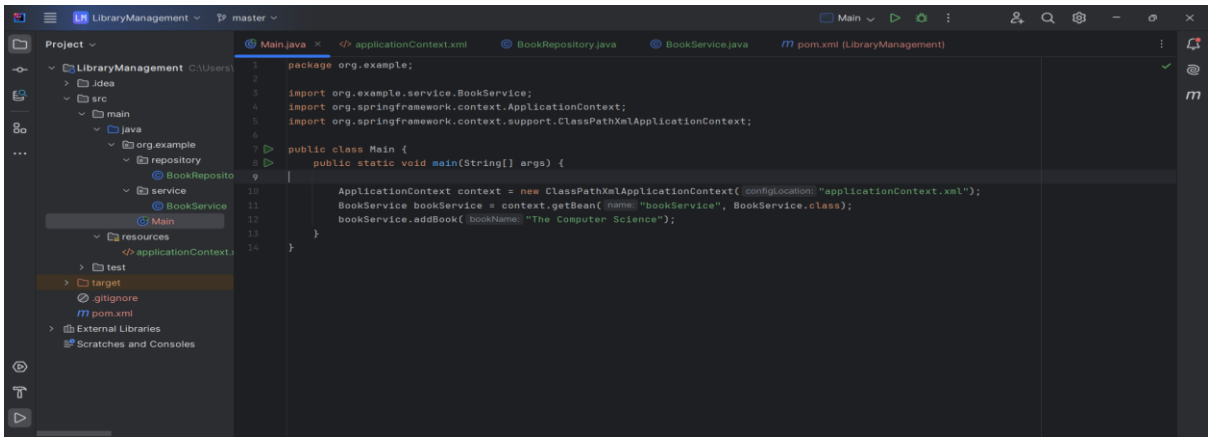
Week - 3

Exercise 1: Configuring a Basic Spring Application

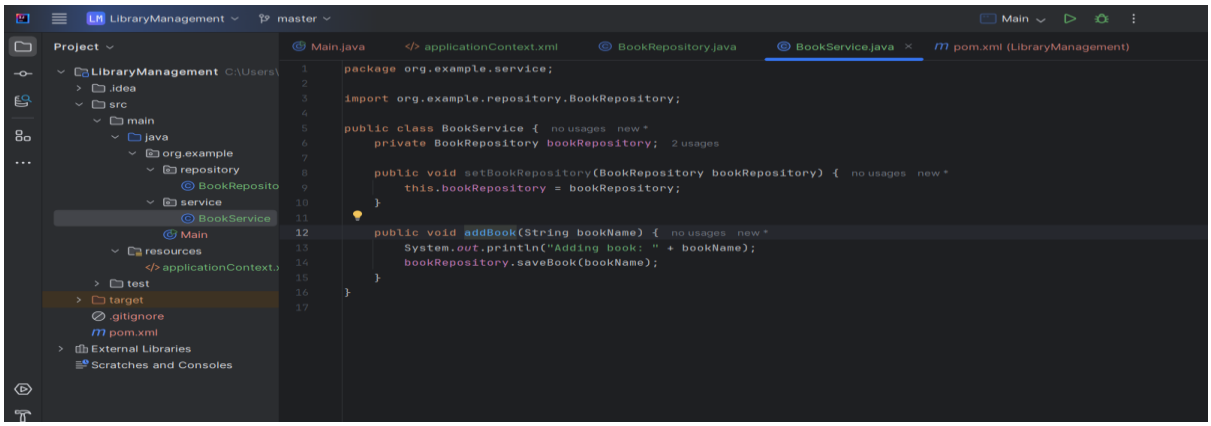
Scenario:

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations

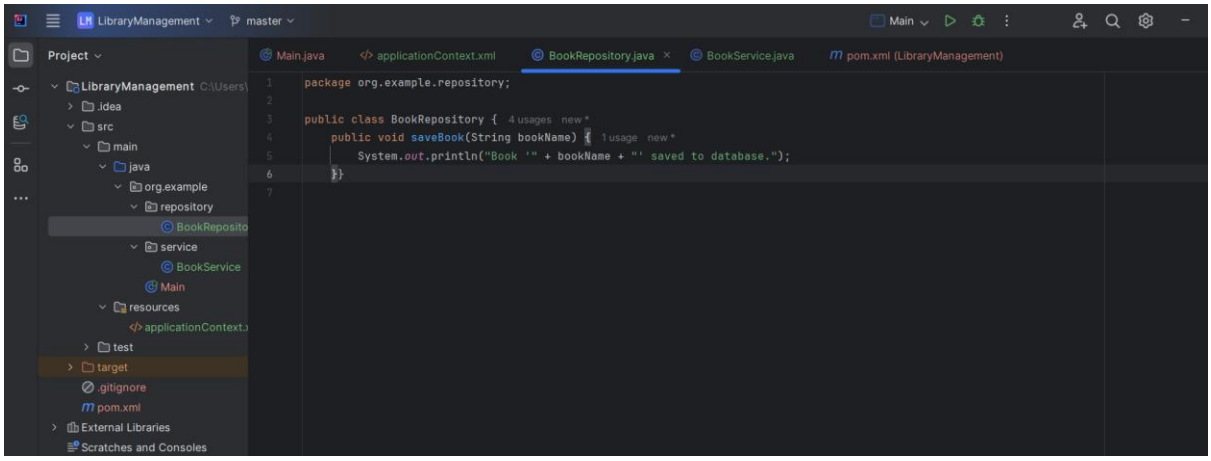
Main.java



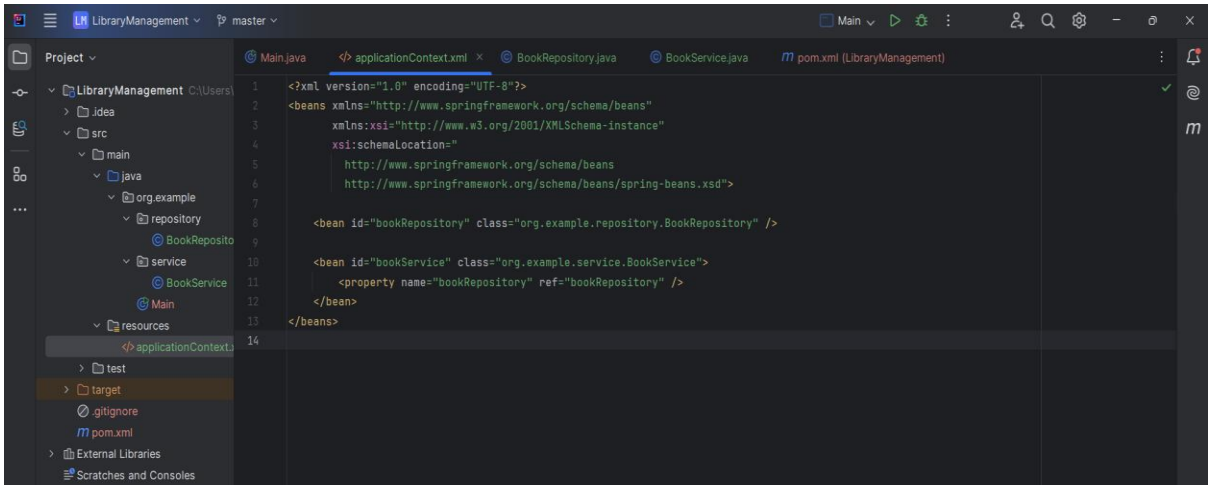
BookService.java



BookRepository.java

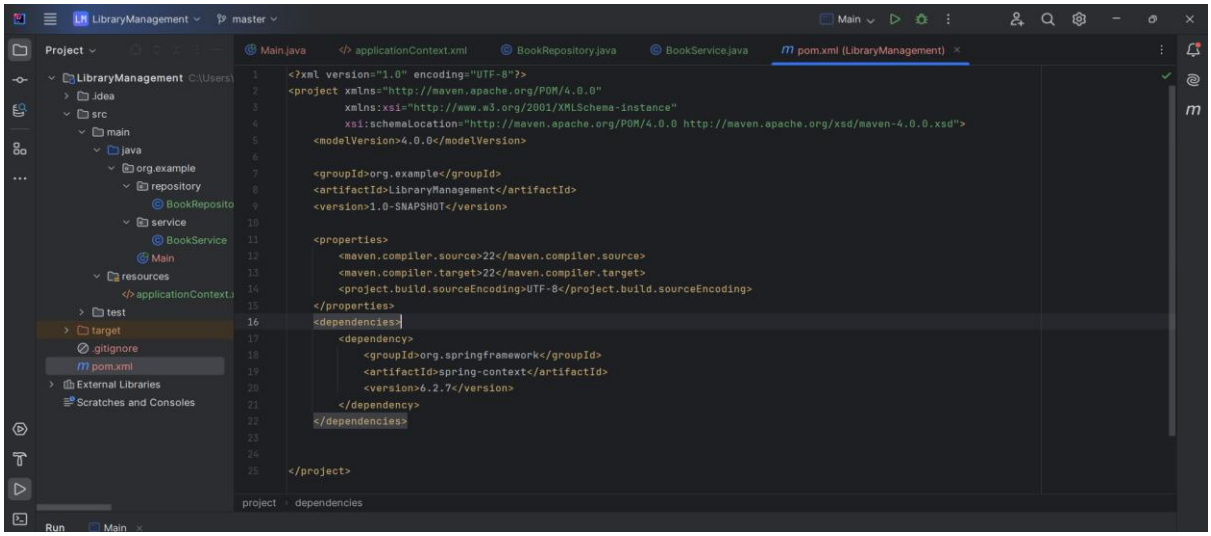


ApplicationContext.xlm



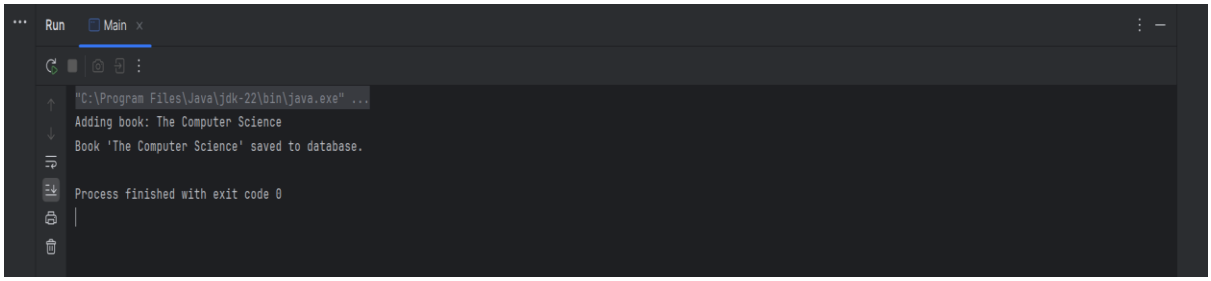
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
         http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd">

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

    <bean id="bookService" class="org.example.service.BookService">
        <property name="bookRepository" ref="bookRepository" />
    </bean>
</beans>
```

Pom.xml



```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>org.example</groupId>
    <artifactId>LibraryManagement</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>22</maven.compiler.source>
        <maven.compiler.target>22</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.7</version>
        </dependency>
    </dependencies>


</project>
```

Output



```
"C:\Program Files\Java\jdk-22\bin\java.exe" ...
Adding book: The Computer Science
Book 'The Computer Science' saved to database.

Process finished with exit code 0
```

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 IoC and DI.

Setter-based Dependency Injection

Main.java

```java
package org.example;

import org.example.service.BookService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext( configLocation: "applicationContext.xml");

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

        bookService.addBook( bookName: "The Compiler Design");
    }
}
```

BookService.java

```java
import org.example.repository.BookRepository;

public class BookService {  4 usages  new *
    private BookRepository bookRepository;  2 usages

    public void setBookRepository(BookRepository bookRepository) {  no usages  new *
        this.bookRepository = bookRepository;
    }

    public void addBook(String bookName) {  1 usage  new *
        System.out.println("Adding book: " + bookName);
        bookRepository.saveBook(bookName);
    }
}
```

BookRepository.java

```java
package org.example.repository;

public class BookRepository {  4 usages  new *
    public void saveBook(String bookName) {  1 usage  new *
        System.out.println("Book '" + bookName + "' saved to database.");
    }}
```

ApplicationContext.xml

```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="org.example.repository.BookRepository" />

    <bean id="bookService" class="org.example.service.BookService">
        <property name="bookRepository" ref="bookRepository" />
    </bean>
</beans>
```

Pom.xlm

```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>org.example</groupId>
    <artifactId>LibraryManagement</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>22</maven.compiler.source>
        <maven.compiler.target>22</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.7</version>
        </dependency>
    </dependencies>


</project>
```

Output

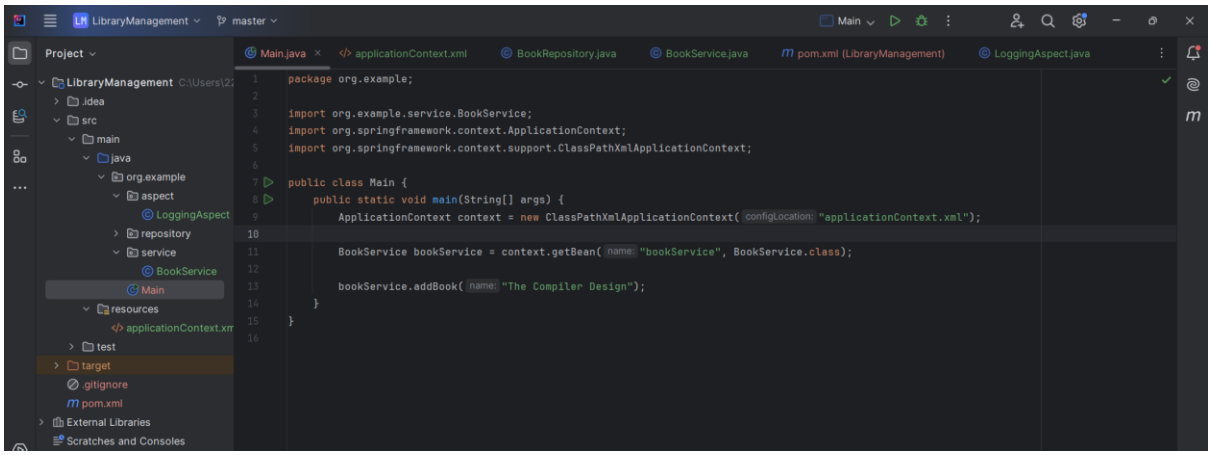
```
"C:\Program Files\Java\jdk-22\bin\java.exe" ...
Setter-based Dependency Injection added
Adding book: The Compiler Design
Book 'The Compiler Design' saved to database.

Process finished with exit code 0
```

Exercise 4: Creating and Configuring a Maven Project

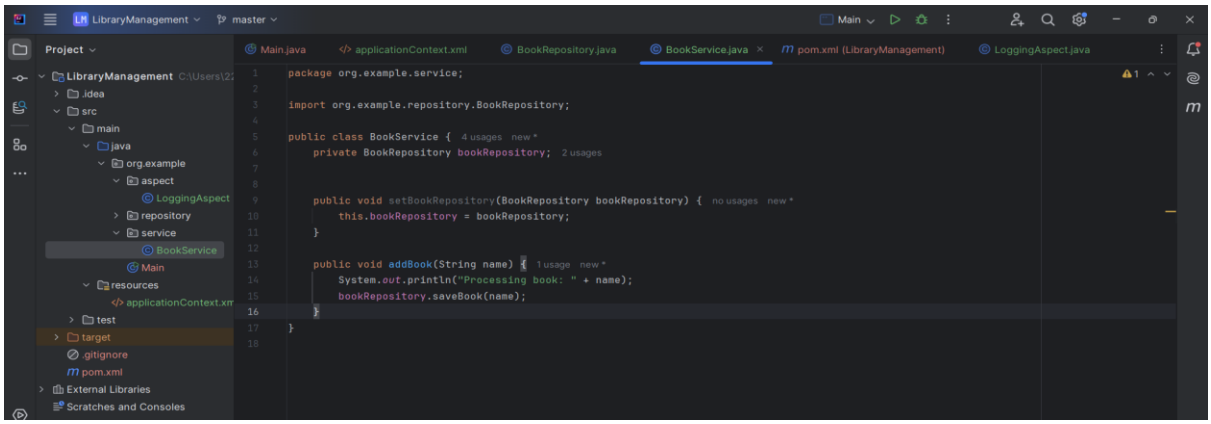
Scenario:

You need to set up a new Maven project for the library management application and add Spring dependencies.

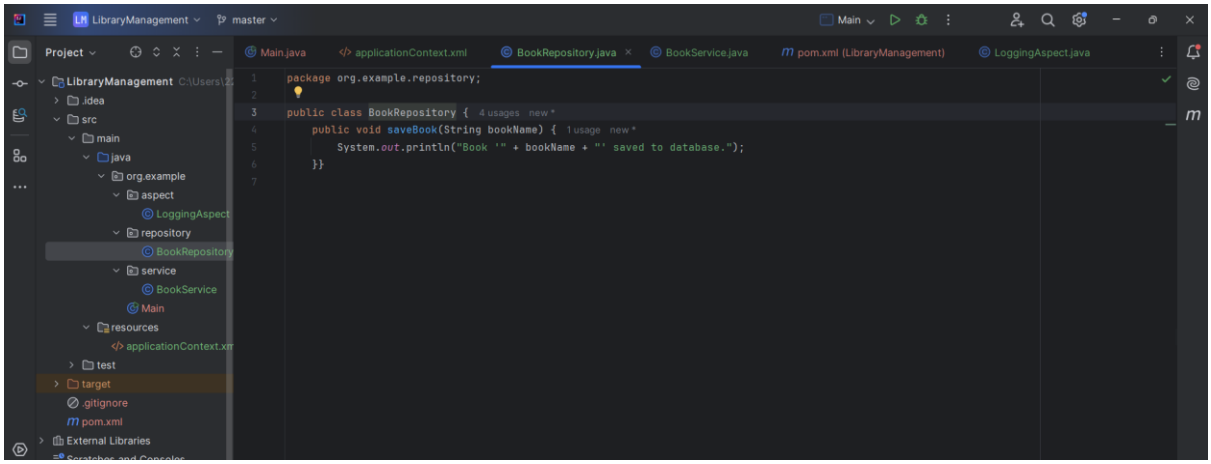
Main.java



BookService.java



BookRepository.java



LoggingAspect.java

```java
package org.example.aspect;


import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
@Aspect 1 usage new *
public class LoggingAspect {
    @Before("execution(* org.example.service.BookService.addBook(..))") no usages new *
    public void log() {
        System.out.println("[AOP] Logging before book addition...");
    }
}
```

ApplicationContext.java

```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"
       xmlns:aop="http://www.springframework.org/schema/aop"
       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="bookRepository" class="org.example.repository.BookRepository" />


    <bean id="bookService" class="org.example.service.BookService">
        <property name="bookRepository" ref="bookRepository" />
    </bean>


    <bean id="loggingAspect" class="org.example.aspect.LoggingAspect" />
</beans>
```

Pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <properties>
    </properties>
    <dependencies>
        <!-- Spring Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.7</version>
        </dependency>

        <!-- Spring AOP -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>5.3.34</version>
        </dependency>

        <!-- Spring WebMVC -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>6.1.14</version>
        </dependency>

        <dependency>
            <groupId>org.aspectj</groupId>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <dependencies>
        </dependency>

        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjweaver</artifactId>
            <version>1.9.21</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <!-- Java 1.8 support -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.10.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

Output

```
"C:\Program Files\Java\jdk-22\bin\java.exe" ...
[AOP] Logging before book addition...
Processing book: The Compiler Design
Book 'The Compiler Design' saved to database.

Process finished with exit code 0
```