**Cognizant Digital Nurture 4.0-Week(3)**

**Name: Gandu Lasya Sri**

**Email: gandulasyasri@gmail.com**

**Superset ID:6428164**

**Mandatory Hands-On Exercises**

**Spring Data JPA with Hibernate**

**Spring-data-jpa-handson**

**1. Project Name:** LibraryJPA

**2. Tools I used:**

* VS Code
* Maven
* Java 17 (but it worked with Java 21+ too)
* Spring Boot (used dependencies from Spring Initializr)
* H2 Database (in-memory)

**How I created this project:**

**Step 1: Created a Maven project using Spring Initializr**

* Added dependencies:
  + Spring Web
  + Spring Data JPA
  + H2 Database

**Folder Structure:**

libraryjpa

├── src

│ └── main

│ └── java

│ └── com.library

│ ├── App.java

│ ├── model

│ │ └── Book.java

│ └── repository

│ └── BookRepository.java

└── pom.xml

**Book.java (Model Class)**

package com.library.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

public Long getId() {

return id;

}

public String getTitle() {

return title;

}

public String getAuthor() {

return author;

}

@Override

public String toString() {

return "Book{id=" + id + ", title='" + title + "', author='" + author + "'}";

}

}

**BookRepository.java**

package com.library.repository;

import com.library.model.Book;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {

}

**App.java**

package com.library;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import com.library.model.Book;

import com.library.repository.BookRepository;

@SpringBootApplication

public class App {

public static void main(String[] args) {

SpringApplication.run(App.class, args);

}

@Bean

CommandLineRunner run(BookRepository repo) {

return args -> {

repo.save(new Book("Effective Java", "Joshua Bloch"));

repo.save(new Book("Clean Code", "Robert C. Martin"));

System.out.println("Books in the database:");

repo.findAll().forEach(System.out::println);

};

}

}

**application.properties**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

spring.jpa.hibernate.ddl-auto=update

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.library</groupId>

<artifactId>LibraryJPA</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>jakarta.persistence</groupId>

<artifactId>jakarta.persistence-api</artifactId>

<version>3.1.0</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**How I ran the project**

mvn clean install

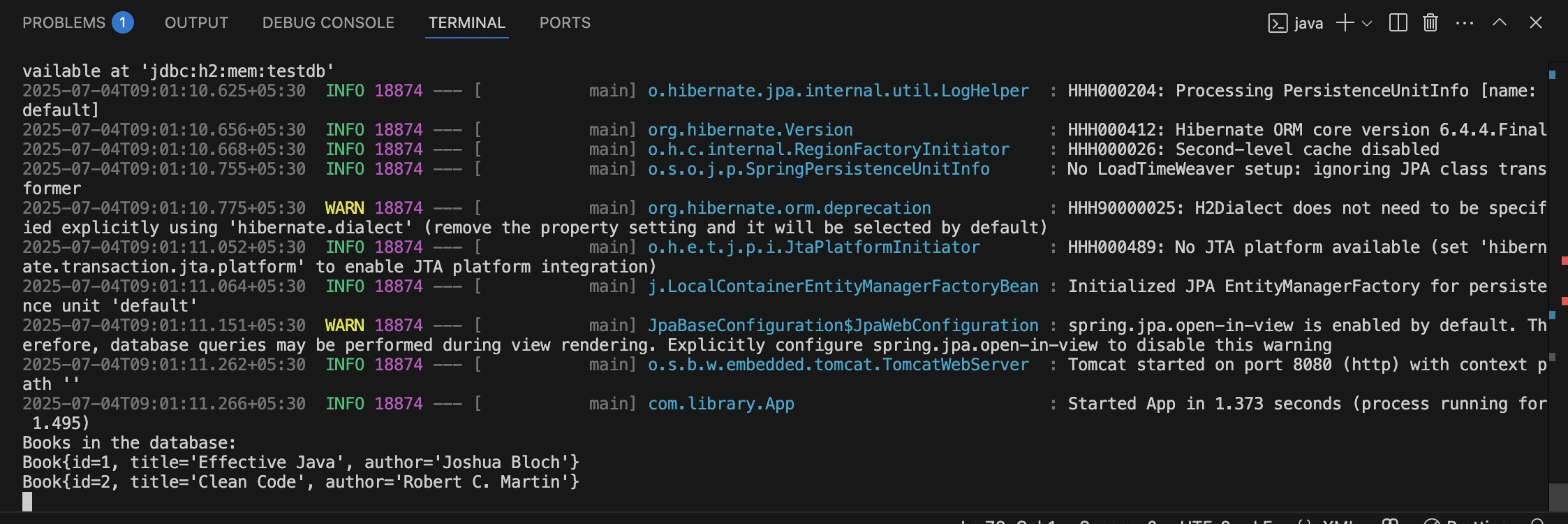
mvn spring-boot:run

**Output in Terminal:**

Books in the database:

Book{id=1, title='Effective Java', author='Joshua Bloch'}

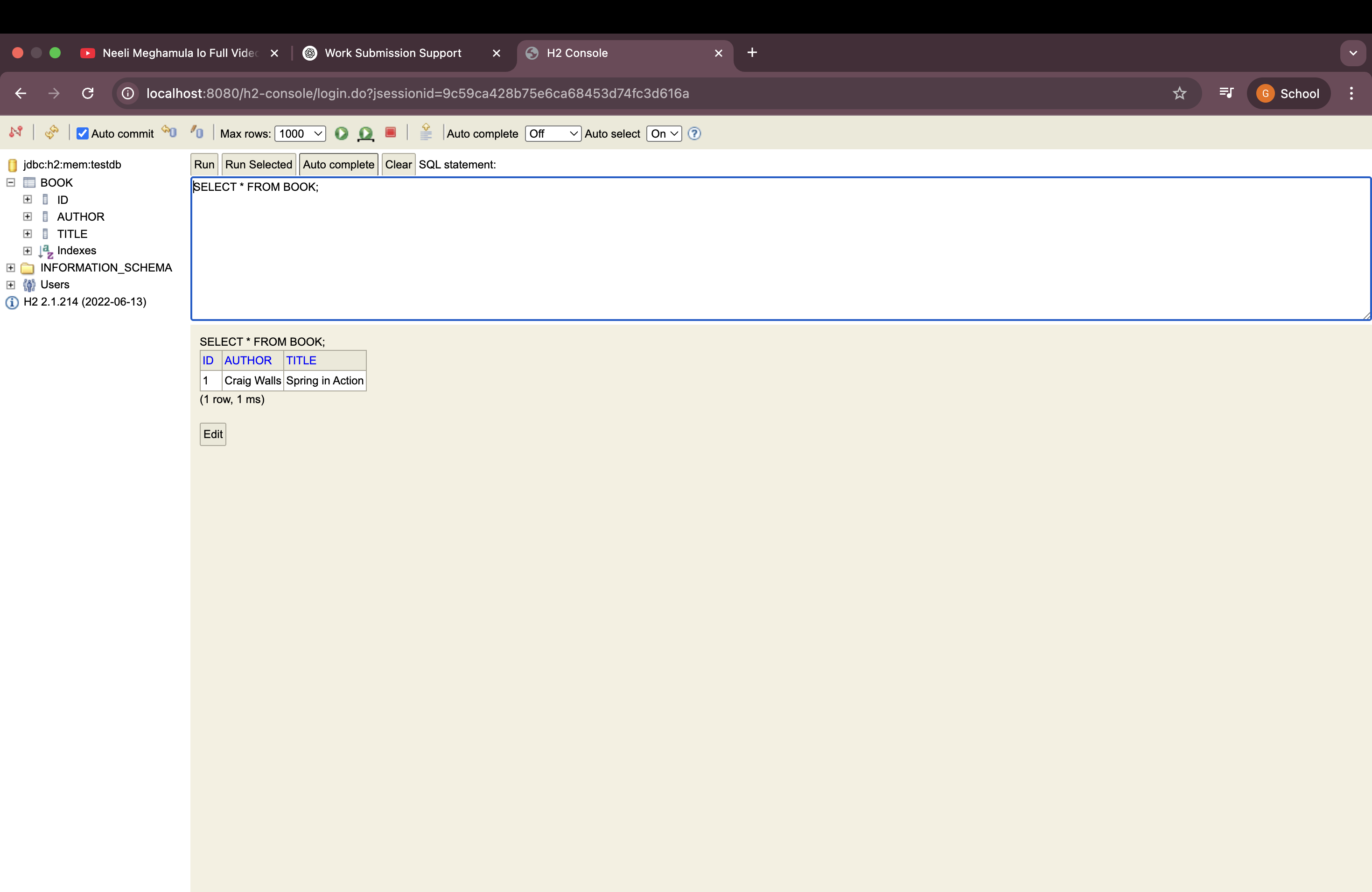
Book{id=2, title='Clean Code', author='Robert C. Martin’}



**Output in H2 Console**

* I opened browser: <http://localhost:8080/h2-console>
* JDBC URL: jdbc:h2:mem:testdb
* Username: sa
* Left password empty
* Clicked Connect → Ran SQL:

SELECT \* FROM BOOK;



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

|  |  |  |
| --- | --- | --- |
| **JPA (Java Persistence API)** | **Hibernate** | **Spring Data JPA** |
| It's just a **specification** (interface), not a tool or library | It's a **framework** that implements JPA | It's a **Spring module** that works on top of JPA & Hibernate |
| Provided by **Java (javax/jakarta.persistence)** | Developed by **Hibernate.org** | Developed by **Spring team** |
| Defines standard methods to interact with databases using ORM | Provides actual working code for ORM operations | Reduces boilerplate code using **Repository interfaces** |
| No, it needs an implementation like Hibernate | Yes, Hibernate can work standalone or with JPA | No, it depends on both JPA and an implementation (like Hibernate) |
| @Entity, @Table, EntityManager | Session, Query, Hibernate-specific config | JpaRepository, CrudRepository, @Repository |