**SPRING CORE AND MAVEN**

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

**Source Code:**

pom.xml

<projectxmlns="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>0.0.1-SNAPSHOT</version>  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 </dependencies>  
</project>

Bookrepository.java

package com.library.repository;

public class BookRepository {

public void saveBook() {

System.out.println("BookRepository: Book saved.");

}

}

BookService.java

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("BookService: Adding book...");

bookRepository.saveBook();

}

}

LibraryManagementApplication.java

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean("bookService", BookService.class);

bookService.addBook();

}

}

applicationContext.html

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

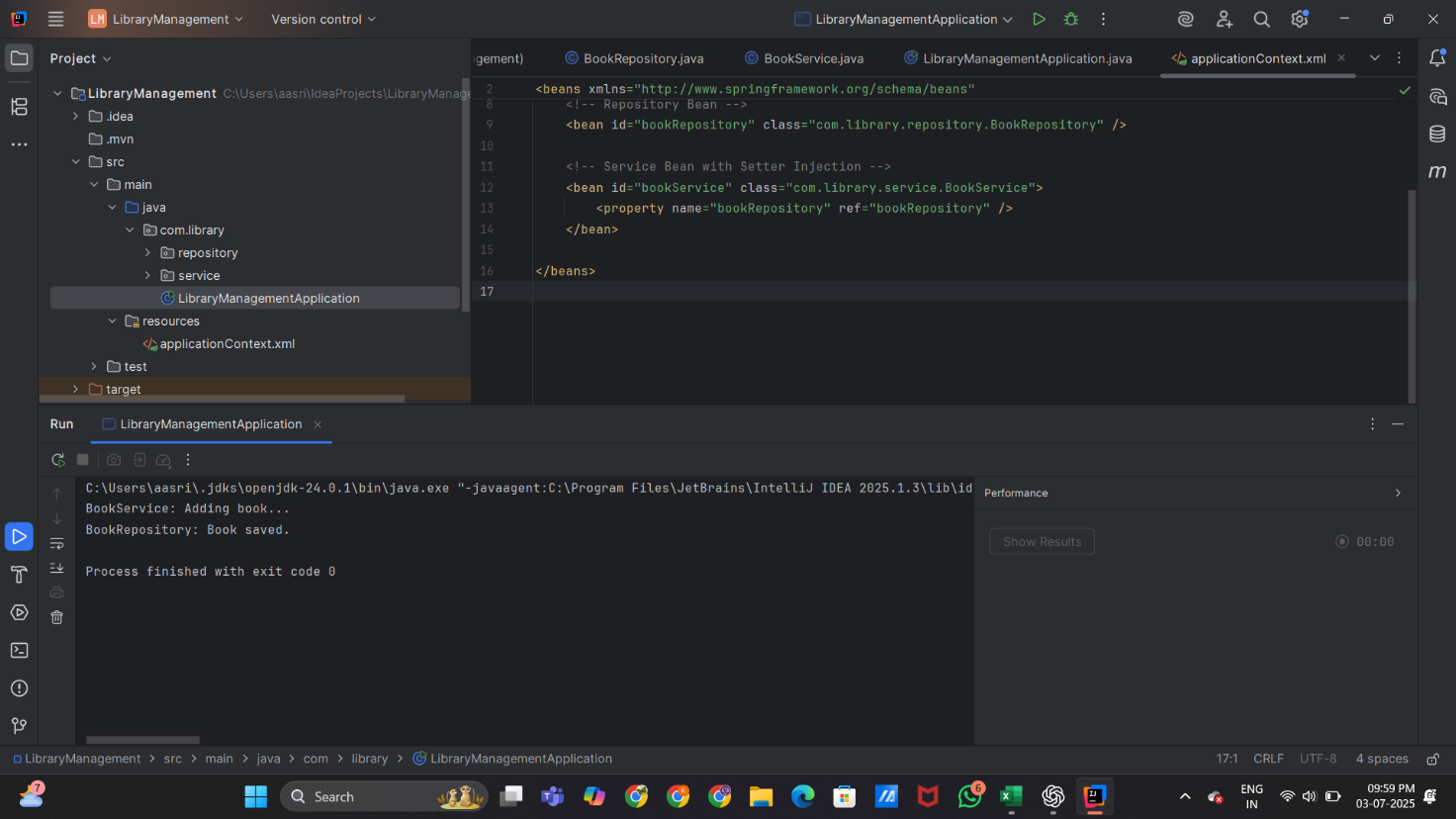
BookService bookService = context.getBean("bookService", BookService.class);

bookService.addBook();

}

}

**Output:**

****

**Exercise 2:** Implementing Dependency Injection

**Source Code:**

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.example</groupId>

<artifactId>UserManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

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

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

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.30</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.30</version>

</dependency>

</dependencies>

</project>

UserRepository.java

package com.example.repository;

public class UserRepository {

public void saveUser() {

System.out.println("UserRepository: User saved to database.");

}

}

UserService.java

package com.example.service;

import com.example.repository.UserRepository;

public class UserService {

private final UserRepository userRepository;

public UserService(UserRepository userRepository) {

this.userRepository = userRepository;

}

public void registerUser() {

System.out.println("UserService: Registering user...");

userRepository.saveUser();

}

}

UserApp.java

package com.example;

import com.example.service.UserService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class UserApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

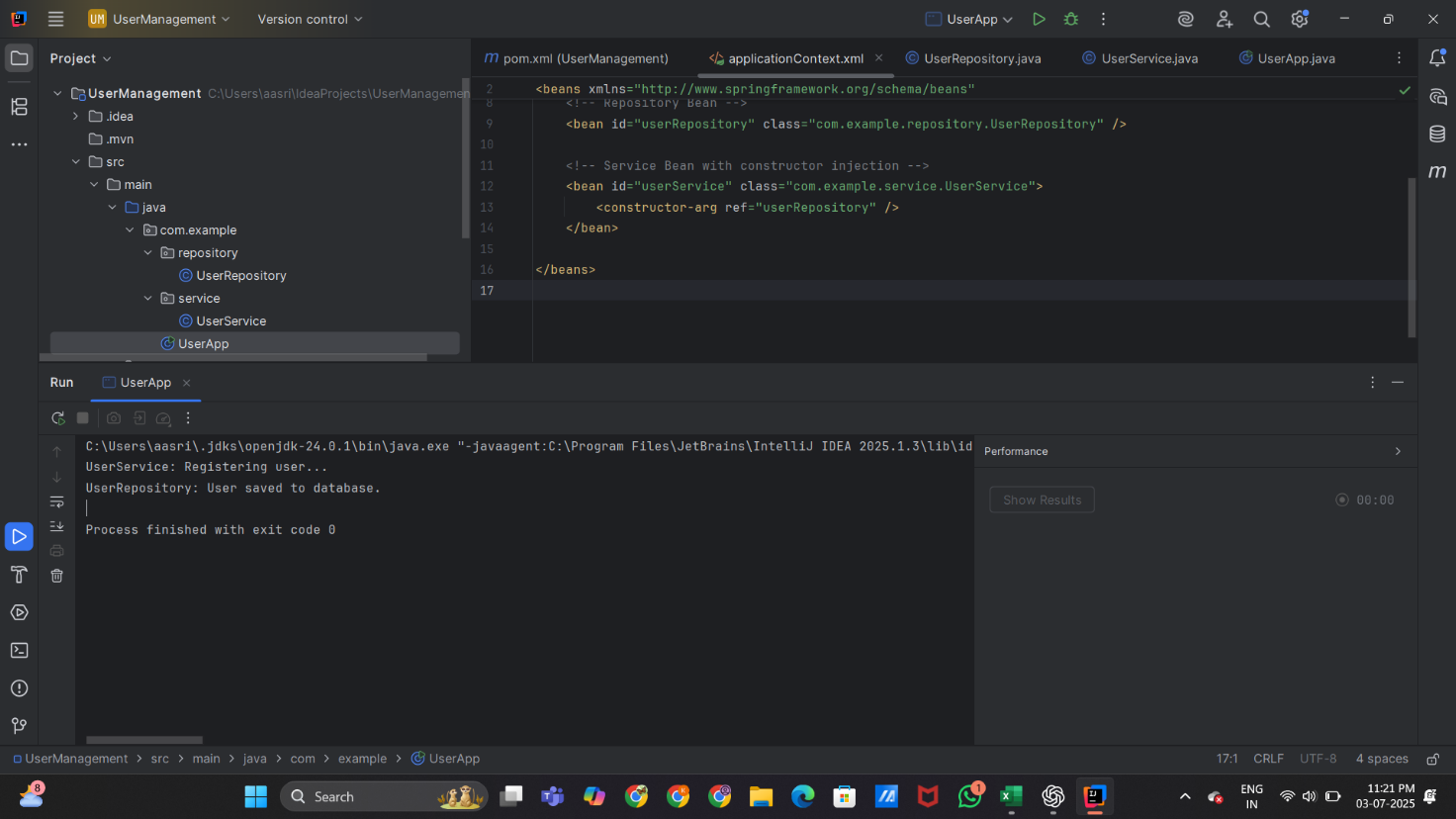
UserService userService = context.getBean("userService", UserService.class);

userService.registerUser();

}

}

**Output:**



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

**Source Code:**

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>org.example</groupId>

<artifactId>NewProject</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>24</maven.compiler.source>

<maven.compiler.target>24</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.30</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.30</version>

</dependency>

</dependencies>

</project>

UserService.java

package com.exercise.service;

import com.exercise.repository.UserRepository;

public class UserService {

private final UserRepository repo;

public UserService(UserRepository repo) {

this.repo = repo;

}

public void registerUser() {

System.out.println("UserService: Registering user...");

repo.save();

}}

UserRepository.java

package com.exercise.repository;

public class UserRepository {

public void save() {

System.out.println("UserRepository: User saved.");

}}

UserController.java

package com.exercise.controller;

import com.exercise.service.UserService;

public class UserController {

private final UserService service;

public UserController(UserService service) {

this.service = service;

}

public void handleRequest() {

System.out.println("UserController: Handling request...");

service.registerUser();

}

}

LoggingAspect.java

package com.exercise.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

@Aspect

public class LoggingAspect {

@Around("execution(\* com.exercise.service.\*.\*(..))")

public Object log(ProceedingJoinPoint joinPoint) throws Throwable {

System.out.println("[LOG] Method: " + joinPoint.getSignature().getName() + " started");

Object result = joinPoint.proceed();

System.out.println("[LOG] Method: " + joinPoint.getSignature().getName() + " ended");

return result;

}

}

Main.java

package com.exercise;

import com.exercise.controller.UserController;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

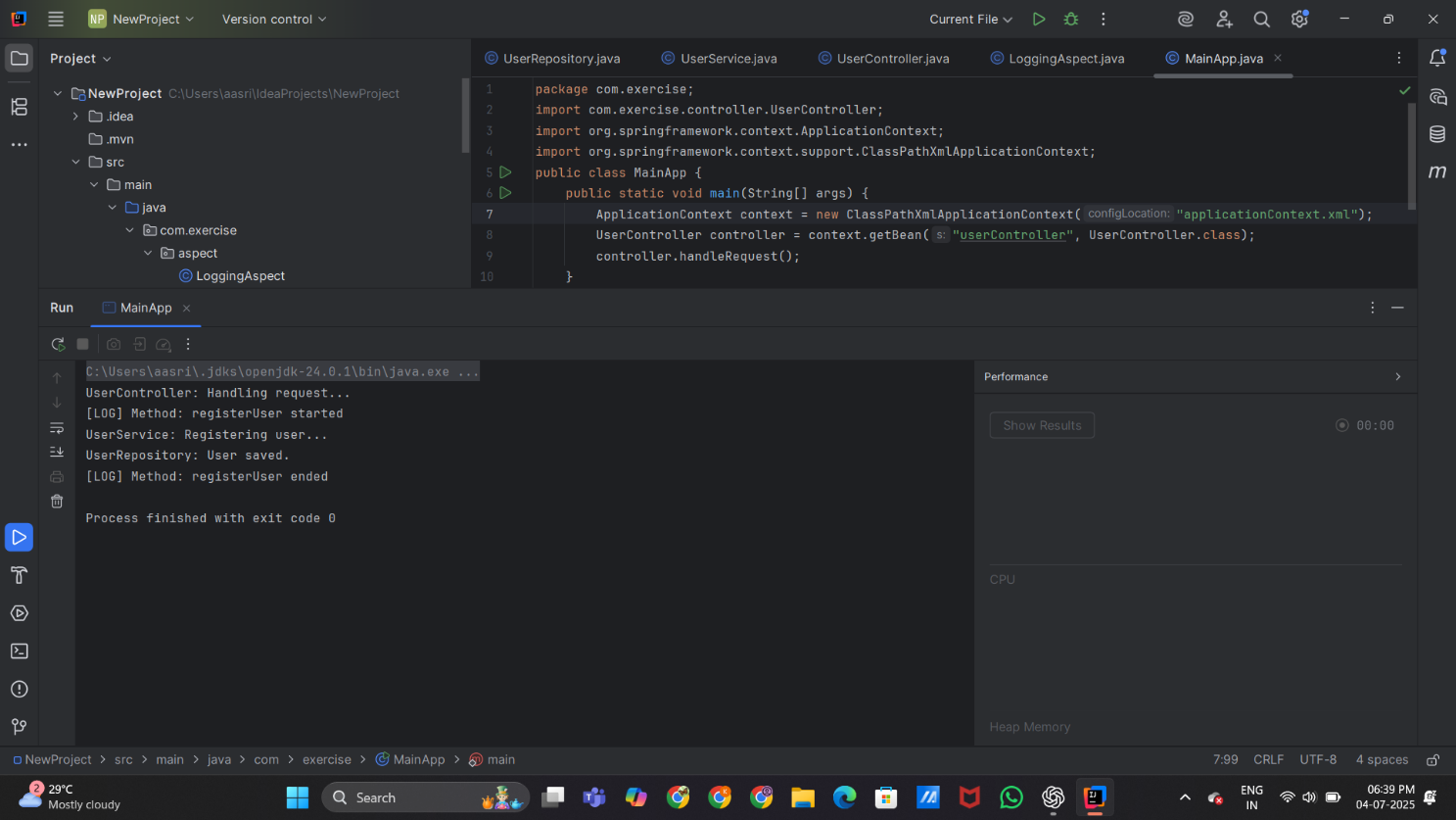
public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

UserController controller = context.getBean("userController", UserController.class);

controller.handleRequest();

}}

**Output:**