**Spring REST using Spring Boot 3**

**Hands-on 1 Submission – Spring Boot Web Application Setup**

**🔹 Objective**

To demonstrate the creation and execution of a simple Spring Boot web application using Maven and verify the output logs to confirm successful startup.

**🔹 Steps Followed**

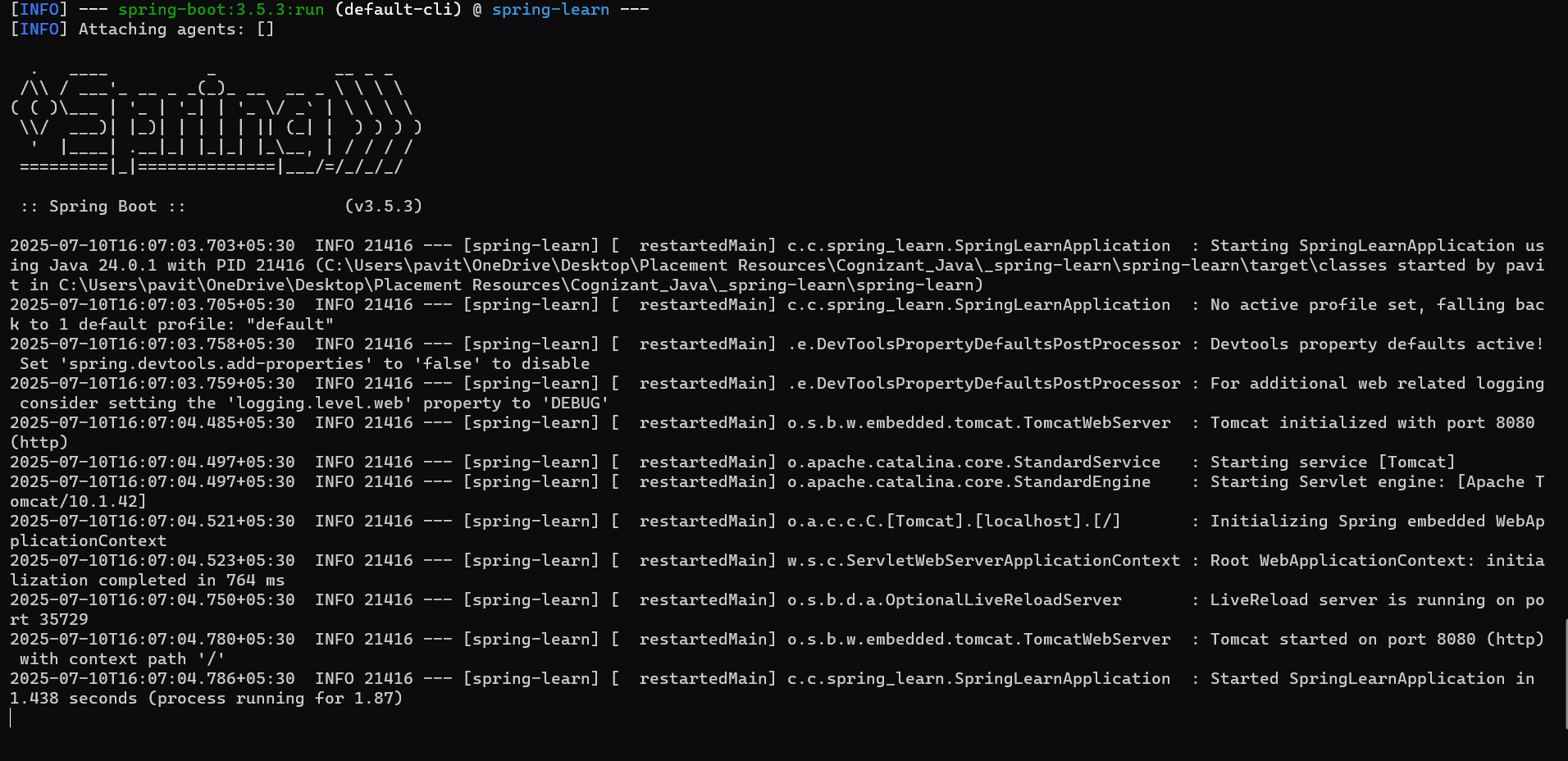
1. **Project Creation**
   * Used <https://start.spring.io> to generate a Spring Boot project.
   * Group: com.cognizant
   * Artifact: spring-learn
   * Selected Dependencies:
     + Spring Web
     + Spring Boot DevTools
2. **Import into Workspace**
   * Extracted the ZIP file and imported the project into **VS Code**.
   * Verified that the pom.xml contained the correct dependencies and the spring-boot-maven-plugin.
3. **Build and Run**
   * Opened terminal and navigated to the correct project folder:

C:\Users\pavit\OneDrive\Desktop\PlacementResources\Cognizant\_Java\\_spring-learn\spring-learn

* + Executed the command:

**🔹 Log Output Verification**

The following log confirms successful execution of the main method and embedded Tomcat server startup:



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

**Country.java**

package com.cognizant.spring\_learn;

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;

}

@Override

public String toString() {

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

}

}

**country.xml (in src/main/resources)**

xml

CopyEdit

<?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="country" class="com.cognizant.spring\_learn.Country">

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

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

</bean>

</beans>

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

public static void main(String[] args) {

displayCountry();

}

public static void displayCountry() {

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

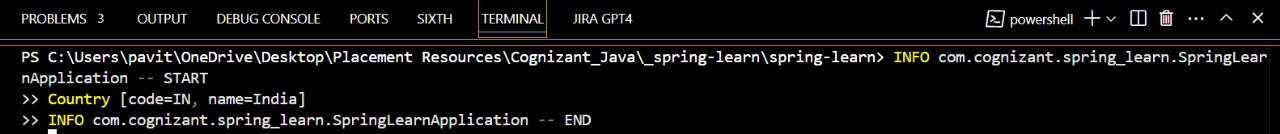
Country country = (Country) context.getBean("country");

System.out.println(country.toString());

}

}

**Output**





**Hello World RESTful Web Service - Spring Boot Application**

**🏠 Project Name:**

spring-learn

**🔎 Objective:**

To implement a simple RESTful web service using Spring Boot that returns the text Hello World!! upon accessing the URL /hello.

**✅ 1. REST Endpoint Implementation**

**File:** HelloController.java **Location:** src/main/java/com/cognizant/spring\_learn/controller/HelloController.java

package com.cognizant.spring\_learn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class HelloController {

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

@GetMapping("/hello")

public String sayHello() {

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

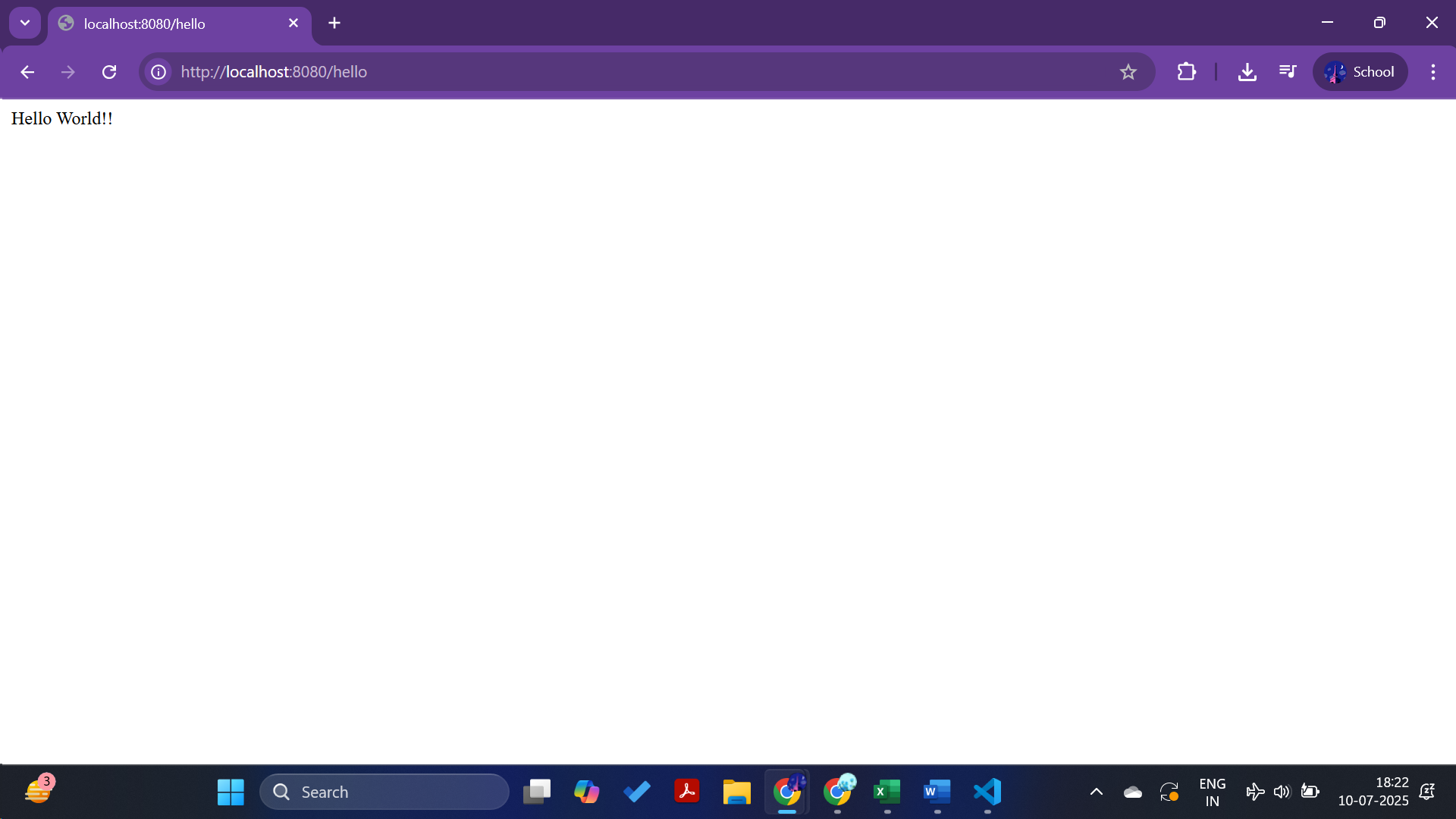
String message = "Hello World!!";

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

return message;

}

}  
**Output**



**REST - Country Web Service**

**REST Endpoint Implementation**

**File:** CountryController.java  
**Location:** src/main/java/com/cognizant/spring\_learn/controller/CountryController.java

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

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

Country country = (Country) context.getBean("country");

LOGGER.info("END - getCountryIndia()");

return country;

}

}

**2. Spring Bean Configuration**

**File:** country.xml  
**Location:** src/main/resources/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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.spring\_learn.Country">

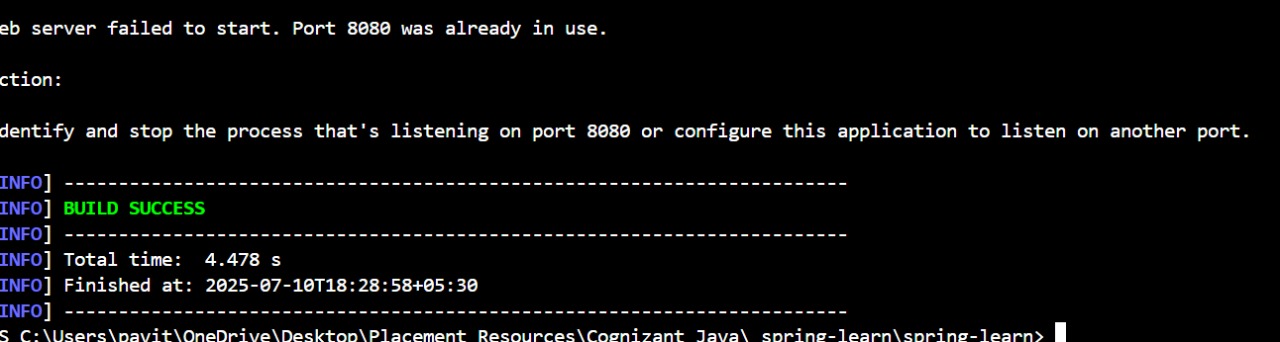
<property name="code" value="IN"/>

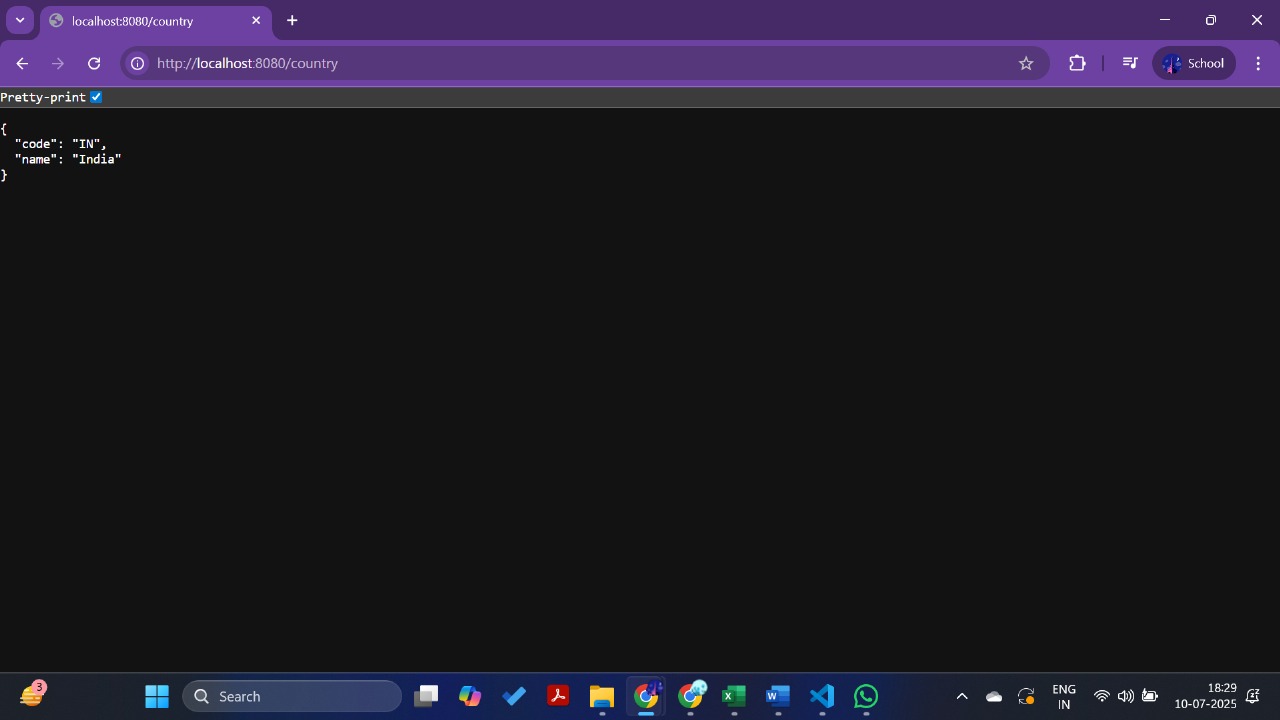
<property name="name" value="India"/>

</bean>

</beans>

**Output**





**REST - Get country based on country code**

**country.xml – Add Multiple Countries**

Make sure your country.xml contains a **List of countries**:

xml

CopyEdit

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

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn.Country">

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

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

</bean>

</list>

</constructor-arg>

</bean>

**CountryService.java**

src/main/java/com/cognizant/spring\_learn/service/CountryService.java

java

CopyEdit

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

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

return countryList.stream()

.filter(country -> country.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null); // Optionally handle with exception

}

}

**Add to CountryController.java**

@GetMapping("/country/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START - getCountry()");

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry()");

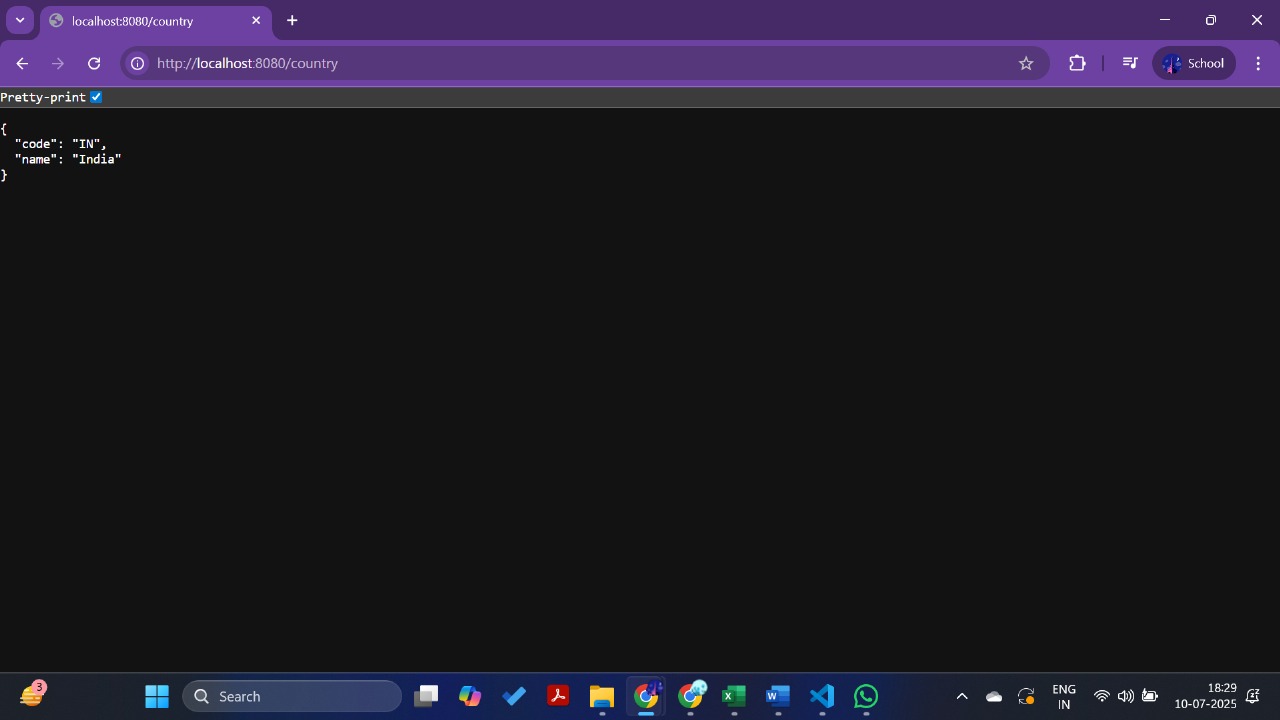
return country;

}

**URL to Test**

<http://localhost:8080/country/in>

**Output**



**Create authentication service that returns JWT**

**🔸 HelloController.java**

@RestController

public class HelloController {

@GetMapping("/hello")

public String sayHello() {

return "Hello World!!";

}

}

**CountryController.java**

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

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

return (Country) context.getBean("country");

}

}

**Country.java**

public class Country {

private String code;

private String name;

// Getters and Setters

}

**country.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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="country" class="com.cognizant.spring\_learn.model.Country">

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

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

</bean>

</beans>

**application.properties**

(Optional, only if port is changed)

properties

CopyEdit

server.port=8080

Output  
  
