**Exercise 2: Implementing Dependency Injection**

**Steps:**

1. **Modify the XML Configuration:**
   * Update **applicationContext.xml** to wire **BookRepository** into **BookService**.

<?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 = *"repositorybean"* class=*"com.library.repository.BookRepository"*/>

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

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

</bean>

</beans>

1. **Update the BookService Class:**
   * Ensure that **BookService** class has a setter method for **BookRepository**.

**package** com.library.service;

**import** com.library.model.Book;

**import** com.library.repository.BookRepository;

**public** **class** BookService

{

**private** BookRepository bookRepository;

**public** **void** setBookRepository(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

}

**public** **void** addBook(Book book) {

bookRepository.saveBook(book);

System.***out***.println("Book Added Successfully");

}

}

1. **Test the Configuration:**
   * Run the **LibraryManagementApplication** main class to verify the dependency injection.

