Lab 4

1. What is Spring?

Spring is a lightweight, modular **Java application framework** that provides infrastructure support for building enterprise applications. It focuses on solving complexities like dependency injection, transaction management, AOP, and integrating with various data sources.

2. What is Spring Boot?

Spring Boot is an opinionated extension of Spring that simplifies application setup by providing defaults, auto-configuration, and an embedded server. It eliminates boilerplate (e.g., web.xml, XML config) and allows you to go from idea to production-ready app faster.

3. Relation: Spring Platform vs. Spring Boot

- The Spring Platform is the broader ecosystem (framework, projects like Spring Data, Spring Security, Spring Cloud, etc.).
- Spring Boot sits on top of the platform, streamlining how you bootstrap and configure those projects.

4. Relation: Spring Platform vs. Spring Framework

- The **Spring Framework** is the **core project** of the spring platform.
- The **platform** includes the framework + additional projects (Spring Data, Security, Cloud, Batch, etc.) as part of the ecosystem

5. Dependency Injection (DI) in spring

- **Concept**: Instead of a class creating its own dependencies, they are **injected externally** (constructor, setter, or field injection).
- **In spring**: The IoC container manages beans and injects dependencies based on configuration (XML, Java config via @Bean, or annotations like @Autowired).

6. Inversion of Control (IoC) and Spring

- **IoC**: The principle that control of object creation and wiring is handed over from the application to the framework/container.
- **In Spring**: The IoC container (ApplicationContext/BeanFactory) is responsible for creating, managing, and wiring beans.
- **Relation**: DI is the *implementation mechanism* of IoC in Spring.

In short: **Spring Framework = core DI + IoC container**,
Spring Platform = full ecosystem of projects,
Spring Boot = fast-track way to build apps using Spring's ecosystem with minimal setup.