**Week 3**

***Spring Core Maven***

**Configuring a Basic Spring Application**

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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>librarymanagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<mainClass>com.library.LibraryManagementApp</mainClass>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.5</version>

</dependency>

</dependencies>

</project>

**applicationConntext.xml**

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="bookRepository" class="com.library.repository.BookRepository"/>

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void save(String title) {

System.out.println("Saving book: " + title);

}

}

**BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository repo) {

this.bookRepository = repo;

}

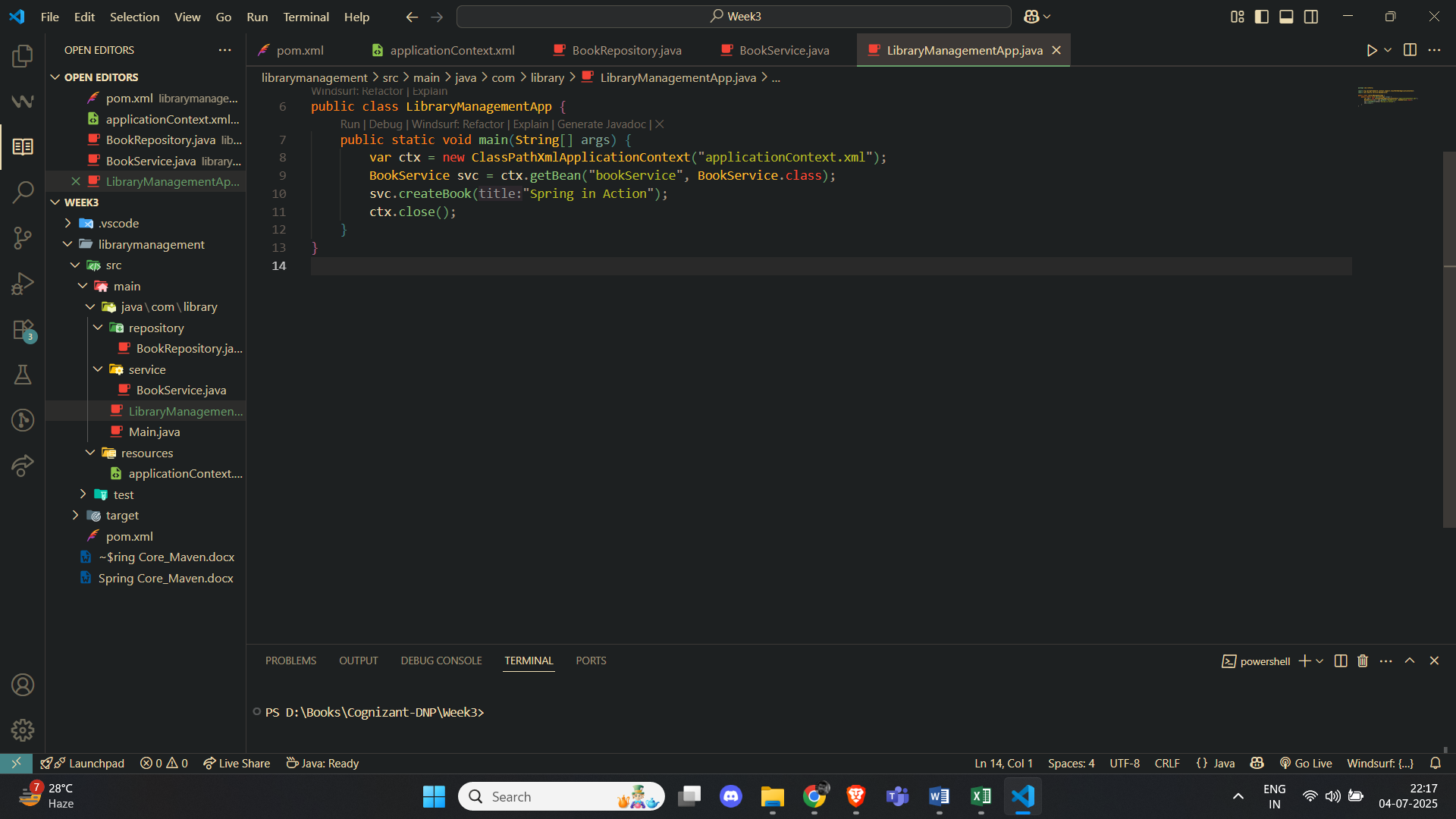
public void createBook(String title) {

bookRepository.save(title);

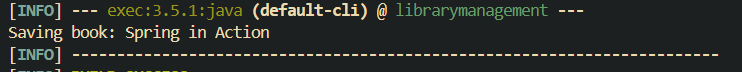
}

}

**LibraryManagementApp.java**



**Output**

****

**Implementing Dependency Injection**

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Repository bean -->

<bean id="bookRepository" class="com.example.library.repository.BookRepository"/>

<!-- Service bean with setter injection -->

<bean id="bookService" class="com.example.library.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**BookRepository.java**

package com.example.library.repository;

public class BookRepository {

public void save(String book) {

System.out.println("Book saved: " + book);

}

}

**BookService.java**

package com.example.library.service;

import com.example.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void save(String book) {

bookRepository.save(book);

}

public BookRepository getBookRepository() {

return bookRepository;

}

}

**LibraryApplication.java**

package com.example.library;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.**SpringBootApplication**;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.**ImportResource**;

@**SpringBootApplication**

@**ImportResource**("classpath:applicationContext.xml")

public class LibraryApplication {

    public static void main(String[] args) {

        try {

            ConfigurableApplicationContext ctx = SpringApplication.run(LibraryApplication.class, args);

            System.out.println("Application started successfully!");

        } catch (Throwable t) {

            t.printStackTrace();        }

    }

}

**Creating and Configuring a Maven Project**

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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.example</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>jar</packaging>

<name>LibraryManagement</name>

<description>Library Management Application using Spring</description>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<!-- Spring Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.36</version>

</dependency>

<!-- Spring WebMVC -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.36</version>

</dependency>

<!-- Servlet API (required for WebMVC) -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Maven Compiler Plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

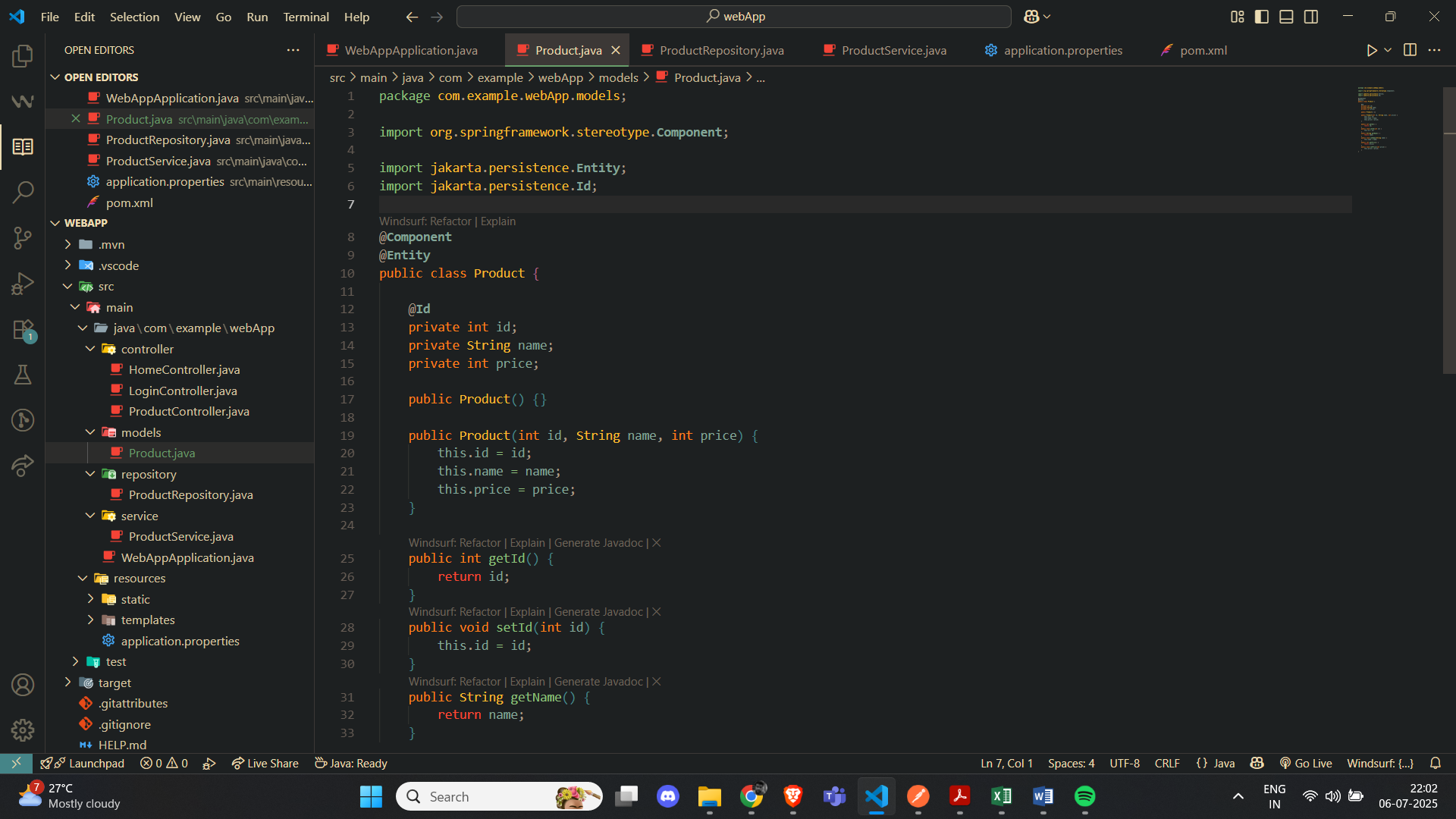
</build>

</project>

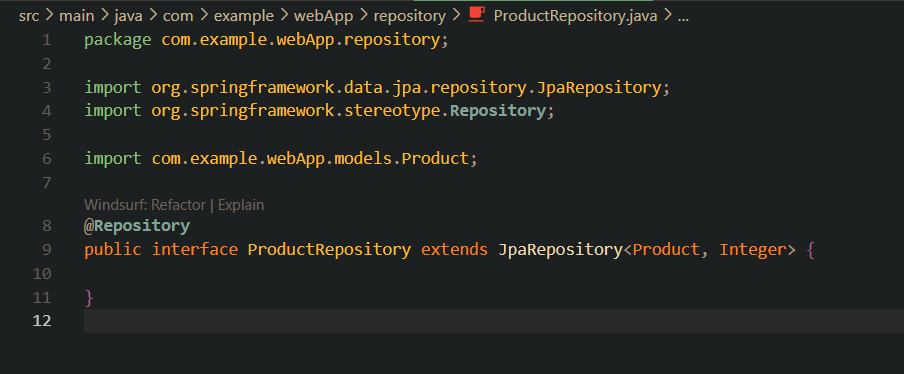
***Spring Data JPA with Spring Boot, Hibernate***

**Spring Data JPA**

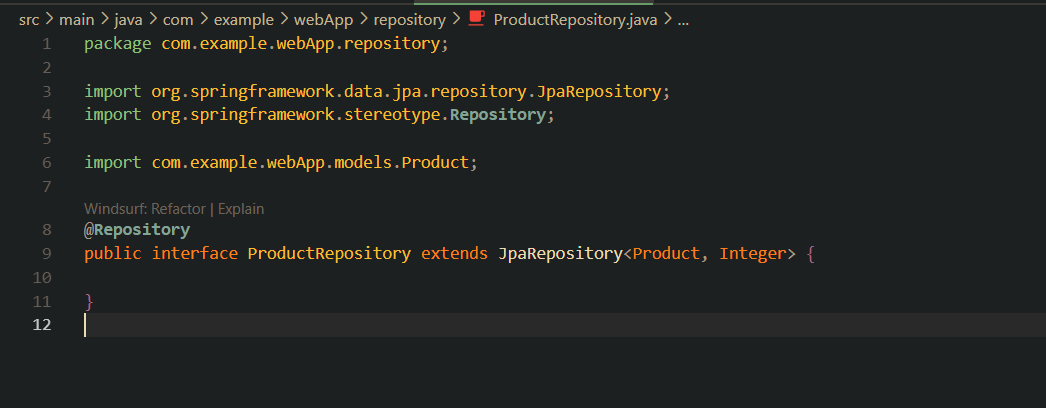
**Product.java**



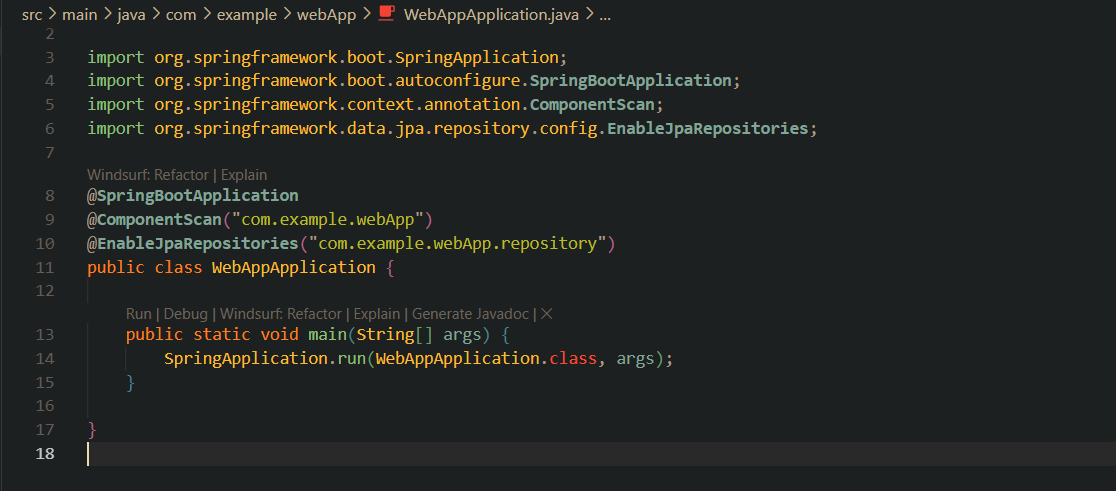
**ProductRepository.java**

****

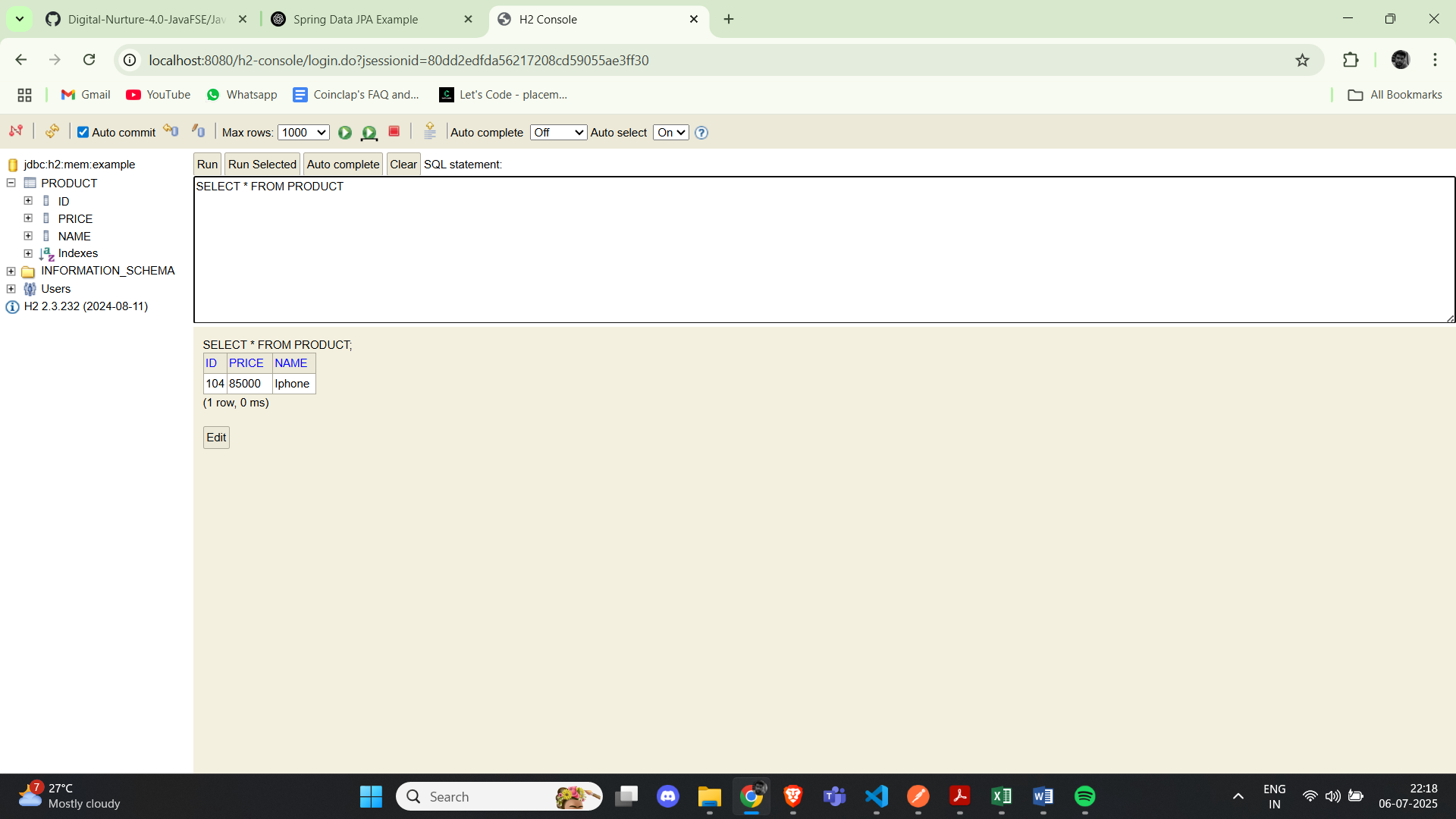
**ProductRepository.java**

****

**WebAppApplication.java**

****

**Output:**



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

**1. JPA (Java Persistence API / Jakarta Persistence)**

* **What it is:** A *specification*, not an implementation. It defines standard interfaces and annotations—like EntityManager, @Entity, @Table, @Id, JPQL, and so on—for mapping and managing relational data.
* **Key Points:**
  + Acts as a set of rules for ORM frameworks.
  + Requires an implementation (like Hibernate or EclipseLink) to function.

## 2. Hibernate

* **What it is:** An ORM framework that implements the JPA spec (along with its own extensions)
* **Capabilities:**
  + Maps Java classes to DB tables.
  + Offers HQL (Hibernate Query Language) and Criteria API.
  + Features like caching, lazy loading, and more.
* **Summary:** It’s the engine that executes JPA standards—widely used as the default provider in Spring Boot

## 3. Spring Data JPA

* **What it is:** A Spring project that builds on top of JPA, providing an abstraction layer to reduce boilerplate in data access
* **What it provides:**
  + CrudRepository, JpaRepository, etc.—interfaces with built-in CRUD, paging, sorting.
  + Derives queries from method names (findByNameOrderByAge(...)).
  + Auto-implements repositories at runtime, backed by whichever JPA provider you configure (typically Hibernate).
* **Note:** It is not a JPA provider — it requires one under the hood