Spring = engine (handles **DI, ORM, and beans**)
Spring Boot = engine + auto-assembly line (configures **DB, server, repositories, and runs app** immediately)

Keyword	Spring	Spring Boot	SMS Example
Auto-configuration	Not available; manual setup required	Automatically configures DB, Hibernate, repositories	Spring Boot auto-configured **MySQL**, **Hibernate**, and **repositories** for **Student & Teacher entities**
Boilerplate	Lots of XML / Java configuration needed	Minimal code, convention over configuration;starter dependencies via **pom.xml**	No SQL or repository implementation needed; just added **Spring Boot starters** in **pom.xml**
Component / Dependency Injection	Provides DI; objects created manually	DI handled automatically via Spring Boot beans	**StudentRepository and TeacherRepository** injected into DataLoader automatically
Deployment	Requires external server and web.xml	Stand-alone, embedded server; no deployment descriptor	SMS runs immediately via **SpringApplication.run()** without external Tomcat or web.xml
Ease of Setup	Slow, setup-heavy	Fast, ready-to-run	SMS CRUD ready in ~1 hour instead of days
Framework Type	Enterprise framework for complex apps	Microservice-friendly, production-ready framework	SMS project can easily be **extended into multiple microservices** if needed
Growth-ready DB	Manual configuration, no in-memory DB	Supports MySQL, H2, auto-configured;minimal setup in **application.properties**	Tables auto-created via **@Entity** and DB URL, username, password set in application.properties
Hosting / Server	Must configure external server	Embedded server included	Spring Boot starts **Tomcat** automatically when SMS runs
Integration	Developer handles integration	Auto-integrates JPA, web, security	**Student and Teacher** repositories worked immediately with database
Just Code Focus	Developer writes both logic + infrastructure	Focus only on business logic	We only wrote **entities, repositories, DataLoader**; Spring Boot handled the rest