**Exercise 1: 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>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>6.1.6</version>  
 </dependency>  
 </dependencies>  
</project>

**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">  
 <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**

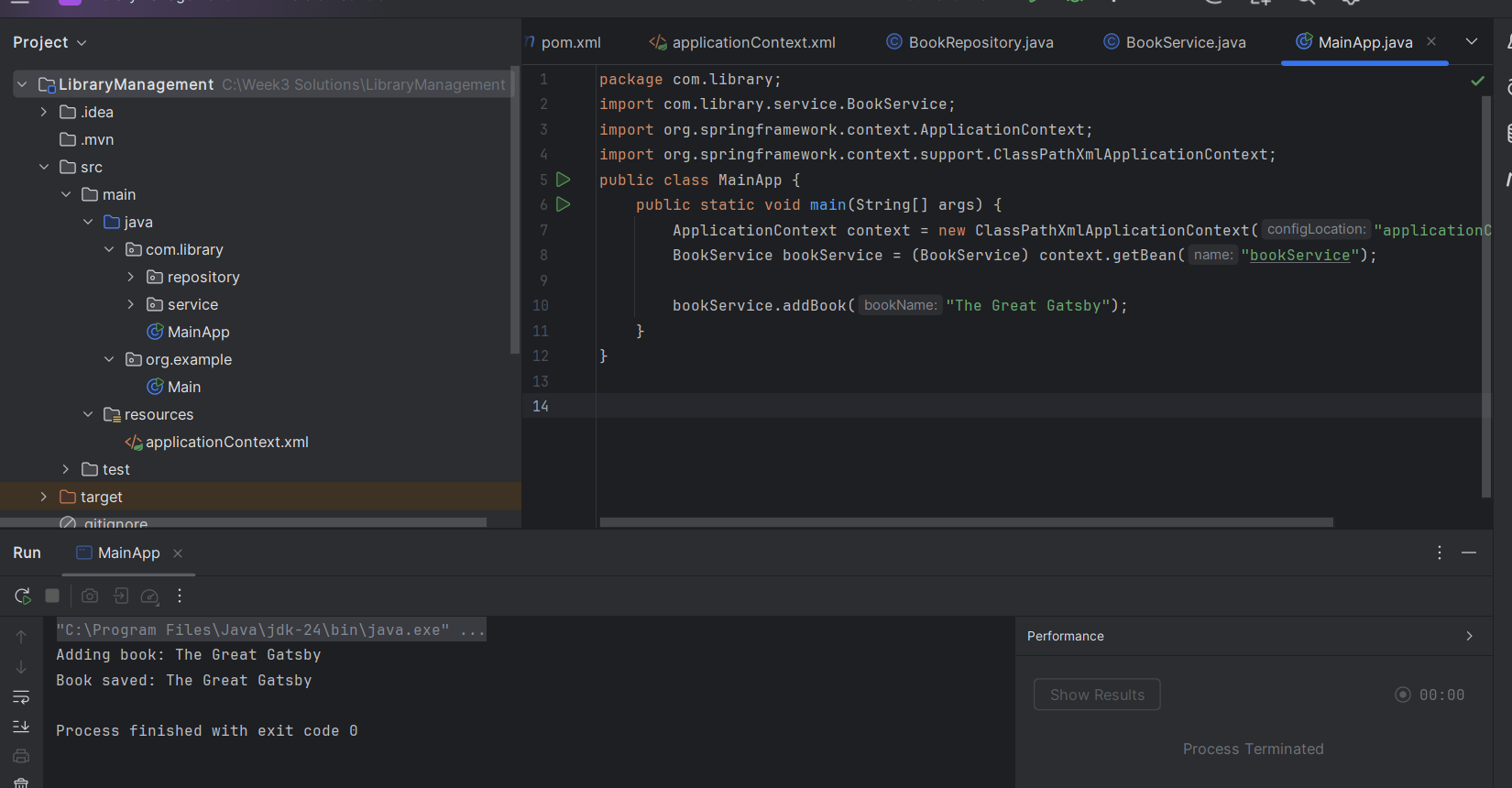
package com.library.repository;  
public class BookRepository {  
 public void saveBook(String bookName) {  
 System.*out*.println("Book saved: " + bookName);  
 }  
}

**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(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

**MainApp.java**

package com.library;  
import com.library.service.BookService;  
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");  
 BookService bookService = (BookService) context.getBean("bookService");  
  
 bookService.addBook("The Great Gatsby");  
 }  
}

**Output**

**Exercise 2: Implementing Dependency Injection**

**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>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>6.1.6</version>  
 </dependency>  
 </dependencies>  
</project>

**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">  
 <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 saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

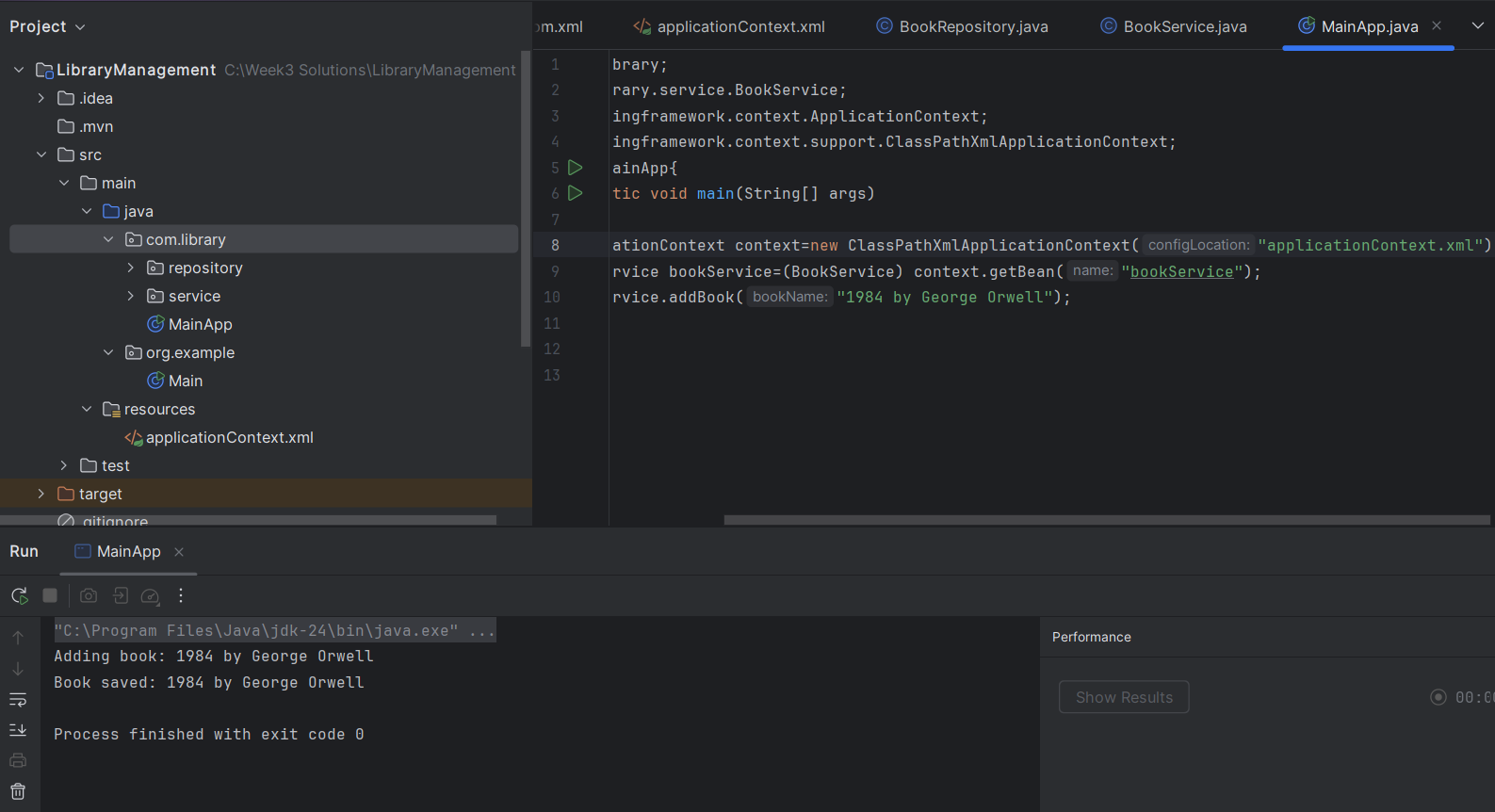
**BookService.java**

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(String bookName)  
 {  
 System.*out*.println("Adding book: " +bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

**MainApp.java**

package com.library;  
import com.library.service.BookService;  
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");  
 BookService bookService=(BookService) context.getBean("bookService");  
 bookService.addBook("1984 by George Orwell");  
 }  
}

**Output**



**Exercise 3: Implementing Logging with Spring AOP**

**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>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 </dependencies>  
</project>

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 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  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
 <aop:aspectj-autoproxy />  
 <bean id="loggingAspect" class="com.library.aspect.LoggingAspect" />  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

**LoggingAspect.java**

package com.library.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.library.service.\*.\*(..))")  
 public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable  
 {  
 long start=System.*nanoTime*();  
 Object result=joinPoint.proceed();  
 long end=System.*nanoTime*();  
 String methodName=joinPoint.getSignature().toShortString();  
 System.*out*.println("[LOG] Execution time of "+methodName+": "+(end-start)/1\_000\_000+" ms");  
 return result;  
 }  
}

**LibraryManagementApplication**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService=(BookService) context.getBean("bookService");  
 bookService.addBook("Brave New World by Aldous Huxley");  
 }  
}

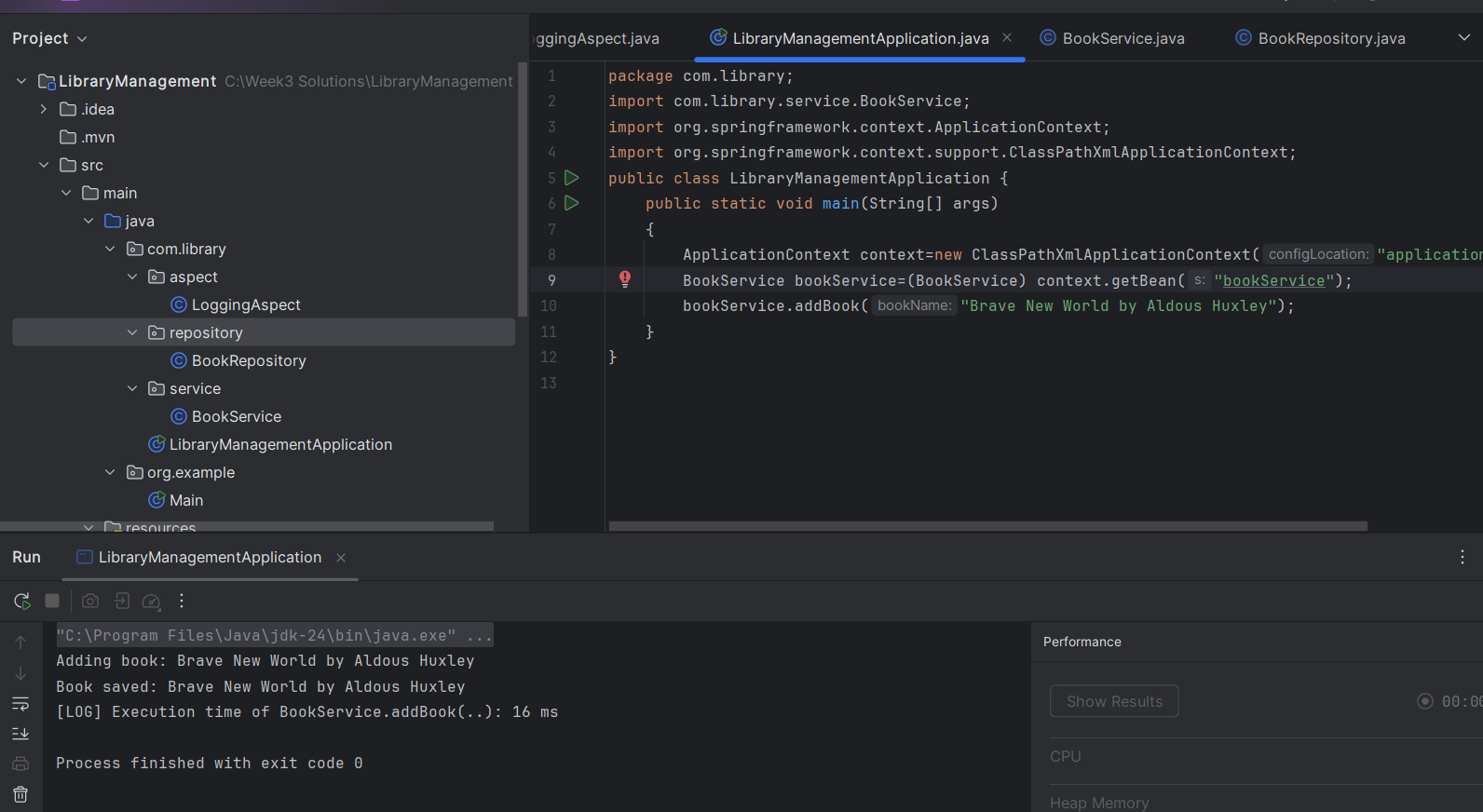
**BookRepository.java**

package com.library.repository;  
public class BookRepository {  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**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(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
 public void removeBook(String bookName)  
 {  
 System.*out*.println("Removing book: "+bookName);  
 }  
}

**Output**

****

**Exercise 4: 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.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <properties>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <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>

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 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  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
 <aop:aspectj-autoproxy />  
 <bean id="loggingAspect" class="com.library.aspect.LoggingAspect" />  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

**LoggingAspect.java**

package com.library.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.library.service.\*.\*(..))")  
 public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable  
 {  
 long start=System.*nanoTime*();  
 Object result=joinPoint.proceed();  
 long end=System.*nanoTime*();  
 String methodName=joinPoint.getSignature().toShortString();  
 System.*out*.println("[LOG] Execution time of "+methodName+": "+(end-start)/1\_000\_000+" ms");  
 return result;  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService = context.getBean(BookService.class);  
 bookService.addBook("The Alchemist");  
 }  
}

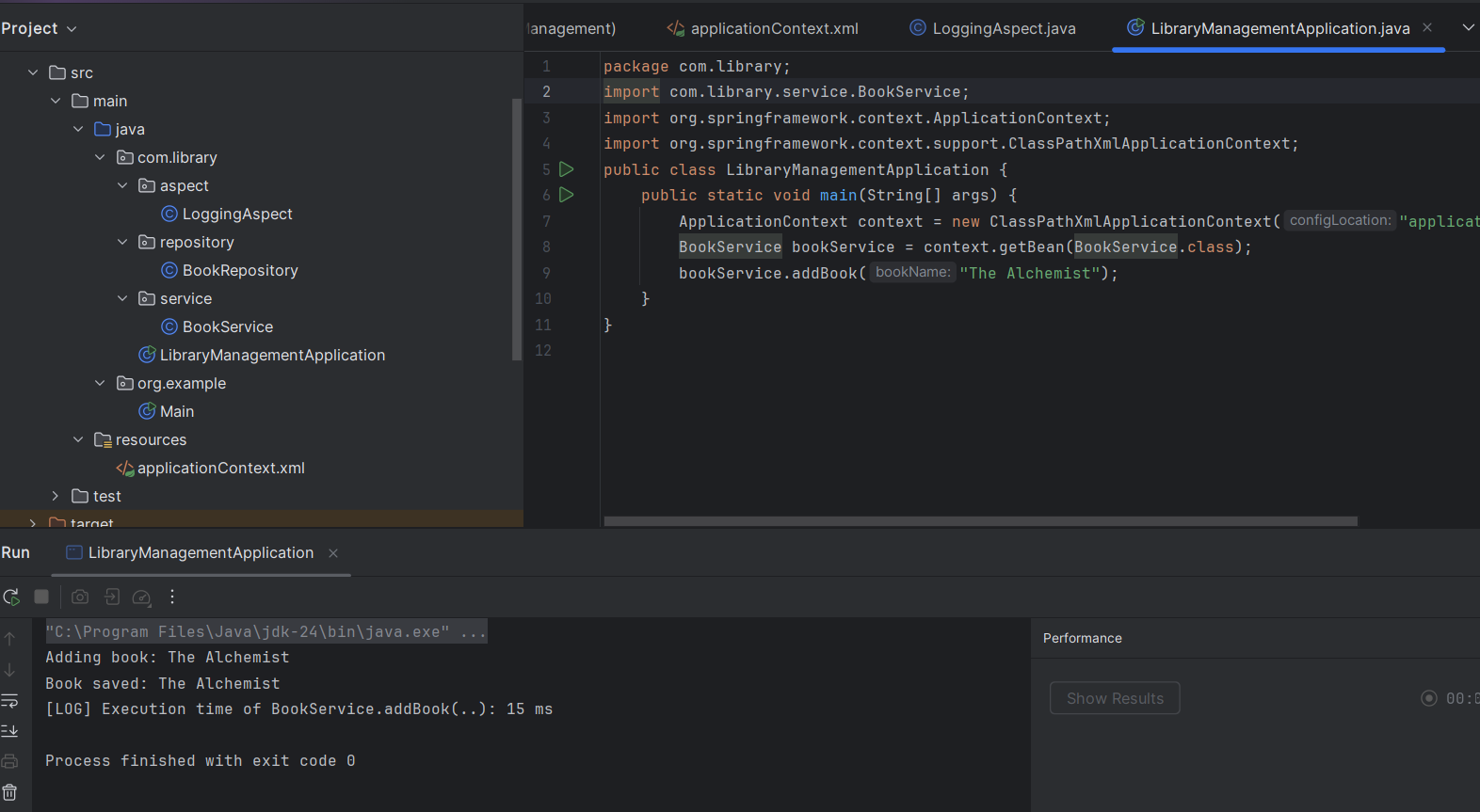
**BookRepository.java**

package com.library.repository;  
public class BookRepository {  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**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(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
 public void removeBook(String bookName)  
 {  
 System.*out*.println("Removing book: "+bookName);  
 }  
}

**Output**



**Exercise 5: Configuring the Spring IoC Container**

**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>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <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>

**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">  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
</beans>

**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(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

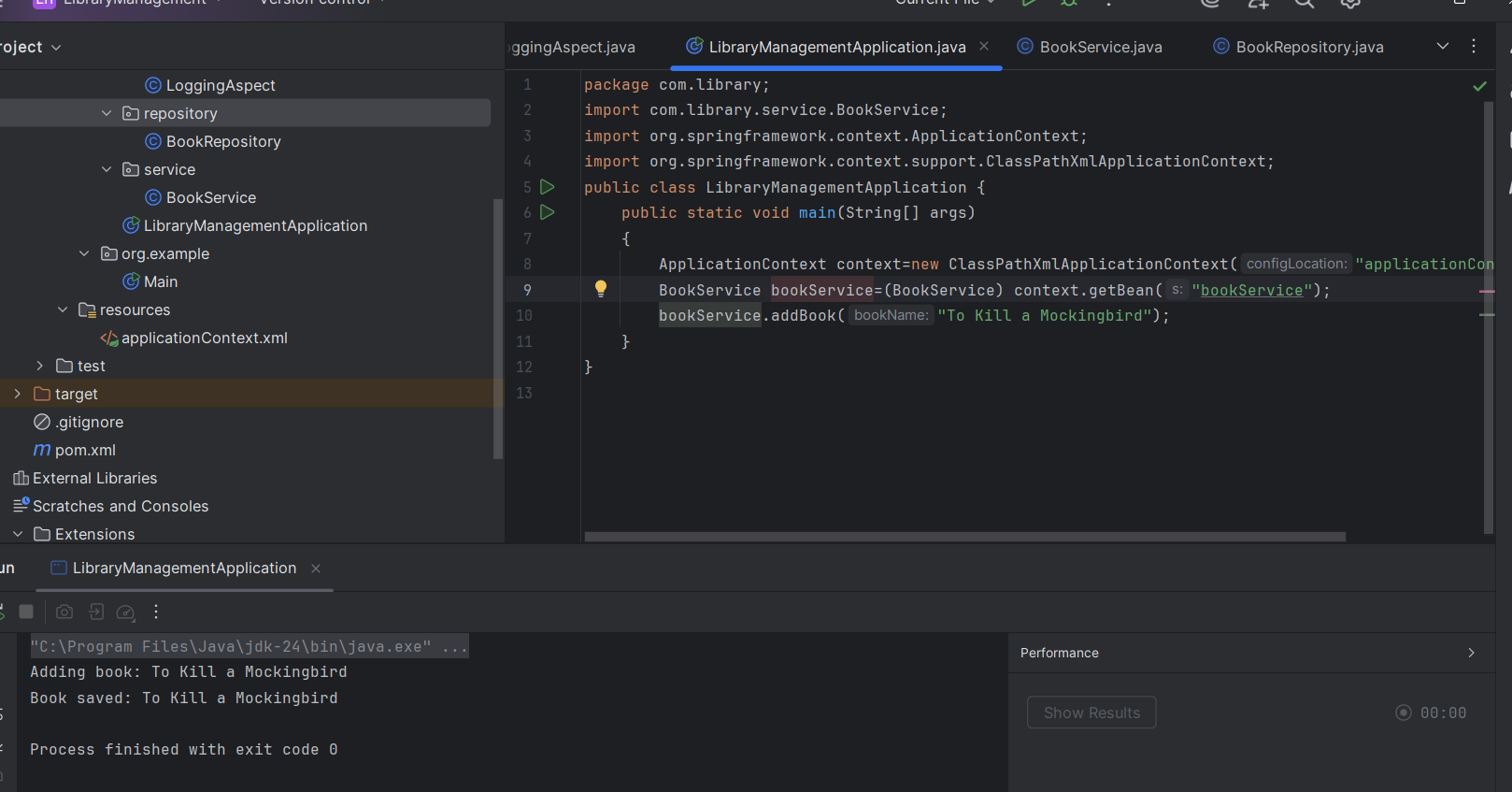
**BookRepository.java**

package com.library.repository;  
public class BookRepository {  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService=(BookService) context.getBean("bookService");  
 bookService.addBook("To Kill a Mockingbird");  
 }  
}

**Output**

****

**Exercise 6: Configuring Beans with Annotations**

**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>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <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>

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:context="http://www.springframework.org/schema/context"  
 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  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd">  
 <context:component-scan base-package="com.library" />  
</beans>

**BookService.java**

package com.library.service;  
import com.library.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
@Service  
public class BookService  
{  
 @Autowired  
 private BookRepository bookRepository;  
 public void addBook(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

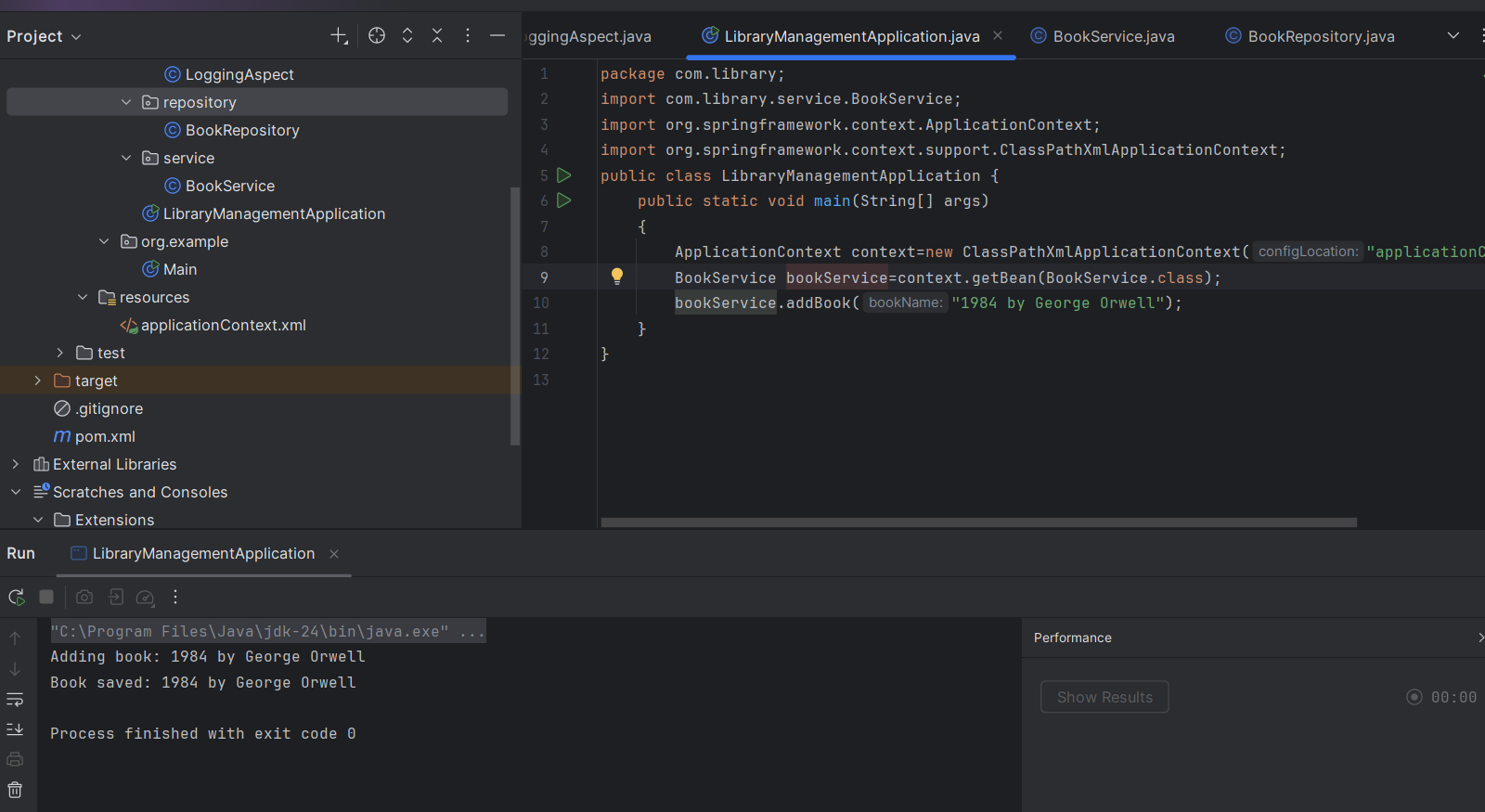
**BookRepository.java**

package com.library.repository;  
import org.springframework.stereotype.Repository;  
@Repository  
public class BookRepository  
{  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService=context.getBean(BookService.class);  
 bookService.addBook("1984 by George Orwell");  
 }  
}

**Output**

****

**Exercise 7: Implementing Constructor and Setter Injection**

**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>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <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>

**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">  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
 <bean id="bookService" class="com.library.service.BookService">  
 <constructor-arg value="LibraryService" />  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
</beans>

**BookService.java**

package com.library.service;  
import com.library.repository.BookRepository;  
public class BookService  
{  
 private String serviceName;  
 private BookRepository bookRepository;  
 public BookService(String serviceName)  
 {  
 this.serviceName=serviceName;  
 }  
 public void setBookRepository(BookRepository bookRepository){  
 this.bookRepository=bookRepository;  
 }  
 public void addBook(String bookName) {  
 System.*out*.println("["+serviceName+"] Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

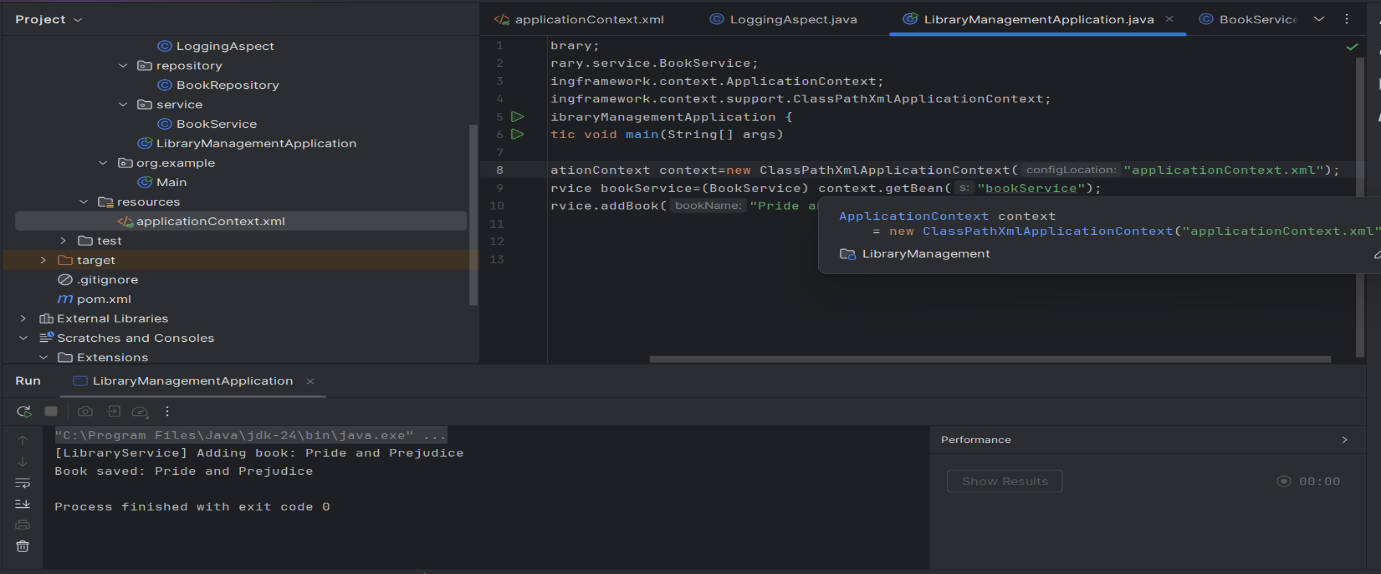
**BookRepository.java**

package com.library.repository;  
public class BookRepository {  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService=(BookService) context.getBean("bookService");  
 bookService.addBook("Pride and Prejudice");  
 }  
}

**Output**



**Exercise 8: Implementing Basic AOP with Spring**

**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>  
 <java.version>1.8</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <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>

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 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  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
 <context:component-scan base-package="com.library" />  
 <aop:aspectj-autoproxy />  
</beans>

**LoggingAspect.java**

package com.library.aspect;  
import org.aspectj.lang.JoinPoint;  
import org.aspectj.lang.annotation.After;  
import org.aspectj.lang.annotation.Aspect;  
import org.aspectj.lang.annotation.Before;  
import org.springframework.stereotype.Component;  
@Aspect  
@Component  
public class LoggingAspect  
{  
 @Before("execution(\* com.library.service.\*.\*(..))")  
 public void logBefore(JoinPoint joinPoint) {  
 System.*out*.println("[LOG] Before executing: "+joinPoint.getSignature().getName());  
 }  
 @After("execution(\* com.library.service.\*.\*(..))")  
 public void logAfter(JoinPoint joinPoint) {  
 System.*out*.println("[LOG] After executing: "+joinPoint.getSignature().getName());  
 }  
}

**BookService.java**

package com.library.service;  
import com.library.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
@Service  
public class BookService {  
 @Autowired  
 private BookRepository bookRepository;  
 public void addBook(String bookName)  
 {  
 System.*out*.println("Adding book: "+bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

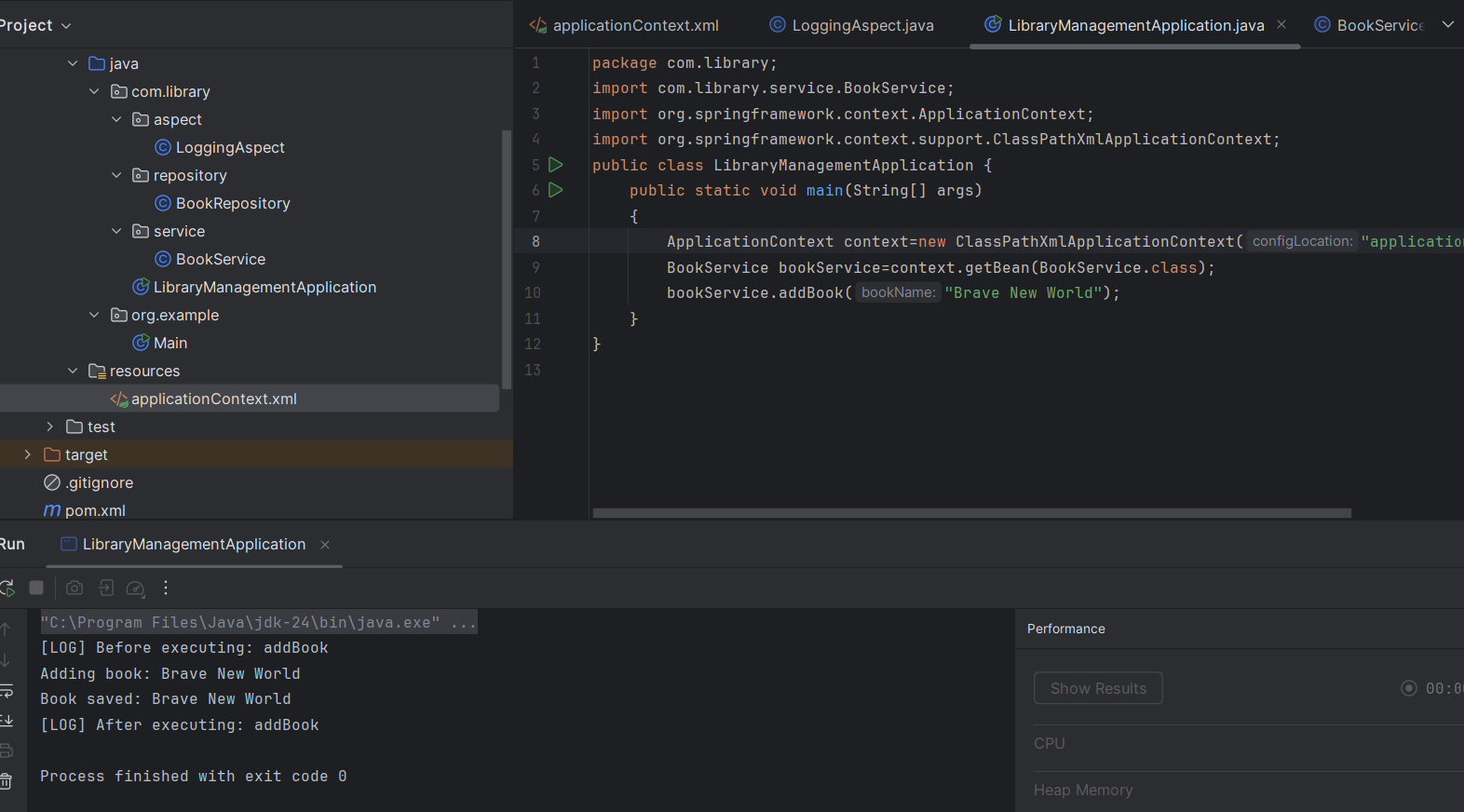
**BookRepository.java**

package com.library.repository;  
import org.springframework.stereotype.Repository;  
@Repository  
public class BookRepository  
{  
 public void saveBook(String bookName)  
 {  
 System.*out*.println("Book saved: "+bookName);  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService=context.getBean(BookService.class);  
 bookService.addBook("Brave New World");  
 }  
}

**Output**

****

**Exercise 9: Creating a Spring Boot 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.example</groupId>  
 <artifactId>library-management</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
 <name>LibraryManagement</name>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.1.9</version>  
 <relativePath/>  
 </parent>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <repositories>  
 <repository>  
 <id>central</id>  
 <url>https://repo.maven.apache.org/maven2</url>  
 </repository>  
 </repositories>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>com.h2database</groupId>  
 <artifactId>h2</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**application.properties**

spring.datasource.url=jdbc:h2:mem:librarydb  
spring.datasource.driverClassName=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect  
spring.jpa.hibernate.ddl-auto=update  
spring.h2.console.enabled=true  
spring.h2.console.path=/h2-console

**BookController.java**

package com.example.library.controller;  
import com.example.library.entity.Book;  
import com.example.library.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
@RestController  
@RequestMapping("/api/books")  
public class BookController  
{  
 @Autowired  
 private BookRepository bookRepository;  
 @GetMapping  
 public List<Book> getAllBooks()  
 {  
 return bookRepository.findAll();  
 }  
 @PostMapping  
 public Book addBook(@RequestBody Book book) {  
 return bookRepository.save(book);  
 }  
 @GetMapping("/{id}")  
 public Book getBookById(@PathVariable Long id) {  
 return bookRepository.findById(id).orElse(null);  
 }  
 @PutMapping("/{id}")  
 public Book updateBook(@PathVariable Long id,@RequestBody Book bookDetails) {  
 Book book=bookRepository.findById(id).orElse(null);  
 if (book!=null)  
 {  
 book.setTitle(bookDetails.getTitle());  
 book.setAuthor(bookDetails.getAuthor());  
 return bookRepository.save(book);  
 }  
 return null;  
 }  
 @DeleteMapping("/{id}")  
 public void deleteBook(@PathVariable Long id) {  
 bookRepository.deleteById(id);  
 }  
}

**Book.java**

package com.example.library.entity;  
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 Long getId() { return id; }  
 public void setId(Long id) { this.id=id; }  
 public String getTitle() { return title; }  
 public void setTitle(String title) { this.title=title; }  
 public String getAuthor() { return author; }  
 public void setAuthor(String author) { this.author=author; }  
}

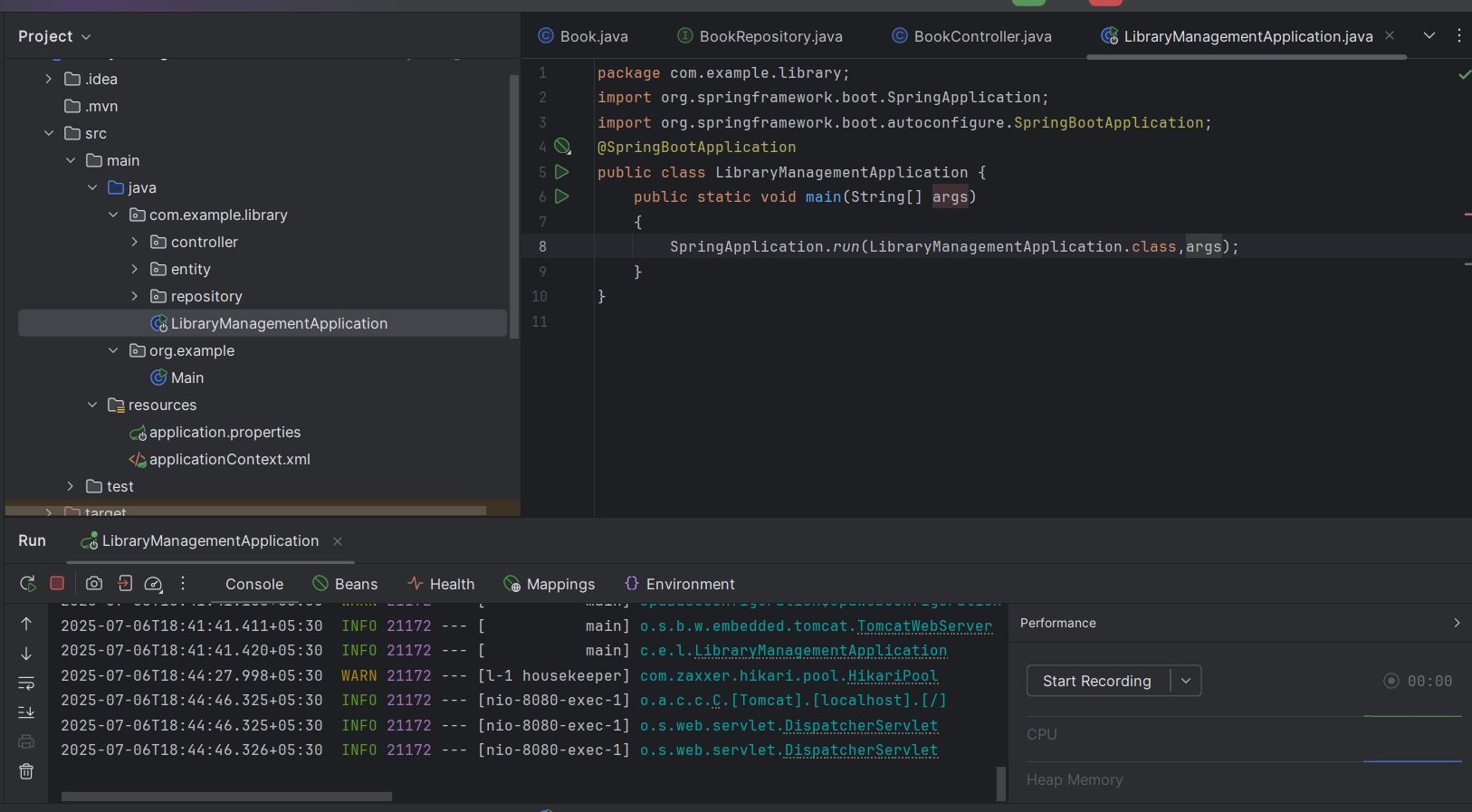
**BookRepository.java**

package com.example.library.repository;  
import com.example.library.entity.Book;  
import org.springframework.data.jpa.repository.JpaRepository;  
public interface BookRepository extends JpaRepository<Book, Long> {}

**LibraryManagementApplication.java**

package com.example.library;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
@SpringBootApplication  
public class LibraryManagementApplication {  
 public static void main(String[] args)  
 {  
 SpringApplication.*run*(LibraryManagementApplication.class,args);  
 }  
}

**Output**

****