**MODULE -- ALGORITHMS\_DATA STRUCTURES**

**Task 1 : Configuring a Basic Spring Application**

**Code :**

**BookRepository.java**

package com.example.demo.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println("Fetching books from the repository...");

}

}

**BookService.java**

package com.example.demo.service;

import com.example.demo.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void listBooks() {

System.out.println("BookService: listing books...");

bookRepository.fetchBooks();

}

}

**BasicApplication.java**

package com.example.demo;

import com.example.demo.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class BasicApplication {

public static void main(String[] args) {

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

// Get BookService bean

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

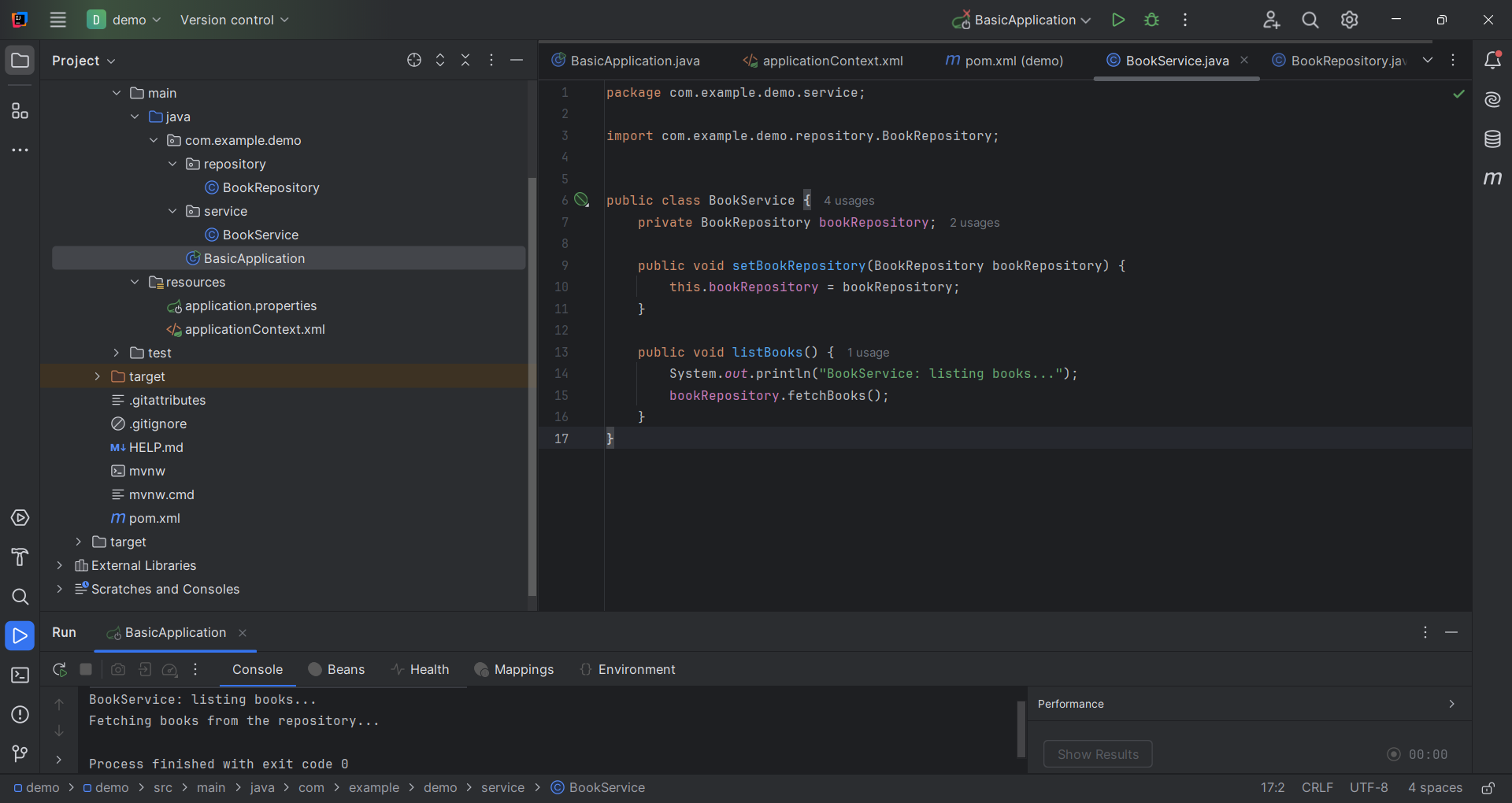
// Call method

bookService.listBooks();

}

}

**Output :**

****

**Task 2 : Implementing Dependency Injection**

**Code:**

**BookRepository.java**

package com.example.demo.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println("BookRepository: Fetching books from the repository...");

System.out.println("BookRepository: Connecting to database...");

System.out.println("BookRepository: Query executed successfully.");

}

}

**BookService.java**

package com.example.demo.service;

import com.example.demo.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

System.out.println("BookService: BookRepository has been set.");

}

public void listBooks() {

System.out.println("BookService: Preparing to list books...");

bookRepository.fetchBooks();

System.out.println("BookService: Book listing complete.");

}

}

**BasicApplication.java**

package com.example.demo;

import com.example.demo.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class BasicApplication {

public static void main(String[] args) {

ApplicationContext context = newClassPathXmlApplicationContext("applicationContext.xml");

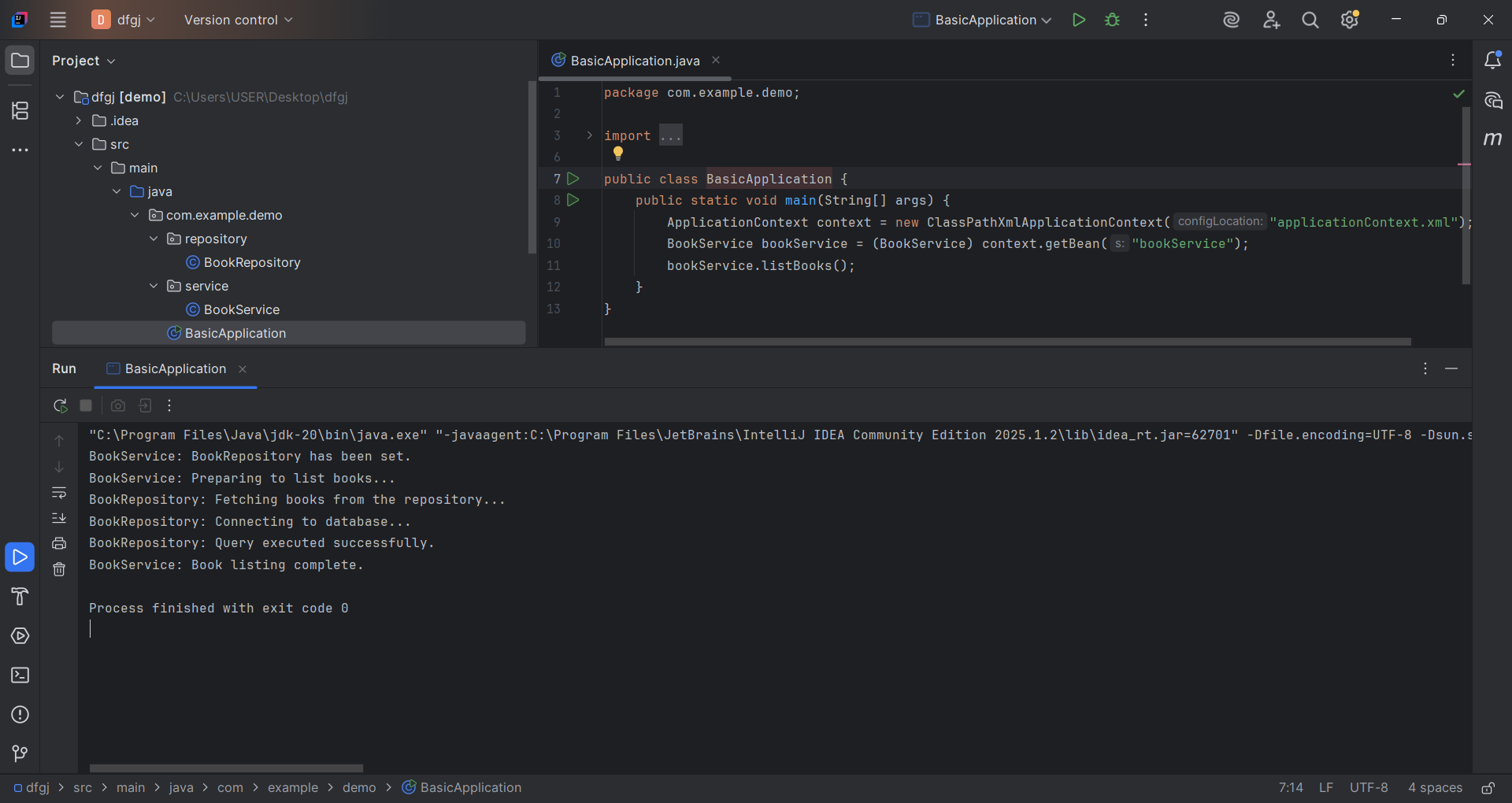
BookService bookService = (BookService) context.getBean("bookService");

bookService.listBooks();

}

}

**Output:**

****

**Task 3 : Creating and Configuring a Maven Project**

**pom.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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>config</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>config</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>24</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- Spring Core/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>

<!-- Servlet API (for Spring MVC) -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<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>

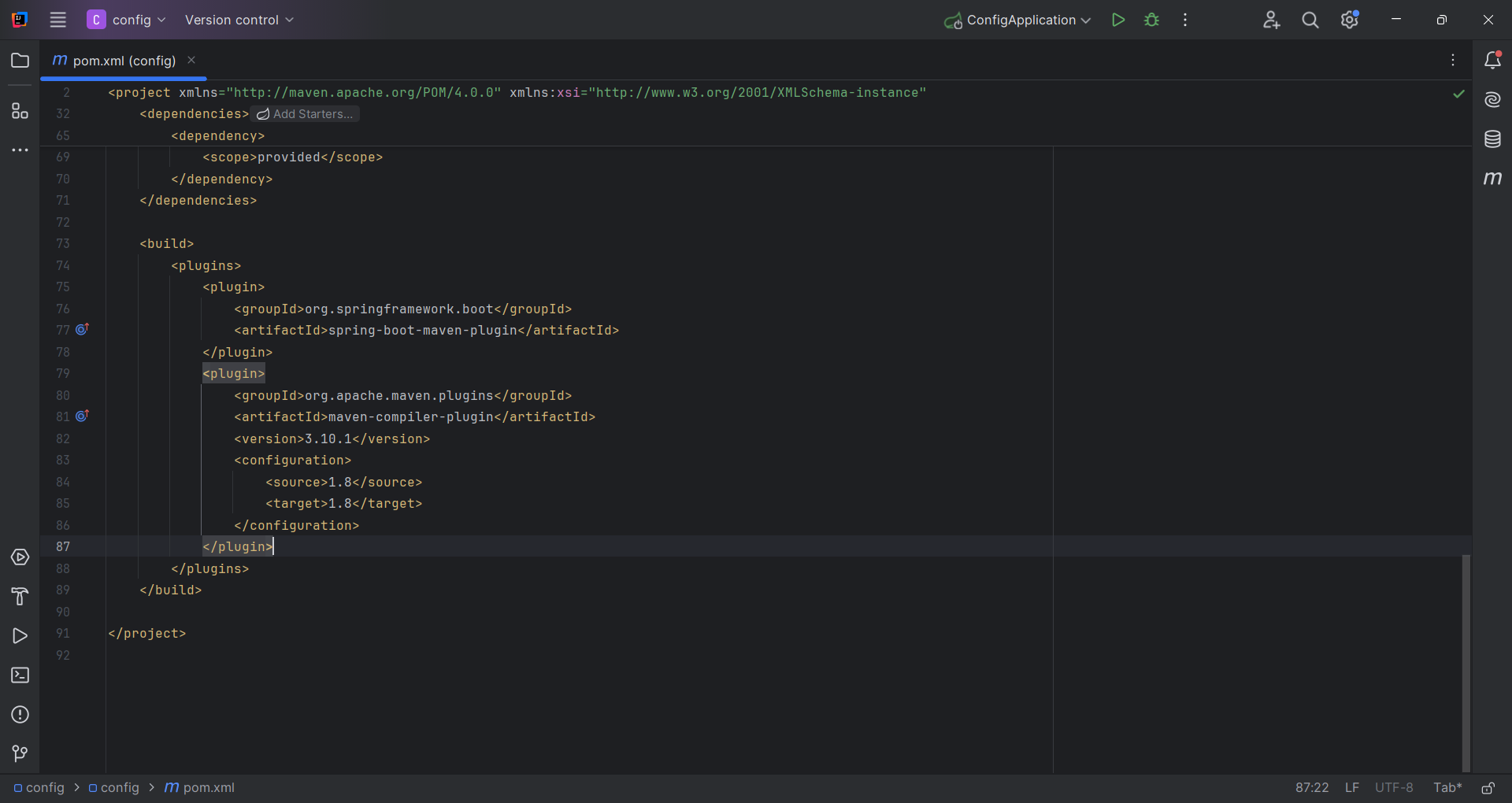
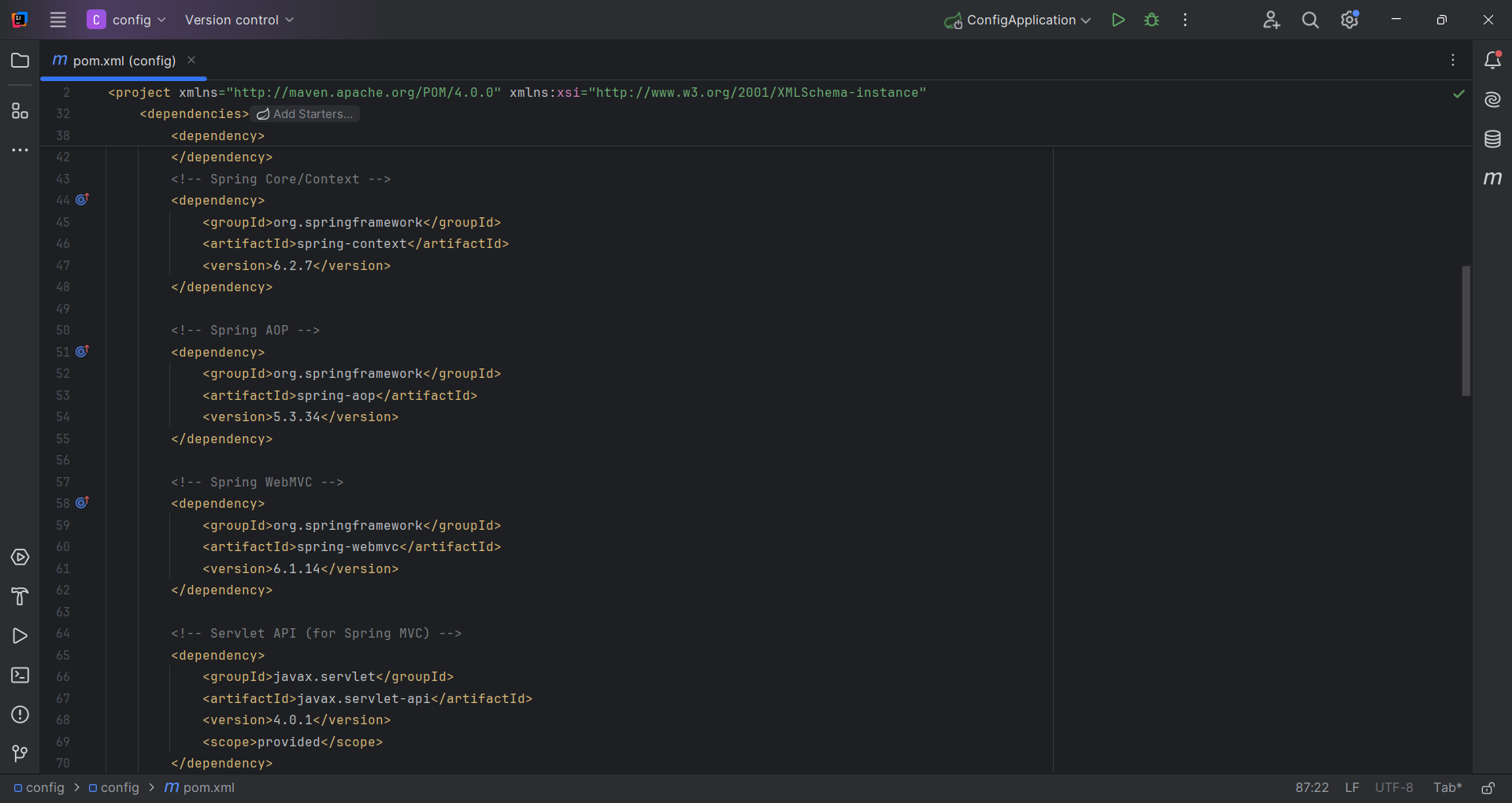
</plugin>

</plugins>

</build>

</project>

**Output:**

****