Exercise 2: Implementing Dependency Injection

Scenario:

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.library</groupId>
   <artifactId>LibraryManagement</artifactId>
   <version>1.0</version>

   <dependencies>
      <!-- Spring Core -->
      <dependency>
         <groupId>org.springframework</groupId>
         <artifactId>spring-context</artifactId>
         <version>5.3.34</version>
      </dependency>
   </dependencies>
</project>
```

2. Configure the Application Context

```
<?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">
```

```
    <!-- Define BookRepository Bean -->
    <bean id="bookRepository" class="com.library.repository.BookRepository"/>

    <!-- Define BookService Bean with dependency injected -->
    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository"/>
    </bean>

</beans>
```

3.**Define Service and Repository Classes**

com/library/repository/BookRepository.java

```
package com.library.repository;

public class BookRepository {
    public void saveBook(String title) {
        System.out.println("BookRepository: Saving book titled \"" + title + "\" to the database.");
    }
}
```

com/library/service/BookService.java

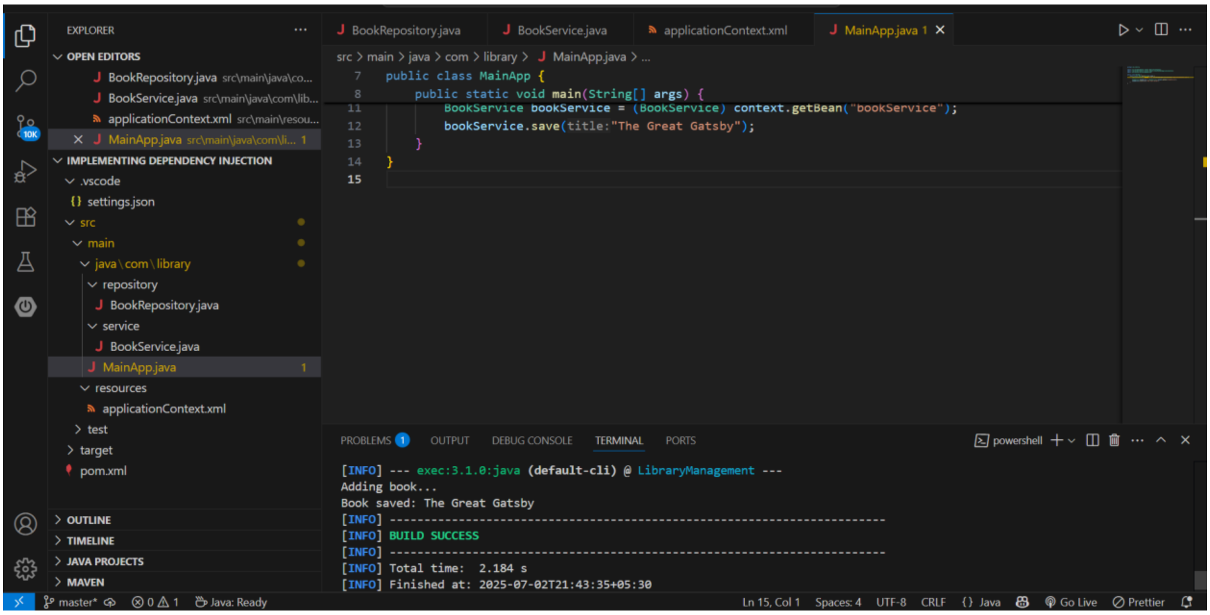
```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    // Setter for Dependency Injection
    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }
    public void save(String bookName) {
        System.out.println("Adding book...");
        bookRepository.save(bookName);
    }
}
```

4. Run the Application

com/library/LibraryApp.java

```
package com.library;

import com.library.service.BookService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryApp {
    public static void main(String[] args) {
         Load Spring context from XML
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

        bookService.addBook("Effective Java");
    }
}
```

Output



```java
public class MainApp {
    public static void main(String[] args) {
        BookService bookService = (BookService) context.getBean("bookService");
        bookService.save(title:"The Great Gatsby");
    }
}

```

```
[INFO] --- exec:3.1.0:java (default-cli) @ LibraryManagement ---
Adding book...
Book saved: The Great Gatsby
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.184 s
[INFO] Finished at: 2025-07-02T21:43:35+05:30
```