**Create a Spring Web Project using Maven**

**Code :**

<dependency>

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

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

</dependency>

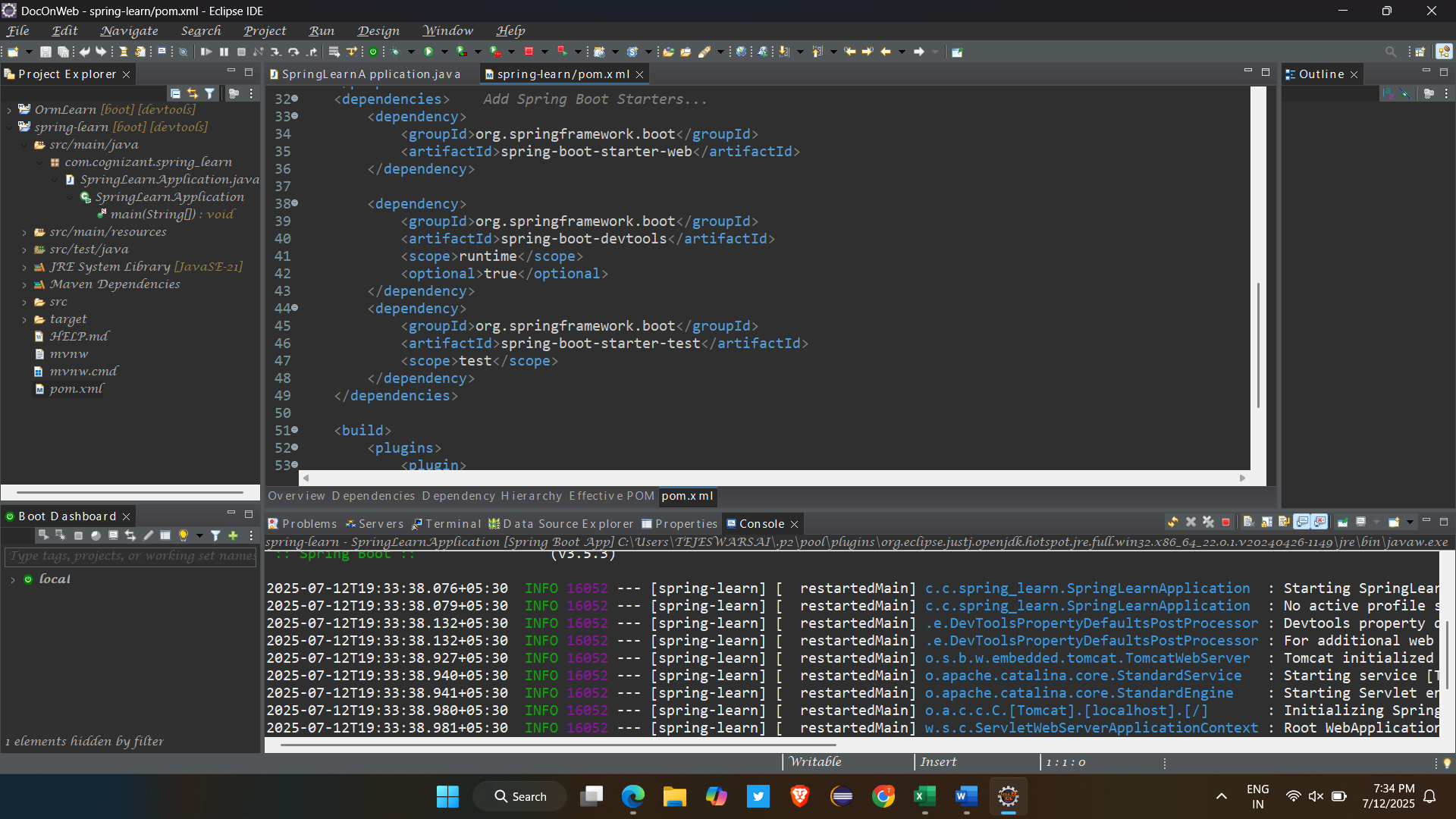
<dependency>

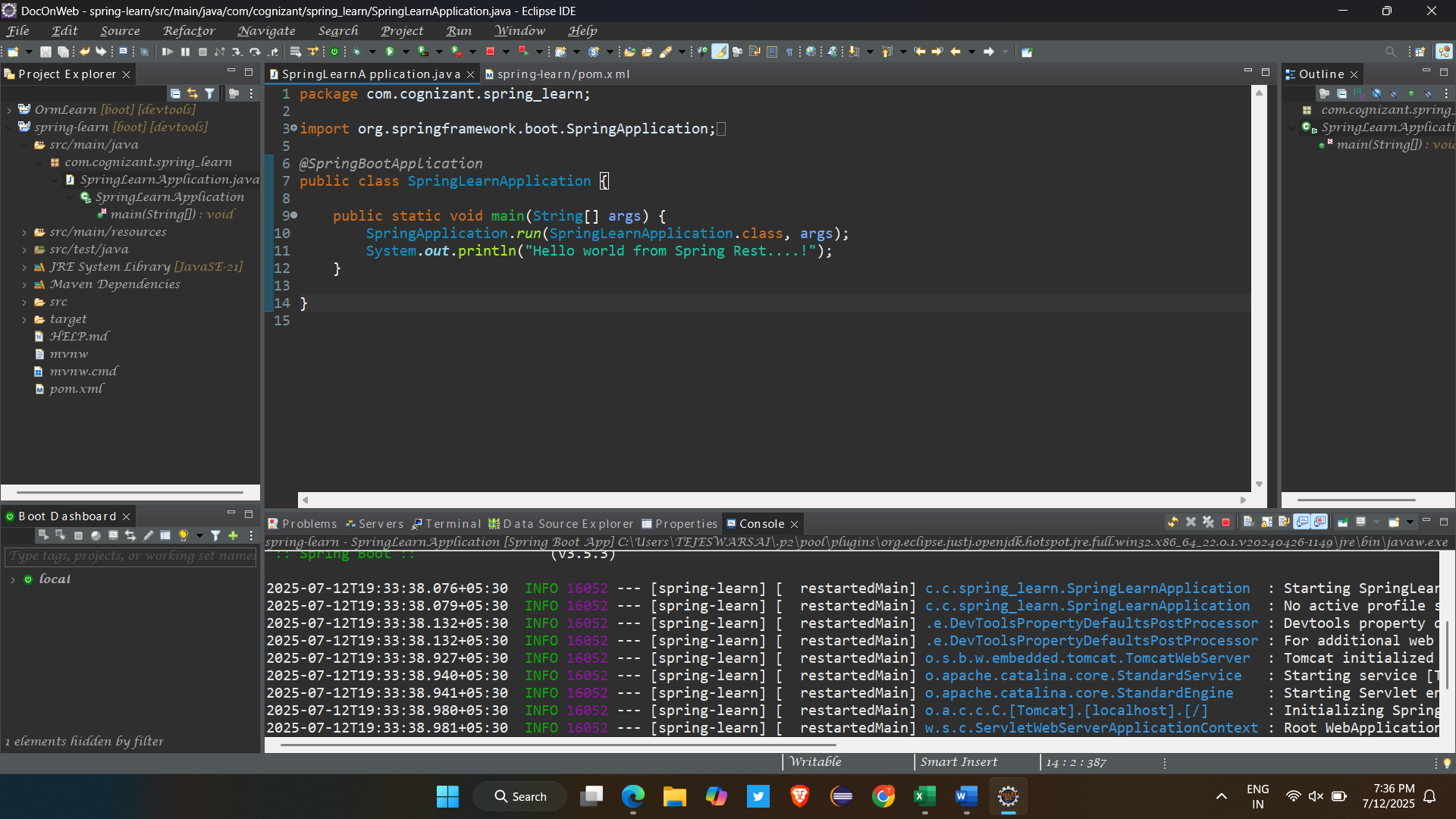
<groupId>org.springframework.boot</groupId>

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>  
  




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

**Code :**

**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.cognizant.spring\_learn;

import java.sql.Date;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

*@SpringBootApplication*

public class SpringLearnApplication {

public static void main(String[] args) {

*displayDate*();

}

public static void displayDate() {

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

java.util.Date date = format.parse("31/12/2018");

System.***out***.println(date);

} catch (ParseException e) {

System.***out***.println("Failed to parse date: " + e.getMessage());

}

}

}

**HelloController.java:**

package com.cognizant.spring\_learn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.\*;

@RestController

public class HelloController {

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

@GetMapping("/hello")

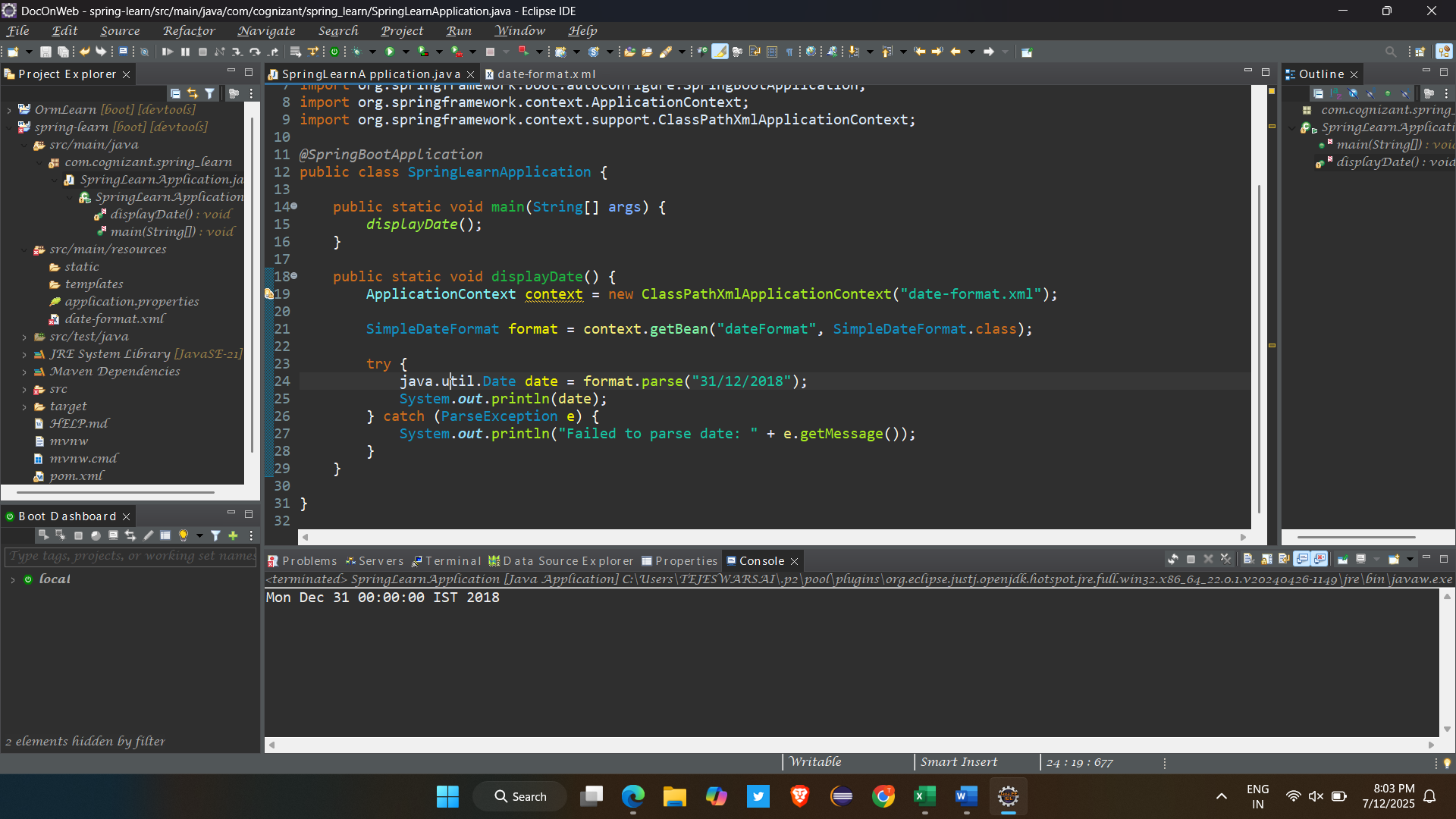
public String sayHello() {

LOGGER.info("START: sayHello()");

LOGGER.info("END: sayHello()");

return "Hello World!!";

}

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**REST - Country Web Service**   
  
Write a REST service that returns India country details in the earlier created spring learn application

**CountryController.java:**package com.cognizant.spring\_learn.controller;

import java.util.ArrayList;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

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

import com.cognizant.model.Country;

@RestController

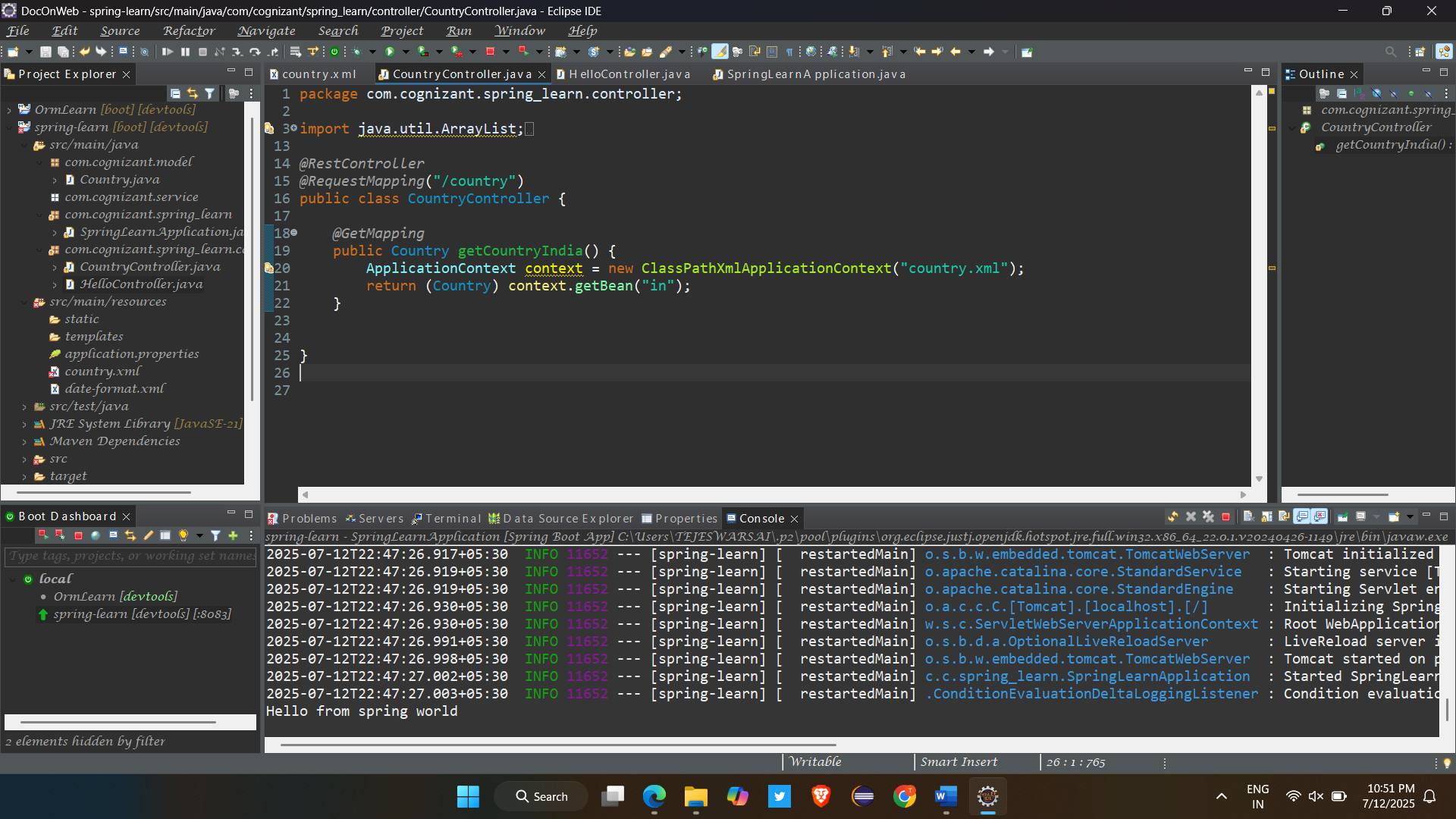
@RequestMapping("/country")

public class CountryController {

@GetMapping

public Country getCountryIndia() {

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

return (Country) context.getBean("in"); }} A screenshot of a computer

AI-generated content may be incorrect.

**REST - Get all countries**   
Write a REST service that returns all the countries.

**CountryController.java:**

*@GetMapping*("/countries")

public List<Country> getAllCountries() {

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

List<Country> countryList = (List<Country>) context.getBean("countryList");

return countryList;

}

**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"*>

<**bean** id=*"in"* class=*"com.cognizant.model.Country"*>

<**property** name=*"code"* value=*"IN"*/>

<**property** name=*"name"* value=*"India"*/>

</**bean**>

<**bean** id=*"countryList"* class=*"java.util.ArrayList"*>

<**constructor-arg**>

<**list**>

<**bean** class=*"com.cognizant.model.Country"*>

<**property** name=*"code"* value=*"IN"*/>

<**property** name=*"name"* value=*"India"*/>

</**bean**>

<**bean** class=*"com.cognizant.model.Country"*>

<**property** name=*"code"* value=*"US"*/>

<**property** name=*"name"* value=*"United States"*/>

</**bean**>

<**bean** class=*"com.cognizant.model.Country"*>

<**property** name=*"code"* value=*"JP"*/>

<**property** name=*"name"* value=*"Japan"*/>

</**bean**>

<**bean** class=*"com.cognizant.model.Country"*>

<**property** name=*"code"* value=*"DE"*/>

<**property** name=*"name"* value=*"Germany"*/>

</**bean**>

</**list**>

</**constructor-arg**>

</**bean**>

</**beans**>

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**REST - Get country based on country code**   
Write a REST service that returns a specific country based on country code. The country code should be case insensitive.

**CountryController.java:**

*@GetMapping*("/countries/{code}")

public Country getCountry(*@PathVariable* String code) {

return countryService.getCountry(code);

}

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.