**100 Days Spring Boot – Weekly Roadmap (LinkedIn Updates)**

**Week 1: Simple REST API**

* **Focus:** Environment setup, Spring Boot basics, simple REST API.
* **Hands-on:**
  + Install Java 17+, Maven/Gradle, IDE (IntelliJ/VS Code).
  + Create Spring Boot project via Spring Initializr.
  + Build a simple HelloController endpoint (/hello).
* **LinkedIn Post Idea:**

"Week 1 of my #100DaysOfSpringBoot: Set up my dev environment and built my first Spring Boot REST API endpoint! 🚀 Excited to dive deeper."

**Week 2: CRUD using JPA**

* **Focus:** JPA entities and basic CRUD operations.
* **Hands-on:**
  + Create Book or Task entity.
  + Implement **CRUD endpoints** (GET, POST, PUT, DELETE) using Spring Data JPA.
  + Test in-memory database (H2) for persistence.
* **LinkedIn Post Idea:**

"Week 2: Implemented CRUD operations using Spring Data JPA. My app can now persist data in-memory and I’m exploring entities & repositories!"

**Week 3: Database Integration**

* **Focus:** Connect Spring Boot to a real database.
* **Hands-on:**
  + Switch from H2 to **PostgreSQL/MySQL**.
  + Configure application.properties for DB connection.
  + Test CRUD APIs with real database persistence.
* **LinkedIn Post Idea:**

"Week 3: Integrated PostgreSQL into my Spring Boot app. Now my CRUD APIs persist data in a real database!"

**Week 4: Service Layer & DTOs**

* **Focus:** Service layer, business logic, DTOs, validation.
* **Hands-on:**
  + Separate Controller → Service → Repository layers.
  + Use DTOs for requests/responses.
  + Add validation (@NotNull, @Email).
* **LinkedIn Post Idea:**

"Week 4: Refactored my app with Service layer and DTOs. Validation and clean architecture make my app more robust!"

**Week 5: Spring Security Basics**

* **Focus:** Authentication, role-based authorization.
* **Hands-on:**
  + Configure Spring Security.
  + Implement simple login/logout.
  + Protect APIs based on roles.
* **LinkedIn Post Idea:**

"Week 5: Added authentication and authorization using Spring Security. My API is now secure and role-aware."

**Week 6: JWT Authentication**

* **Focus:** JWT tokens, refresh tokens.
* **Hands-on:**
  + Generate JWT on login.
  + Secure endpoints using JWT.
  + Implement token refresh.
* **LinkedIn Post Idea:**

"Week 6: Implemented JWT authentication in my Spring Boot app. Securing REST APIs has never been this practical!"

**Week 7: Exception Handling & Logging**

* **Focus:** Global exception handling, SLF4J logging.
* **Hands-on:**
  + Use @ControllerAdvice for custom exceptions.
  + Add logging for requests/responses and errors.
* **LinkedIn Post Idea:**

"Week 7: Added centralized exception handling and logging to my app. Debugging and monitoring is much easier now!"

**Week 8: Testing Basics**

* **Focus:** Unit & integration testing.
* **Hands-on:**
  + Write unit tests for services (JUnit + Mockito).
  + Controller tests with MockMvc.
  + Integration tests with H2.
* **LinkedIn Post Idea:**

"Week 8: Started testing my Spring Boot app. Unit and integration tests ensure my code is reliable."

**Week 9: Advanced Testing**

* **Focus:** Testcontainers, contract testing, CI integration.
* **Hands-on:**
  + Use Testcontainers for DB tests.
  + Contract tests for APIs.
  + Add tests in CI/CD pipeline.
* **LinkedIn Post Idea:**

"Week 9: Took testing to the next level with Testcontainers and API contract tests. CI/CD ready!"

**Week 10: Microservices Basics**

* **Focus:** Monolith → Microservices intro, RestTemplate/OpenFeign.
* **Hands-on:**
  + Split app into 2 services (User & Orders).
  + Communicate via RestTemplate/OpenFeign.
* **LinkedIn Post Idea:**

"Week 10: Started exploring microservices with Spring Boot. Services now communicate seamlessly!"

**Week 11: Config, Discovery & Resilience**

* **Focus:** Config Server, Eureka, Resilience4j.
* **Hands-on:**
  + Centralized configuration with Config Server.
  + Service discovery with Eureka.
  + Circuit breaker & retries with Resilience4j.
* **LinkedIn Post Idea:**

"Week 11: Added centralized config, service discovery, and resilience patterns. My microservices are more robust!"

**Week 12: Async & Messaging**

* **Focus:** @Async, @Scheduled, Kafka/RabbitMQ messaging.
* **Hands-on:**
  + Async methods for long tasks.
  + Scheduled tasks (cron jobs).
  + Event-driven messaging (Kafka or RabbitMQ).
* **LinkedIn Post Idea:**

"Week 12: Implemented asynchronous processing and messaging in my Spring Boot app. Excited about event-driven design!"

**Week 13: Deployment & Monitoring**

* **Focus:** Docker, cloud deployment, Actuator, metrics.
* **Hands-on:**
  + Dockerize app and deploy locally or cloud.
  + Monitor app with Spring Boot Actuator.
  + Collect metrics with Micrometer/Prometheus.
* **LinkedIn Post Idea:**

"Week 13: Deployed my Spring Boot app using Docker and added monitoring. Ready for production-like setup!"

**Week 14: Capstone Project & Showcase**

* **Focus:** Build a complete project and document it.
* **Hands-on:**
  + Integrate all learned features into one app.
  + Authentication, CRUD, microservices, messaging, testing, deployment.
  + Write documentation (README, Swagger, Postman collection).
* **LinkedIn Post Idea:**

"Week 14: Completed my Capstone Spring Boot project! From REST APIs to microservices and deployment, it's been an amazing 100-day journey. #100DaysOfSpringBoot"