**Exercise 2: Implementing Dependency Injection**

**Step 1: Modify the XML Configuration**

Update applicationContext.xml to wire BookRepository into BookService:

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

<beans xmlns="(link unavailable)"

xmlns:xsi="(link unavailable)"

xsi:schemaLocation="(link unavailable)

(link unavailable)">

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

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

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

</bean>

</beans>

**Step 2: Update the BookService Class**

Ensure that BookService class has a setter method for BookRepository:

package com.library.service;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

// Service methods

}

**Step 3: Test the Configuration**

Run the LibraryManagementApplication main class to verify the dependency injection:

package com.library;

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);

System.out.println("Book Service: " + bookService);

System.out.println("Book Repository: " + bookService.getBookRepository());

}

}

**Output:**

Book Service: com.library.service.BookService@12345678

Book Repository: com.library.repository.BookRepository@90123456