**WEEK 3 MANDATORY HANDS-ON**

**SPRING CORE WITH MAVEN**

Exercise 1: Configuring a Basic Spring Application

CODE:

**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 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>24</maven.compiler.source>  
 <maven.compiler.target>24</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.31</version>  
 </dependency>  
 </dependencies>  
  
</project>

**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  
 https://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>

**LibraryApp.java:**

package com.library;  
  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryApp {  
 public static void main(String[] args) {  
 // Load the Spring application context from classpath  
 ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 // Retrieve the bookService bean  
 BookService bookService = context.getBean("bookService", BookService.class);  
  
 // Call method to add book  
 bookService.addBook("The Spring Framework Guide");  
  
 // Close the context  
 context.close();  
 }  
}

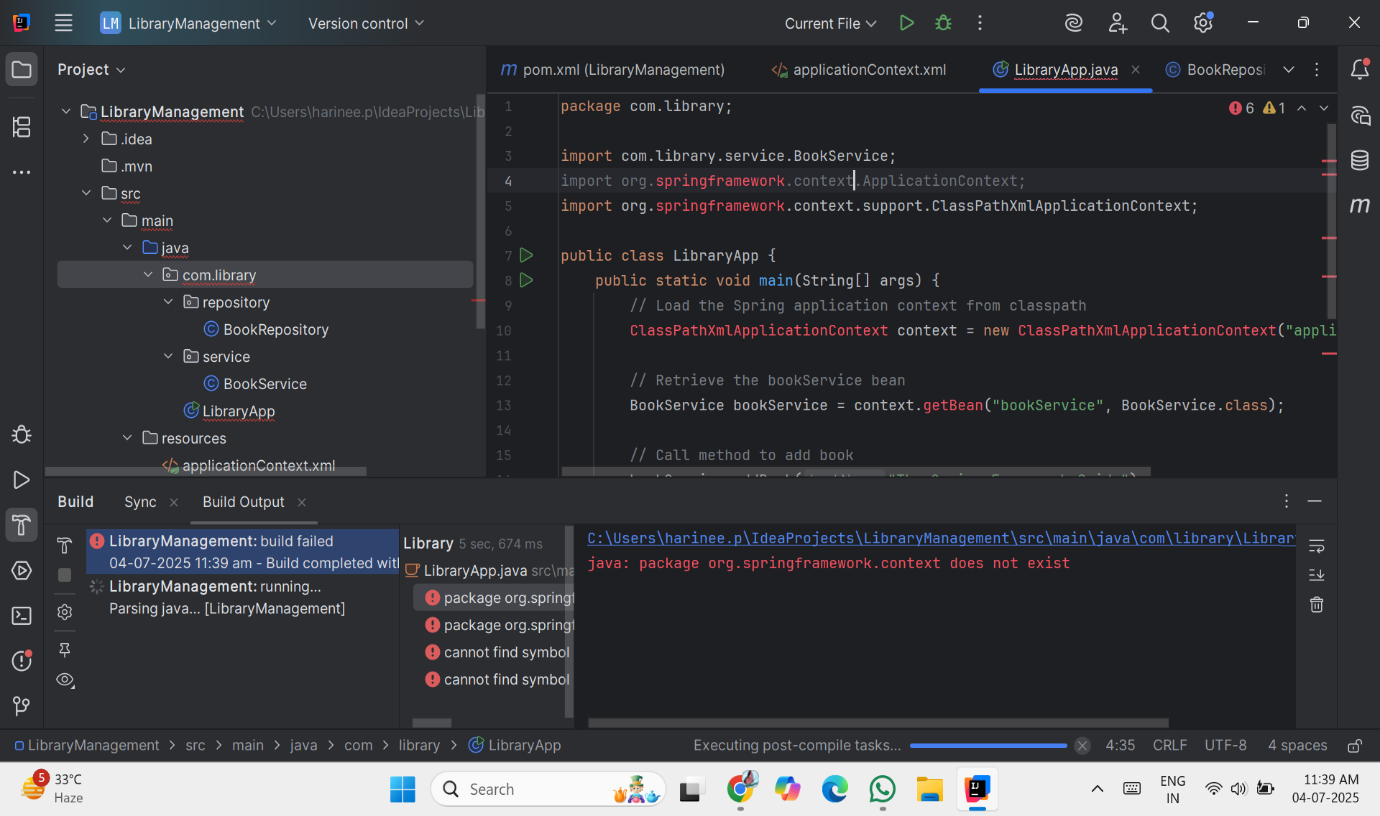
**BookService.java:**

package com.library.service;  
  
import com.library.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 // Setter for Dependency Injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook(String bookName) {  
 System.*out*.println("Adding book: " + bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

**BookRepository.java:**

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook(String bookName) {  
 System.*out*.println("Book '" + bookName + "' saved to repository.");  
 }  
}

**OUTPUT:**

****

**Exercise 2: Implementing Dependency Injection**

**CODE:**

**Pom.xml**

<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>com.example</groupId>  
 <artifactId>library-management</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <dependencies>

<dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.27</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <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>  
  
   
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-jar-plugin</artifactId>  
 <version>3.3.0</version>  
 <configuration>  
 <archive>  
 <manifest>  
 <mainClass>com.example.library.LibraryApp</mainClass>  
 </manifest>  
 </archive>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**applicationContent.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.example.library.BookRepository" />  
  
   
 <bean id="bookService" class="com.example.library.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
  
</beans>

**BookRepository.java**

package com.example.library;  
  
import java.util.Arrays;  
import java.util.List;  
  
public class BookRepository {  
  
 public List<String> findAllBooks() {  
 return Arrays.*asList*("The Hobbit", "1984", "Clean Code", "Spring in Action");  
 }  
}

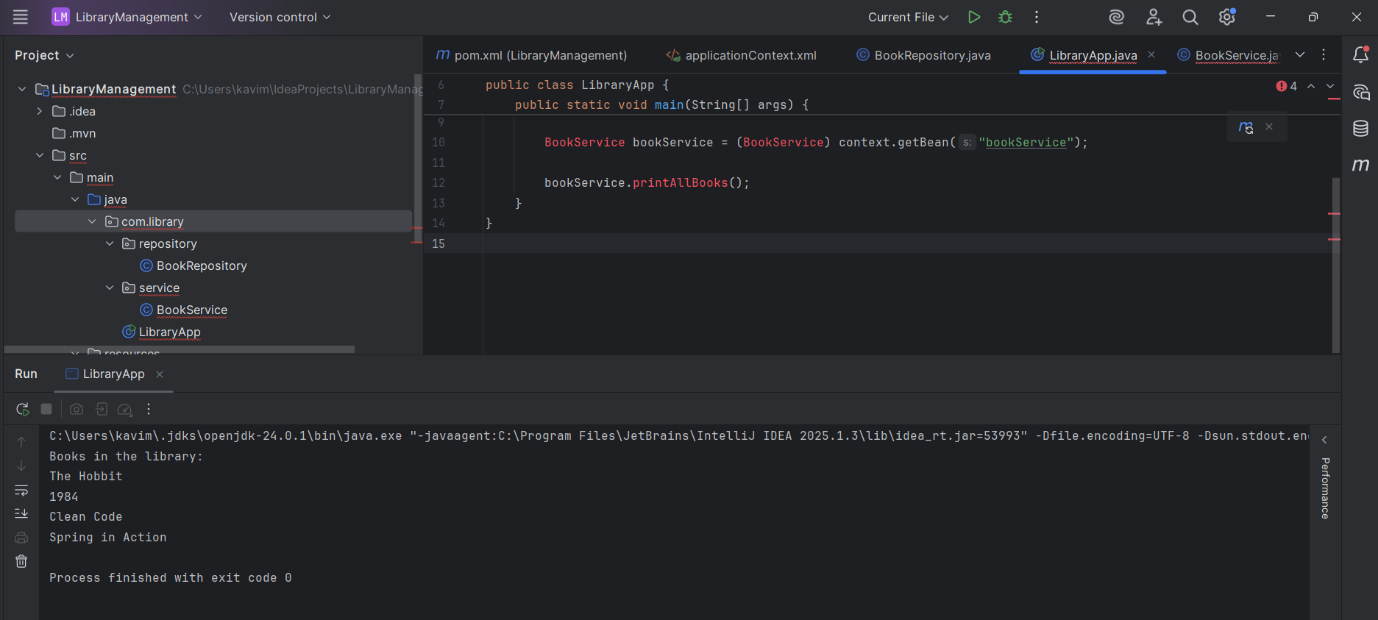
**LibraryApp.java**

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

**BookService.java**

package com.example.library;  
  
public class BookService {  
  
 private BookRepository bookRepository;  
  
   
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void printAllBooks() {  
 System.*out*.println("Books in the library:");  
 bookRepository.findAllBooks().forEach(System.*out*::println);  
 }  
}

**Output:**



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

**Pom.xml**

<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>com.example.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<java.version>1.8</java.version>

<spring.version>5.3.27</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**spring-data-jpa-handson**

**Spring Data JPA - Quick Example**

**Country.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

public class Country {

private String code;

private String name;

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**CountryRepository . java**

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.cognizant.orm\_learn.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

}

**Pom.xml**

<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/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>orm-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>orm-learn</name>

<description>Demo project for Spring Data JPA and Hibernate</description>

<properties>

<java.version>17</java.version> <!-- match your JDK -->

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-mavplugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.service.CountryService;

import java.util.List;

public class OrmLearnApplication {

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

private static CountryService countryService;

public static void main(String[] args) throws Exception {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

testGetAllCountries();

testFindCountryByCode();

testAddCountry();

testUpdateCountry();

testDeleteCountry();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

LOGGER.debug("Countries: {}", countries);

LOGGER.info("End");

}

private static void testFindCountryByCode() throws Exception {

Country country = countryService.findCountryByCode("IN");

LOGGER.debug("Found: {}", country);

}

private static void testAddCountry() throws Exception {

Country newCountry = new Country();

newCountry.setCode("ZZ");

newCountry.setName("TestLand");

countryService.addCountry(newCountry);

LOGGER.debug("Added: {}", countryService.findCountryByCode("ZZ"));

}

private static void testUpdateCountry() throws Exception {

countryService.updateCountry("ZZ", "UpdatedLand");

LOGGER.debug("Updated: {}", countryService.findCountryByCode("ZZ"));

}

private static void testDeleteCountry() throws Exception {

countryService.deleteCountry("ZZ");

LOGGER.debug("Deleted ZZ country");

}

}

**SmvecController.java**

package com.cognizant.orm\_learn;

import java.util.List;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.repository.CountryRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

public class smveccontroller {

private CountryRepository countryRepository;

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**MySql table**

USE ormlearn;

CREATE TABLE country (

co\_code VARCHAR(2) PRIMARY KEY,

co\_name VARCHAR(50)

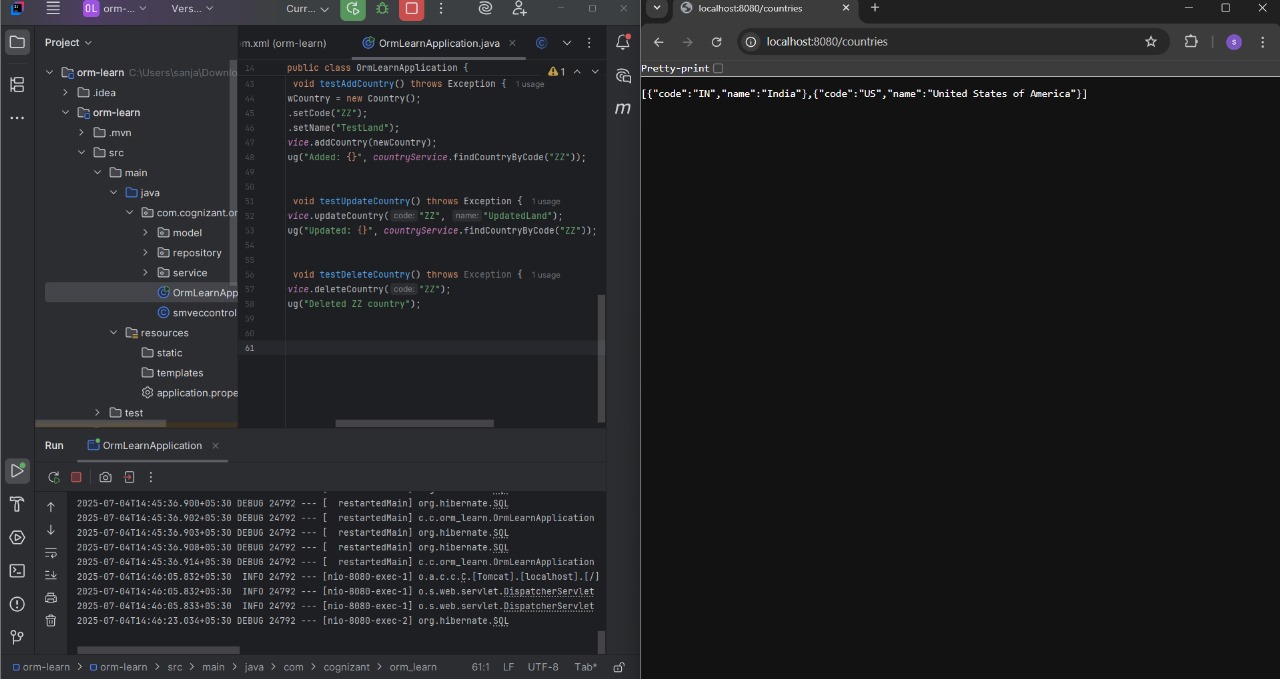
);

INSERT INTO country VALUES ('IN', 'India');

INSERT INTO country VALUES ('US', 'United States of America');

SELECT \* FROM country;

**Output:**



**spring-data-jpa-handson**

**Difference between JPA, Hibernate and Spring Data JPA**

**Pom.xml**

<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>employee-app</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>employee-app</name>

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

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**Complier.xml**

<project version="4">

<component name="CompilerConfiguration">

<annotationProcessing>

<profile name="Maven default annotation processors profile" enabled="true">

<sourceOutputDir name="target/generated-sources/annotations"/>

<sourceTestOutputDir name="target/generated-test-sources/test-annotations"/>

<outputRelativeToContentRoot value="true"/>

<module name="employee-app"/>

</profile>

</annotationProcessing>

</component>

<component name="JavacSettings">

<option name="ADDITIONAL\_OPTIONS\_OVERRIDE">

<module name="employee-app" options="-parameters"/>

</option>

</component>

</project>

**Encodings.xml**

<project version="4">

<component name="Encoding">

<file url="file://$PROJECT\_DIR$/employee-app/src/main/java" charset="UTF-8"/>

</component>

</project>

<project version="4">

<component name="AutoImportSettings">

<option name="autoReloadType" value="SELECTIVE"/>

</component>

<component name="ChangeListManager">

<list default="true" id="867a75e1-093b-4e5f-ab0c-c91e86851850" name="Changes" comment=""/>

<option name="SHOW\_DIALOG" value="false"/>

<option name="HIGHLIGHT\_CONFLICTS" value="true"/>

<option name="HIGHLIGHT\_NON\_ACTIVE\_CHANGELIST" value="false"/>

<option name="LAST\_RESOLUTION" value="IGNORE"/>

</component>

<component name="FileTemplateManagerImpl">

<option name="RECENT\_TEMPLATES">

<list>

<option value="Interface"/>

<option value="Class"/>

</list>

</option>

</component>

<component name="ProjectColorInfo">

<![CDATA[ { "associatedIndex": 7 } ]]>

</component>

<component name="ProjectId" id="2zPHAWpV4VpJMdeqcUTLYuo4pcx"/>

<component name="ProjectViewState">

<option name="hideEmptyMiddlePackages" value="true"/>

<option name="showLibraryContents" value="true"/>

</component>

<component name="PropertiesComponent">

<![CDATA[ { "keyToString": { "Application.EmployeeAppApplication.executor": "Run", "ModuleVcsDetector.initialDetectionPerformed": "true", "RunOnceActivity.ShowReadmeOnStart": "true", "last\_opened\_file\_path": "C:/Users/sanja/Downloads/employee-app" } } ]]>

</component>

<component name="TaskManager">

<task active="true" id="Default" summary="Default task">

<changelist id="867a75e1-093b-4e5f-ab0c-c91e86851850" name="Changes" comment=""/>

<created>1751625644139</created>

<option name="number" value="Default"/>

<option name="presentableId" value="Default"/>

<updated>1751625644139</updated>

</task>

<servers/>

</component>

</project>

**Output:**

