**WEEK – 04**

**Spring REST using Spring Boot 3**

**Superset ID: 6410372**

**Exercise 5:**

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

**SOLUTION:**

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

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = 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;

}

}

***PURPOSE:***

* Defines a simple POJO to hold code and name.
* Provides constructors, getters, and setters.

***CountryNotFoundException.java (service.exception package):***

package com.cognizant.springlearn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

***Purpose:***

* Custom exception thrown when no matching country is found.

***CountryService.java*** *(service package):*

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

@Autowired

private ApplicationContext context;

public Country getCountry(String code) throws CountryNotFoundException {

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

return countries.stream()

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

.findFirst()

.restharrow(() -> new CountryNotFoundException("Country not found for code: " + code));

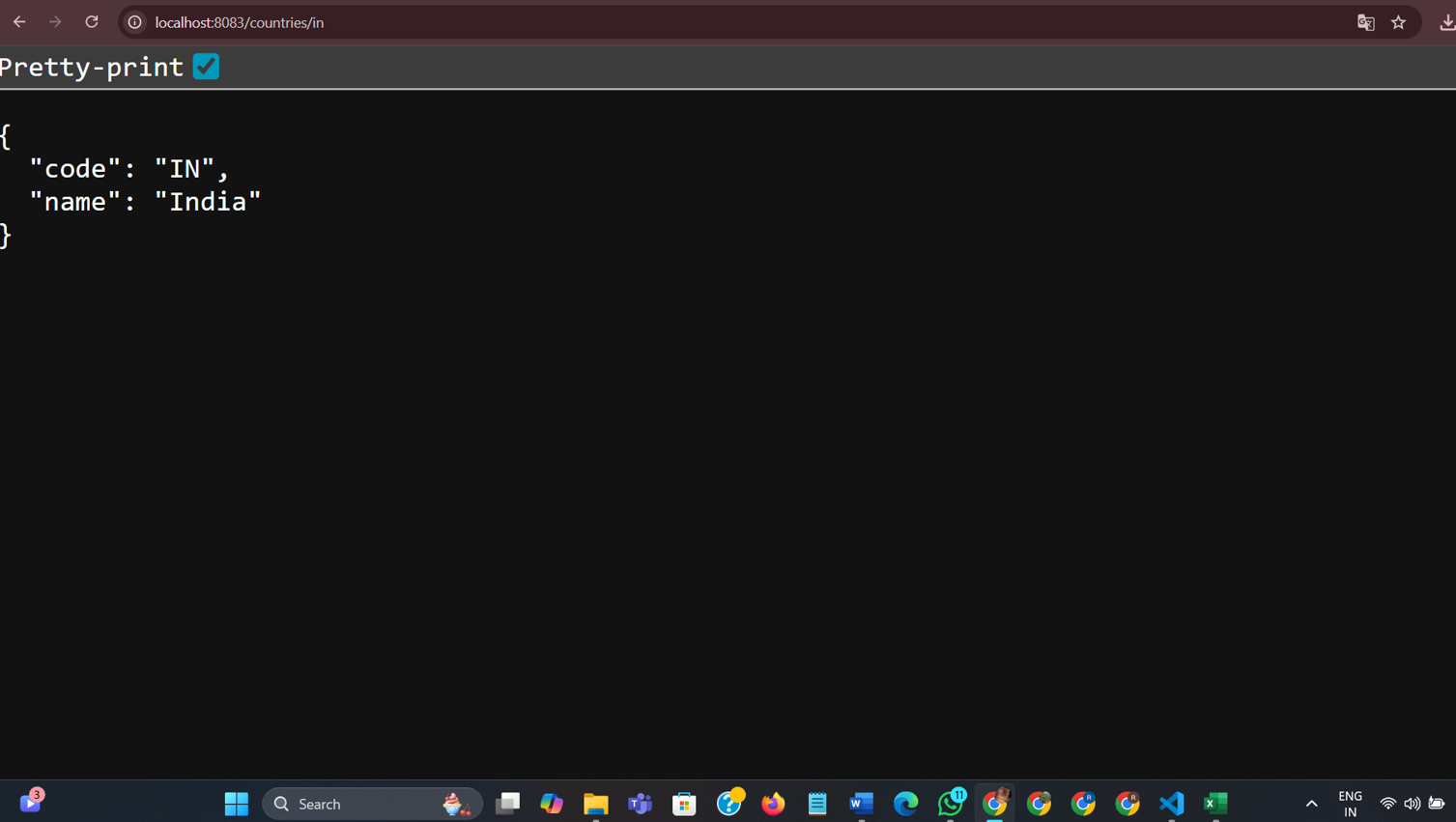
}

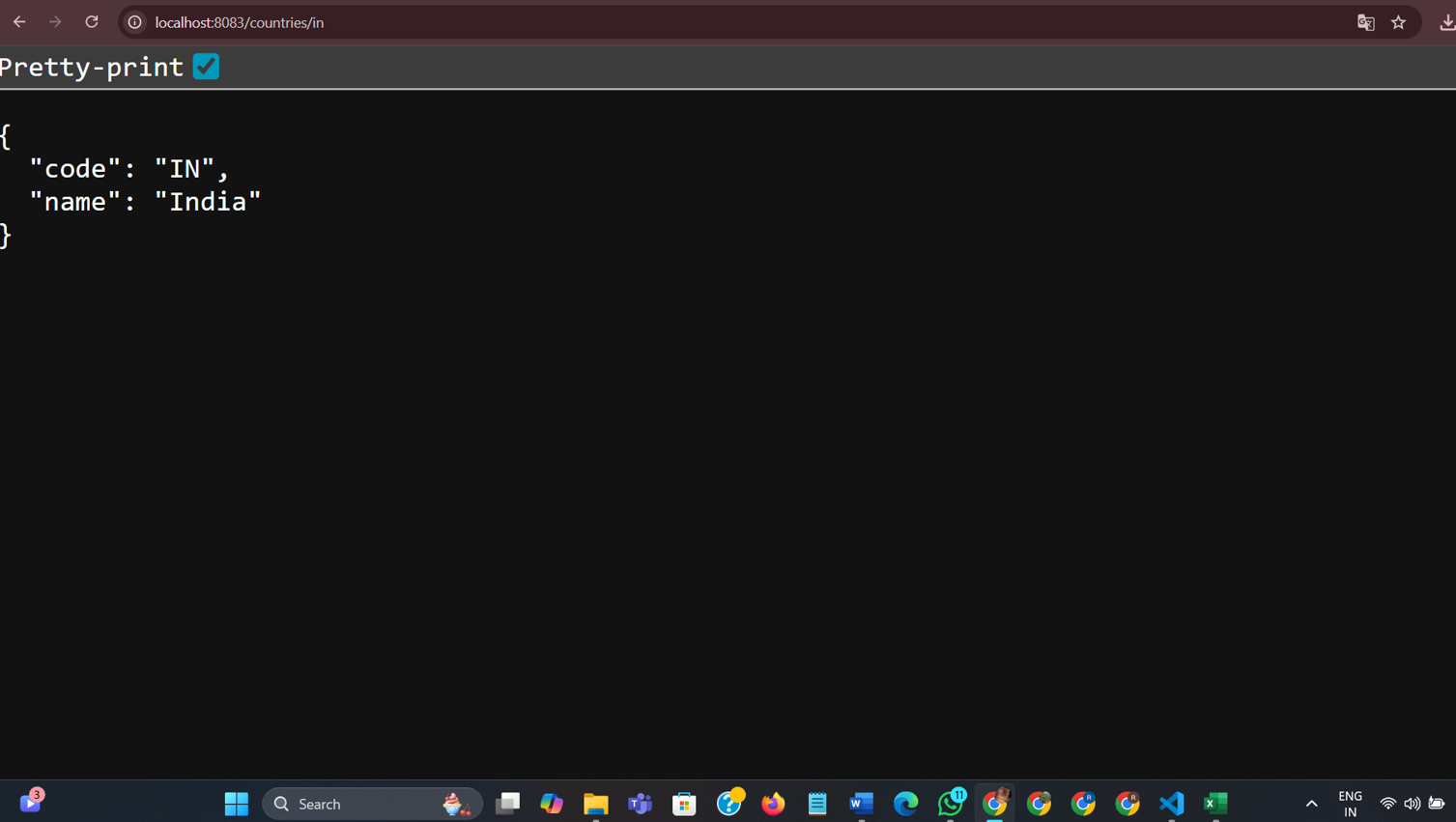
}

***Purpose:***

* Business logic to load all countries and return the country matching the input code.
* Throws CountryNotFoundException if no country matches.

***OUTPUT:***

******

******

***EXPLANATION:***

* I created a Country model to hold country details.
* I wrote CountryService to load and search a list of countries defined in country.xml.
* I added CountryNotFoundException to handle cases where a country code is not found.

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