Digital Nurture 4.0 – Deep Skilling Program

Java Full Stack Engineer (FSE)

Week 3 – Mandatory Exercises Submission

Name: Khushi

GitHub Profile: https://github.com/KhushiSGowda

Email:khushisnagraj12@gmail.com

Skills Covered:

Spring core maven

Spring Core – Exercises 1, 2 and 4

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

Objective: Set up a basic Spring application using XML-based configuration and demonstrate bean creation and usage.

Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" ...>  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.29</version>  
 </dependency>  
 </dependencies>  
</project>

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">  
  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

BookRepository.java:

package com.library.repository;  
  
public class BookRepository {  
 public String getBookDetails() {  
 return "Spring in Action";  
 }  
}

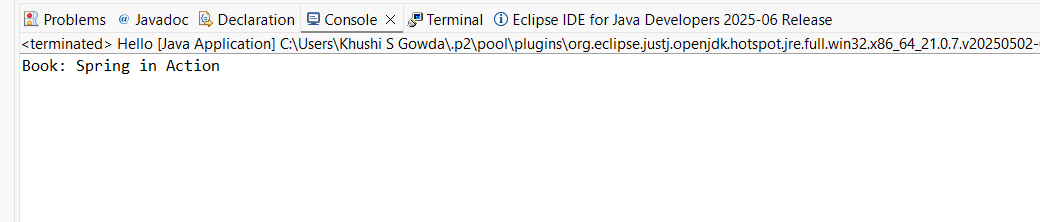
BookService.java:

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 displayBook() {  
 System.out.println("Book: " + bookRepository.getBookDetails());  
 }  
}

LibraryManagementApplication.java:

package com.library;  
  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryManagementApplication {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.displayBook();  
 }  
}

**Output:**

**Exercise 2: Implementing Dependency Injection**

Objective: Use Spring's IoC container to inject dependencies. This exercise reuses the setup from Exercise 1 using setter-based injection in 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">  
  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

applicationContext.xml:

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 displayBook() {  
 System.out.println("Book: " + bookRepository.getBookDetails());  
 }  
}

BookService.java:

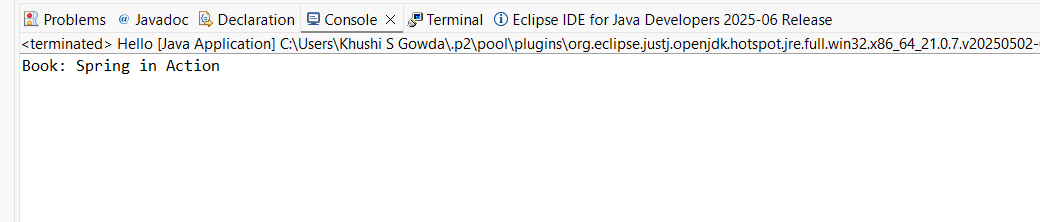
package com.library.repository;  
  
public class BookRepository {  
 public String getBookDetails() {  
 return "Spring in Action";  
 }  
}

BookRepository.java:

Implementation:

Already implemented in Exercise 1 using the setter method in BookService and <property> tag in XML.

**Output:**

****

**Exercise 4: Creating and Configuring a Maven Project**

Objective: Create a Maven project and configure Spring dependencies and compiler plugin.

pom.xml:

<project xmlns="http://maven.apache.org/POM/4.0.0" ...>  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.29</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>5.3.29</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.29</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

Output:

