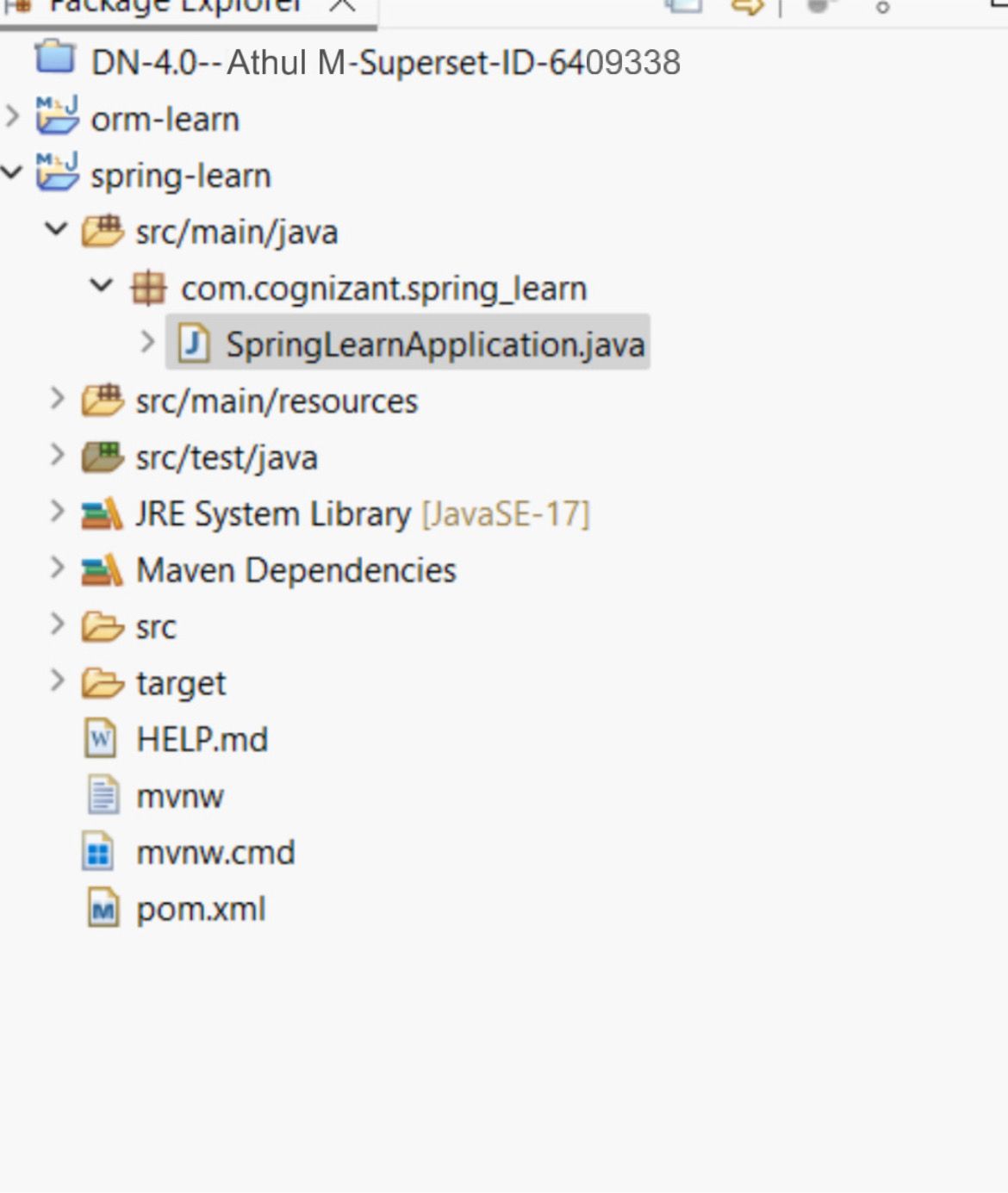
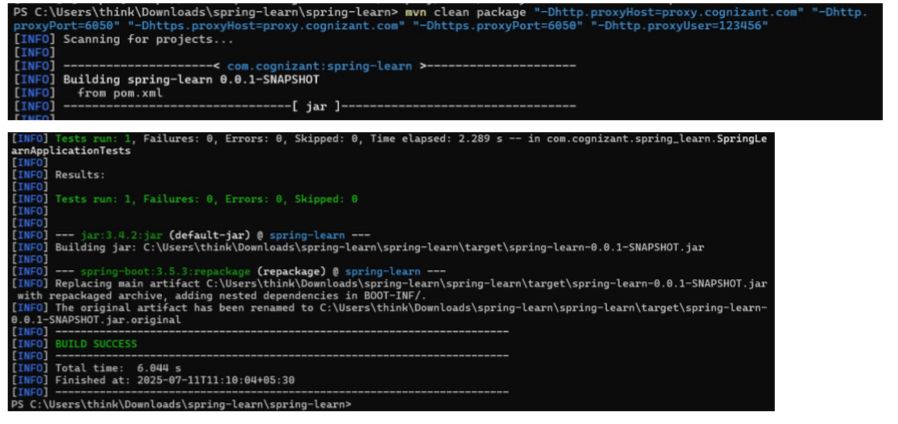
**1.NAME OF THE QUESTION**

Create a Spring Web Project using Maven

****

**Proxy build**



**CODE**

**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) {

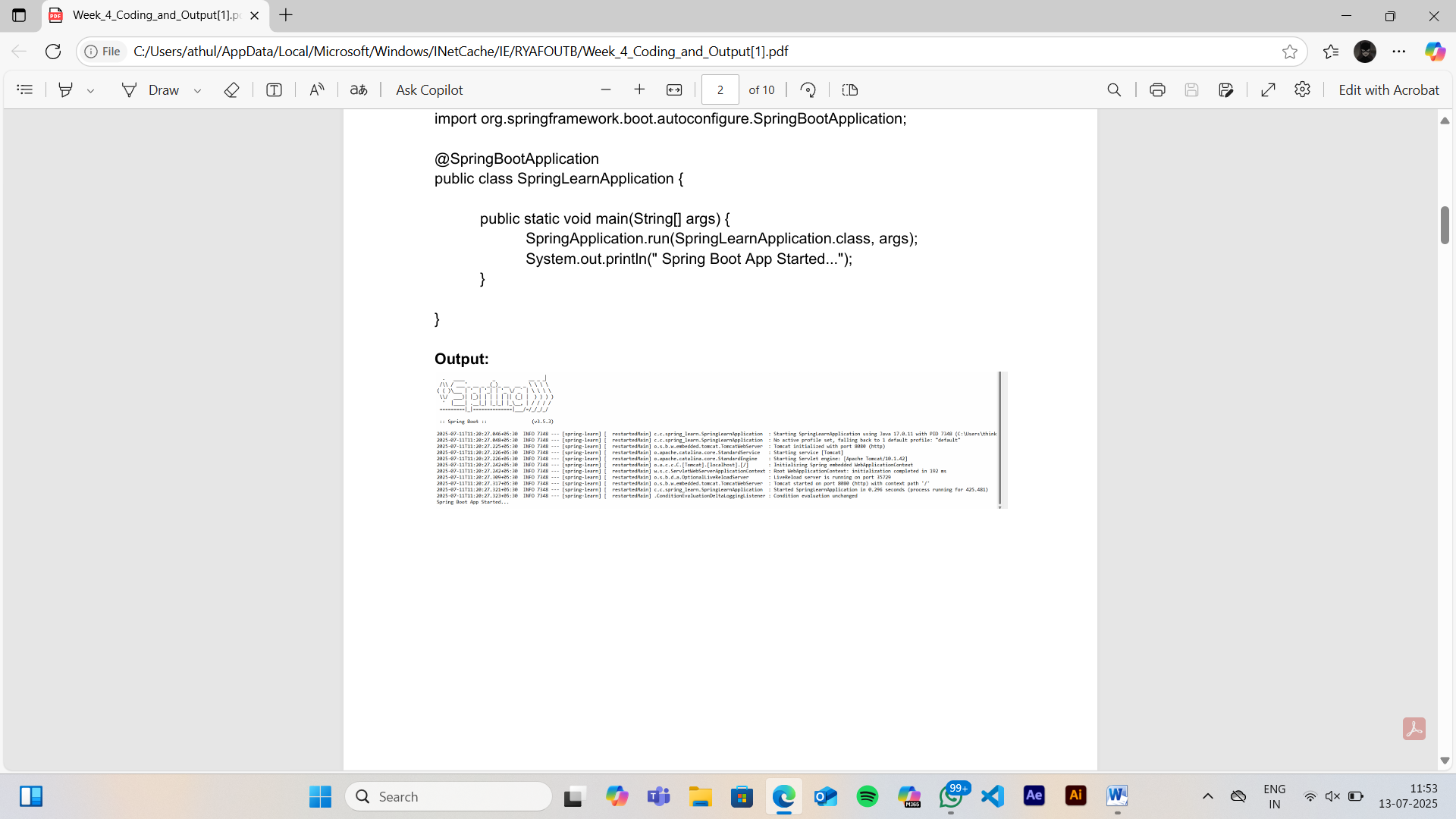
SpringApplication.run(SpringLearnApplication.class, args);

System.out.println(" Spring Boot App Started...");

}

}

**OUTPUT**



**2.NAME OF THE QUESTION**

Spring Core – Load Country from Spring Configuration XML

**CODE**

**SpringLearnApllication.java:**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

displayCountry();

LOGGER.info("END");

}

public static void displayCountry() {

try (ClassPathXmlApplicationContext context = new

ClassPathXmlApplicationContext("country.xml")) {

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

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

}

}

}

**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("Inside getCode()");

return code;

}

public void setCode(String code) {

LOGGER.debug("Inside setCode()");

this.code = code;

}

public String getName() {

LOGGER.debug("Inside getName()");

return name;

}

public void setName(String name) {

LOGGER.debug("Inside setName()");

this.name = name;

}

@Override

public String toString() {

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

}

}

**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">

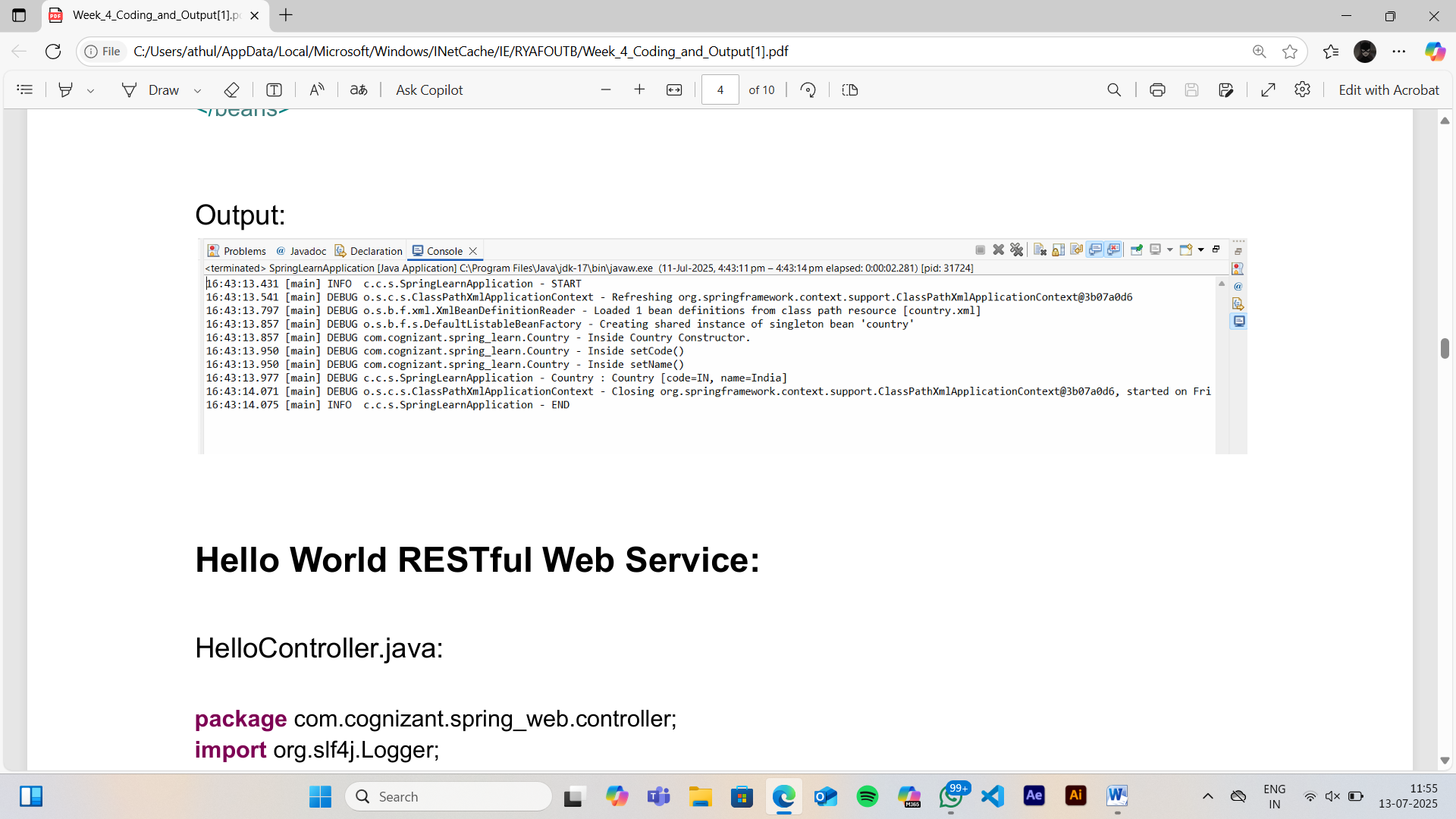
<property name="code" value="IN"/>

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

</bean>

</beans>

**OUTPUT**

****

**3.NAME OF THE QUESTION**

Hello World RESTful Web Service

**CODE**

**HelloController.java:**

package com.cognizant.spring\_web.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!!";

}

}

**SpringWebApplication.java:**

package com.cognizant.spring\_web;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringWebApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringWebApplication.class, args);

LOGGER.info("END");

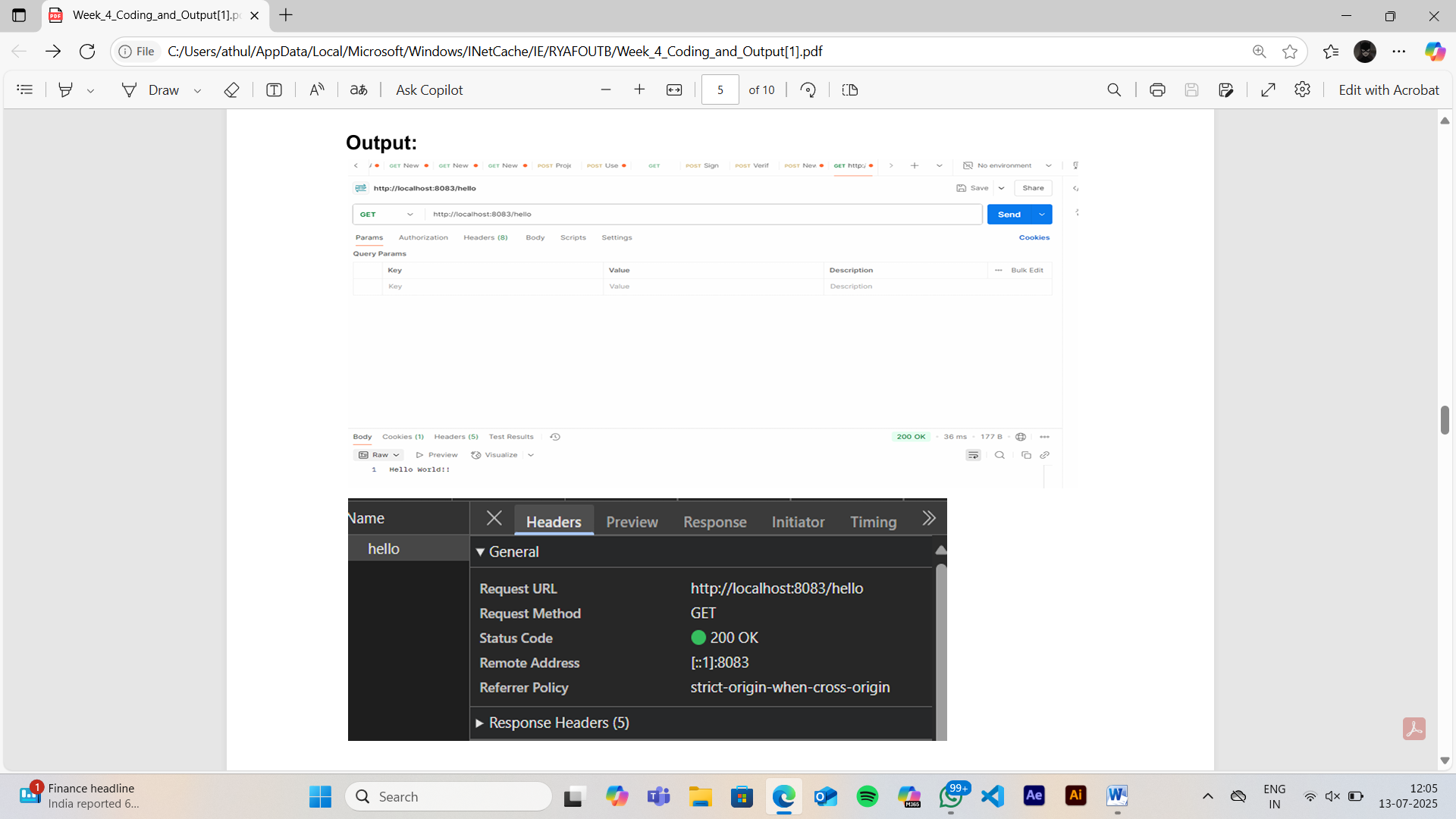
}

}

Port:

8083

**OUTPUT**

****

**4.NAME OF THE QUESTION**

REST - country web service

**CODE**

**Country.java:**

package com.cognizant.spring\_web.model;

public class Country {

private String code;

private String name;

public Country() {}

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\_web.controller;

import com.cognizant.spring\_web.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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");

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

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

context.close();

LOGGER.info("END");

return country;

}

}

**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.spring\_web.model.Country">

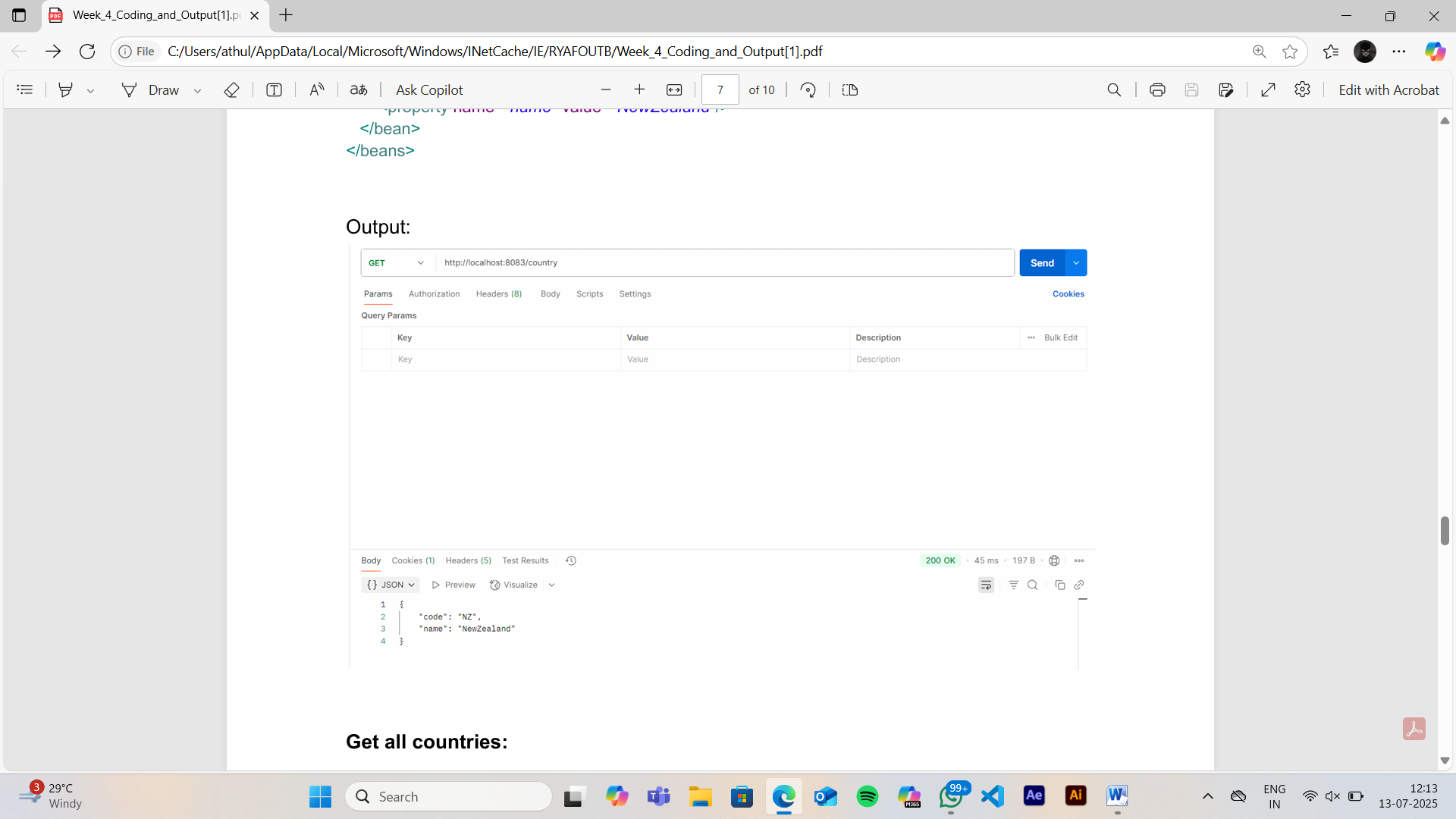
<property name="code" value="NZ"/>

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

</bean>

</beans>

**OUTPUT**



**5.NAME OF THE QUESTION**

REST – Get country based on country code

**CODE**

**country.xml:**

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

<constructor-arg>

<list>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

</list>

</constructor-arg>

</bean>

**Endpoint:**

@GetMapping("/countries")

public List<Country> getAllCountries() {

LOGGER.info("START");

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

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

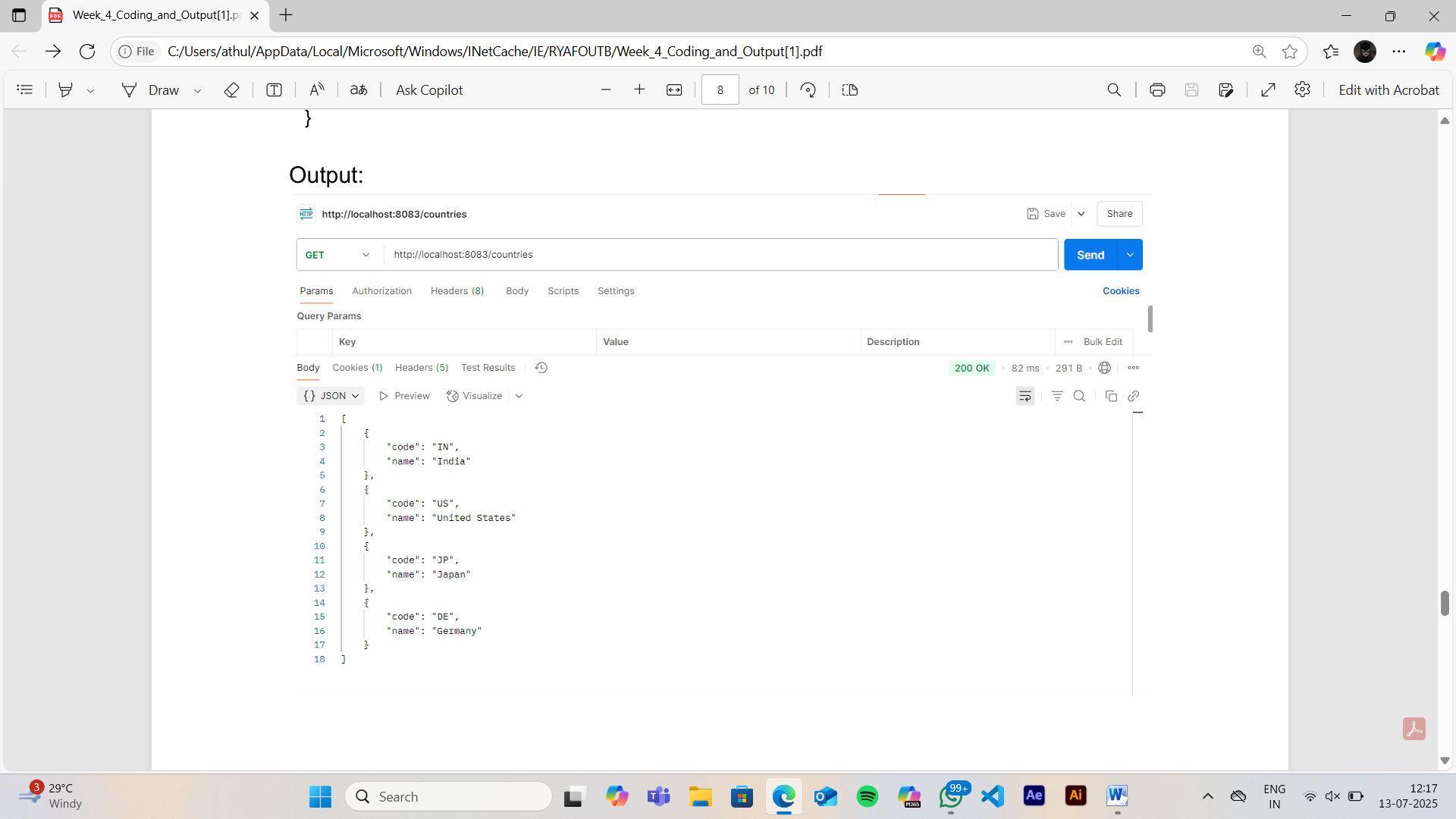
context.close();

LOGGER.info("END");

return countryList;

}

**OUTPUT**



**6.NAME OF THE QUESTION**

Create authentication service that returns JWT

**CODE**

**AuthController.java:**

package com.cognizant.jwt\_auth.controller;

import com.cognizant.jwt\_auth.util.JwtUtil;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import java.util.Base64;

import java.util.Collections;

@RestController

public class AuthController {

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(

@RequestHeader(value = "Authorization", required = false) String authHeader) {

if (authHeader == null) {

return ResponseEntity.status(HttpStatus.BAD\_REQUEST)

.body("Missing Authorization header");

}

try {

String[] credentials = extractAndDecodeHeader(authHeader);

String username = credentials[0];

String password = credentials[1];

if (isValidUser(username, password)) {

String token = JwtUtil.generateToken(username);

return ResponseEntity.ok(Collections.singletonMap("token", token));

}

return ResponseEntity.status(HttpStatus.UNAUTHORIZED)

.body("Invalid credentials");

} catch (Exception e) {

return ResponseEntity.status(HttpStatus.BAD\_REQUEST)

.body("Error" + e.getMessage());

}

}

private boolean isValidUser(String username, String password) {

return "Mugilan".equals(username) && "Hero".equals(password);

}

private String[] extractAndDecodeHeader(String header) {

if (!header.startsWith("Basic ")) {

throw new RuntimeException("Invalid Authorization");

}

byte[] base64Token = Base64.getDecoder().decode(header.substring(6));

String token = new String(base64Token);

String[] parts = token.split(":", 2);

if (parts.length != 2) {

throw new RuntimeException("Invalid format");

}

return parts;

}

}

**Jwtutil.java:**

package com.cognizant.jwt\_auth.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import java.security.Key;

import java.util.Date;

public class JwtUtil {

private static final String SECRET = "my-key";

private static final Key KEY = Keys.hmacShaKeyFor(SECRET.getBytes());

private static final long EXPIRY\_MS = 60 \* 60 \* 1000;

private JwtUtil() {}

public static String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + EXPIRY\_MS))

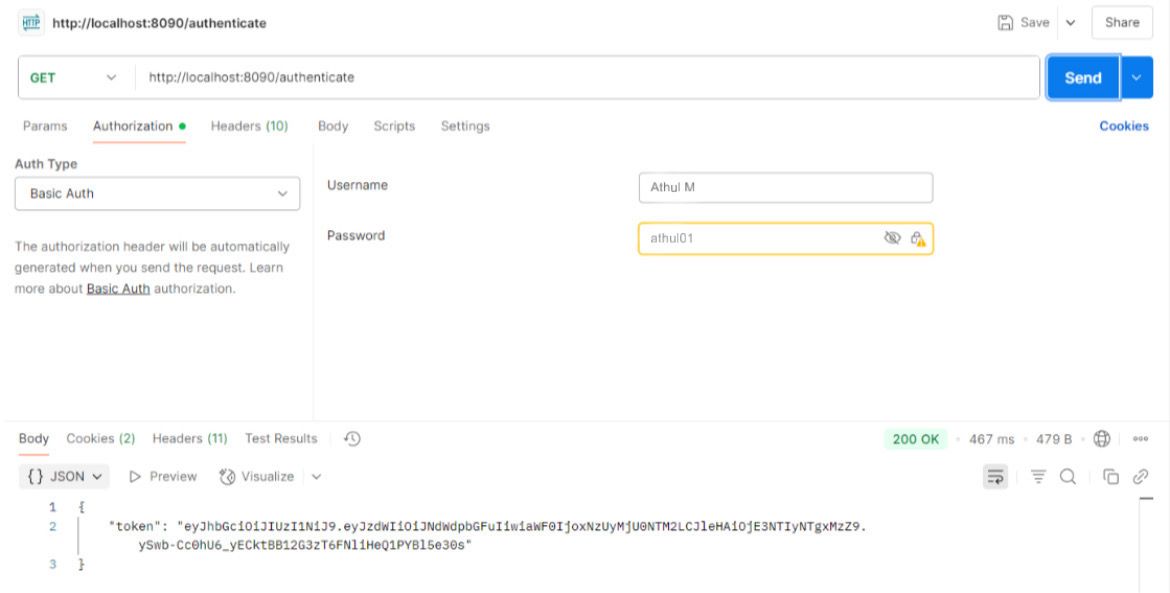
.signWith(KEY, SignatureAlgorithm.HS256)

.compact();

}

}

**OUTPUT**

****