**Exercise 1: Hello World RESTful WEB Service**

**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");

LOGGER.info("END");

return "Hello World!!";

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

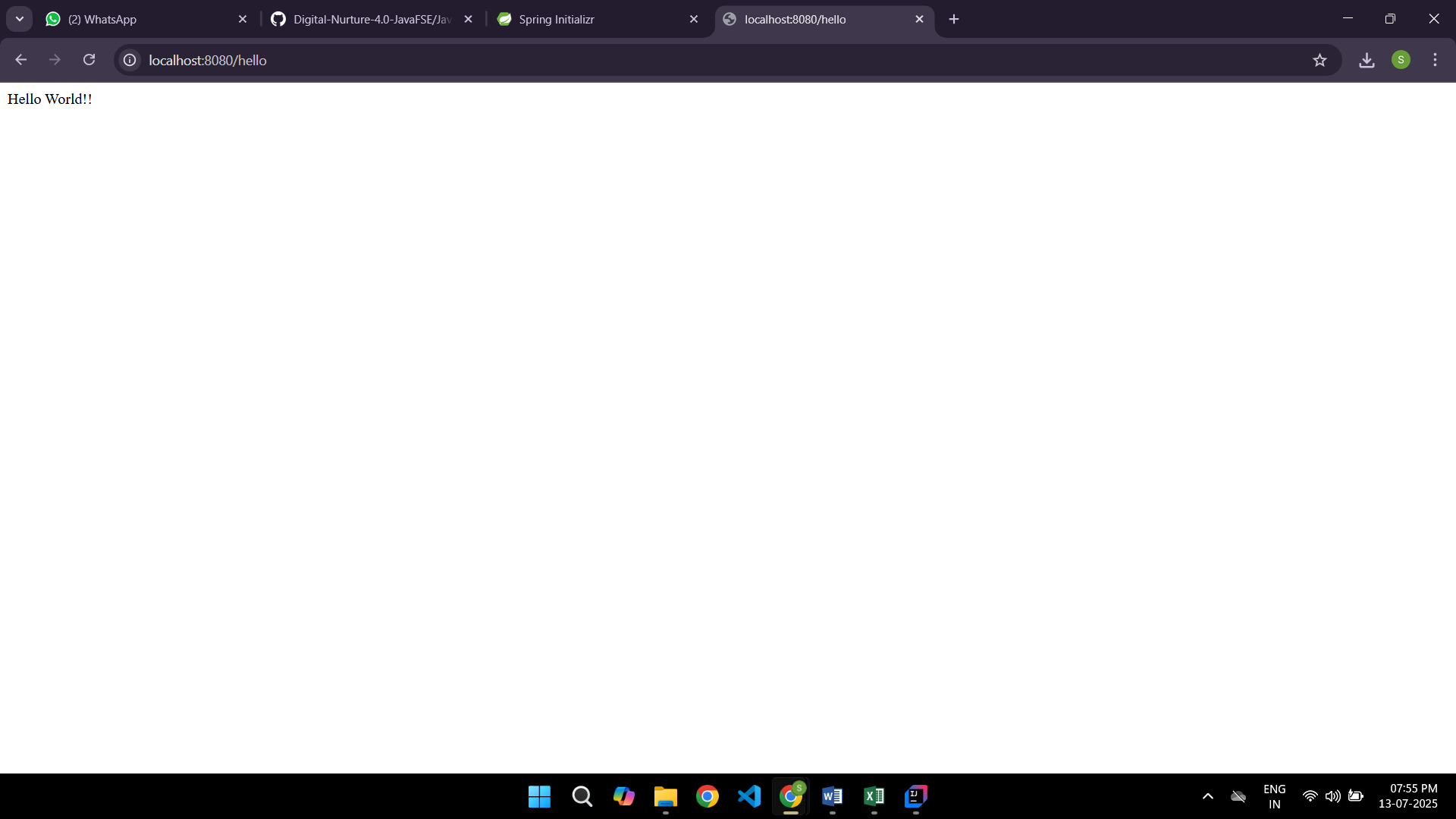
public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

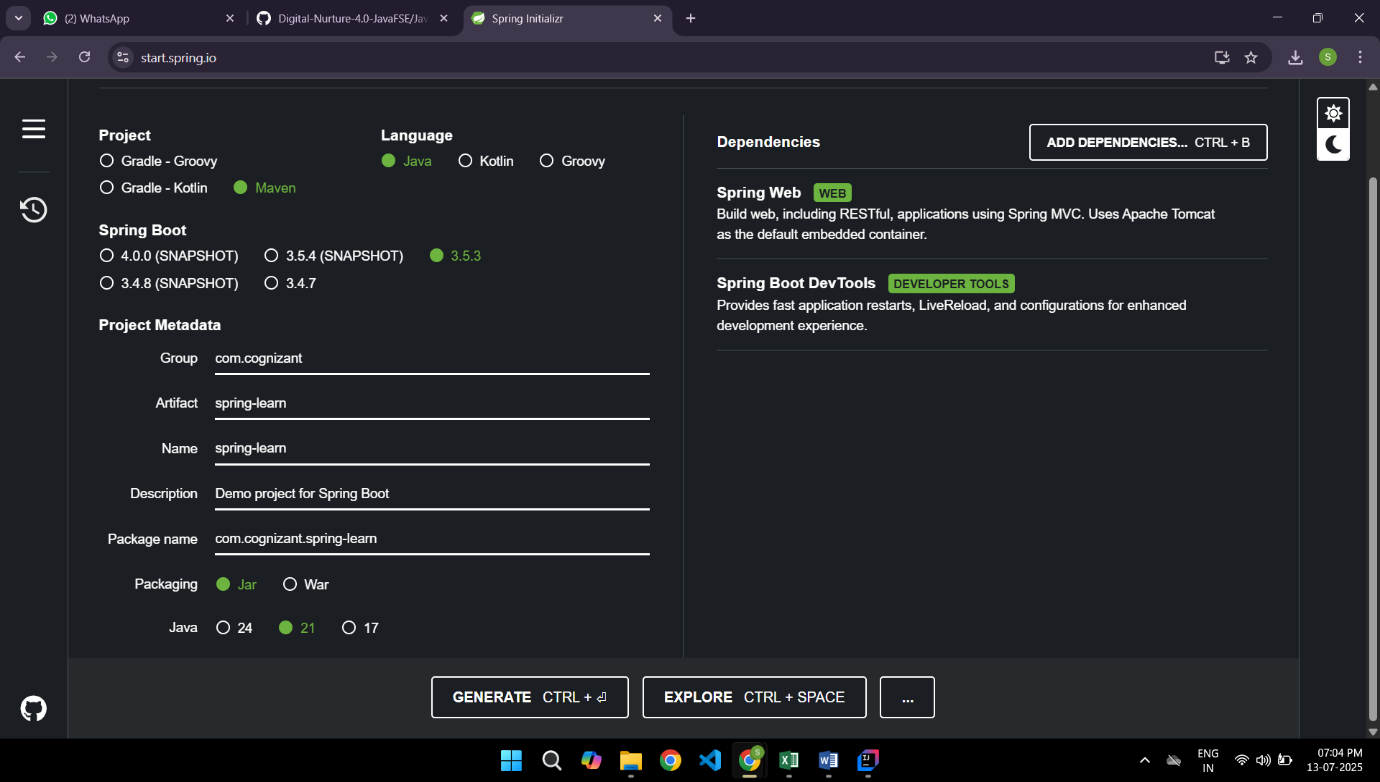
}

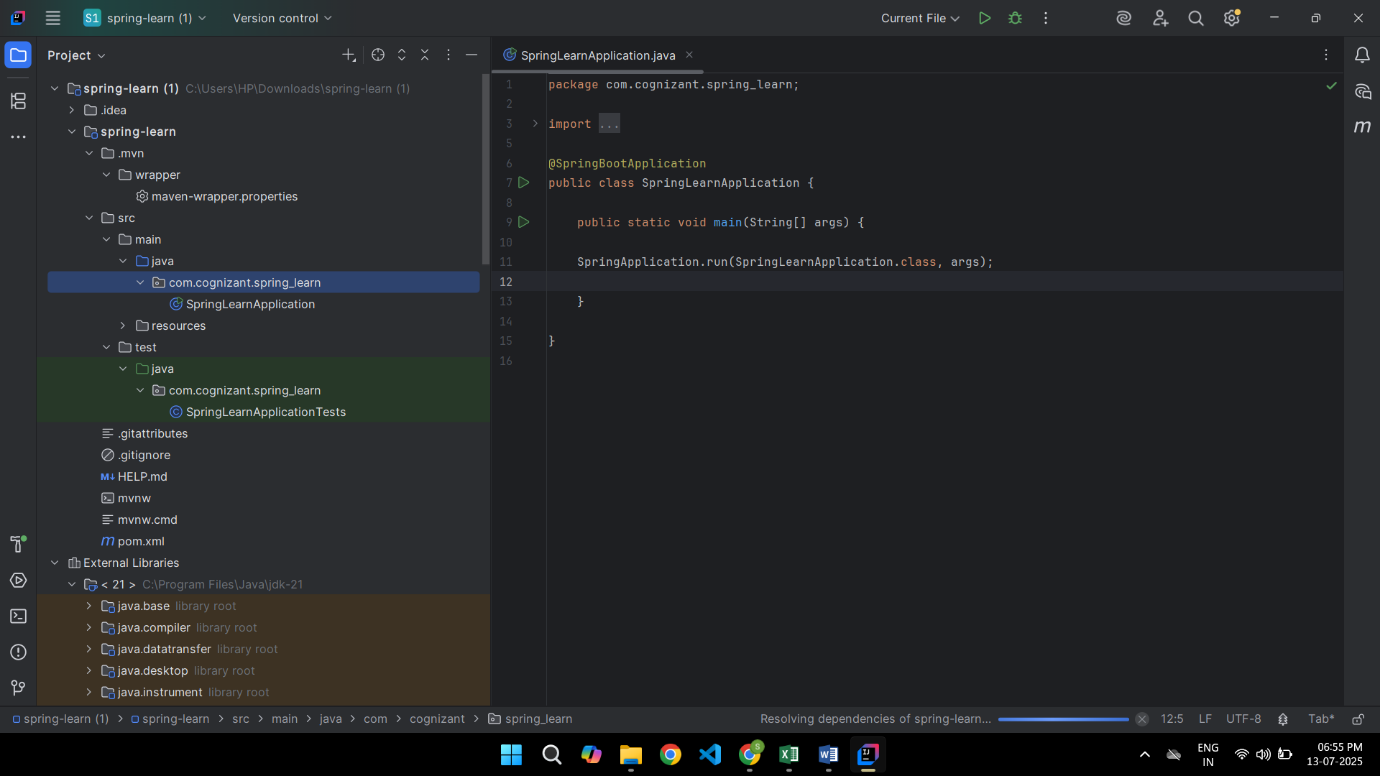
}

**OUTPUT:**

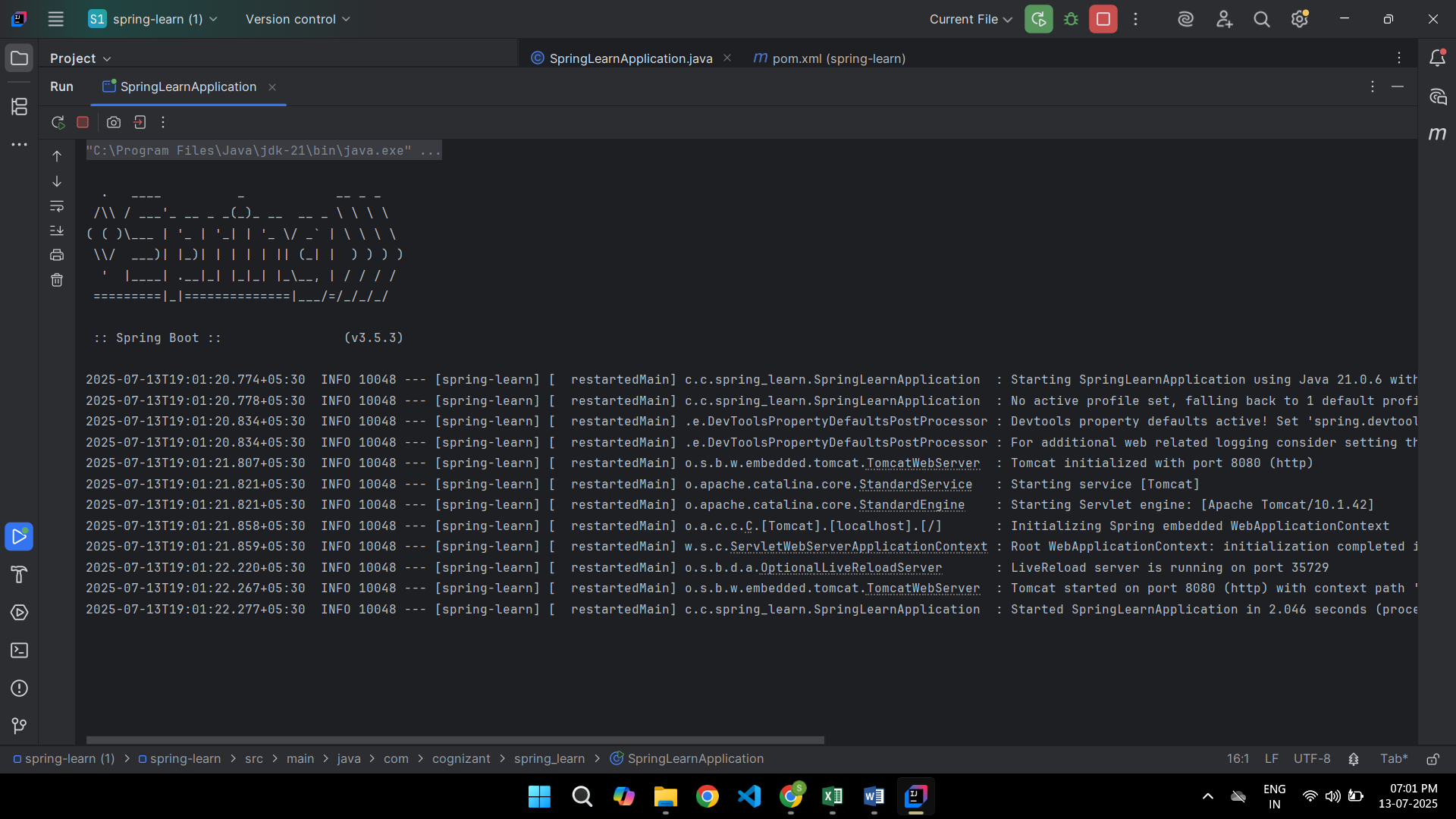


**Exercise 1 : Create a Spring Web Project Using Maven**

****

****

**OUTPUT:**



**Exercise 2: REST- Country Web Service**

**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="in" class="com.cognizant.spring\_learn.model.Country">**

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

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

**</bean>**

**</beans>**

**Country.java**

**package com.cognizant.spring\_learn.model;**

**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;**

**}**

**}**

**CountryController.java**

**package com.cognizant.spring\_learn.Controller;**

**import com.cognizant.spring\_learn.model.Country;**

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

**import org.springframework.context.ApplicationContext;**

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

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

**@RestController**

**public class CountryController {**

**@Autowired**

**private ApplicationContext context;**

**@RequestMapping("/country")**

**public Country getCountryIndia() {**

**return (Country) context.getBean("in");**

**}**

**}**

**SpringLearnApplication.java**

**package com.cognizant.spring\_learn;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.context.annotation.ImportResource;**

**@SpringBootApplication**

**@ImportResource("classpath:country.xml")**

**public class SpringLearnApplication {**

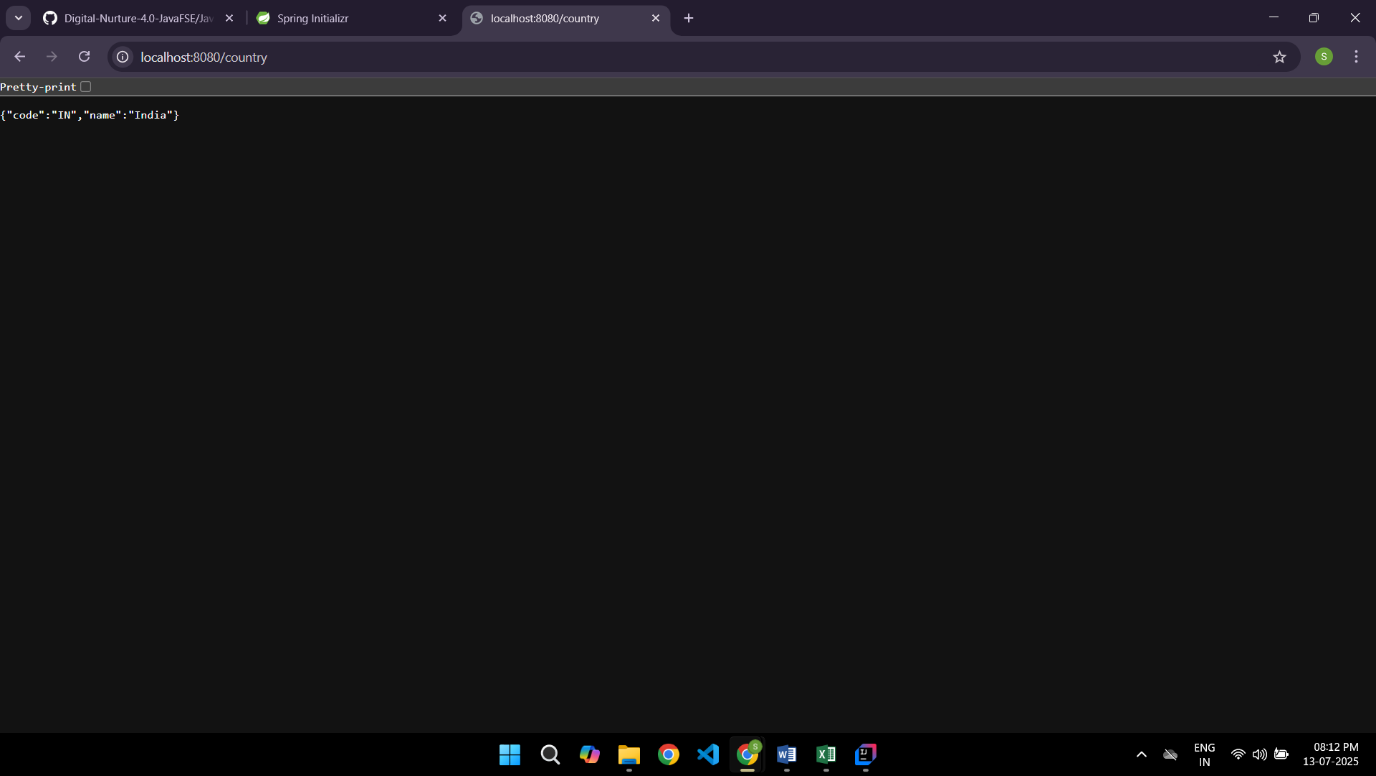
**public static void main(String[] args) {**

**SpringApplication.run(SpringLearnApplication.class, args);**

**}**

**}**

**OUTPUT:**

****

**Exercise 2: Spring Core Load Country 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.cognizant</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>17</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>**

**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>**

**Country.java**

**package com.cognizant.spring\_learn;**

**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;**

**public Country() {**

**LOGGER.debug("Inside Country Constructor");**

**}**

**public String getCode() {**

**LOGGER.debug("Getting code: {}", code);**

**return code;**

**}**

**public void setCode(String code) {**

**LOGGER.debug("Setting code: {}", code);**

**this.code = code;**

**}**

**public String getName() {**

**LOGGER.debug("Getting name: {}", name);**

**return name;**

**}**

**public void setName(String name) {**

**LOGGER.debug("Setting name: {}", name);**

**this.name = name;**

**}**

**@Override**

**public String toString() {**

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

**}**

**}**

**SpringLearnApplication.java**

**package com.cognizant.spring\_learn;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.context.ApplicationContext;**

**import org.springframework.context.support.ClassPathXmlApplicationContext;**

**public class SpringLearnApplication {**

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

**public static void main(String[] args) {**

**SpringLearnApplication app = new SpringLearnApplication();**

**app.displayCountry();**

**}**

**public void displayCountry() {**

**LOGGER.info("START");**

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

**Country country = context.getBean("country", Country.class);**

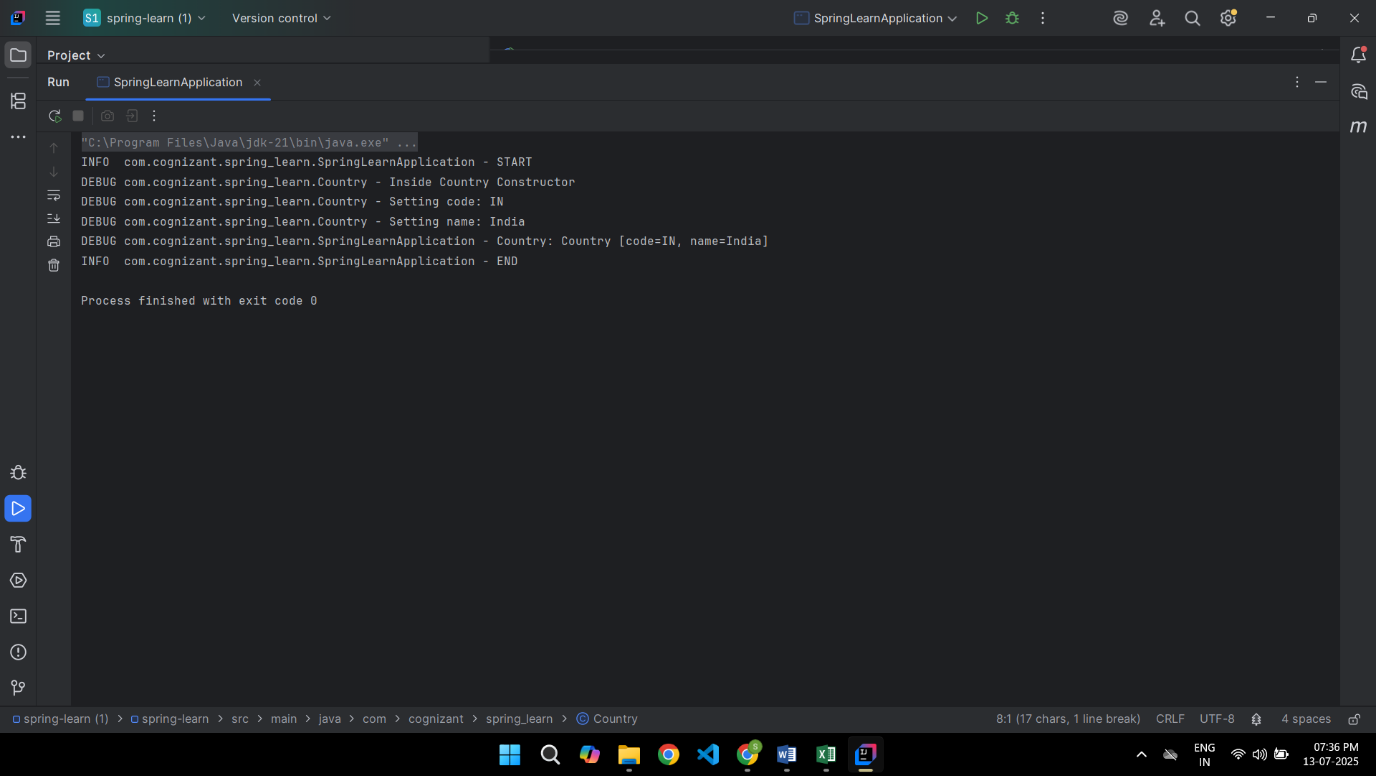
**LOGGER.debug("Country : {}", country.toString());**

**LOGGER.info("END");**

**}**

**}**

**OUTPUT:**

****

**Exercise 3: REST –GET Country Based on Country Code**

**CountryController.java**

**package com.cognizant.spring\_learn.Controller;**

**import com.cognizant.spring\_learn.model.Country;**

**import com.cognizant.spring\_learn.service.exception.CountryService;**

**import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;**

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

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

**@RestController**

**public class CountryController {**

**@Autowired**

**private CountryService countryService;**

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

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

**return countryService.getCountry(code);**

**}**

**}**

**country.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<beans xmlns="http://www.springframework.org/schema/beans"**

**xmlns:util="http://www.springframework.org/schema/util"**

**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**

**http://www.springframework.org/schema/util**

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

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

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

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

**</bean>**

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

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

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

**</bean>**

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

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

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

**</bean>**

**<util:list id="countryList">**

**<ref bean="in"/>**

**<ref bean="us"/>**

**<ref bean="jp"/>**

**</util:list>**

**</beans>**

**CountryNotFoundException.java**

**package com.cognizant.spring\_learn.service.exception;**

**import org.springframework.http.HttpStatus;**

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

**@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Country not found")**

**public class CountryNotFoundException extends Exception {**

**public CountryNotFoundException(String message) {**

**super(message);**

**}**

**}**

**CountryService.java**

**package com.cognizant.spring\_learn.service.exception;**

**import com.cognizant.spring\_learn.model.Country;**

**import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;**

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

**import org.springframework.stereotype.Service;**

**import java.util.List;**

**@Service**

**public class CountryService {**

**@Autowired**

**private List<Country> countryList;**

**public Country getCountry(String code) throws CountryNotFoundException {**

**return countryList.stream()**

**.filter(c -> c.getCode().equalsIgnoreCase(code))**

**.findFirst()**

**.orElseThrow(() -> new CountryNotFoundException("Country not found"));**

**}**

**}**

**Country.java**

**package com.cognizant.spring\_learn.model;**

**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;**

**}**

**}**

**SpringLearnApplication.java**

**package com.cognizant.spring\_learn;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.context.annotation.ImportResource;**

**@SpringBootApplication**

**@ImportResource("classpath:country.xml")**

**public class SpringLearnApplication {**

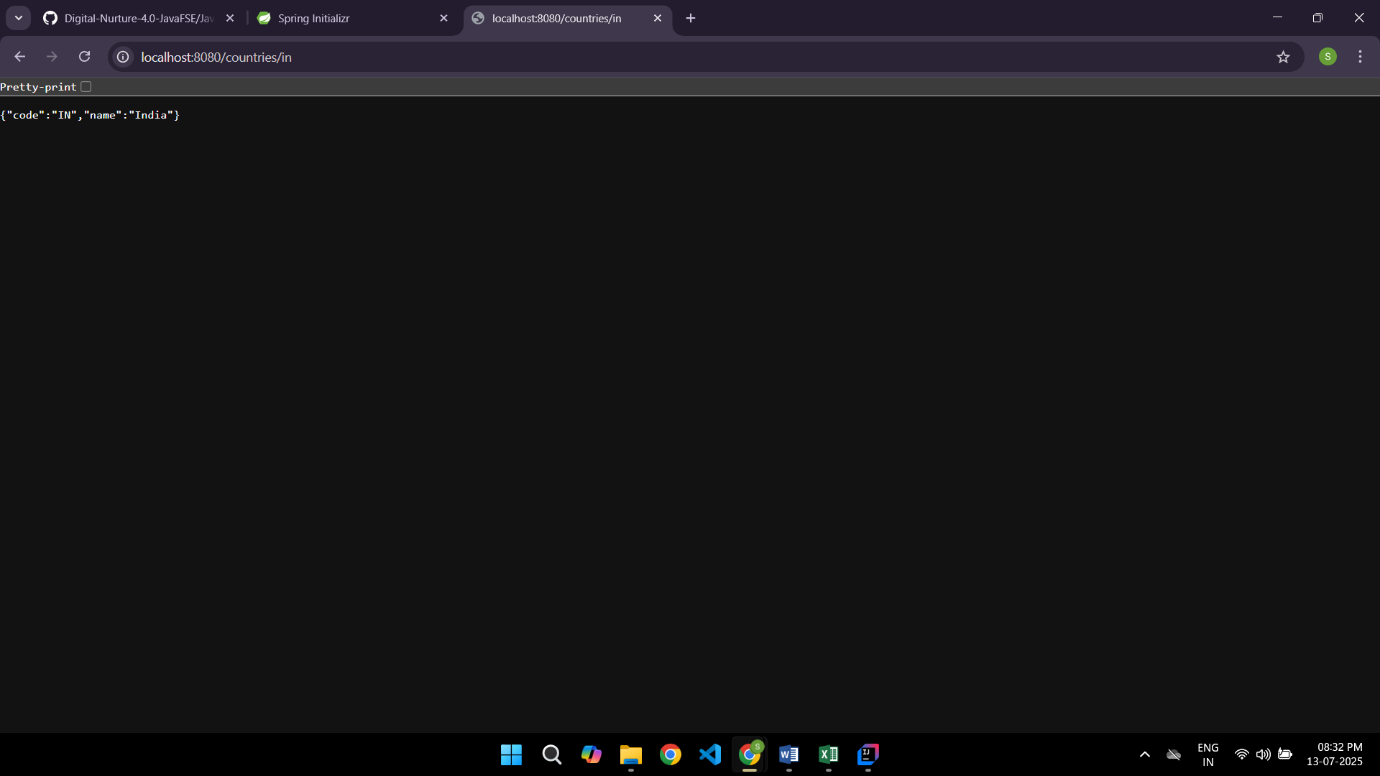
**public static void main(String[] args) {**

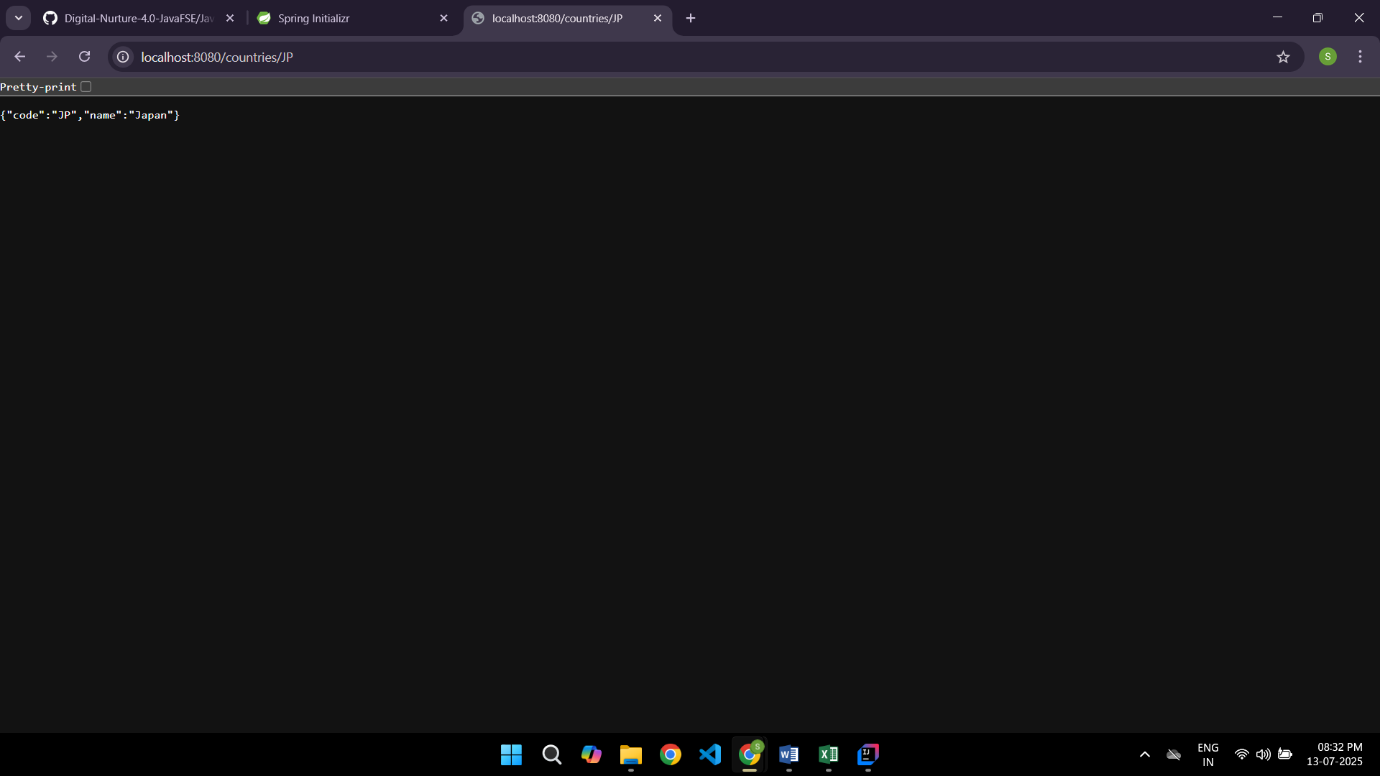
**SpringApplication.run(SpringLearnApplication.class, args);**

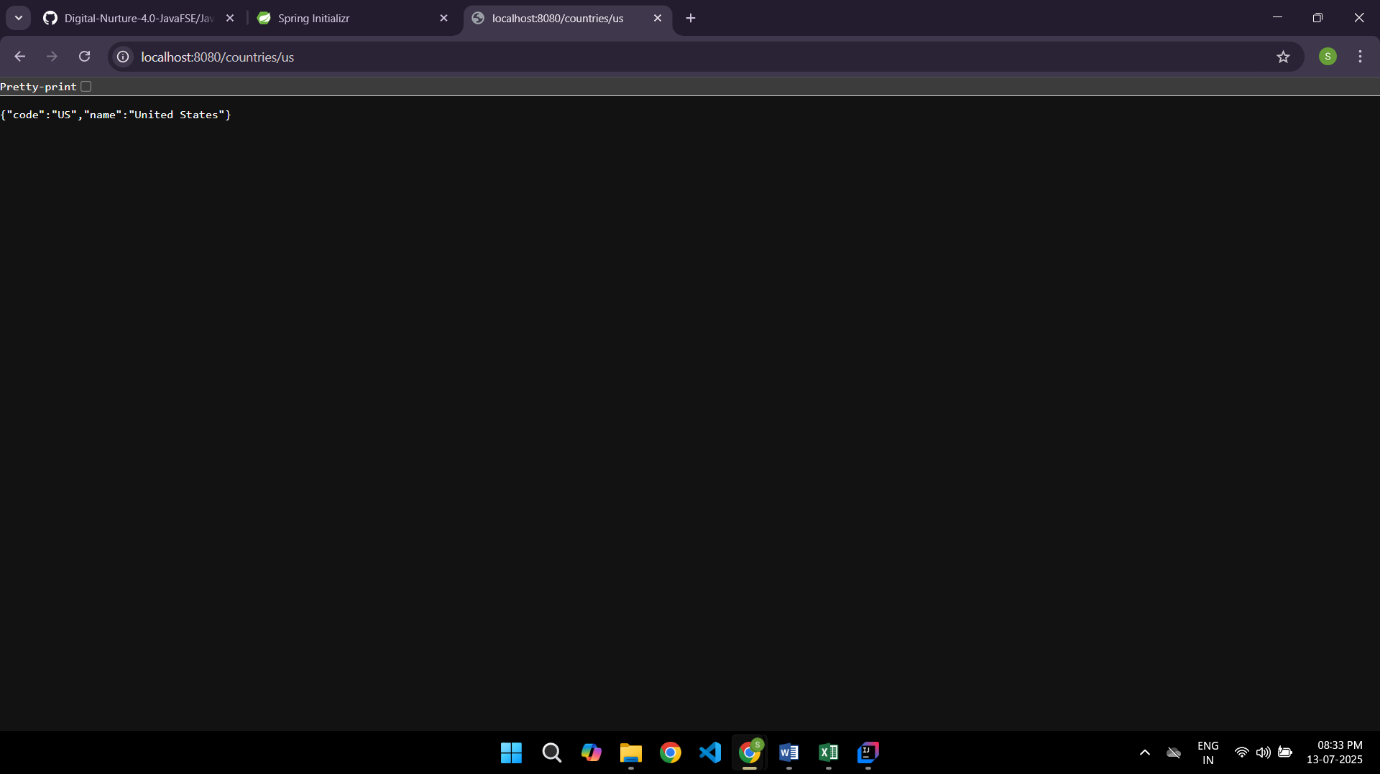
**}**

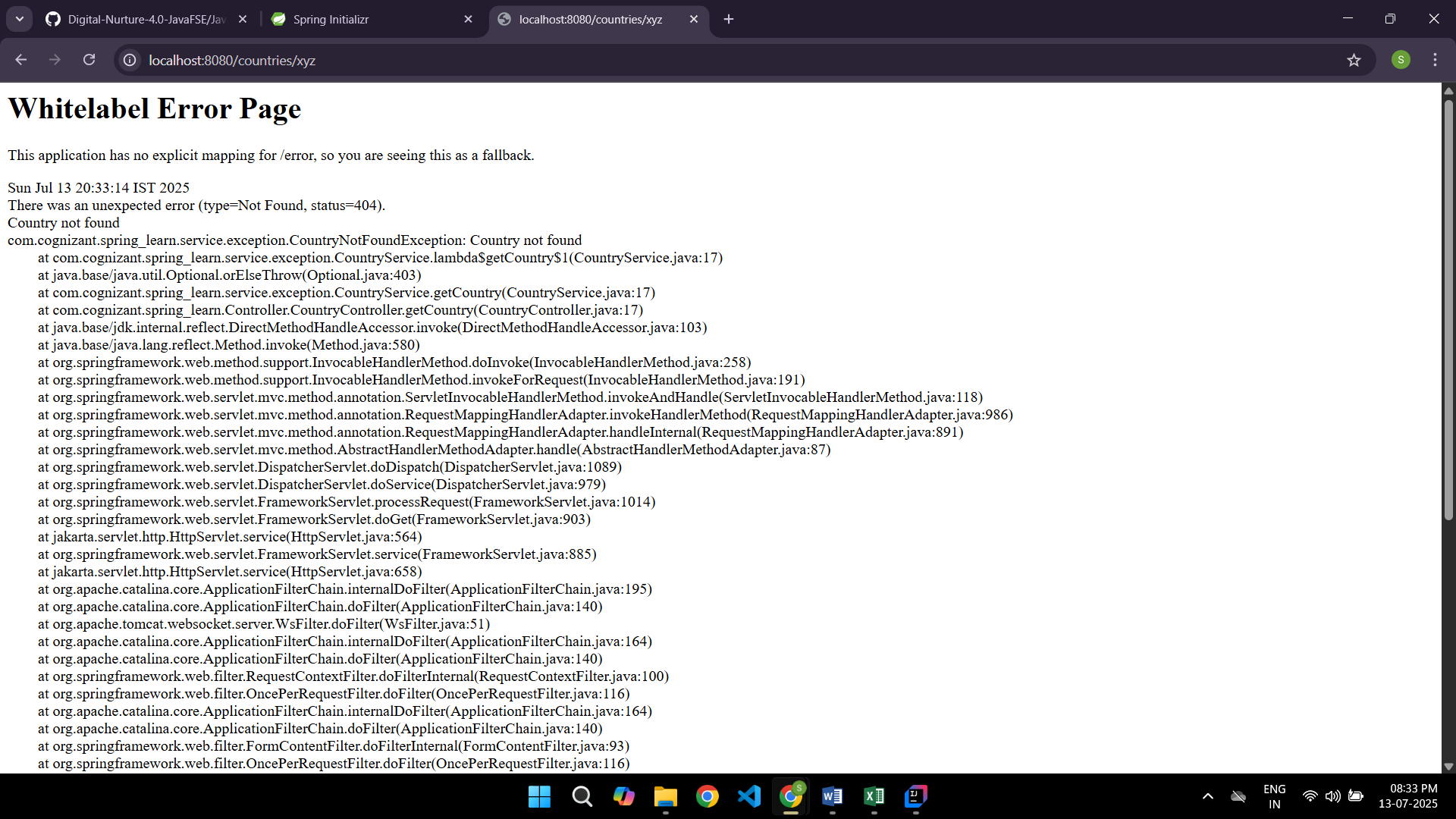
**}**

**OUTPUT:**

****

****

****

****

**Create Authentication Service that returns JWT**

**CountryConfig.java:**

**package com.cognizant.spring\_learn.config;**

**import com.cognizant.spring\_learn.model.Country;**

**import org.springframework.context.annotation.Bean;**

**import org.springframework.context.annotation.Configuration;**

**import java.util.Arrays;**

**import java.util.List;**

**@Configuration**

**public class CountryConfig {**

**@Bean**

**public List<Country> countryList() {**

**return Arrays.asList(**

**new Country("IN", "India"),**

**new Country("US", "United States"),**

**new Country("JP", "Japan")**

**);**

**}**

**}**

**SecurityConfig.java**

**package com.cognizant.spring\_learn.security;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.context.annotation.Bean;**

**import org.springframework.context.annotation.Configuration;**

**import org.springframework.security.authentication.AuthenticationManager;**

**import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;**

**import org.springframework.security.config.annotation.web.builders.HttpSecurity;**

**import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;**

**import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;**

**import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;**

**import org.springframework.security.crypto.password.PasswordEncoder;**

**@Configuration**

**@EnableWebSecurity**

**public class SecurityConfig extends WebSecurityConfigurerAdapter {**

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

**@Override**

**protected void configure(AuthenticationManagerBuilder auth) throws Exception {**

**auth.inMemoryAuthentication()**

**.withUser("admin").password(passwordEncoder().encode("pwd")).roles("ADMIN")**

**.and()**

**.withUser("user").password(passwordEncoder().encode("pwd")).roles("USER");**

**}**

**@Bean**

**public PasswordEncoder passwordEncoder() {**

**return new BCryptPasswordEncoder();**

**}**

**@Override**

**protected void configure(HttpSecurity httpSecurity) throws Exception {**

**httpSecurity.csrf().disable().httpBasic().and()**

**.authorizeRequests()**

**.antMatchers("/authenticate").hasAnyRole("USER", "ADMIN")**

**.anyRequest().authenticated()**

**.and()**

**.addFilter(new JwtAuthorizationFilter(authenticationManager()));**

**}**

**@Override**

**@Bean**

**public AuthenticationManager authenticationManagerBean() throws Exception {**

**return super.authenticationManagerBean();**

**}**

**}**

**JwtAuthorizationFilter.java**

**package com.cognizant.spring\_learn.security;**

**import java.io.IOException;**

**import java.util.ArrayList;**

**import javax.servlet.FilterChain;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.security.authentication.AuthenticationManager;**

**import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;**

**import org.springframework.security.core.context.SecurityContextHolder;**

**import org.springframework.security.web.authentication.www.BasicAuthenticationFilter;**

**import io.jsonwebtoken.Claims;**

**import io.jsonwebtoken.Jws;**

**import io.jsonwebtoken.JwtException;**

**import io.jsonwebtoken.Jwts;**

**public class JwtAuthorizationFilter extends BasicAuthenticationFilter {**

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

**public JwtAuthorizationFilter(AuthenticationManager authManager) {**

**super(authManager);**

**}**

**@Override**

**protected void doFilterInternal(HttpServletRequest req, HttpServletResponse res,**

**FilterChain chain) throws IOException, ServletException {**

**LOGGER.info("Start JWT filter");**

**String header = req.getHeader("Authorization");**

**LOGGER.debug("JWT Header: {}", header);**

**if (header == null || !header.startsWith("Bearer ")) {**

**chain.doFilter(req, res);**

**return;**

**}**

**UsernamePasswordAuthenticationToken auth = getAuthentication(req);**

**SecurityContextHolder.getContext().setAuthentication(auth);**

**chain.doFilter(req, res);**

**}**

**private UsernamePasswordAuthenticationToken getAuthentication(HttpServletRequest request) {**

**String token = request.getHeader("Authorization");**

**if (token != null) {**

**try {**

**Jws<Claims> jws = Jwts.parser()**

**.setSigningKey("secretkey")**

**.parseClaimsJws(token.replace("Bearer ", ""));**

**String user = jws.getBody().getSubject();**

**if (user != null) {**

**return new UsernamePasswordAuthenticationToken(user, null, new ArrayList<>());**

**}**

**} catch (JwtException e) {**

**LOGGER.error("Invalid token", e);**

**}**

**}**

**return null;**

**}**

**}**

**pom.xml**

**<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">**

**<modelVersion>4.0.0</modelVersion>**

**<groupId>com.cognizant</groupId>**

**<artifactId>spring-learn</artifactId>**

**<version>1.0.0</version>**

**<packaging>jar</packaging>**

**<name>spring-learn</name>**

**<parent>**

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

**<artifactId>spring-boot-starter-parent</artifactId>**

**<version>2.7.18</version>**

**<relativePath/>**

**</parent>**

**<properties>**

**<java.version>1.8</java.version> <!-- or 11 if you're using Java 11 -->**

**</properties>**

**<dependencies>**

**<dependency>**

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

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

**</dependency>**

**<dependency>**

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

**<artifactId>spring-boot-starter-security</artifactId>**

**</dependency>**

**<dependency>**

**<groupId>io.jsonwebtoken</groupId>**

**<artifactId>jjwt</artifactId>**

**<version>0.9.0</version>**

**</dependency>**

**<dependency>**

**<groupId>javax.servlet</groupId>**

**<artifactId>javax.servlet-api</artifactId>**

**<version>4.0.1</version>**

**<scope>provided</scope>**

**</dependency>**

**<dependency>**

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

**<artifactId>spring-boot-starter-logging</artifactId>**

**</dependency>**

**<dependency>**

**<groupId>org.projectlombok</groupId>**

**<artifactId>lombok</artifactId>**

**<optional>true</optional>**

**</dependency>**

**<dependency>**

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

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

**<scope>runtime</scope>**

**</dependency>**

**<dependency>**

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

**<artifactId>spring-boot-starter-test</artifactId>**

**<scope>test</scope>**

**</dependency>**

**<dependency>**

**<groupId>javax.xml.bind</groupId>**

**<artifactId>jaxb-api</artifactId>**

**<version>2.3.1</version>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

**<plugin>**

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

**<artifactId>spring-boot-maven-plugin</artifactId>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

**Country.java**

**package com.cognizant.spring\_learn.model;**

**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;**

**}**

**}**

**CountryController.java**

**package com.cognizant.spring\_learn.Controller;**

**import com.cognizant.spring\_learn.model.Country;**

**import com.cognizant.spring\_learn.service.exception.CountryService;**

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

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

**import java.util.List;**

**@RestController**

**public class CountryController {**

**@Autowired**

**private CountryService countryService;**

**@GetMapping("/countries")**

**public List<Country> getAllCountries() {**

**return countryService.getAllCountries();**

**}**

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

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

**return countryService.getCountry(code);**

**}**

**}**

**CountryService.java**

**package com.cognizant.spring\_learn.service.exception;**

**import com.cognizant.spring\_learn.model.Country;**

**import org.springframework.stereotype.Service;**

**import java.util.Arrays;**

**import java.util.List;**

**@Service**

**public class CountryService {**

**public List<Country> getAllCountries() {**

**return Arrays.asList(**

**new Country("IN", "India"),**

**new Country("US", "United States"),**

**new Country("JP", "Japan")**

**);**

**}**

**public Country getCountry(String code) throws Exception {**

**return getAllCountries().stream()**

**.filter(c -> c.getCode().equalsIgnoreCase(code))**

**.findFirst()**

**.orElseThrow(() -> new Exception("Country not found: " + code));**

**}**

**}**

**AuthenticationController.java**

**package com.cognizant.spring\_learn.Controller;**

**import java.util.Base64;**

**import java.util.Date;**

**import java.util.HashMap;**

**import java.util.Map;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

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

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

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

**import io.jsonwebtoken.JwtBuilder;**

**import io.jsonwebtoken.Jwts;**

**import io.jsonwebtoken.SignatureAlgorithm;**

**@RestController**

**public class AuthenticationController {**

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

**@GetMapping("/authenticate")**

**public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {**

**LOGGER.info("Start authentication");**

**LOGGER.debug("Header: {}", authHeader);**

**String user = getUser(authHeader);**

**String token = generateJwt(user);**

**Map<String, String> map = new HashMap<>();**

**map.put("token", token);**

**LOGGER.info("End authentication");**

**return map;**

**}**

**private String getUser(String authHeader) {**

**String encoded = authHeader.substring("Basic ".length());**

**byte[] decodedBytes = Base64.getDecoder().decode(encoded);**

**String decoded = new String(decodedBytes);**

**return decoded.split(":")[0];**

**}**

**private String generateJwt(String user) {**

**JwtBuilder builder = Jwts.builder();**

**builder.setSubject(user);**

**builder.setIssuedAt(new Date());**

**builder.setExpiration(new Date(System.currentTimeMillis() + 1200000)); // 20 minutes**

**builder.signWith(SignatureAlgorithm.HS256, "secretkey");**

**return builder.compact();**

**}**

**}**

**SpringLearnApplication.java**

**package com.cognizant.spring\_learn;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**@SpringBootApplication**

**public class SpringLearnApplication {**

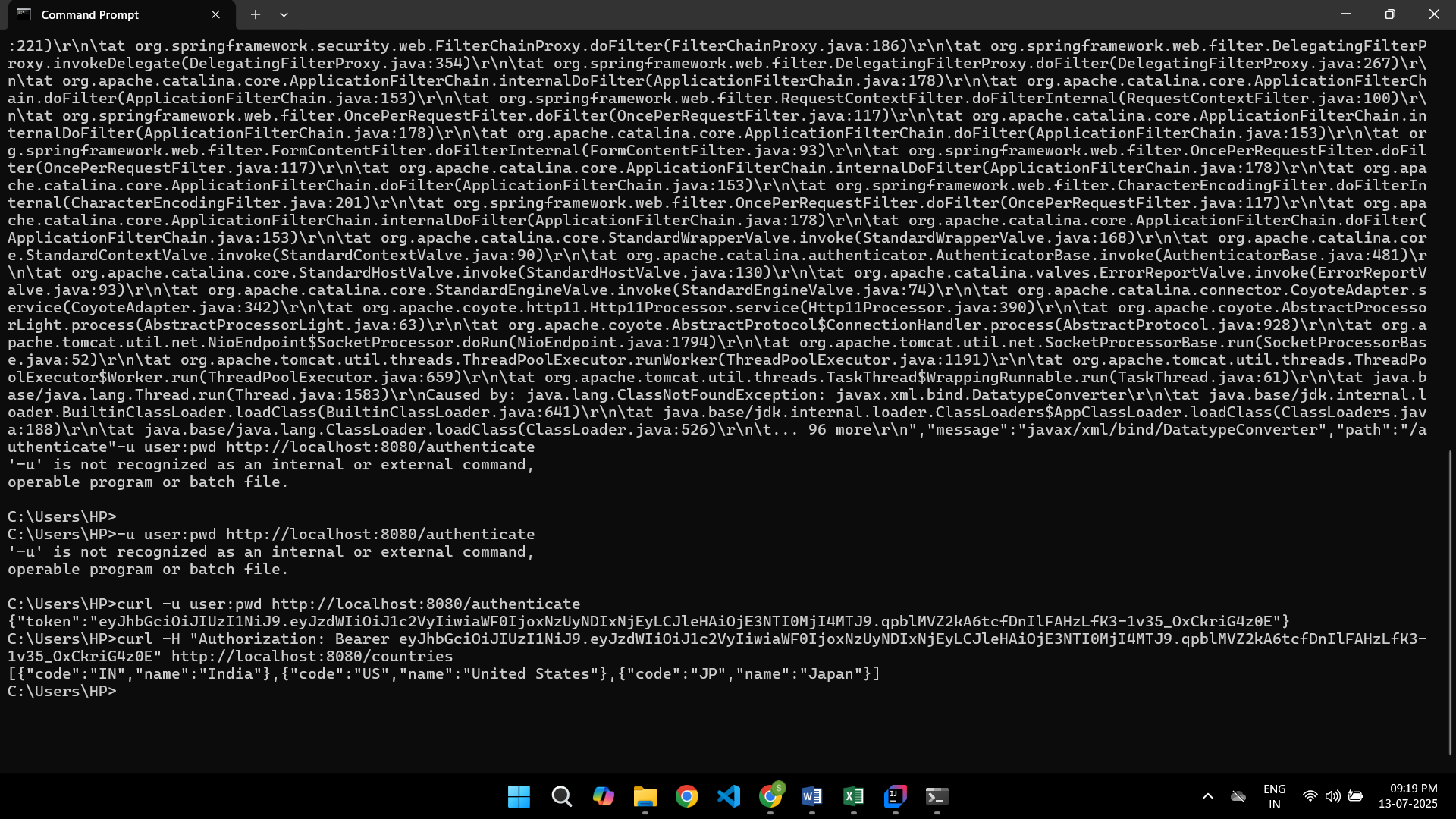
**public static void main(String[] args) {**

**SpringApplication.run(SpringLearnApplication.class, args);**

**}**

**}**

**OUTPUT:**

****