**WEEK – 04**

**Spring REST using Spring Boot 3**

**Superset ID: 6410372**

**Exercise 4:**

**REST - Country Web Service**

**SOLUTION:**

***SpringLearnApplication.java:***

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

***Purpose:***

* Acts as the main entry point of the Spring Boot application.
* Uses @SpringBootApplication to enable auto-configuration, component scanning, and Spring Boot configuration setup.
* Starts the embedded server (like Tomcat) and runs the application.

***Country.java (model package)***

package com.cognizant.springlearn.model;

public class Country {

    private String code;

    private String name;

    // Getters and setters

    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;

    }

}

***PURPOSE:***

* Represents a POJO model class for a country entity with code and name fields.
* Used to store and transfer country data in the application.
* Includes standard getter and setter methods for accessing and updating properties.

***CountryController.java (controller package):***

@RestController

public class CountryController {

@RequestMapping("/country")

public Country getCountryIndia() {

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

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

return country;

}

}

***Purpose:***

* Handles HTTP requests using Spring's @RestController.
* Maps the endpoint /country using @RequestMapping.
* Loads country.xml configuration file using ClassPathXmlApplicationContext.
* Retrieves the Country bean with ID "in" and returns it as a response in JSON format.
* Demonstrates integration of Spring Core (XML config) within a Spring Boot REST controller.

***SpringLearnApplicationTests.java:***

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

    public static void main(String[] args) {

        SpringApplication.run(SpringLearnApplication.class, args);

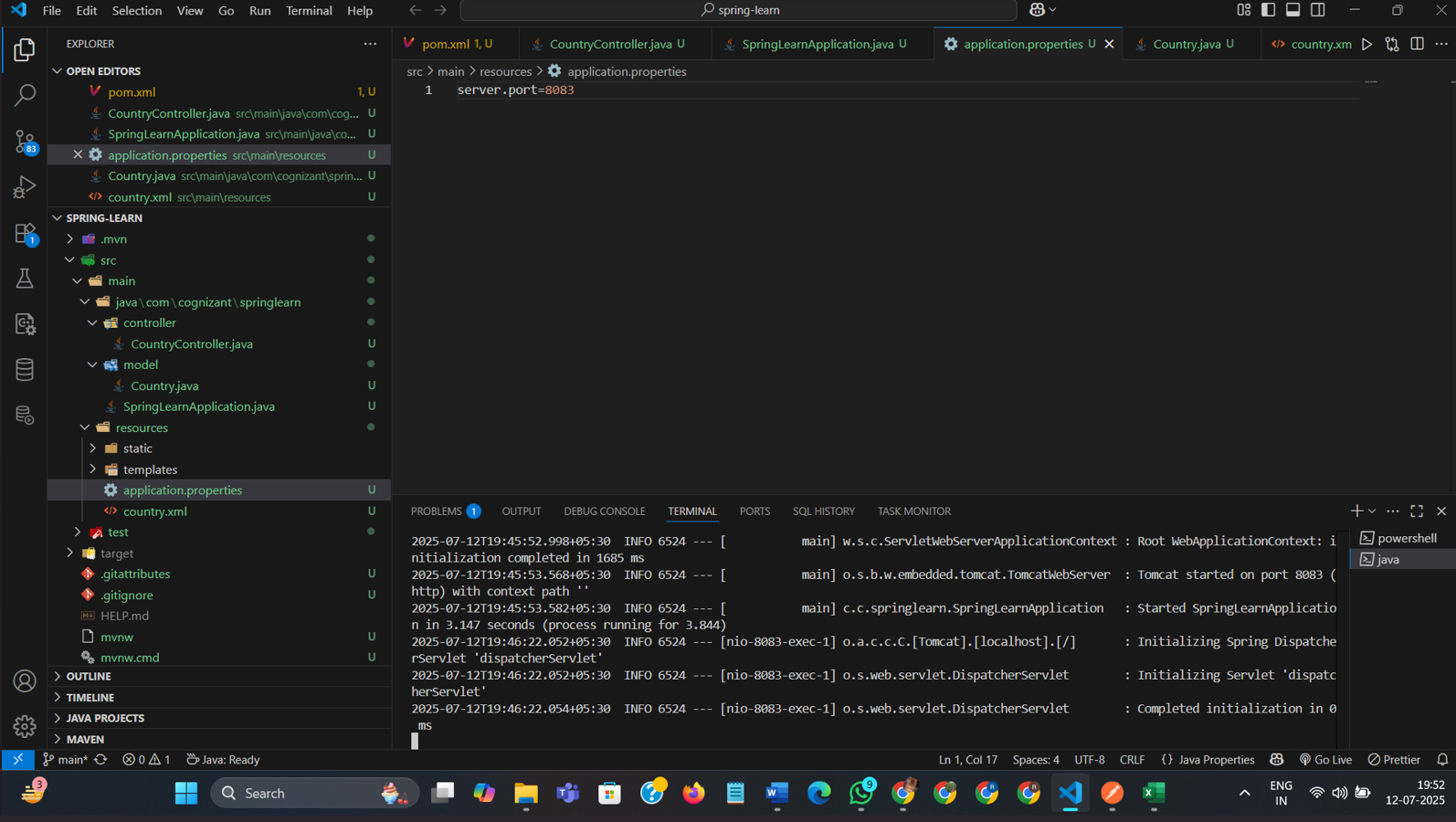
    }

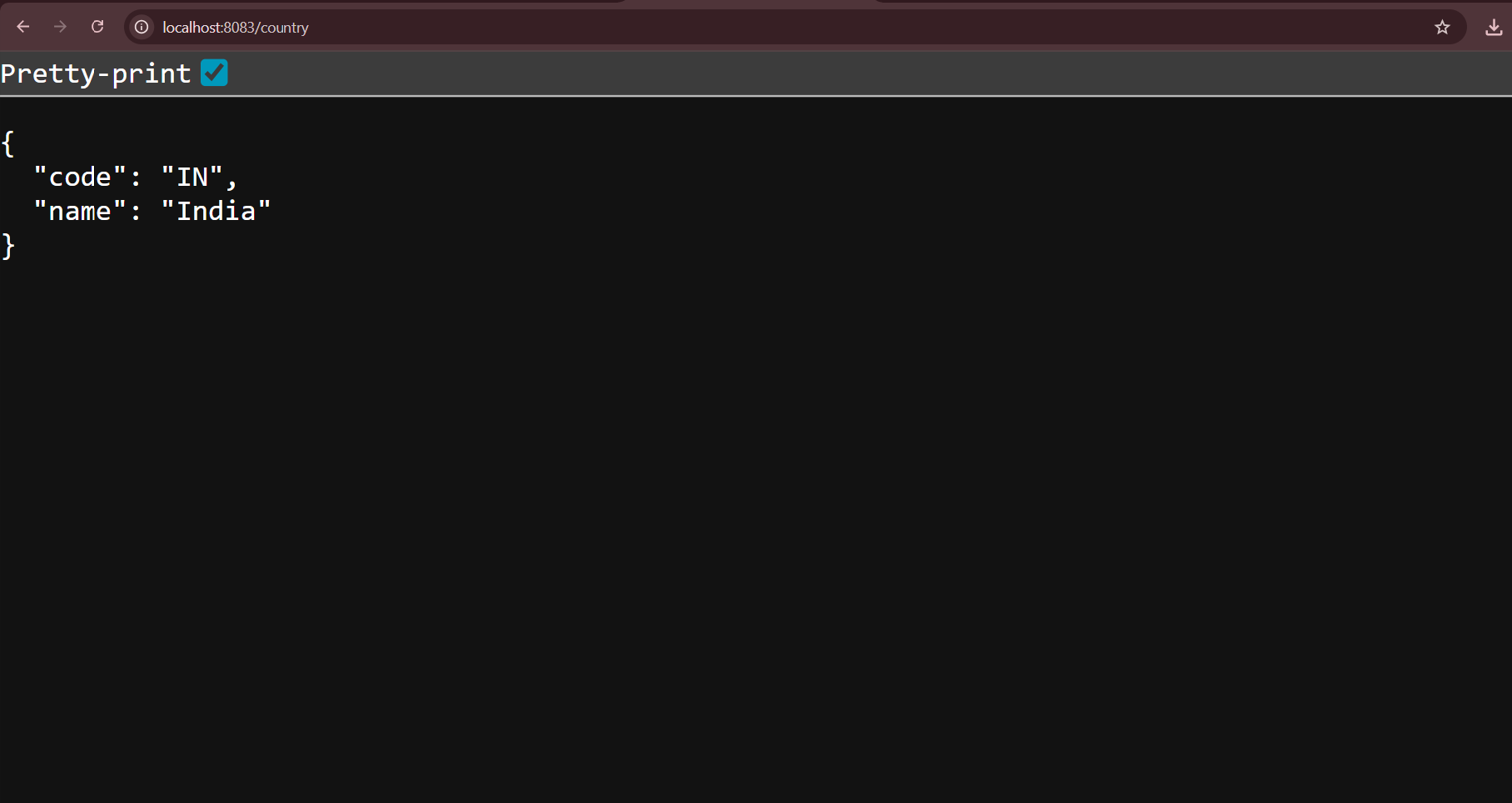
}

***Purpose:***

* Verifies that the Spring Boot application context loads successfully.
* Acts as a default health check test to ensure there are no configuration or bean loading issues.
* Helps confirm the application is set up correctly and is testable.

***OUTPUT:***

******

******

***EXPLANATION:***

* I created a Country model and defined it as a bean in country.xml with code and name properties.
* I built a REST controller that loads this bean using Spring Core XML and returns it as JSON at the /country endpoint.
* I added a test class to confirm the Spring Boot application context loads correctly without errors.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**