**Week\_3 spring-data-jpa-handson**

**Que:** Difference between JPA, Hibernate and Spring Data JPA

**Ans:**

In enterprise Java development, working with databases is a crucial part. To simplify this, developers use Object Relational Mapping (ORM) tools such as **JPA**, **Hibernate**, and **Spring Data JPA**. Although these technologies are closely related, each plays a different role in building robust Java applications. This assignment explains the differences between them with examples and comparisons.

**JPA (Java Persistence API):**

* JPA is a **Java specification** that defines how to map Java objects to relational databases.
* It is **only an interface**; it does not provide any implementation.
* JPA is part of **Jakarta EE** (formerly Java EE).
* It includes annotations like @Entity, @Id, @OneToMany, @Table, etc.
* To actually persist data, JPA requires a provider (like Hibernate).

**Hibernate:**

Hibernate is an **open-source ORM framework** that implements the JPA specification. It is the most widely used JPA provider in the Java ecosystem.

**Key Characteristics:**

* Implements all features of JPA
* Offers **additional non-JPA features**, such as:
  + HQL (Hibernate Query Language)
  + Caching (1st and 2nd level)
  + Lazy Loading
  + Interceptors
  + Batch processing
* Uses XML or annotation-based configuration

**Spring Data JPA:**

Spring Data JPA is a **Spring Framework project** that simplifies the use of JPA in Spring applications by **reducing boilerplate code**.

It sits on top of JPA and Hibernate and provides:

* Pre-built CRUD methods
* Query method generation from method names
* Support for JPQL, native queries, and dynamic queries

**Key Characteristics:**

* Part of the larger **Spring Data** family
* Uses repositories like JpaRepository, CrudRepository
* Reduces the need to write DAO classes