WEEK 03

Exercise 1: Configuring a Basic Spring Application

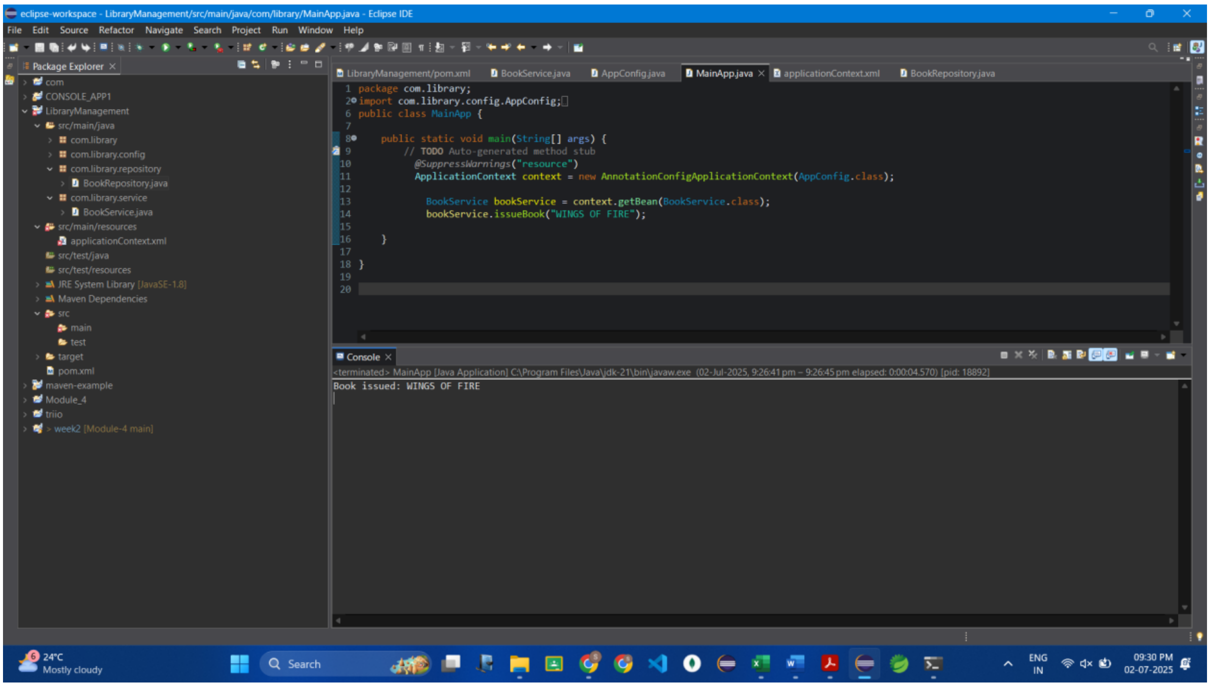
Scenario:

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

Steps:

1. **Set Up a Spring Project:**
   - Create a Maven project named **LibraryManagement**.
   - Add Spring Core dependencies in the **pom.xml** file.
2. **Configure the Application Context:**
   - Create an XML configuration file named **applicationContext.xml** in the **src/main/resources** directory.
   - Define beans for **BookService** and **BookRepository** in the XML file.
3. **Define Service and Repository Classes:**
   - Create a package **com.library.service** and add a class **BookService**.
   - Create a package **com.library.repository** and add a class **BookRepository**.
4. **Run the Application:**
   - Create a main class to load the Spring context and test the configuration.

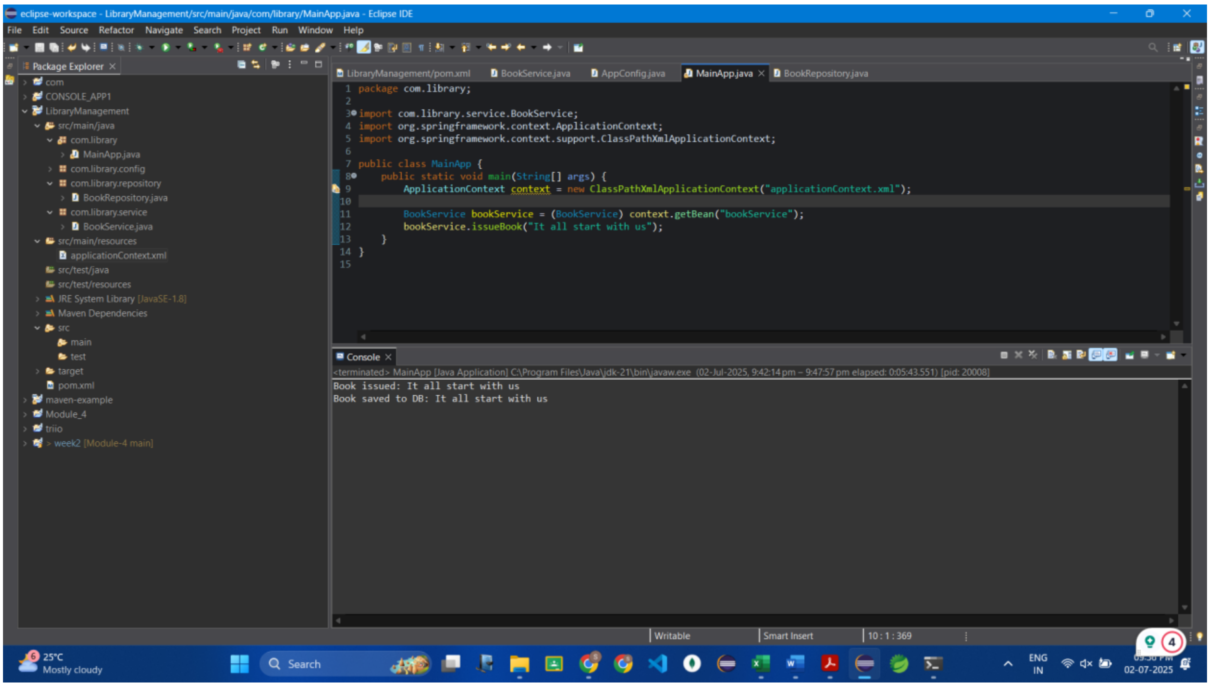
1) Set up a spring project



```java
package com.library;
import com.library.config.AppConfig;
public class MainApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
          @SuppressWarnings("resource")
          ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

            BookService bookService = context.getBean(BookService.class);
            bookService.issueBook("WINGS OF FIRE");

    }

}
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (02-Jul-2025, 9:26:41 pm – 9:26:45 pm elapsed: 0:00:04.570) [pid: 18892]
Book issued: WINGS OF FIRE
```

2) Configure the Application Context:



eclipse-workspace - LibraryManagement/src/main/java/com/library/MainApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LibraryManagement/pom.xml
BookService.java
AppConfig.java
MainApp.java
BookRepository.java
Console

```java
package com.library;

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

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

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.issueBook("It all start with us");
    }
}
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (02-Jul-2025, 9:42:14 pm – 9:47:57 pm elapsed: 0:05:43.551) [pid: 20008]
Book issued: It all start with us
Book saved to DB: It all start with us
```

Exercise 2: Implementing Dependency Injection

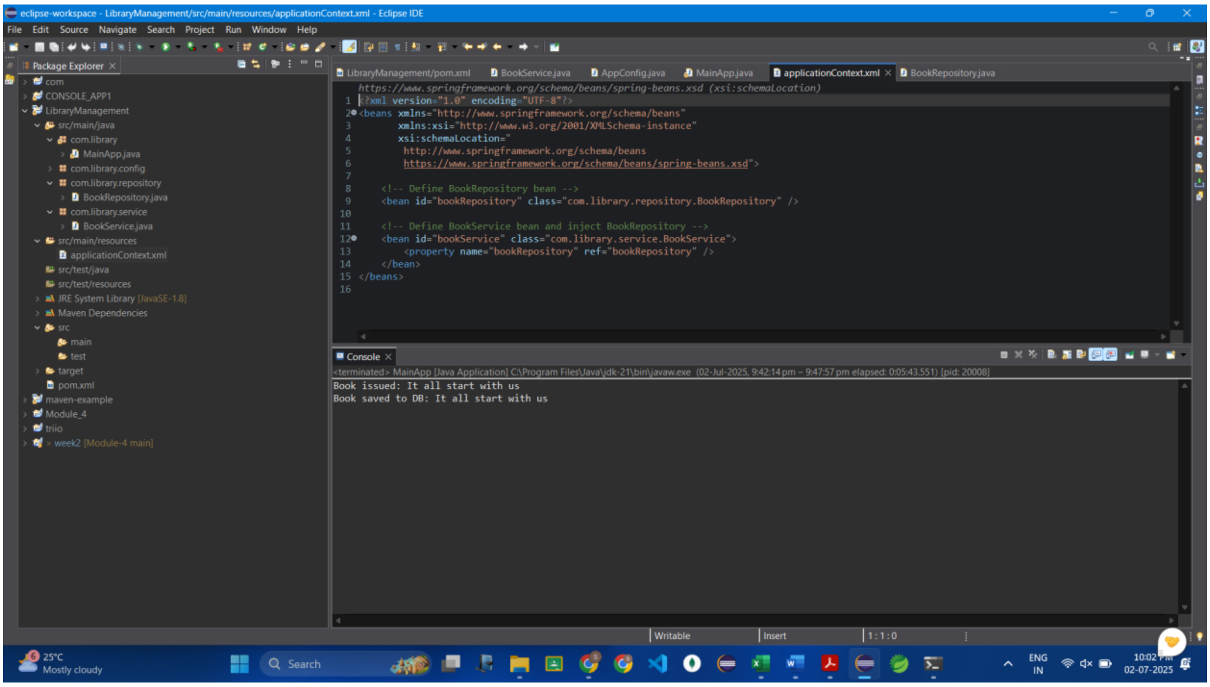
Scenario:

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

Steps:

1. **Modify the XML Configuration:**
   - Update **applicationContext.xml** to wire **BookRepository** into **BookService**.
2. **Update the BookService Class:**
   - Ensure that **BookService** class has a setter method for **BookRepository**.
3. **Test the Configuration:**
   - Run the **LibraryManagementApplication** main class to verify the dependency injection.

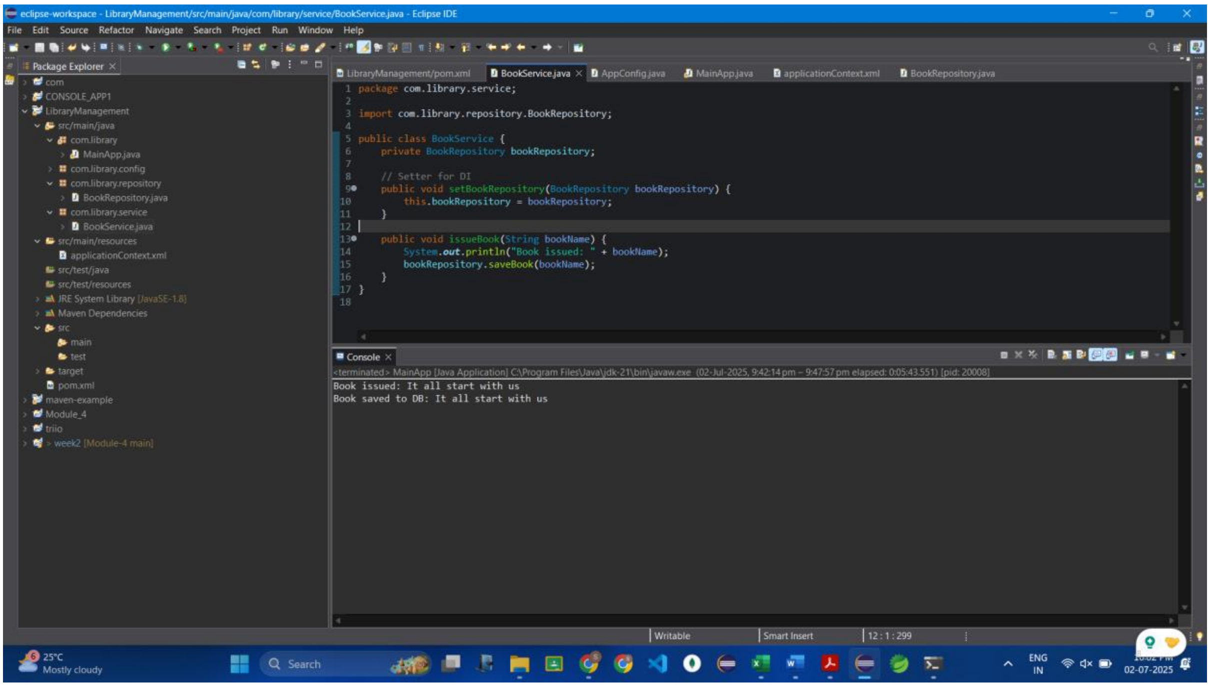
XML CONFIGURATION:


eclipse-workspace - LibraryManagement/src/main/resources/applicationContext.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
LibraryManagement/pom.xml
BookService.java
AppConfig.java
MainApp.java
applicationContext.xml
BookRepository.java
Console
Writable
Insert
1 : 1 : 0

```
https://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?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
        https://www.springframework.org/schema/beans/spring-beans.xsd">

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

    <!-- Define BookService bean and inject BookRepository -->
    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository" />
    </bean>
</beans>
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (02-Jul-2025, 9:42:14 pm – 9:47:57 pm elapsed: 0:05:43.551) [pid: 20008]
Book issued: It all start with us
Book saved to DB: It all start with us
```

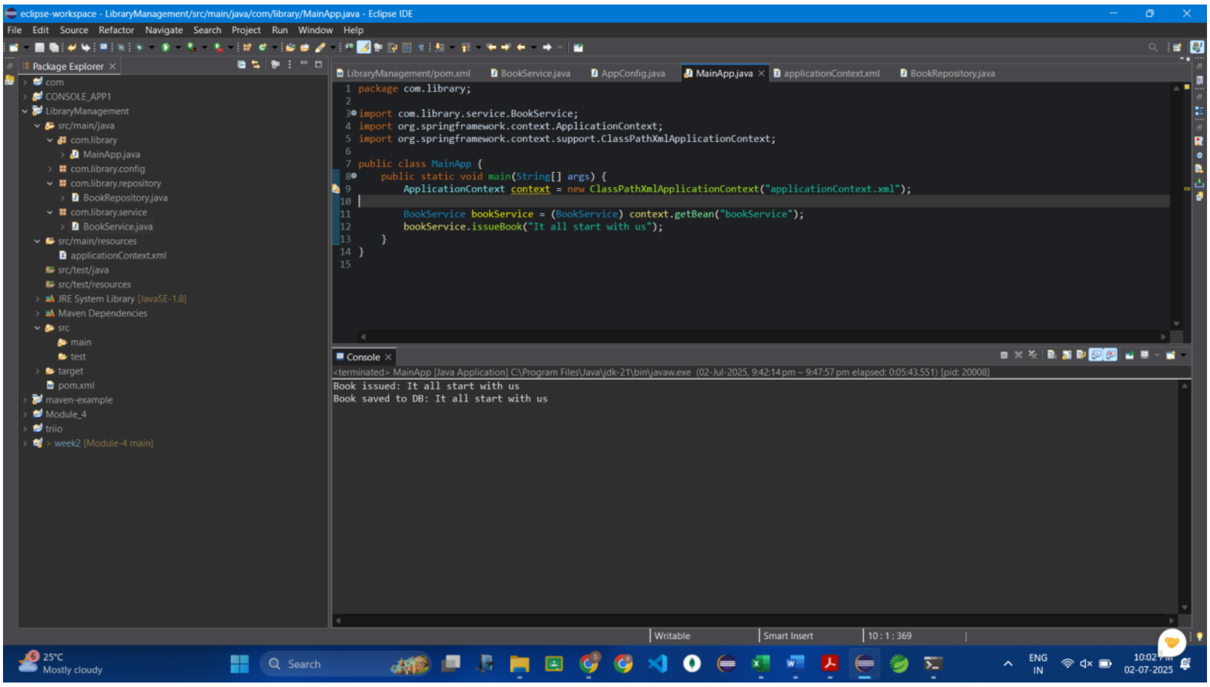
BOOKSERVICE.JAVA


eclipse-workspace - LibraryManagement/src/main/java/com/library/service/BookService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
LibraryManagement/pom.xml
BookService.java
AppConfig.java
MainApp.java
applicationContext.xml
BookRepository.java
Console
Writable
Smart Insert
12 : 1 : 299

```java
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    // Setter for DI
    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void issueBook(String bookName) {
        System.out.println("Book issued: " + bookName);
        bookRepository.saveBook(bookName);
    }
}
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (02-Jul-2025, 9:42:14 pm – 9:47:57 pm elapsed: 0:05:43.551) [pid: 20008]
Book issued: It all start with us
Book saved to DB: It all start with us
```

MAINAPP.JAVA



```java
package com.library;

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

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

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.issueBook("It all start with us");
    }
}

```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (02-Jul-2025, 9:42:14 pm – 9:47:57 pm elapsed: 0:05:43.551) [pid: 20008]
Book issued: It all start with us
Book saved to DB: It all start with us
```

Exercise 4: Creating and Configuring a Maven Project

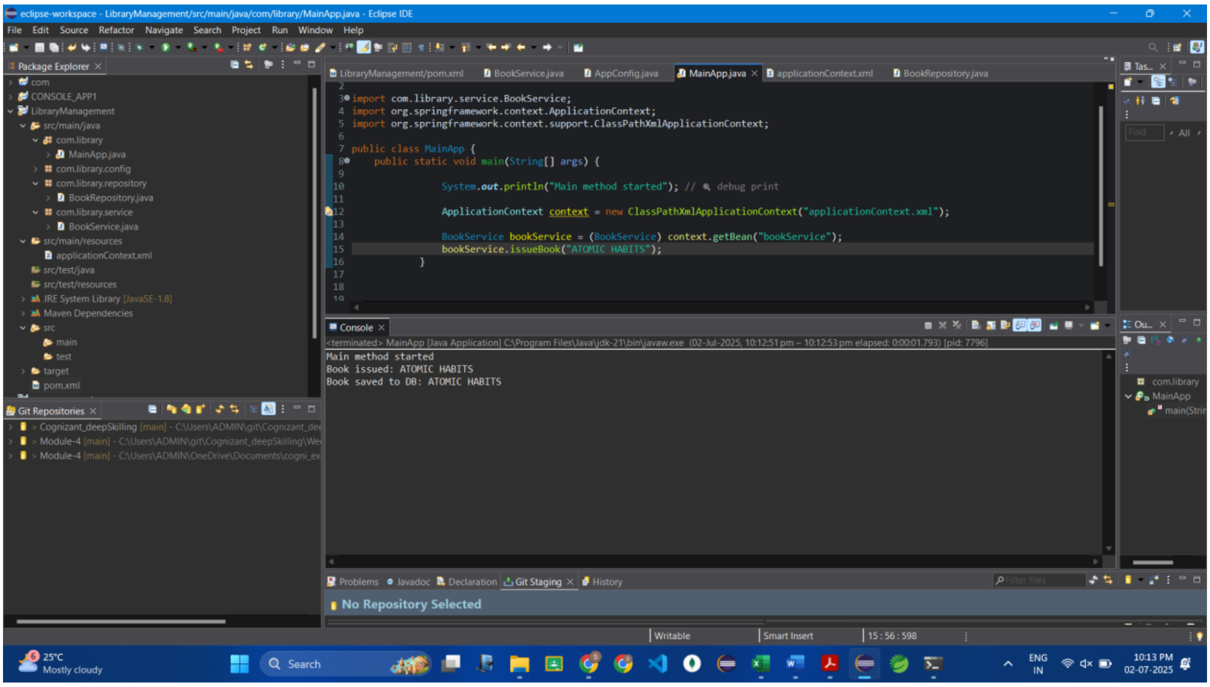
Scenario:

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

Steps:

1. **Create a New Maven Project:**
   - Create a new Maven project named **LibraryManagement**.
2. **Add Spring Dependencies in pom.xml:**
   - Include dependencies for Spring Context, Spring AOP, and Spring WebMVC.
3. **Configure Maven Plugins:**
   - Configure the Maven Compiler Plugin for Java version 1.8 in the pom.xml file.

MAINAPP.JAVA

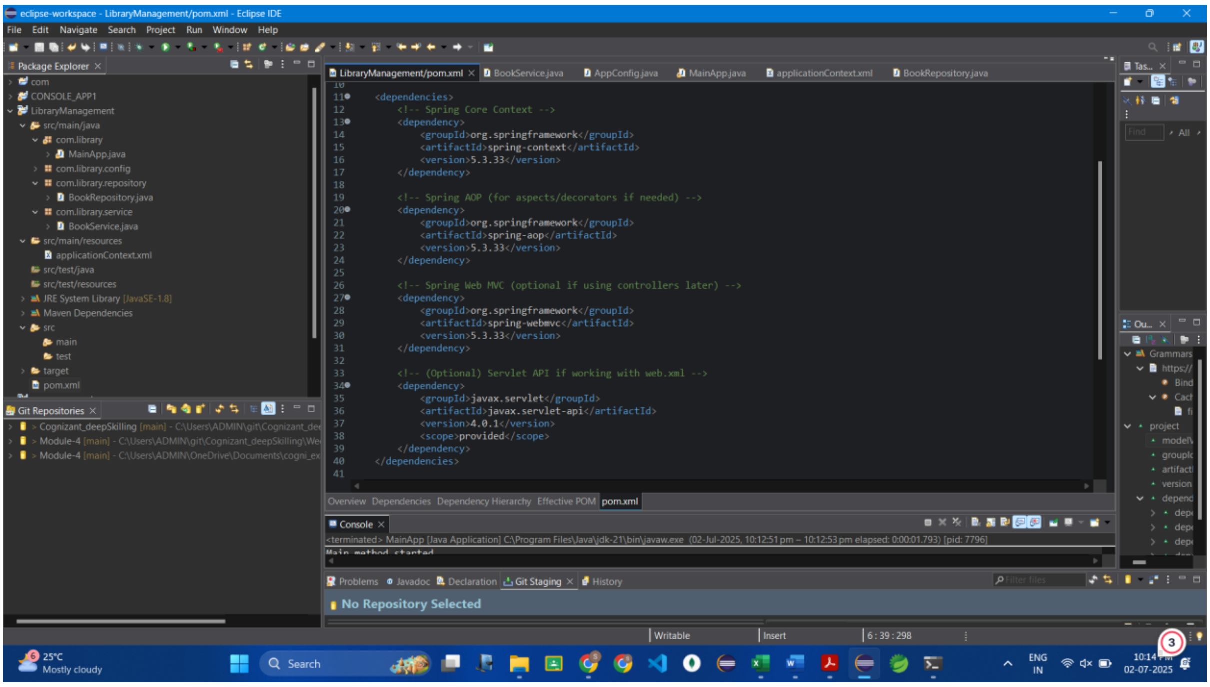


eclipse-workspace - LibraryManagement/src/main/java/com/library/MainApp.java - Eclipse IDE

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

public class MainApp {
    public static void main(String[] args) {

                System.out.println("Main method started"); // 🔍 debug print

                ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

                BookService bookService = (BookService) context.getBean("bookService");
                bookService.issueBook("ATOMIC HABITS");
            }
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (02-Jul-2025, 10:12:51 pm – 10:12:53 pm elapsed: 0:00:01.793) [pid: 7796]
Main method started
Book issued: ATOMIC HABITS
Book saved to DB: ATOMIC HABITS
```

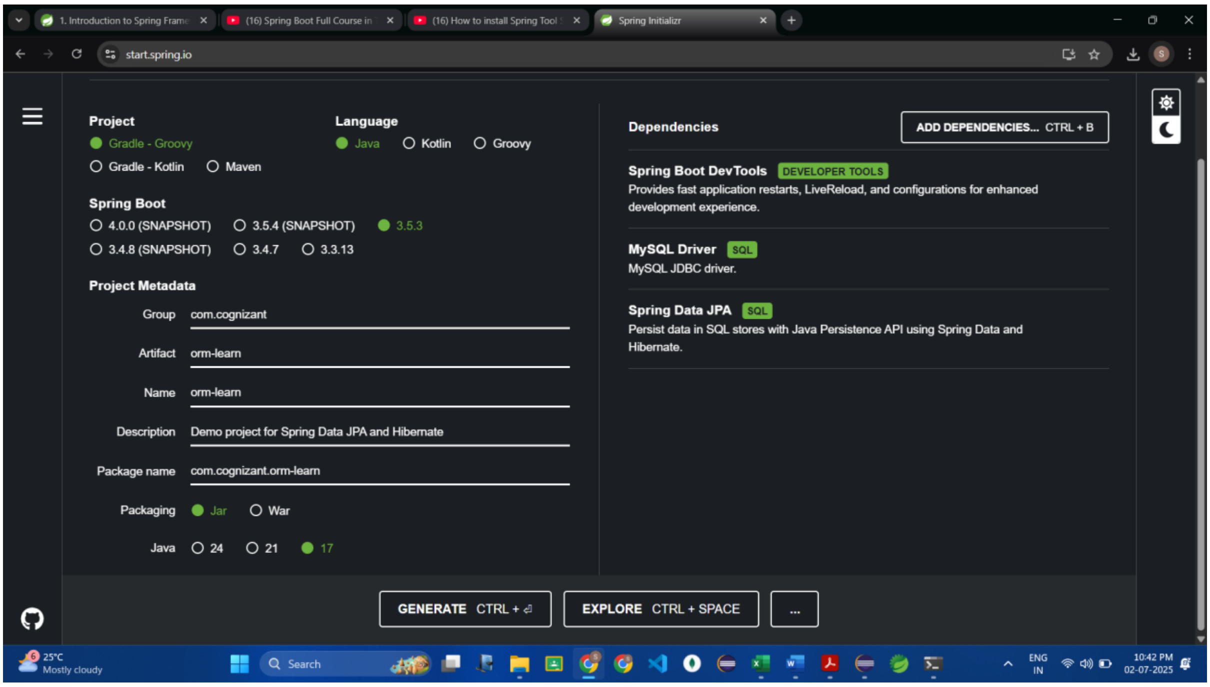
POM.XML



```xml
    <dependencies>
        <!-- Spring Core Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.33</version>
        </dependency>

        <!-- Spring AOP (for aspects/decorators if needed) -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>5.3.33</version>
        </dependency>

        <!-- Spring Web MVC (optional if using controllers later) -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>5.3.33</version>
        </dependency>

        <!-- (Optional) Servlet API if working with web.xml -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>4.0.1</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
```

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (02-Jul-2025, 10:12:51 pm – 10:12:53 pm elapsed: 0:00:01.793) [pid: 7796]
Main method started
```

Spring Data JPA - Quick Example



1. Introduction to Spring Frame
(16) Spring Boot Full Course in
(16) How to install Spring Tool
Spring Initializr
start.spring.io

Project
● Gradle - Groovy ○ Gradle - Kotlin ○ Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 4.0.0 (SNAPSHOT) ○ 3.5.4 (SNAPSHOT) ● 3.5.3
○ 3.4.8 (SNAPSHOT) ○ 3.4.7 ○ 3.3.13

Project Metadata

Group: com.cognizant

Artifact: orm-learn

Name: orm-learn

Description: Demo project for Spring Data JPA and Hibernate

Package name: com.cognizant.orm-learn

Packaging: ● Jar ○ War

Java: ○ 24 ○ 21 ● 17

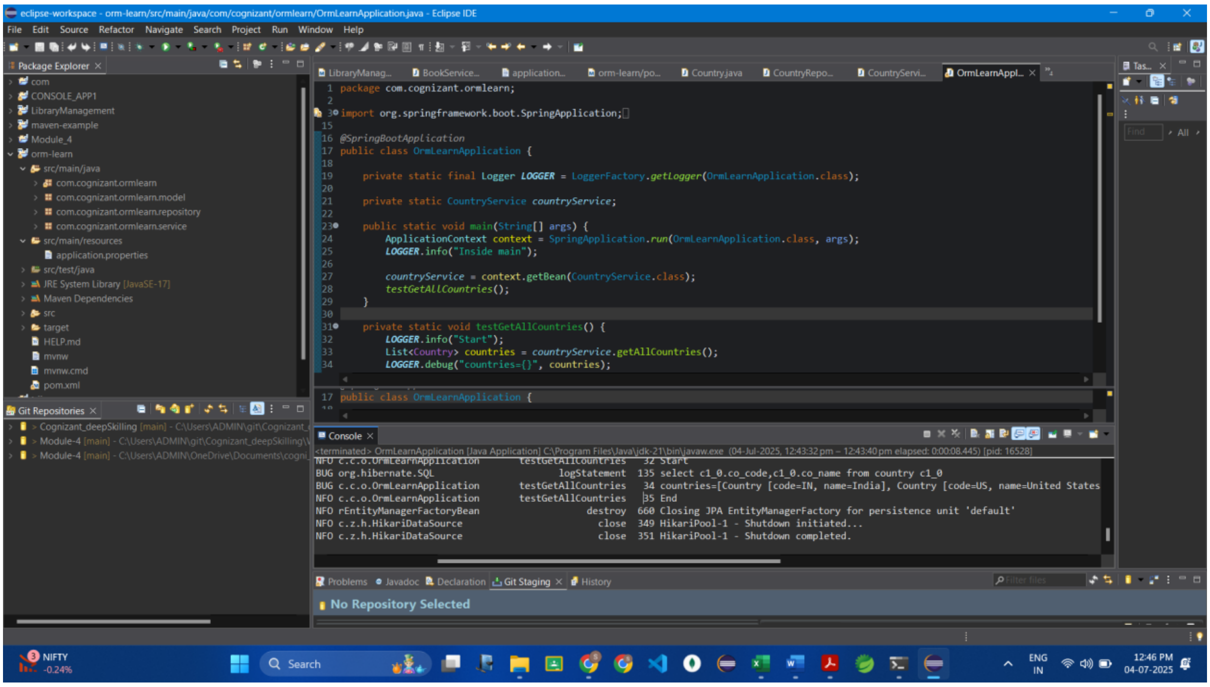
Dependencies

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

MySQL Driver SQL
MySQL JDBC driver.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

ADD DEPENDENCIES... CTRL + B
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
...
25°C
Mostly cloudy
Search
ENG
IN
10:42 PM
02-07-2025



eclipse-workspace - orm-learn/src/main/java/com/cognizant/ormlearn/OrmLearnApplication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
com
CONSOLE_APP1
LibraryManagement
maven-example
Module_4
orm-learn
src/main/java
com.cognizant.ormlearn
com.cognizant.ormlearn.model
com.cognizant.ormlearn.repository
com.cognizant.ormlearn.service
src/main/resources
application.properties
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Git Repositories
Cognizant_deepSkilling [main] - C:\Users\ADMIN\git\Cognizant_
Module-4 [main] - C:\Users\ADMIN\git\Cognizant_deepSkilling\
Module-4 [main] - C:\Users\ADMIN\OneDrive\Documents\cogni
LibraryManag...
BookService...
application...
orm-learn/po...
Country.java
CountryRepo...
CountryServi...
OrmLearnAppl...

```java
package com.cognizant.ormlearn;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class OrmLearnApplication {

    private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

    private static CountryService countryService;

    public static void main(String[] args) {
        ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);
        LOGGER.info("Inside main");

        countryService = context.getBean(CountryService.class);
        testGetAllCountries();
    }

    private static void testGetAllCountries() {
        LOGGER.info("Start");
        List<Country> countries = countryService.getAllCountries();
        LOGGER.debug("countries={}", countries);
```

```java
public class OrmLearnApplication {
```

Console

```
<terminated> OrmLearnApplication [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (04-Jul-2025, 12:43:32 pm – 12:43:40 pm elapsed: 0:00:08.445) [pid: 16528]
NFO c.c.o.OrmLearnApplication      testGetAllCountries   32 Start
BUG org.hibernate.SQL                      logStatement  135 select c1_0.co_code,c1_0.co_name from country c1_0
BUG c.c.o.OrmLearnApplication      testGetAllCountries   34 countries=[Country [code=IN, name=India], Country [code=US, name=United States
NFO c.c.o.OrmLearnApplication      testGetAllCountries   35 End
NFO rEntityManagerFactoryBean                   destroy  660 Closing JPA EntityManagerFactory for persistence unit 'default'
NFO c.z.h.HikariDataSource                        close  349 HikariPool-1 - Shutdown initiated...
NFO c.z.h.HikariDataSource                        close  351 HikariPool-1 - Shutdown completed.
```