**WEEK -3**

**ADDITIONAL HANDS ON**

**Exercise 5: Configuring the Spring IoC Container**

**Scenario:**

The library management application requires a central configuration for beans and dependencies.

Steps:

1. Create Spring Configuration File:

o Create an XML configuration file named applicationContext.xml in the src/main/resources directory.

o Define beans for BookService and BookRepository in the XML file.

2. Update the BookService Class:

o Ensure that the BookService class has a setter method for BookRepository.

3. Run the Application:

o Create a main class to load the Spring context and test the configuration.

**CODE:**

**Main.java**

package com.library.main;

import com.library.service.BookService;

import com.library.service.MemberService;

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

MemberService memberService = context.getBean("memberService", MemberService.class);

bookService.addBook("The Great Gatsby");

memberService.addMember("Keerthana Jothi");

}

}

**ApplicationContext.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="com.library.repository.BookRepository" />

<bean id="memberRepository" class="com.library.repository.MemberRepository" />

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

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

</bean>

<bean id="memberService" class="com.library.service.MemberService">

<property name="memberRepository" ref="memberRepository" />

</bean>

</beans>

**MemberRepository.java**

package com.library.repository;

public class MemberRepository {

public void saveMember(String name) {

System.out.println("Saving member: " + name);

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook(String title) {

System.out.println("Saving book: " + title);

}

}

**MemberService.java**

package com.library.service;

import com.library.repository.MemberRepository;

public class MemberService {

private MemberRepository memberRepository;

public void setMemberRepository(MemberRepository memberRepository) {

this.memberRepository = memberRepository;

}

public void addMember(String name) {

System.out.println("Adding member: " + name);

memberRepository.saveMember(name);

}

}

**BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

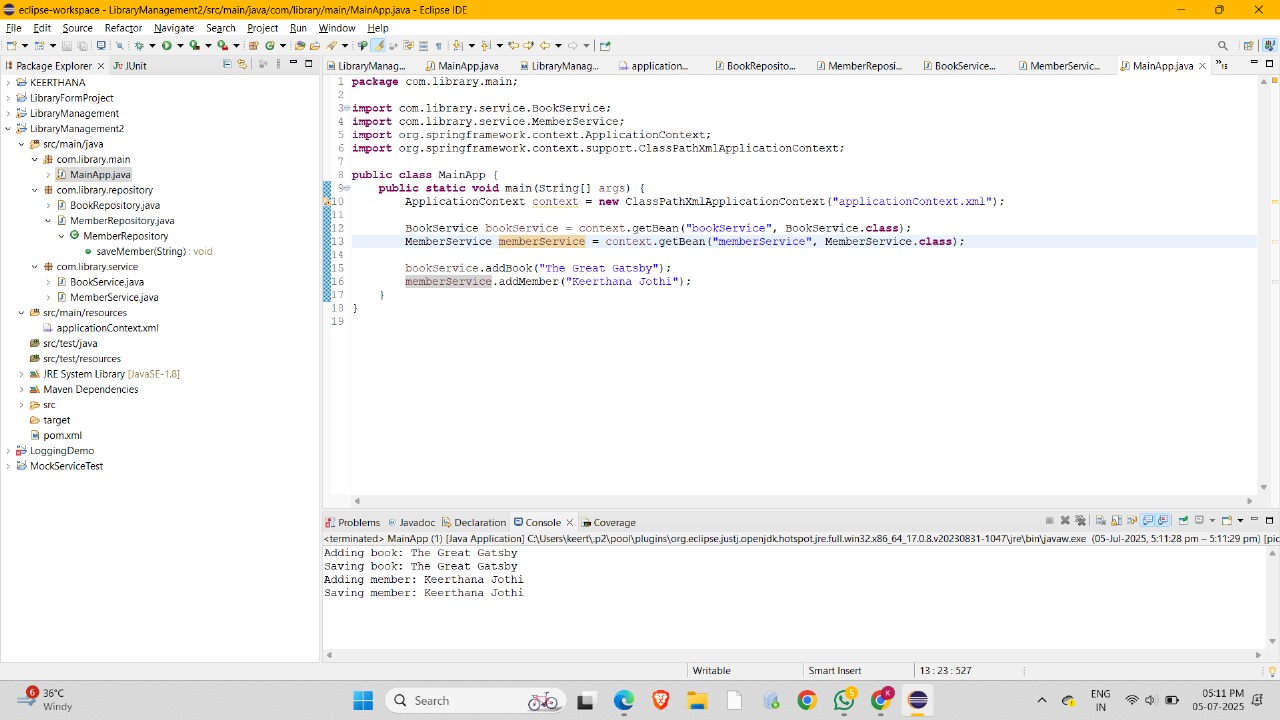
System.out.println("Adding book: " + title);

bookRepository.saveBook(title);

}

}

**OUPUT:**



**Exercise 7: Implementing Constructor and Setter Injection**

**Scenario:**

The library management application requires both constructor and setter injection for better control over bean initialization.

Steps:

1. Configure Constructor Injection:

o Update applicationContext.xml to configure constructor injection for BookService.

2. Configure Setter Injection:

o Ensure that the BookService class has a setter method for BookRepository and configure it in applicationContext.xml.

3. Test the Injection:

o Run the LibraryManagementApplication main class to verify both constructor and setter injection.

**CODE:**

**Main.java**

package com.library.main;

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

bookService.addBook("The Great Gatsby");

}

}

**BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Constructor Injection

public BookService(BookRepository bookRepository) {

System.out.println("Constructor injection called");

this.bookRepository = bookRepository;

}

// Setter Injection

public void setBookRepository(BookRepository bookRepository) {

System.out.println("Setter injection called");

this.bookRepository = bookRepository;

}

public void addBook(String title) {

System.out.println("Adding book: " + title);

bookRepository.save(title);

}

}

**ApplicationContext.xml**

<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">

<!-- Repository Bean -->

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

<!-- BookService Bean with constructor + setter injection -->

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

<!-- Constructor injection -->

<constructor-arg ref="bookRepository" />

<!-- Setter injection -->

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

</bean>

</beans>

**OUTPUT:**

