**Hands on 1**

**Create a Spring Web Project using Maven**

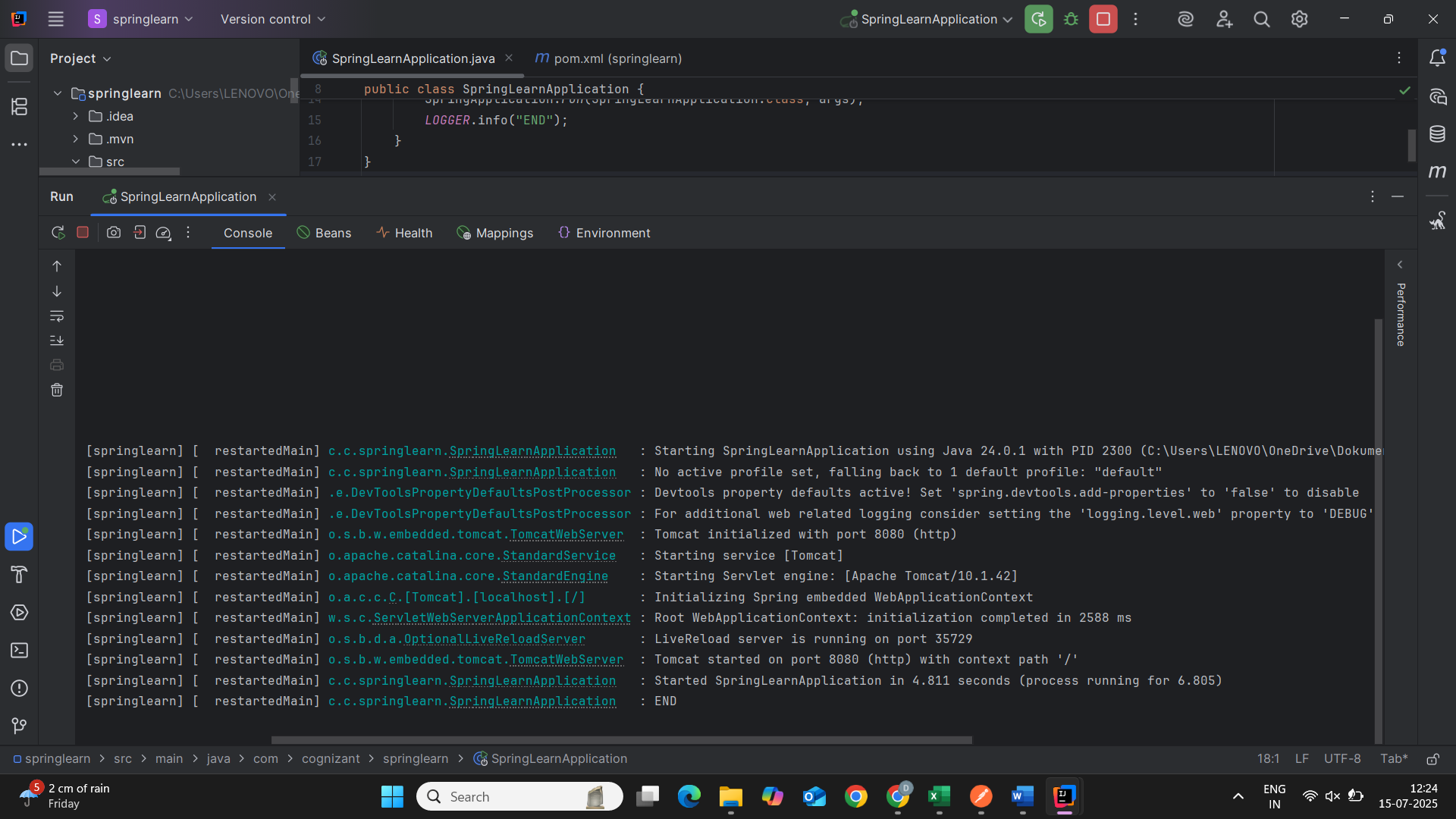
**pom.xml :**

<?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.cognizant</groupId>  
 <artifactId>springlearn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>springlearn</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-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </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>

**SpringLearnApplication.java :**

package com.cognizant.springlearn;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 *LOGGER*.info("START");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *LOGGER*.info("END");  
 }  
}

**OUTPUT :**

****

**Hands on 2**

**Spring Core – Load SimpleDateFormat from Spring Configuration XML**

**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>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</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>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.20</version> <!-- or your Spring Boot's version -->  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

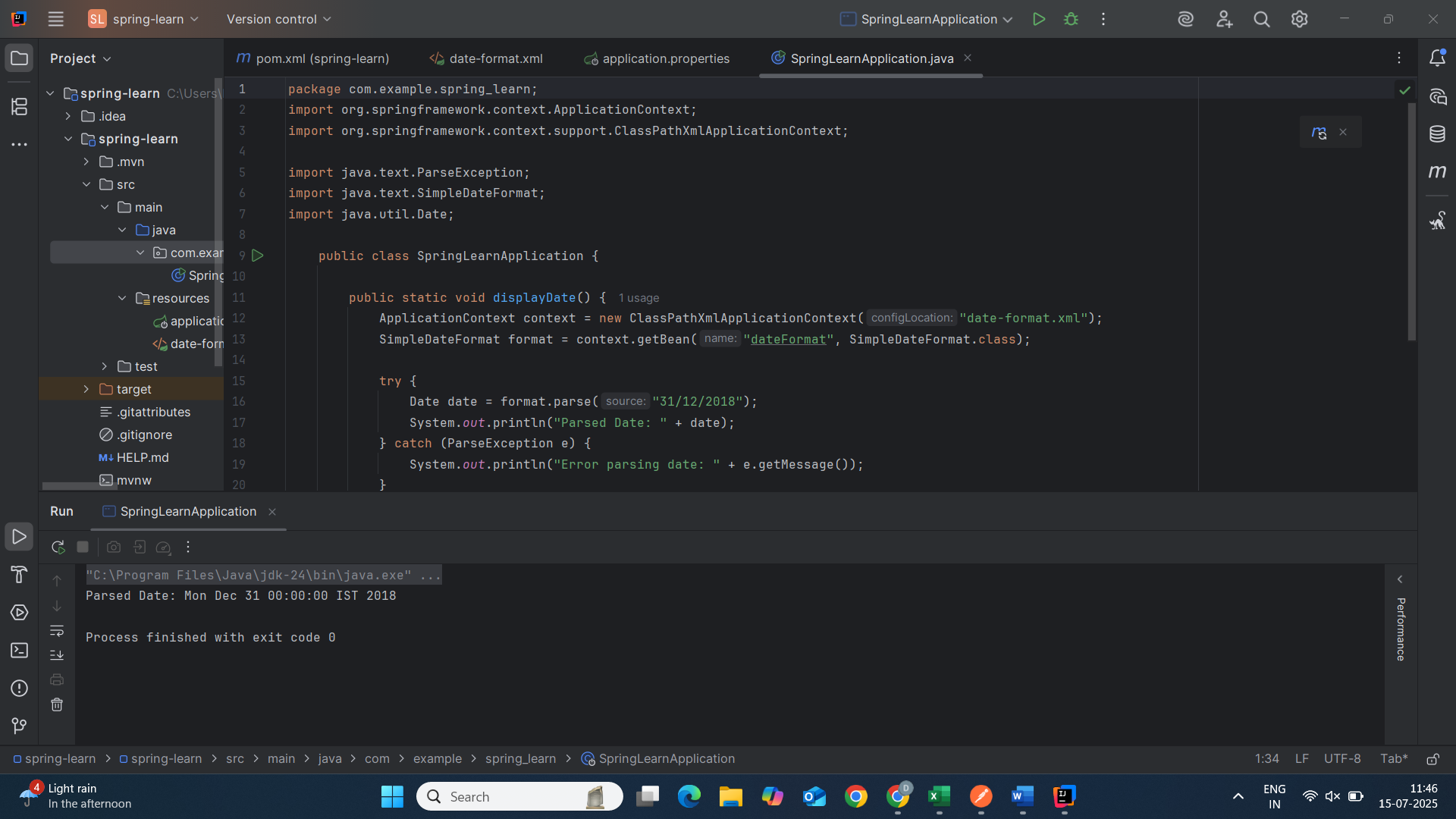
**date-format.xml :**

<?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="dateFormat" class="java.text.SimpleDateFormat">  
 <constructor-arg value="dd/MM/yyyy"/>  
 </bean>  
</beans>

**SpringLearnApplication.java :**

package com.example.spring\_learn;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
 public class SpringLearnApplication {  
  
 public static void displayDate() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
  
 try {  
 Date date = format.parse("31/12/2018");  
 System.*out*.println("Parsed Date: " + date);  
 } catch (ParseException e) {  
 System.*out*.println("Error parsing date: " + e.getMessage());  
 }  
 }  
  
 public static void main(String[] args) {  
 *displayDate*();  
 }  
 }

**OUTPUT :**

****

**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**

**Country.java :**

package com.example.country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Country.class);  
  
 private String code;  
 private String name;  
  
 // Constructor  
 public Country() {  
 *LOGGER*.debug("Inside Country Constructor.");  
 }  
  
 // Getters and Setters with debug logs  
 public String getCode() {  
 *LOGGER*.debug("Inside getCode() method. Code: {}", code);  
 return code;  
 }  
  
 public void setCode(String code) {  
 *LOGGER*.debug("Inside setCode() method. Setting code to: {}", code);  
 this.code = code;  
 }  
  
 public String getName() {  
 *LOGGER*.debug("Inside getName() method. Name: {}", name);  
 return name;  
 }  
  
 public void setName(String name) {  
 *LOGGER*.debug("Inside setName() method. Setting name to: {}", name);  
 this.name = name;  
 }  
  
 // toString() method  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

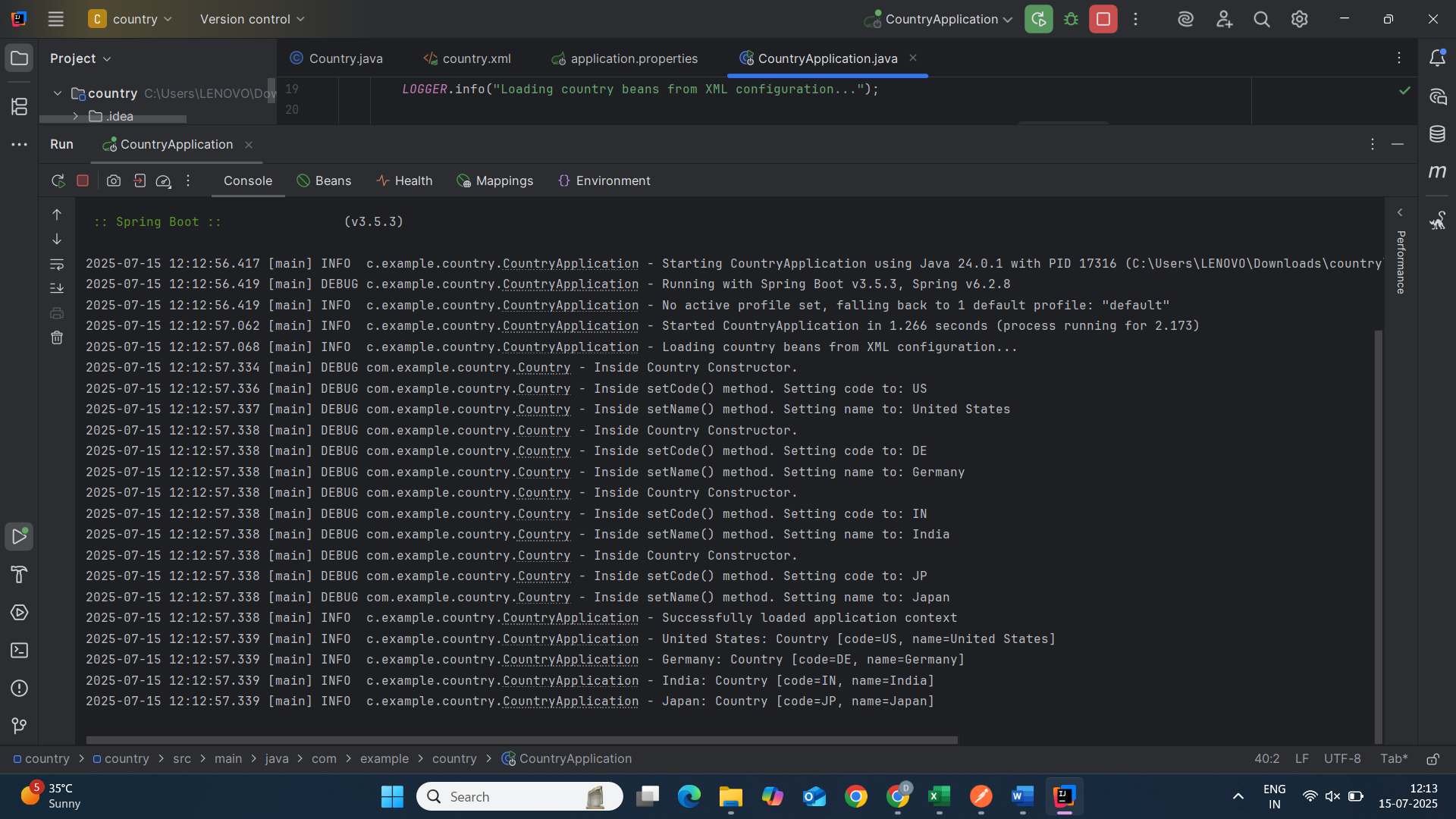
**country.xml :**

<?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">  
  
 <!-- United States -->  
 <bean id="us" class="com.example.country.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <!-- Germany -->  
 <bean id="de" class="com.example.country.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <!-- India -->  
 <bean id="in" class="com.example.country.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <!-- Japan -->  
 <bean id="jp" class="com.example.country.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
</beans>

**CountryApplication.java :**

package com.example.country;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
@SpringBootApplication  
public class CountryApplication {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryApplication.class);  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(CountryApplication.class, args);  
 *displayCountries*();  
 }  
  
 public static void displayCountries() {  
 *LOGGER*.info("Loading country beans from XML configuration...");  
  
 try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml")) {  
 *LOGGER*.info("Successfully loaded application context");  
  
 // Retrieve all country beans  
 Country us = context.getBean("us", Country.class);  
 Country de = context.getBean("de", Country.class);  
 Country in = context.getBean("in", Country.class);  
 Country jp = context.getBean("jp", Country.class);  
  
 // Log country information  
 *LOGGER*.info("United States: {}", us);  
 *LOGGER*.info("Germany: {}", de);  
 *LOGGER*.info("India: {}", in);  
 *LOGGER*.info("Japan: {}", jp);  
  
 } catch (Exception e) {  
 *LOGGER*.error("Error loading country configuration: ", e);  
 }  
 }  
}

**OUTPUT :**

****