**Exercise 2 : Implementing Dependency Injection**

Scenario :

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's loC and DI.

**1. XML Configuration**

(applicationContext.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

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

<!-- Define the BookRepository bean -->

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

<!-- Define the BookService bean and inject BookRepository -->

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

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

</bean>

</beans>

**2. Update BookService Class**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter method for DI

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void displayBooks() {

bookRepository.findAllBooks().forEach(System.out::println);

}

}

**3. BookRepository Class**

package com.example.repository;

import java.util.Arrays;

import java.util.List;

public class BookRepository {

public List<String> findAllBooks() {

return Arrays.asList("The Alchemist", "1984", "Clean Code");

}

}

**4. Main Class to Test**

(LibraryManagementApplication.java)

package com.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.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.displayBooks();

}

}

**Output :**

