**Hands-on 4: Difference Between JPA, Hibernate, and Spring Data JPA**

**🔹 Java Persistence API (JPA)**

* **Specification only**, no implementation
* Defines annotations and interfaces (like **@Entity, EntityManager**)
* Needs a provider (like Hibernate)

**🔹 Hibernate**

* **ORM framework** that **implements JPA**
* Can work with both XML and annotation configuration
* Offers extra features (like caching, HQL)

**🔹 Spring Data JPA**

* Abstraction layer over JPA and Hibernate
* **Reduces boilerplate code** (e.g., CRUD operations via interfaces)
* Uses JpaRepository, CrudRepository, etc.
* Automatically handles transactions and session management

**🔹 Code Comparison:**

| **Feature** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- |
| Insert | Manual session/transaction code | repository.save(entity) |
| DAO Layer | Needs to be manually written | Interface-based with Spring magic |
| Boilerplate | High | Low |
| Transaction | Manual or declarative | Automatic with @Transactional |