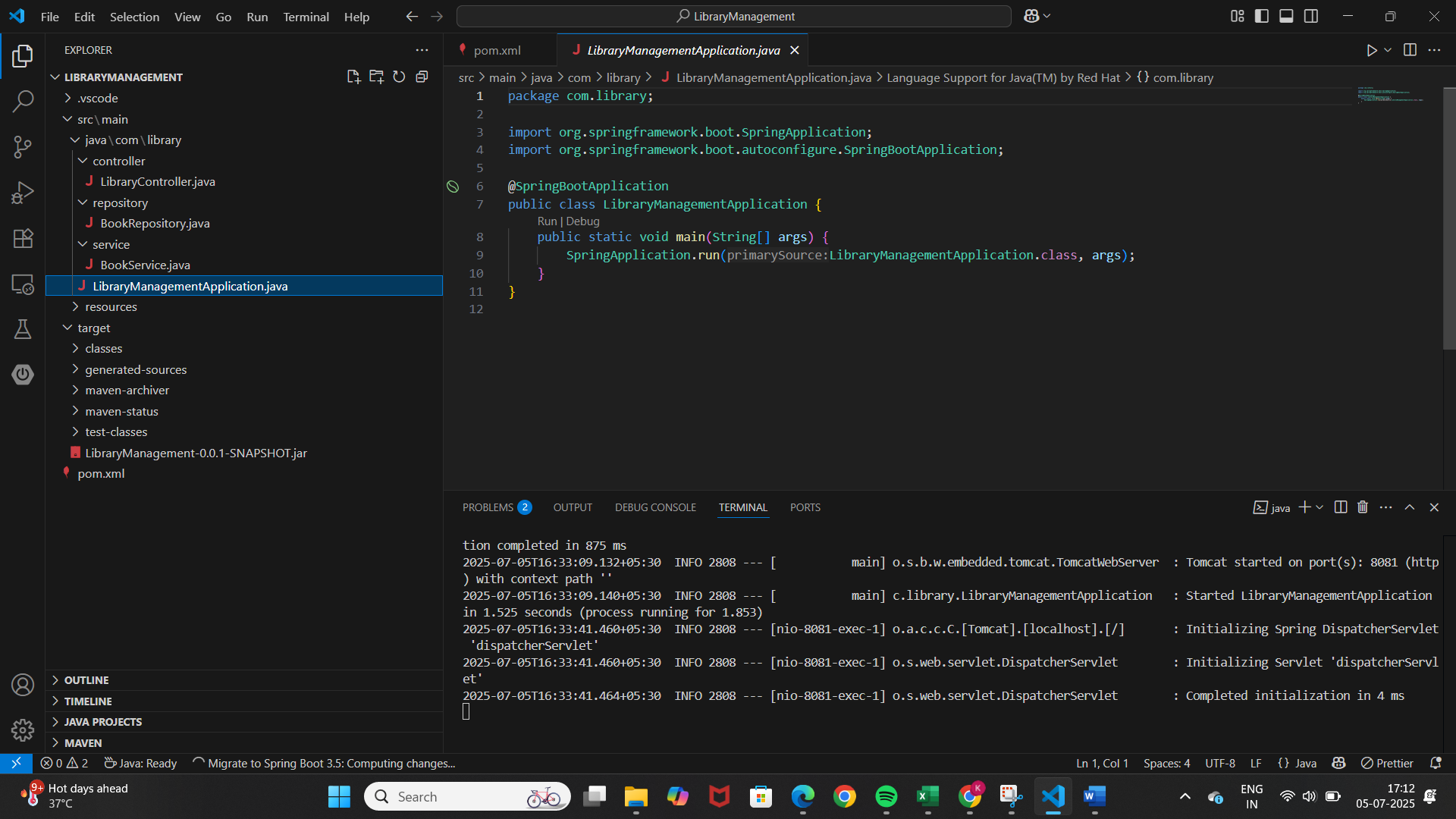
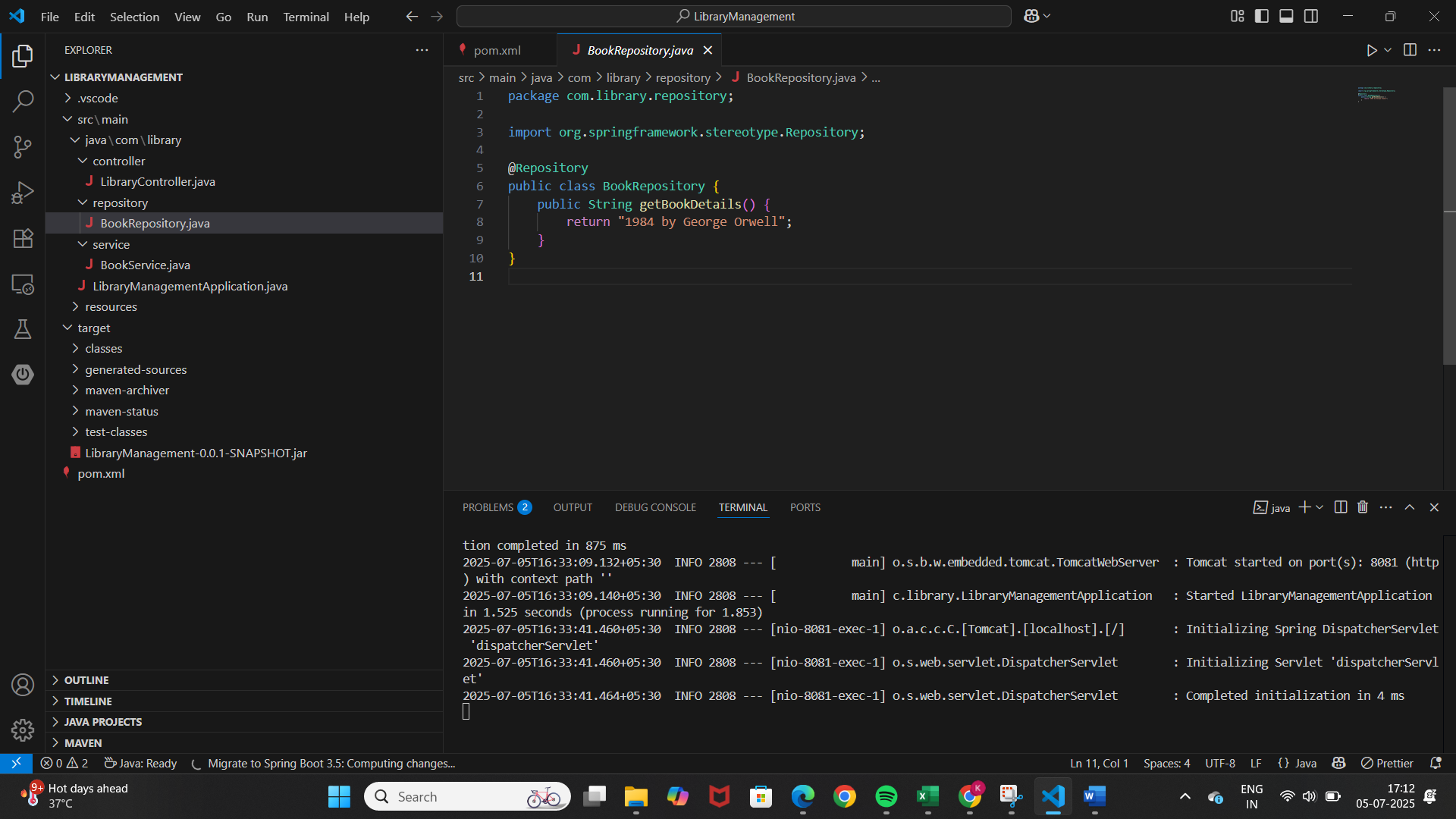
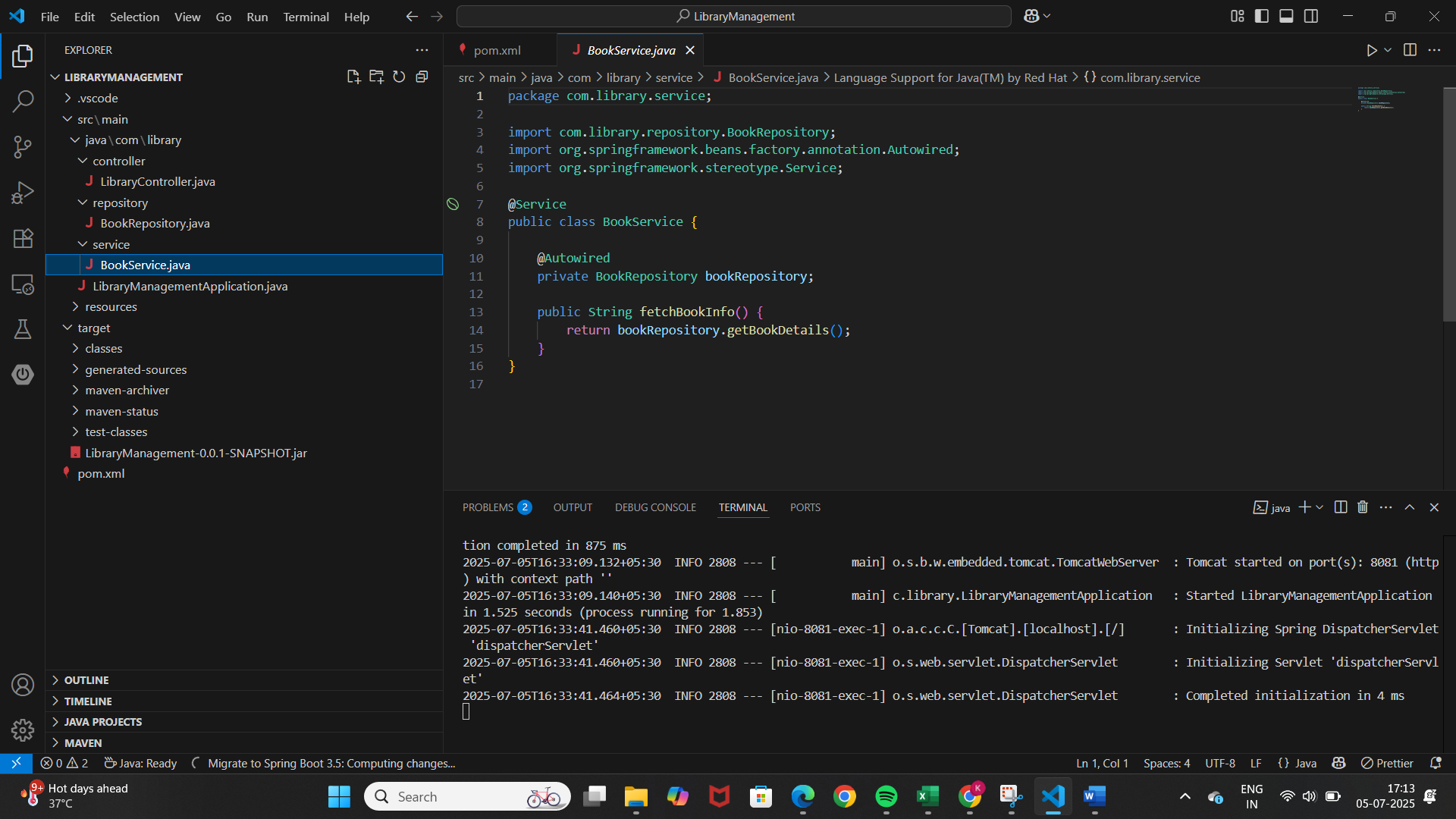
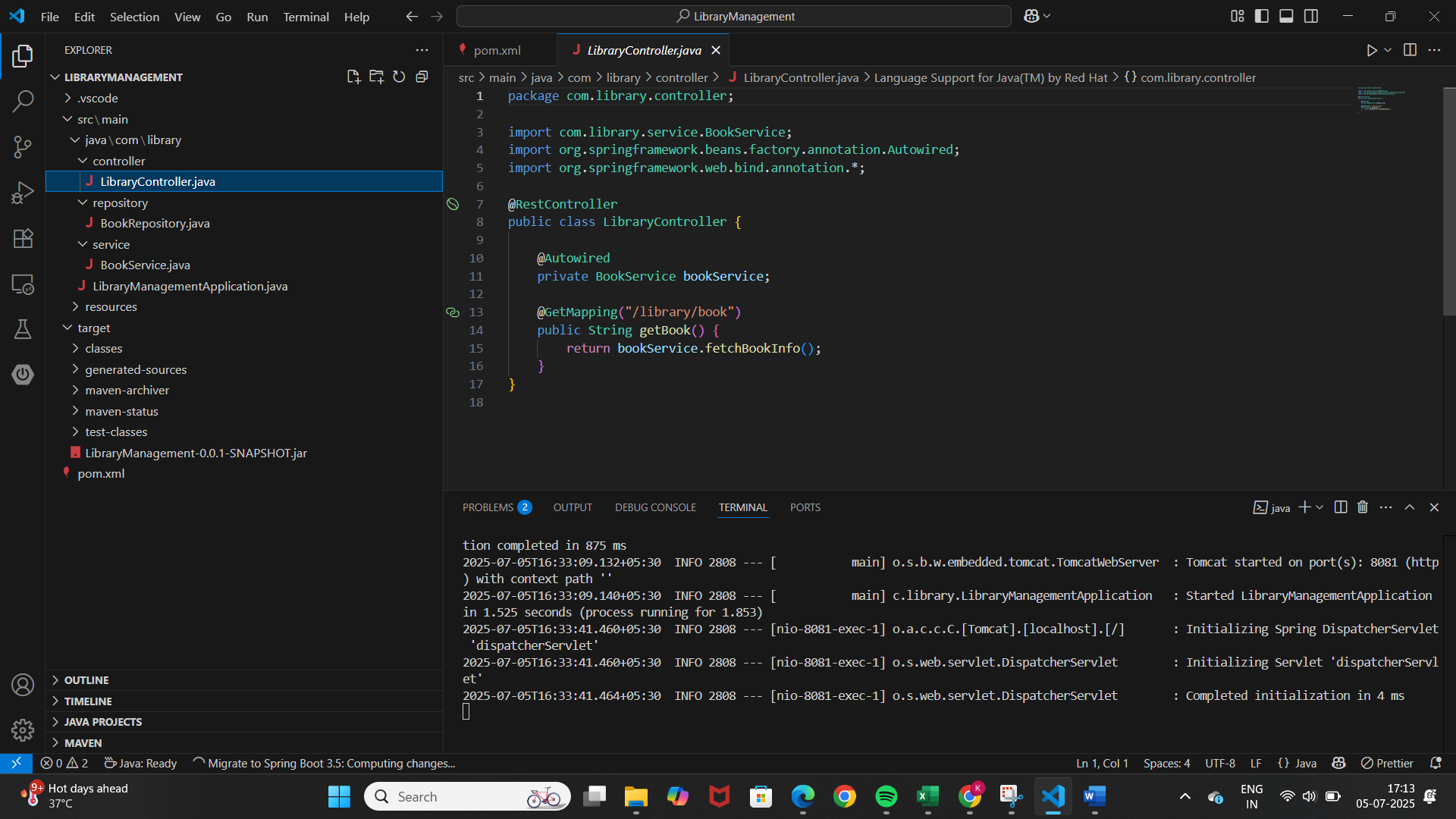
**Week-3: Spring core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

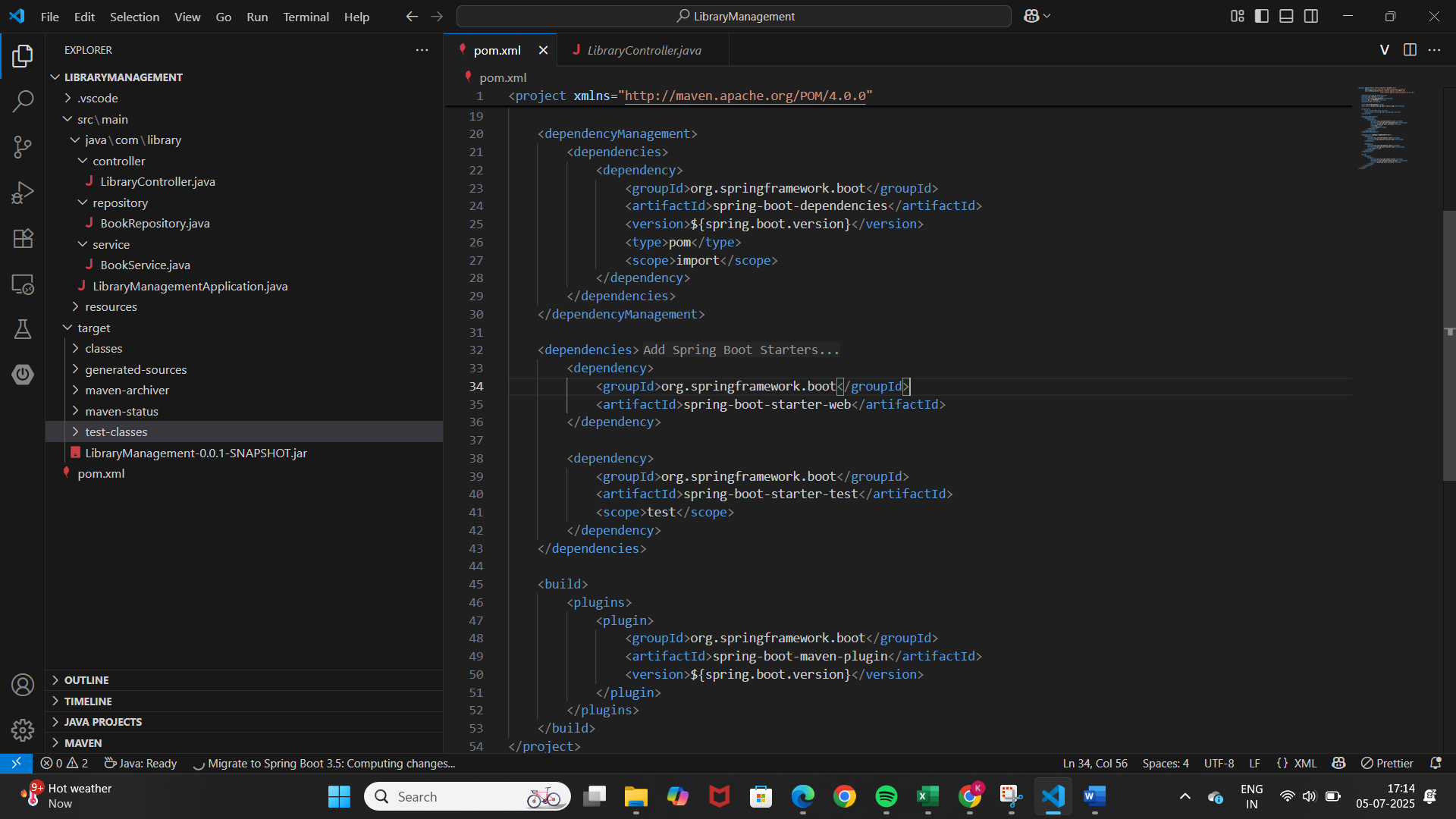
LibraryManagementApplication.java

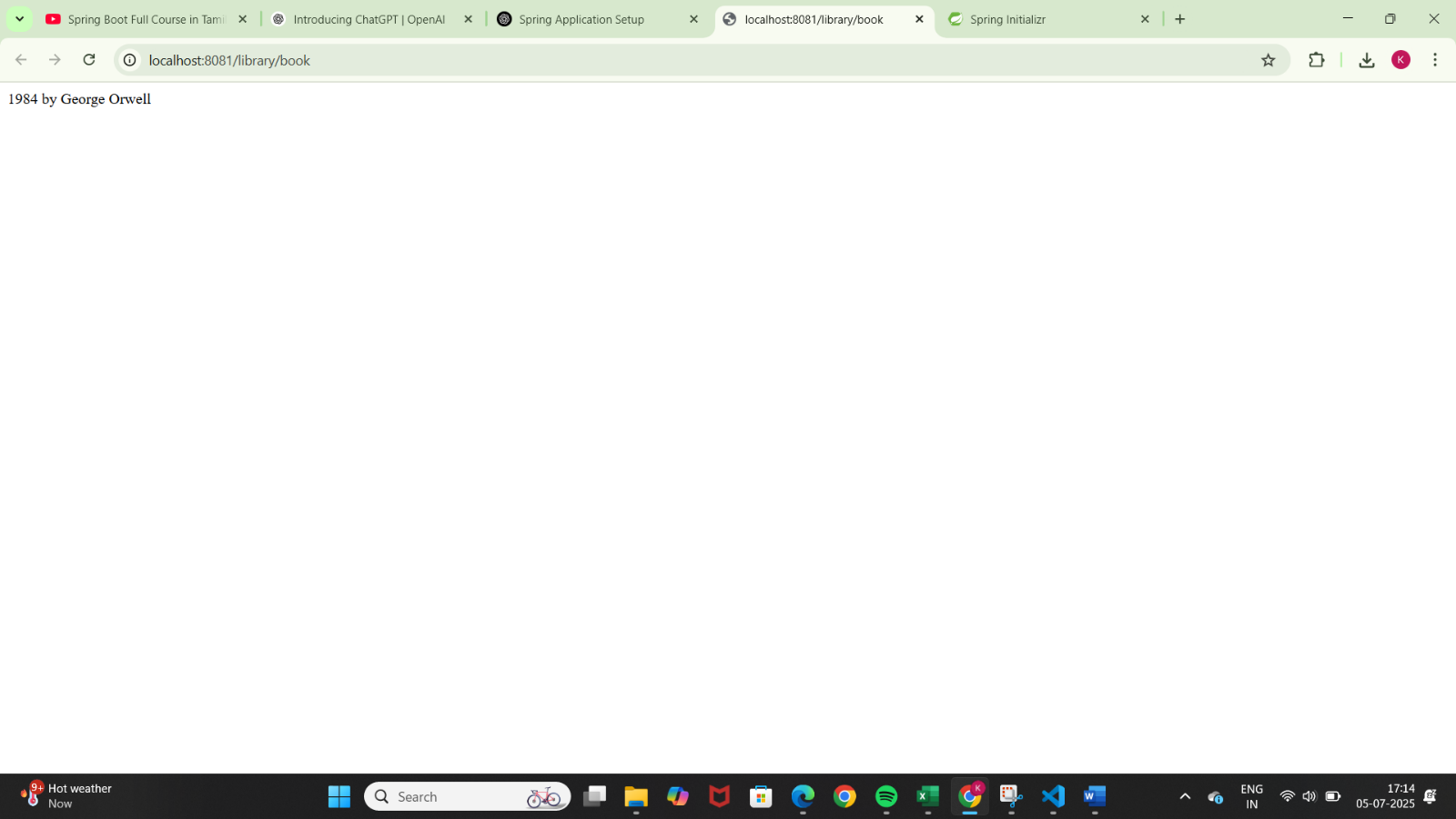
BookRepository.java

BookService.java

LibraryController.java

Pom.xml

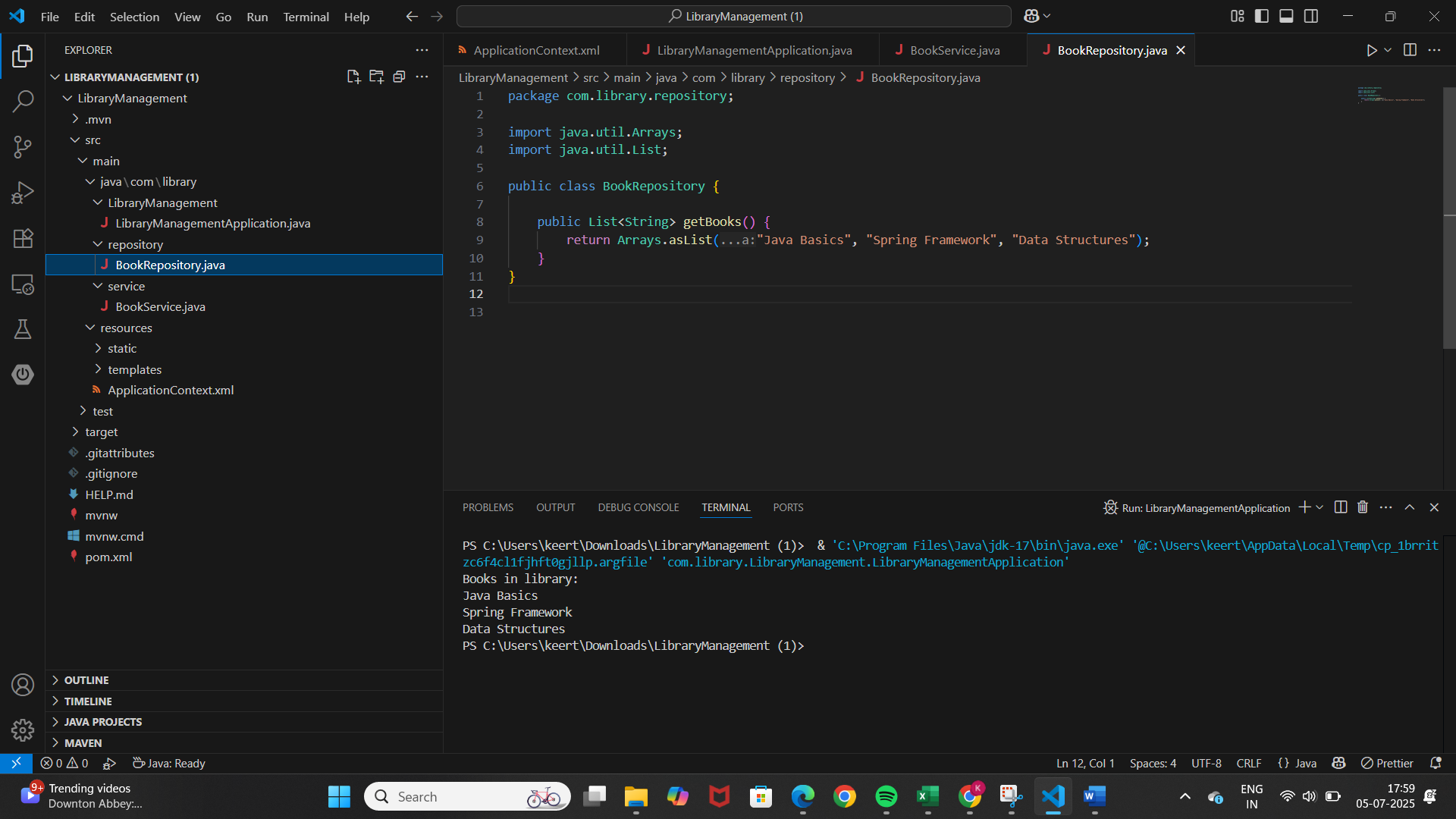


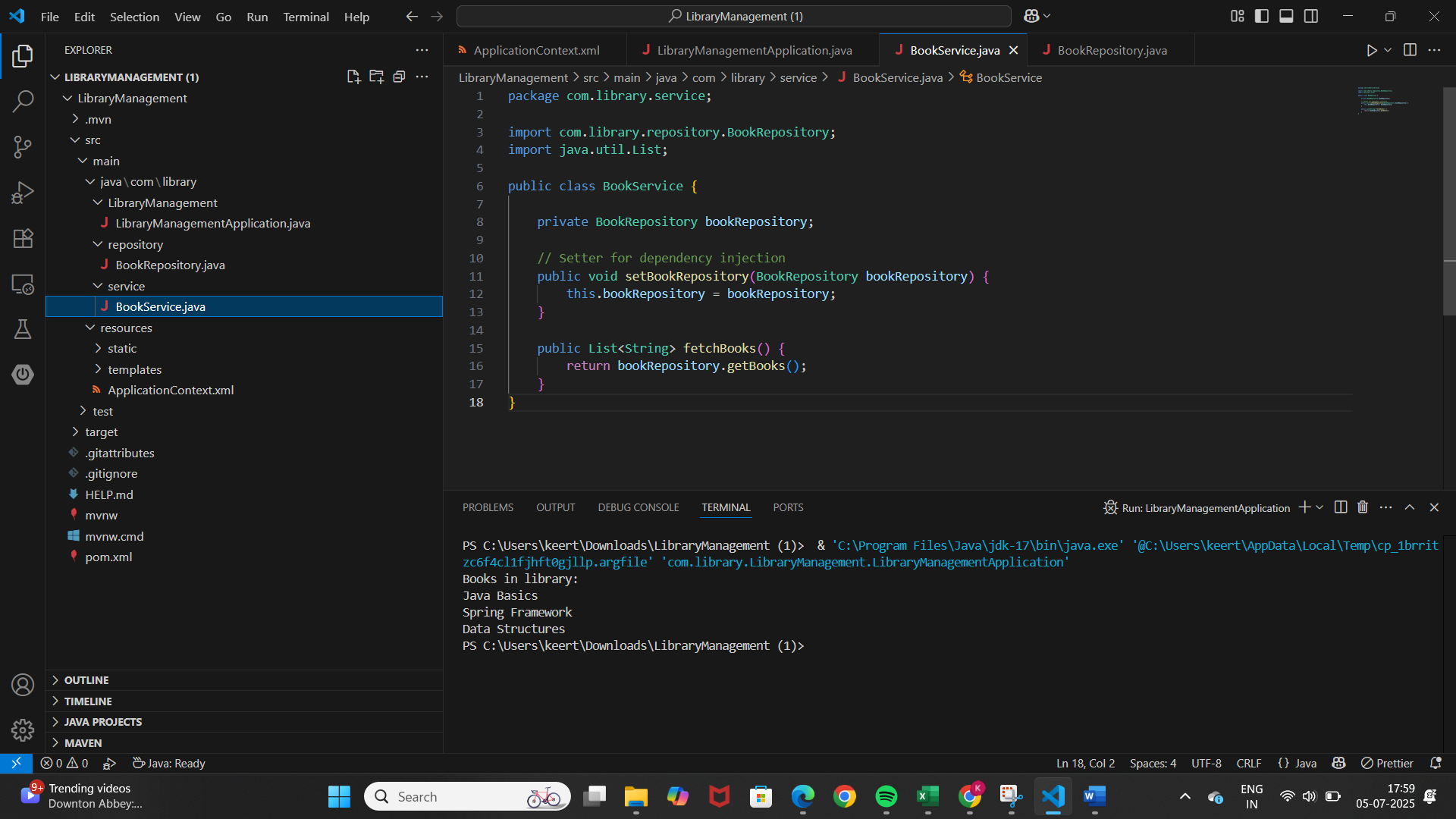
Output:

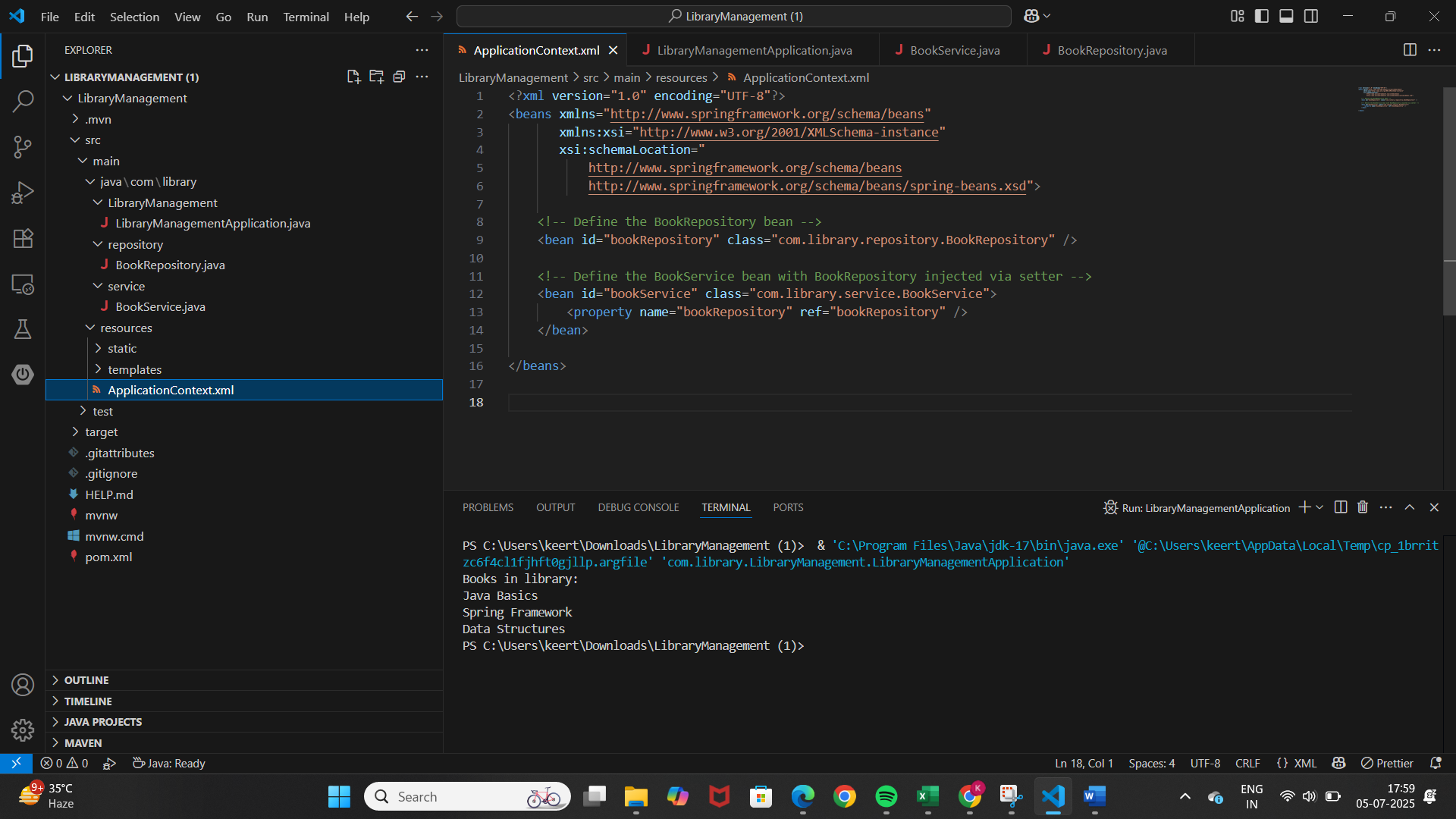
**Exercise 2: Implementing Dependency Injection**

LibraryManagementApplication.java

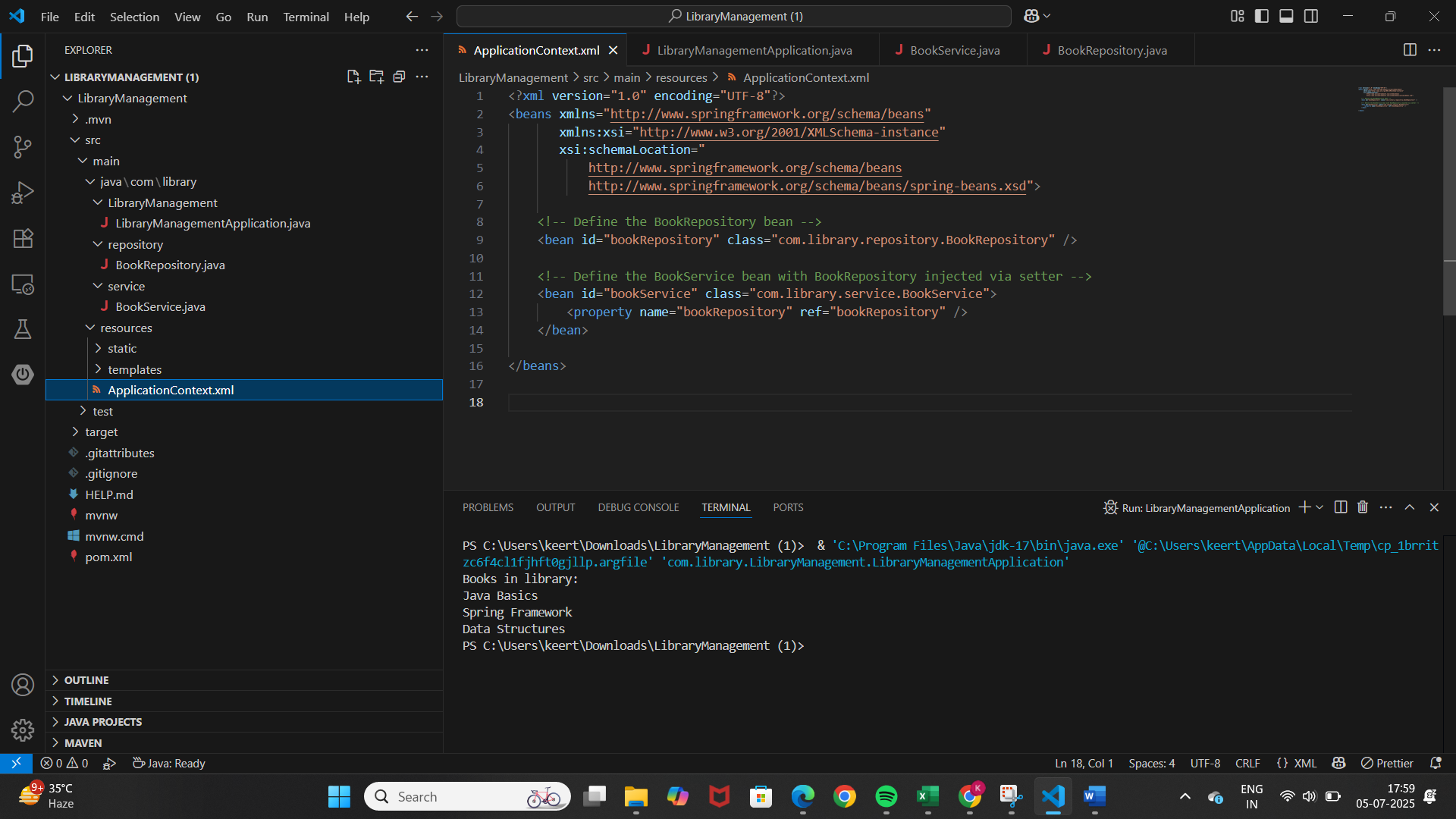
BookRepository.java

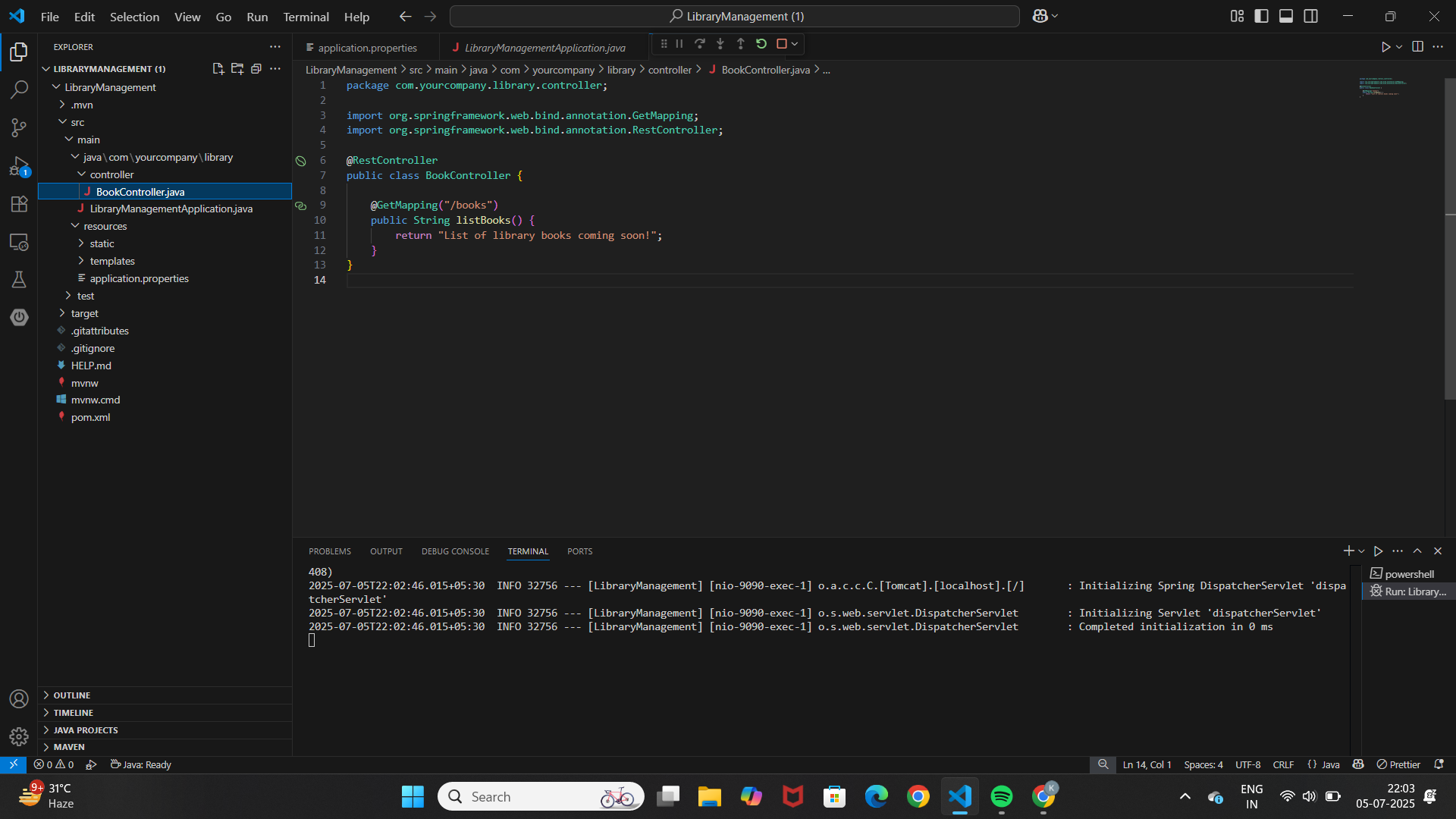


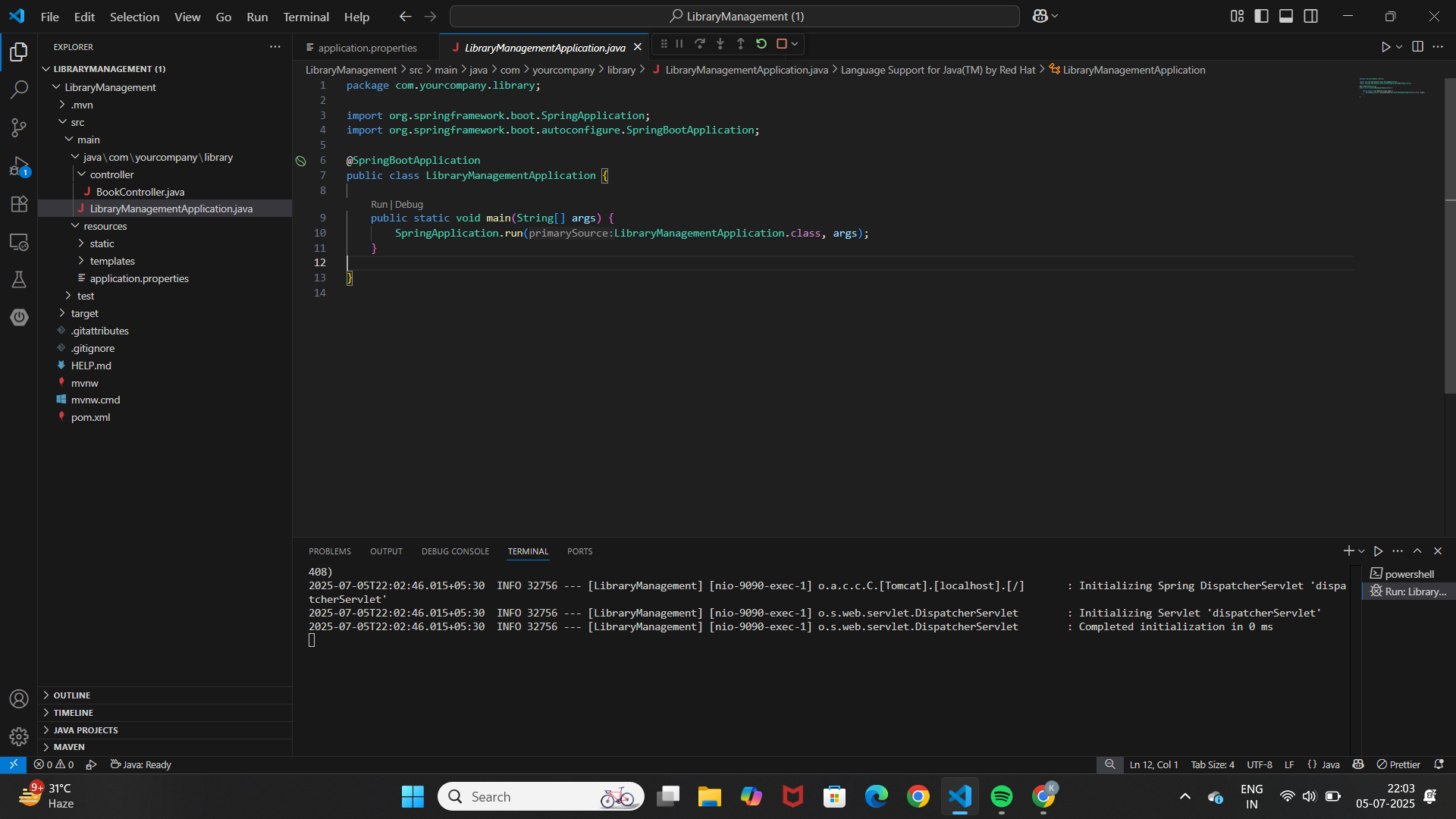
BookService.java

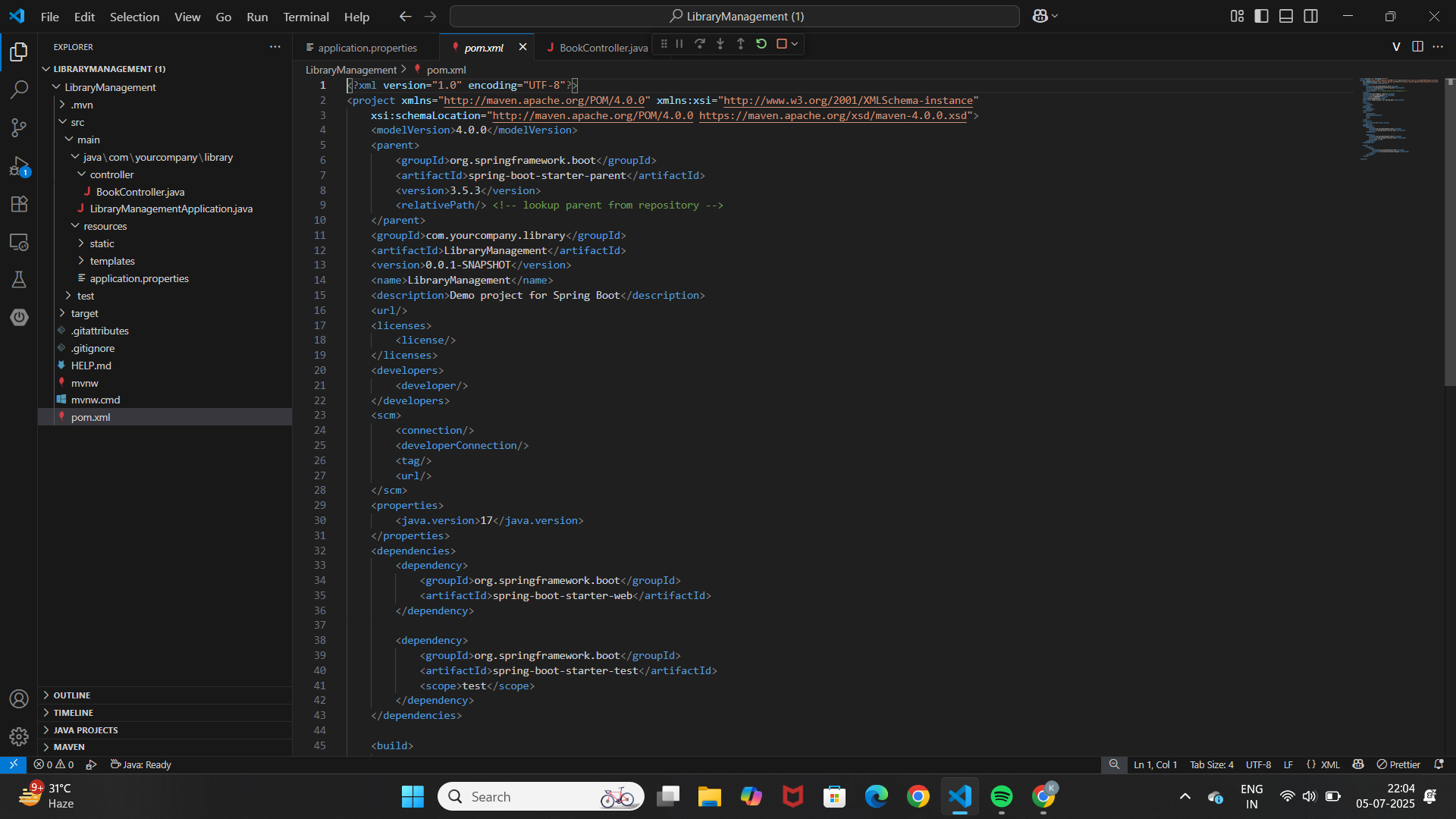
ApplicationContext.xml

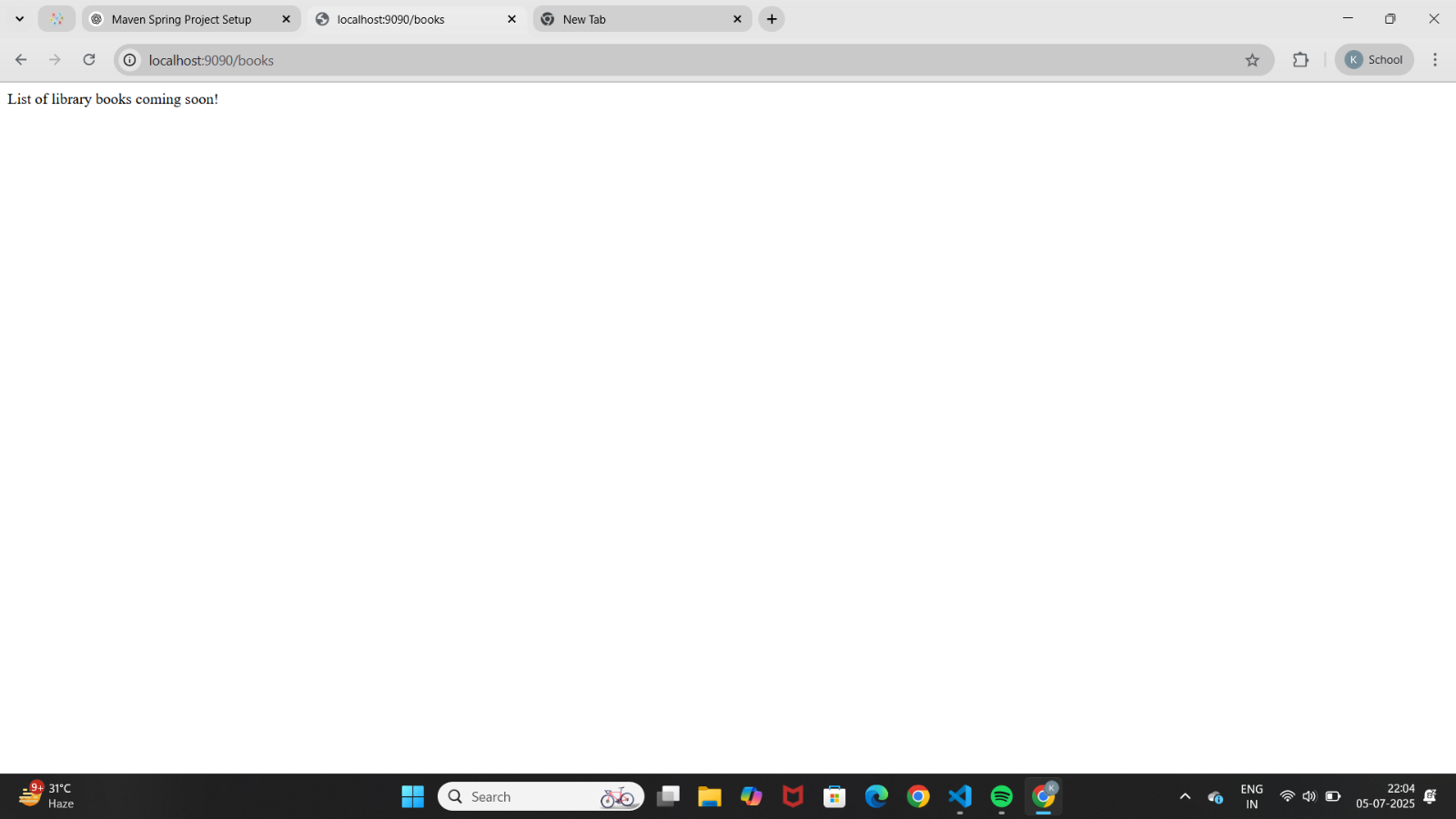
Output:



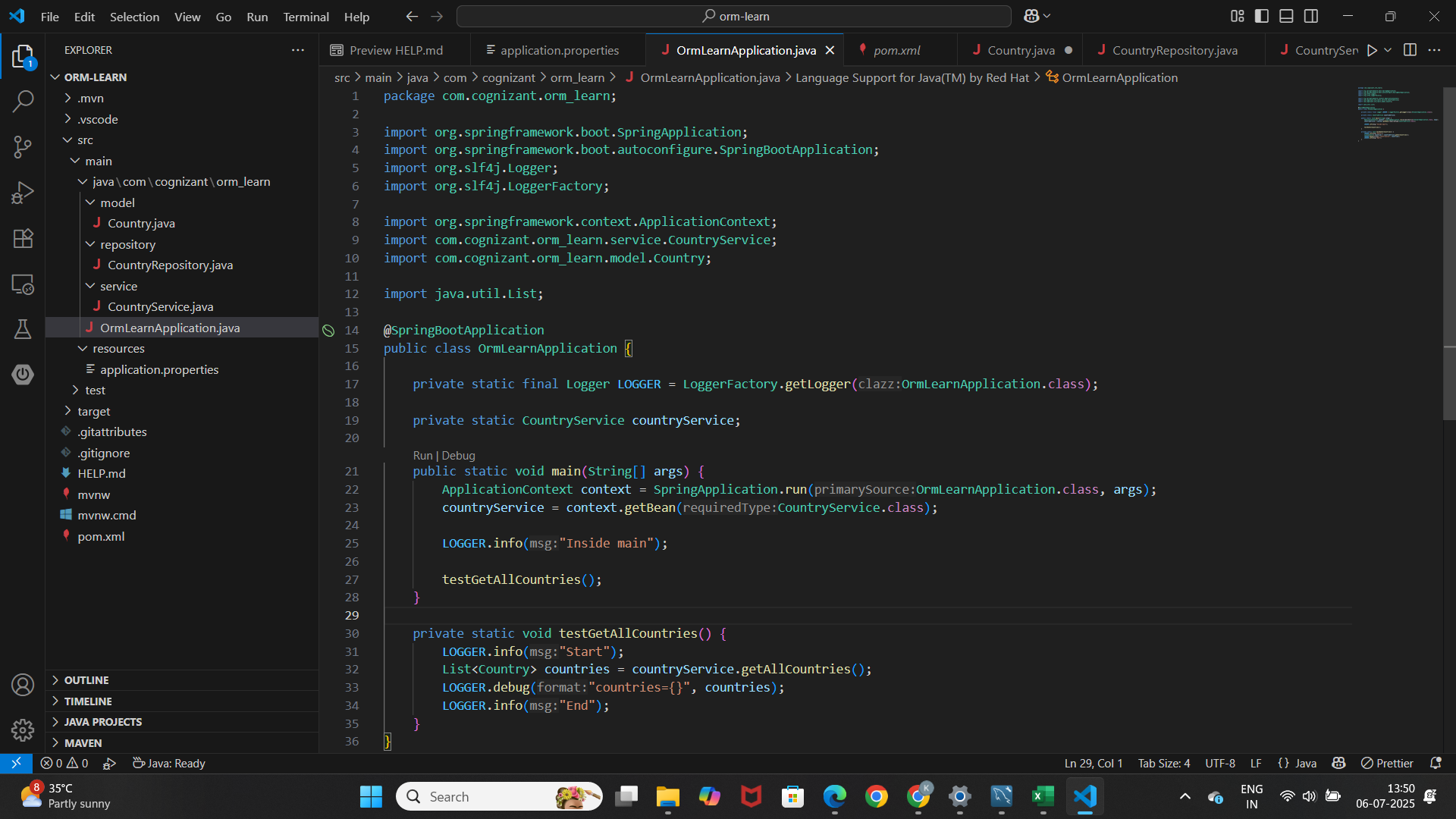
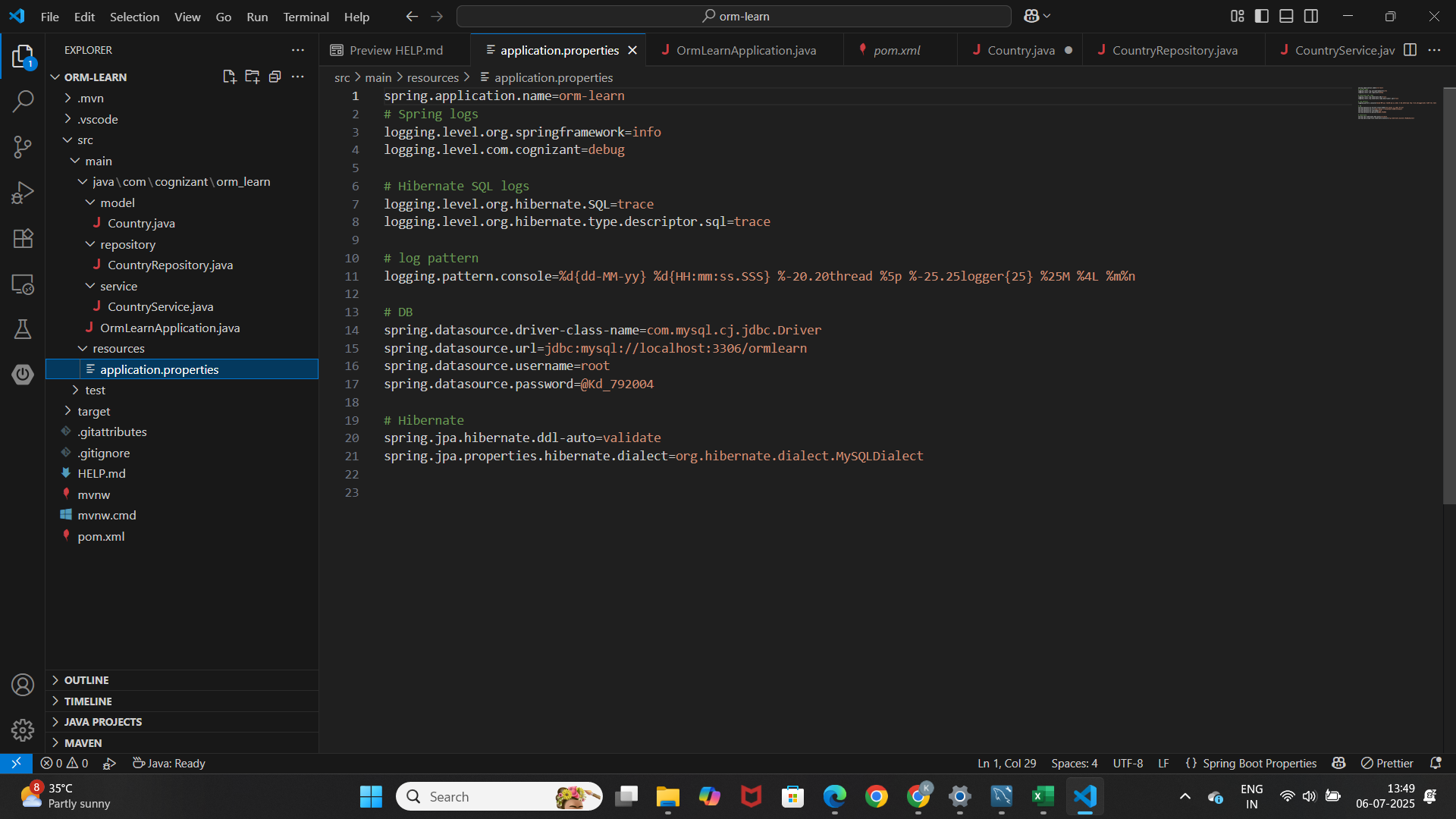
**Exercise 4: Creating and Configuring a Maven Project**

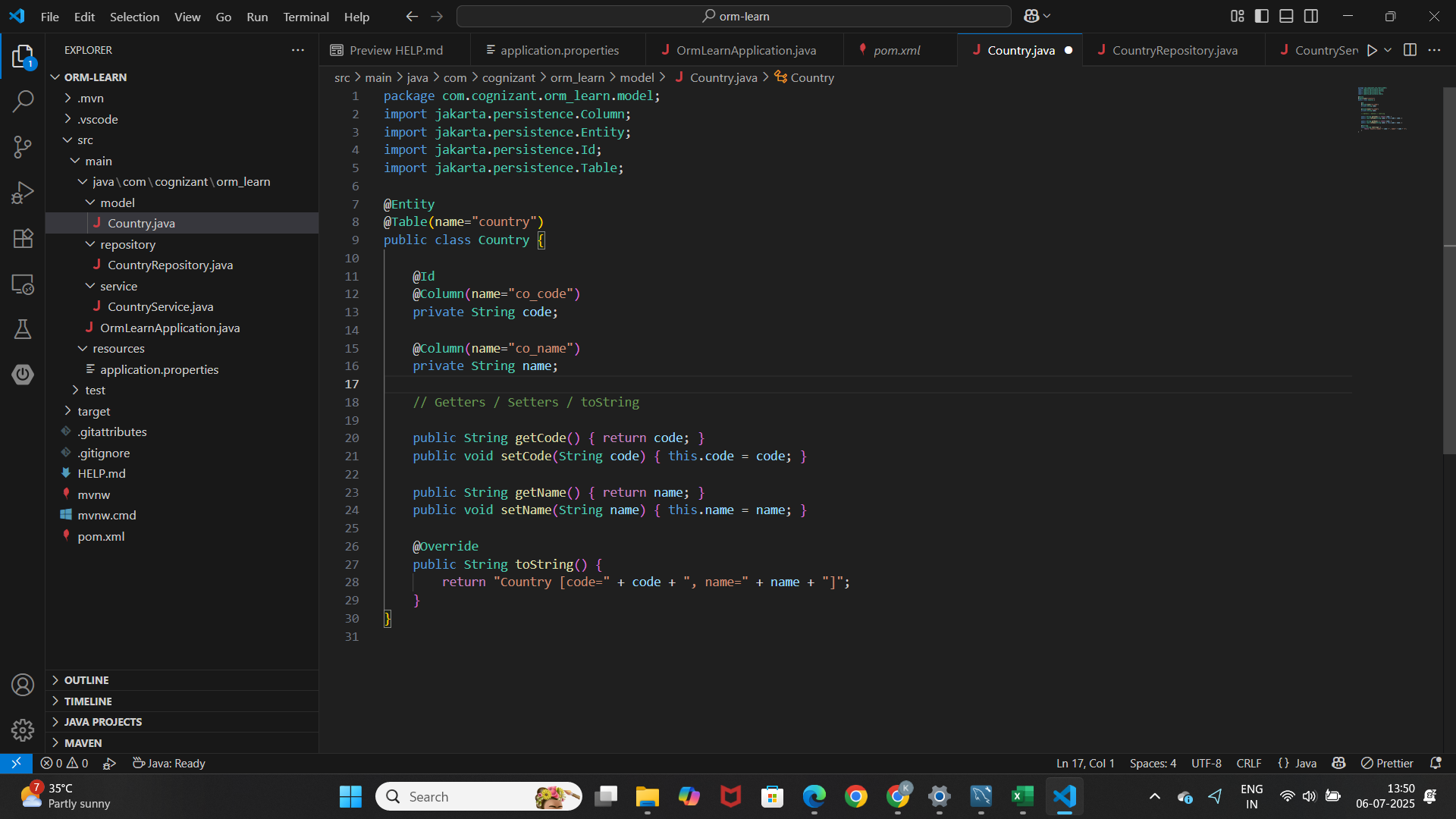
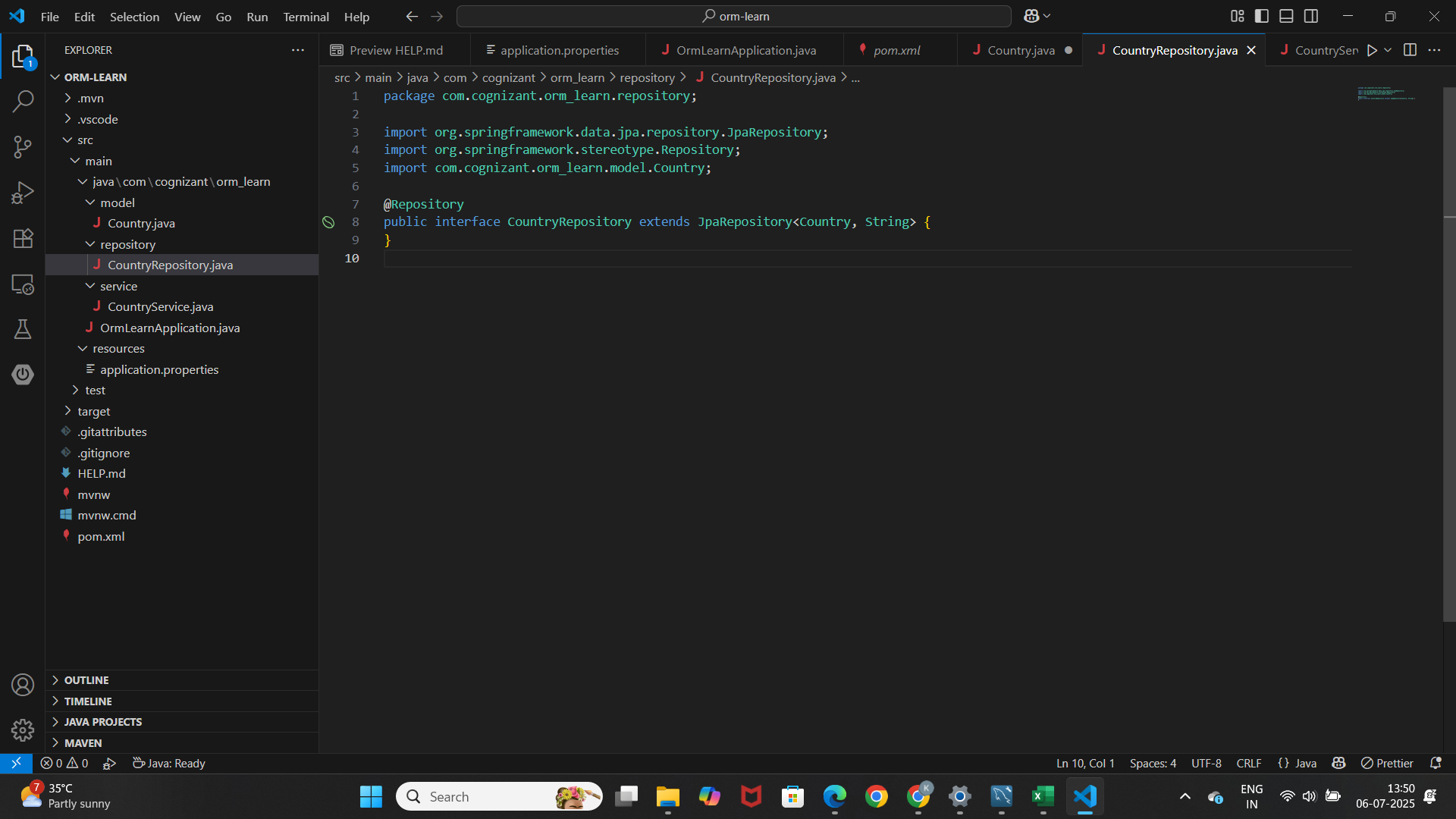


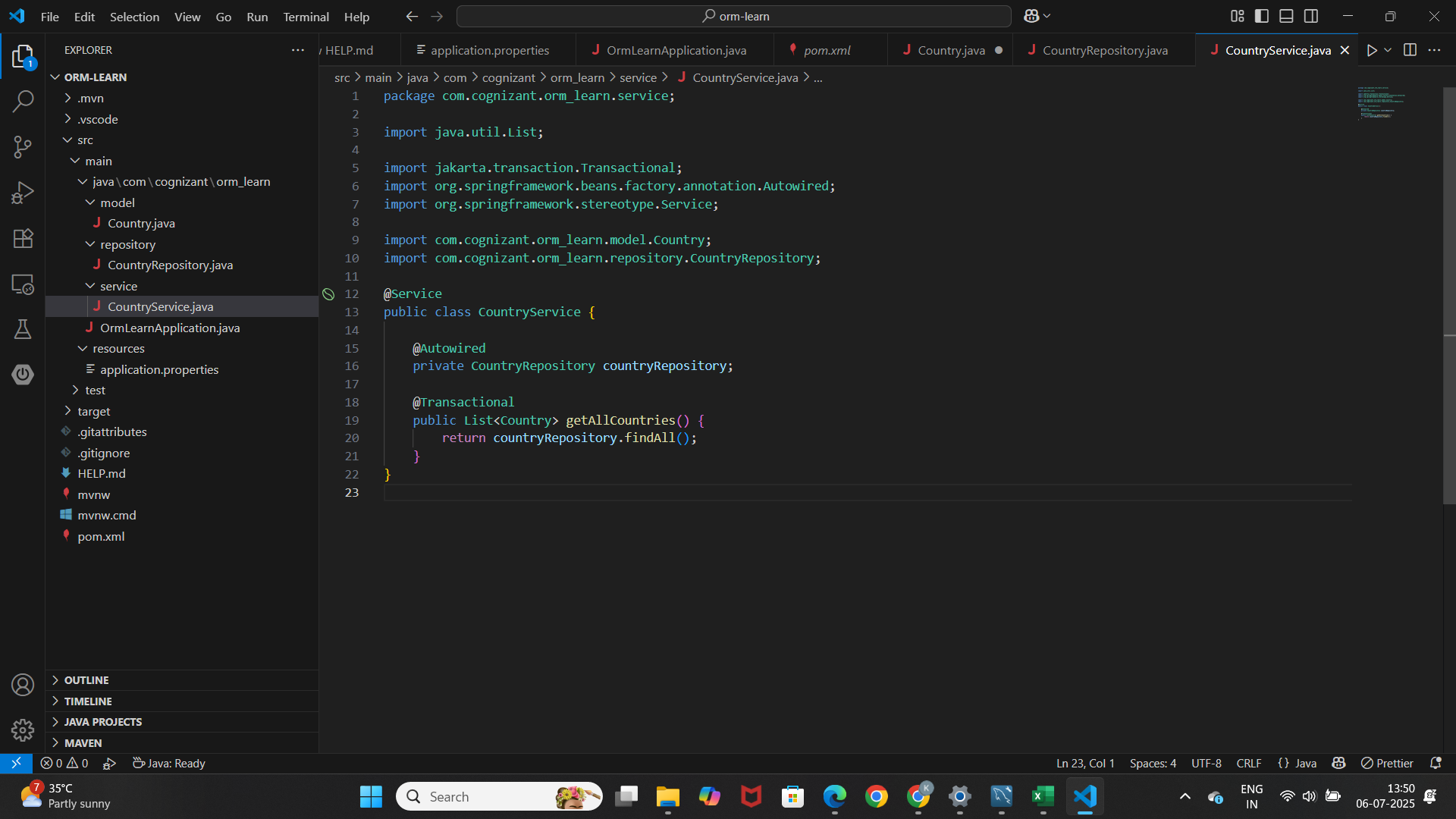
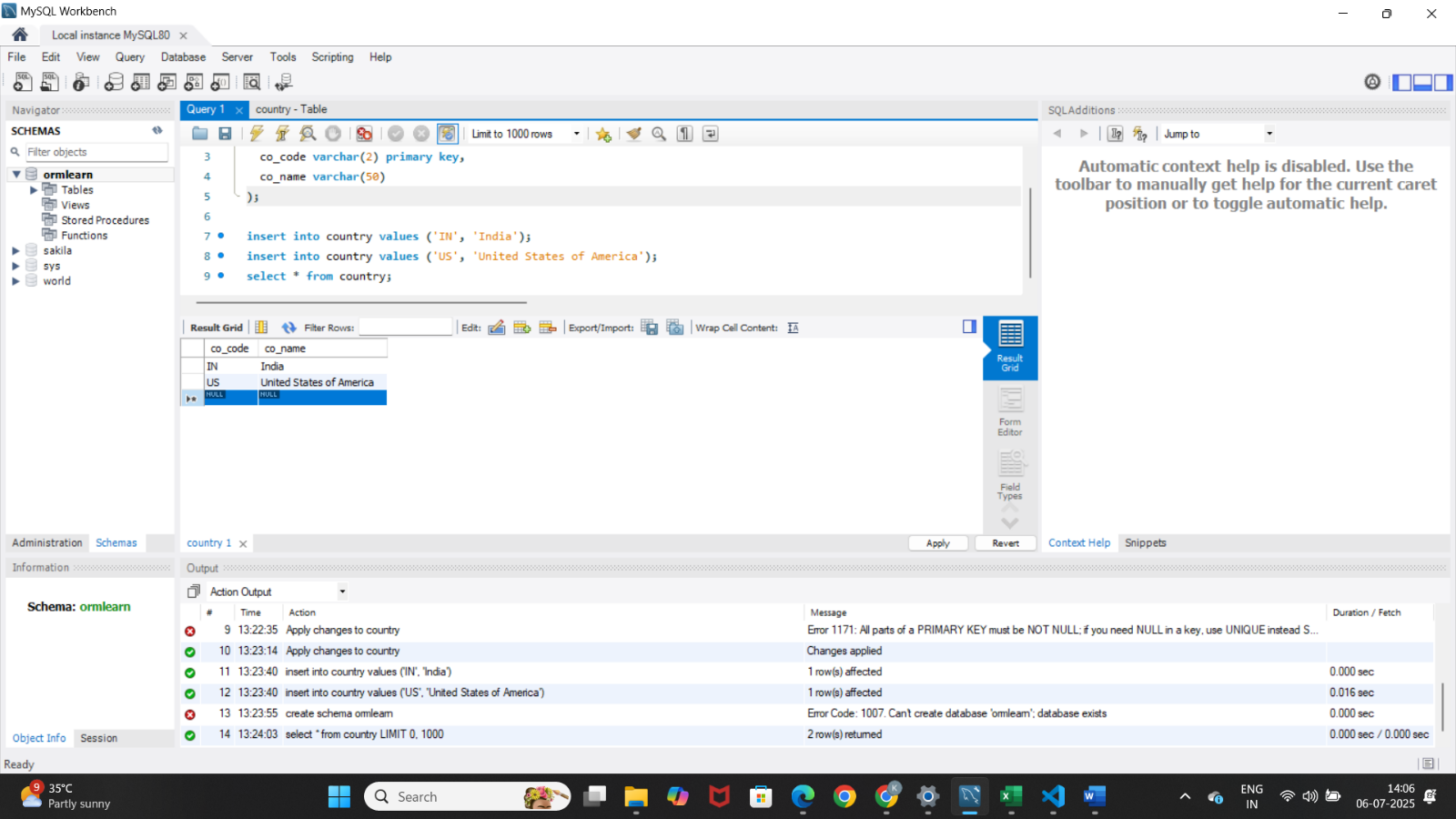


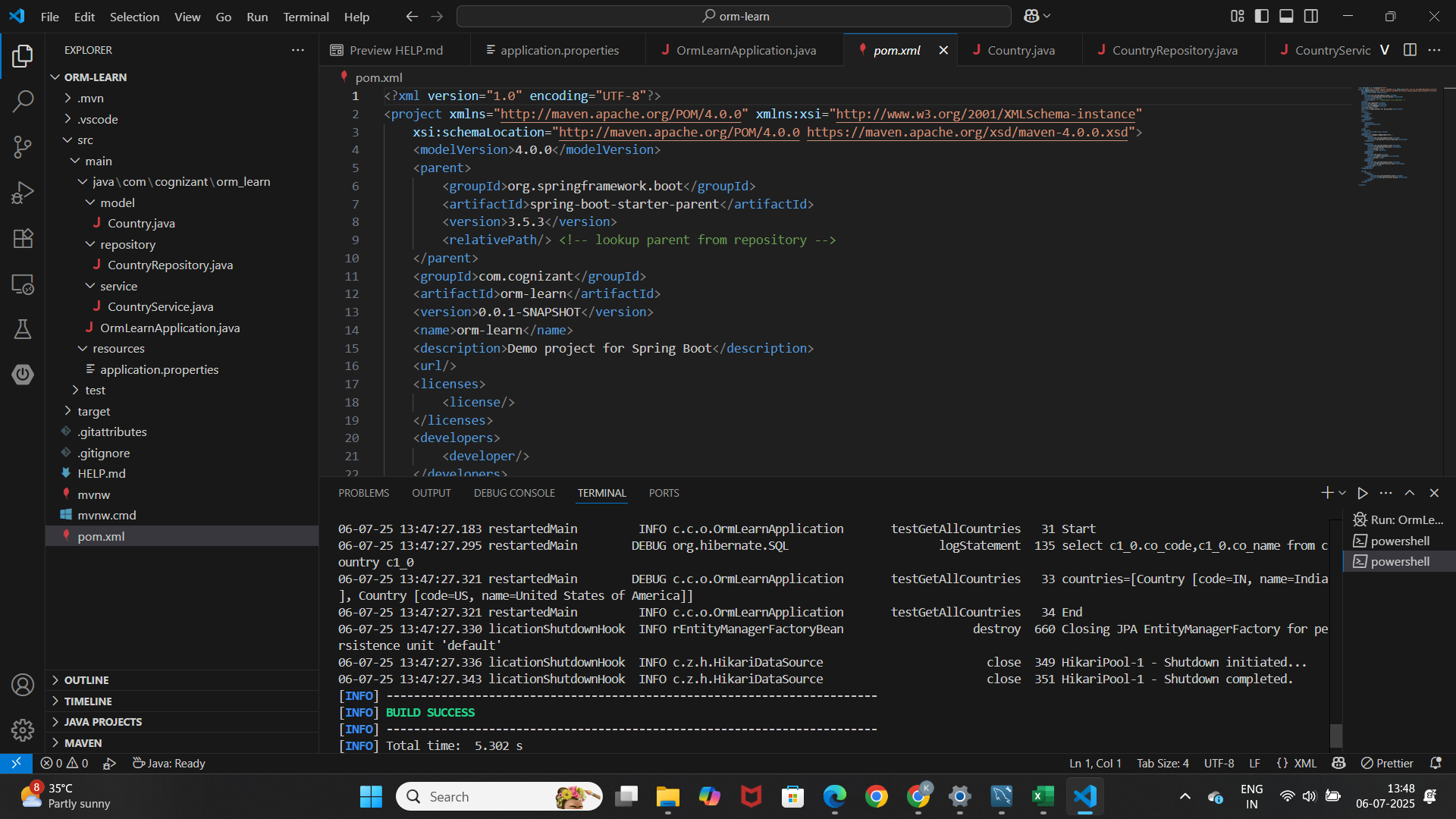
Output:

**1 Spring Data JPA - Quick Example**

****





Output:

**Exercise 4-Difference between JPA, Hibernate, and Spring Data JPA**

Jpa

* JPA is a specification that standardizes object-relational mapping (ORM) in Java.
* It defines a set of annotations and interfaces for mapping Java objects to relational database tables.
* JPA itself does not provide any concrete implementation, but serves as a contract for ORM frameworks to follow.
* Common JPA annotations include @Entity, @Id, and @Table.

Hibernate

* Hibernate is a widely adopted open-source ORM framework that implements the JPA specification.
* It provides the actual working code to perform object-relational mapping, query execution, and transaction management.
* Prior to Spring Data JPA, developers had to write substantial boilerplate code in Hibernate, managing sessions and transactions manually.

Spring data JPA

* Spring Data JPA is a higher-level abstraction built on top of JPA and Hibernate.
* It significantly reduces the amount of boilerplate code by providing ready-made repository interfaces such as JpaRepository.
* Spring Data JPA automatically manages transactions, sessions, and standard CRUD operations, increasing developer productivity and reducing error-prone code.
* Internally, Spring Data JPA still relies on Hibernate as the default JPA provider.

Code comparison

Hibernate

public Integer addEmployee(Employee employee){

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

} finally {

session.close();

}

return employeeID;

}

Spring data Jpa

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

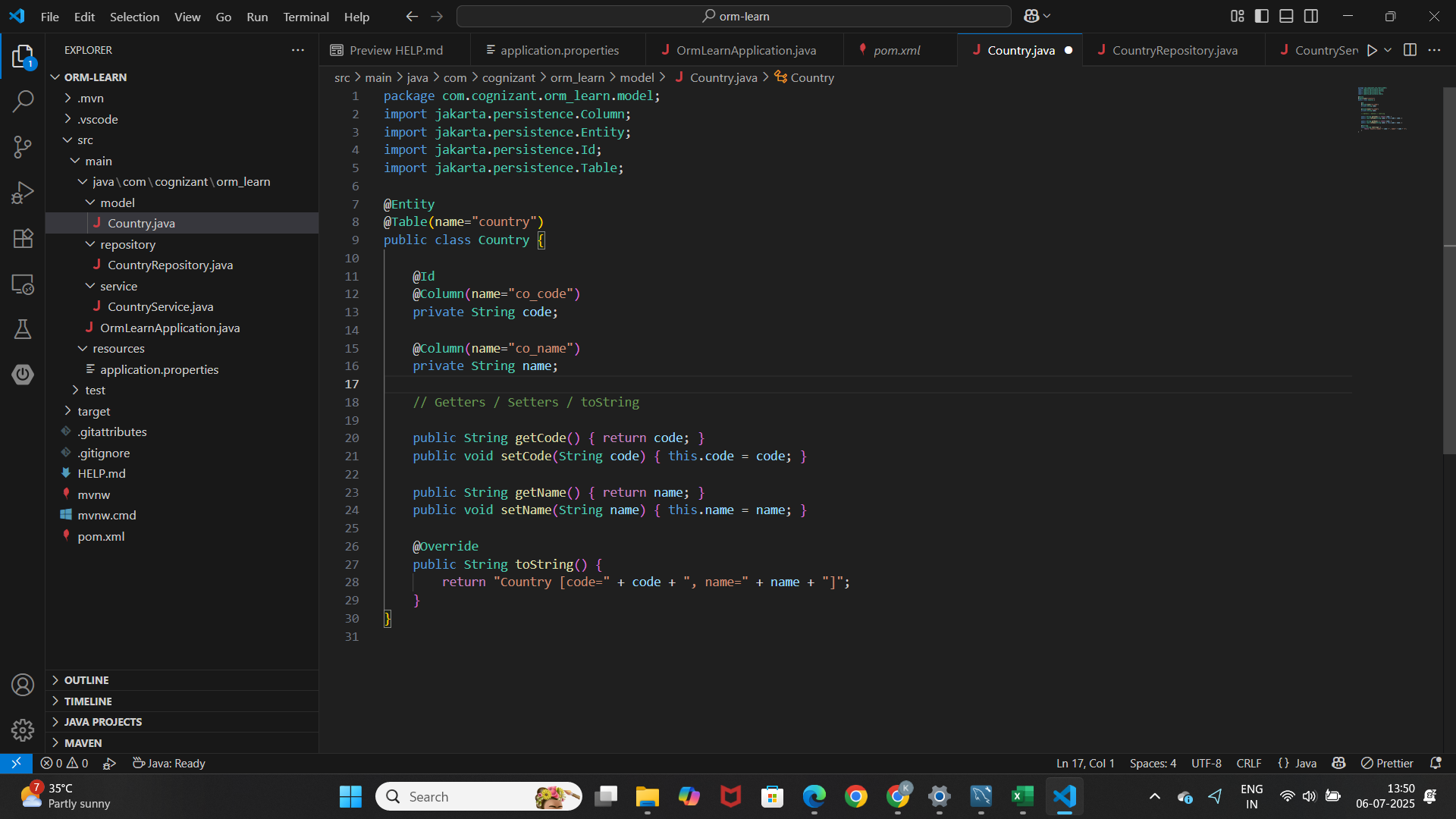
public void addEmployee(Employee employee) {

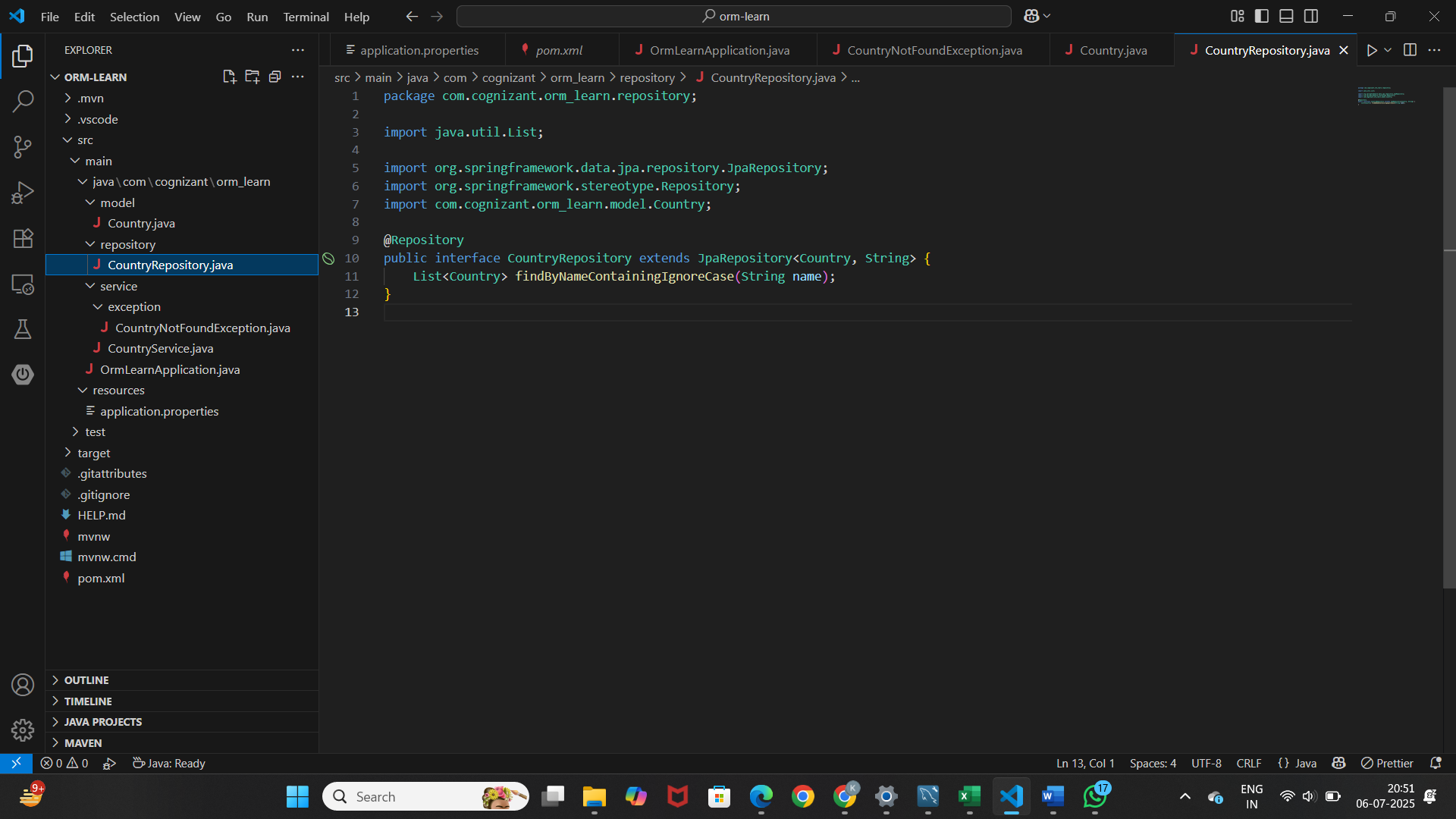
employeeRepository.save(employee);

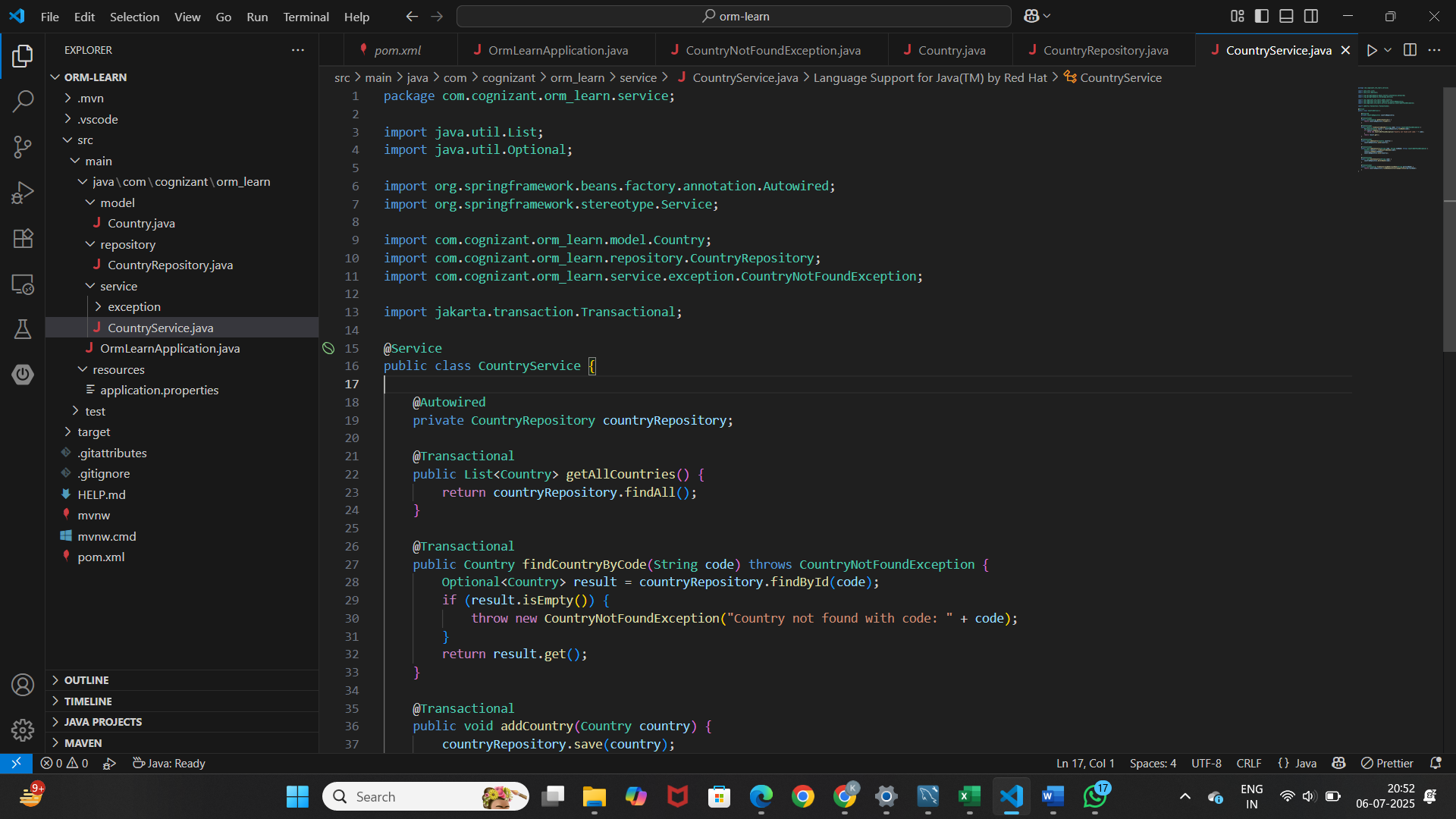
}

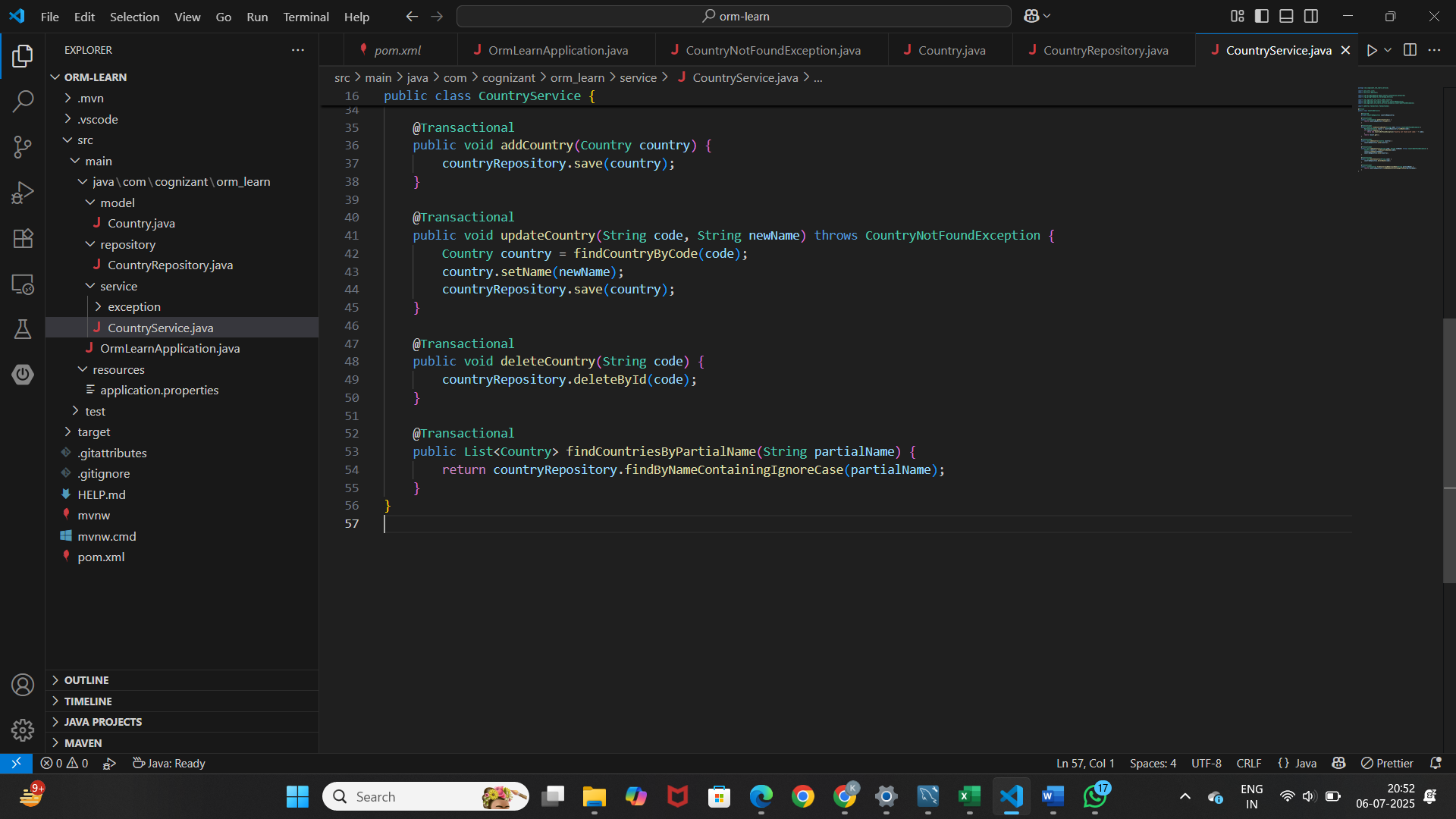
* JPA is a specification for ORM in Java.
* Hibernate is an implementation of JPA, handling low-level details.
* Spring Data JPA is an abstraction that simplifies and automates repository patterns using JPA and Hibernate under the hood.

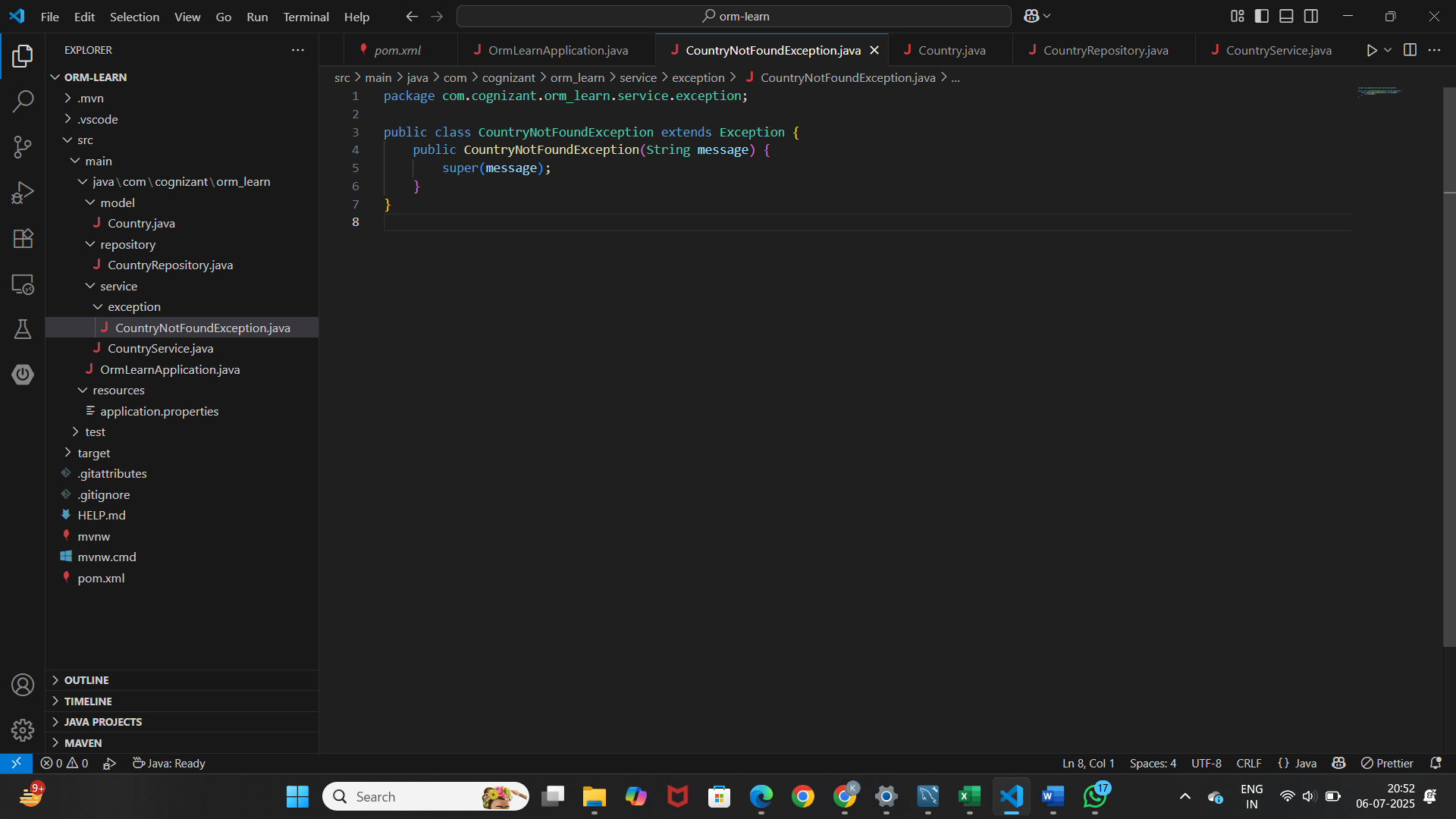
**Additional important hands-on**

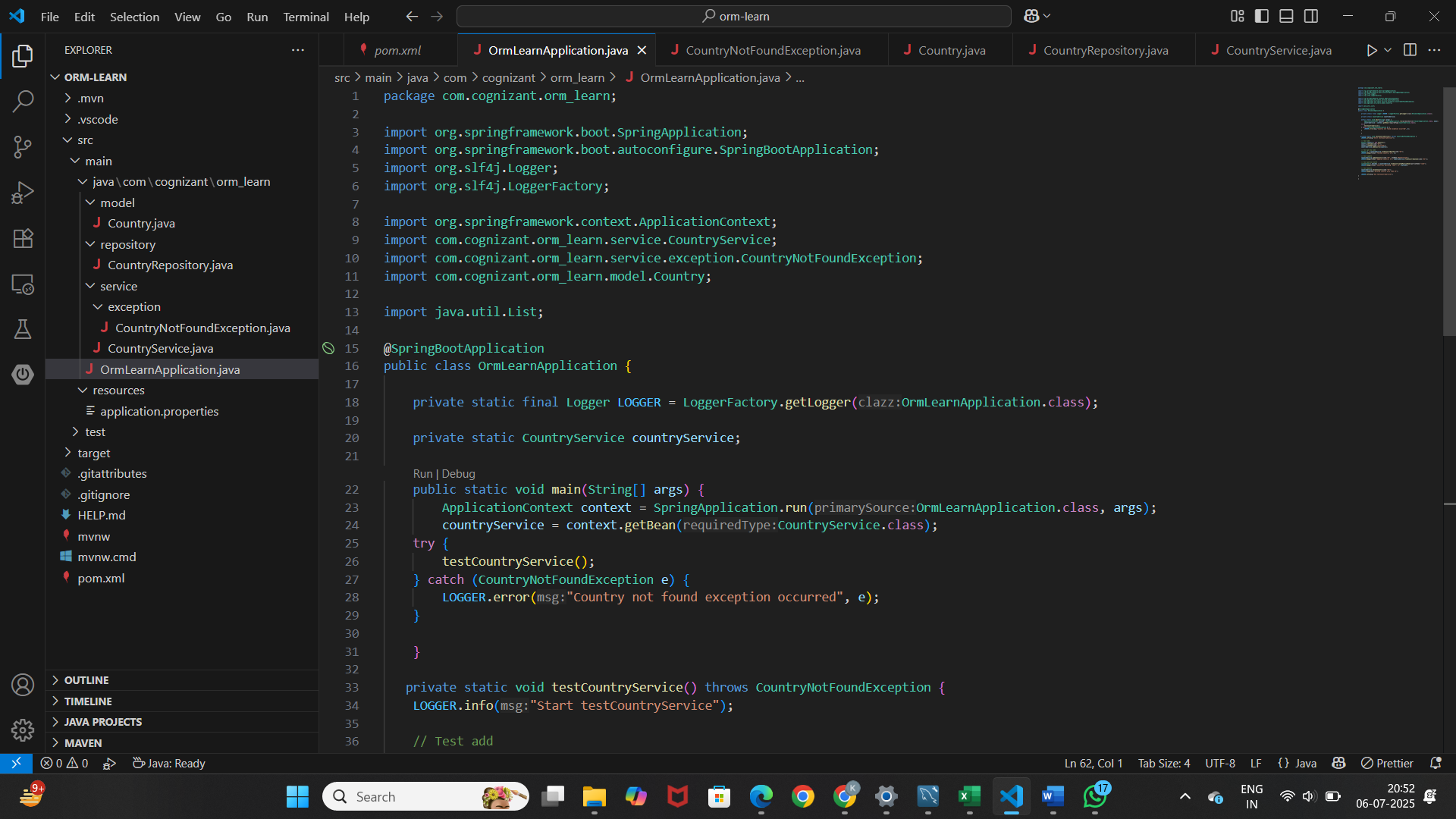
Exercise 5,6,7,8,9-Implement services for managing Country, Find a country based on country code, Add a new country, Update a country based on code, Delete a country based on code

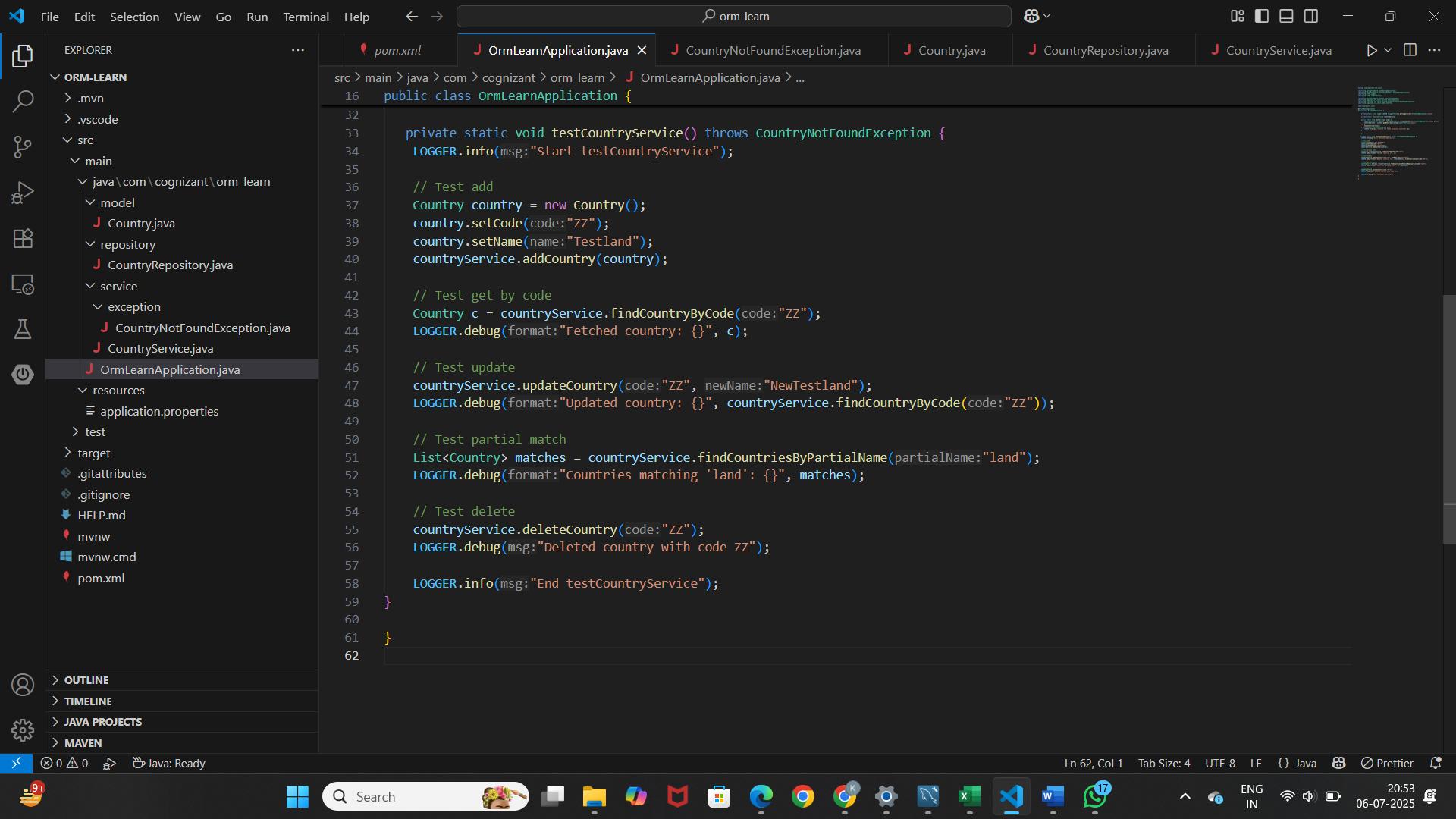












Output:

