema Artifacts:**
   Tables and views
   Indexes, constraints, and triggers
   Entity-relationship (ER) diagrams: visualizing schema design

   Schema Design Best Practices:
   Normalization: achieving optimal data structure
   Denormalization: balancing performance and storage needs
   Handling complex relationships with joins and constraints

   Hands-on Activities:
   Designing a database schema for a sample use case
   Implementing primary and foreign key constraints
   Validating referential integrity using triggers and rules

   Common Challenges and Solutions:
   Avoiding redundancy and anomalies
   Strategies for evolving schemas without compromising data integrity

7. **Fundamentals and Strategic Design**
   Introduction to Domain-Driven Design (DDD):
   What is DDD and why it matters?
   Understanding domains, subdomains, and bounded contexts

   Strategic Design Principles:
   Identifying core, supporting, and generic domains
   Designing bounded contexts and context mapping
   Cultivating collaboration between business and technical teams

8. **Tactical Design and Practical Applications**
   Tactical Design Principles:
   Entities, Value Objects, and Aggregates
   Domain Events and Repositories
   Leveraging factories and application services

   Implementing DDD:
   Real-world examples of DDD in action
   Transitioning from a legacy system to a DDD approach
   Common pitfalls and how to avoid them

   Hands-on Exercises and Case Studies:
   Building a domain model for a sample business scenario
   Context mapping workshop to align team understanding

9. **Understanding BPMN 2.0 Fundamentals**
   Overview and importance of BPMN in process modeling.
   Key principles and benefits of using BPMN 2.0.
   Understanding the BPMN 2.0 standard and its elements.

   BPMN Core Components
   Basic shapes: Events, activities, and gateways.
   Types of events: Start, intermediate, and end events.
   Understanding sequence flows and message flows.
   Pools, lanes, and collaboration diagrams for team processes.

   Building Simple Workflows
   Hands-on: Creating your first BPMN diagram.
   Best practices for clear and effective process models.

10. **Advanced BPMN Techniques and Workflow Creation**
    Advanced BPMN Elements
    Sub-processes: Reusable components for complex workflows.
    Conditional flows and looping constructs.
    Timer, message, and error events.

    Workflow Design and Optimization
    Creating end-to-end workflows using BPMN notations.
    Identifying bottlenecks and inefficiencies in processes.
    Designing workflows for automation and scalability.

    Practical Applications of BPMN
    Case studies: Real-world examples of BPMN workflows.
    Hands-on: Building and optimizing a workflow from scratch.

11. **Collabortion and Documentation**
    Using BPMN diagrams for team collaboration.
    Documenting processes for stakeholders.

12. **Containers**

13. **Kubernetes**
    Performance
    Scalability - Auto Scaling

14. **Cloud Computing**
    Infrastructure

    Cloud Security Fundamentals

15. **DevOps**
    Revision