Week 4

EXERCISE 1: Create a Spring Web Project using Maven

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);
	}
}
```

SpringLearnApplicationTest.java

```
package com.cognizant.spring_learn;
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
@SpringBootTest
class SpringLearnApplicationTests {
   @Test
   void contextLoads() {
      System.out.println("SpringLearnApplication started successfully.");
   }
}
```

Digital Nuture Program

Week 4

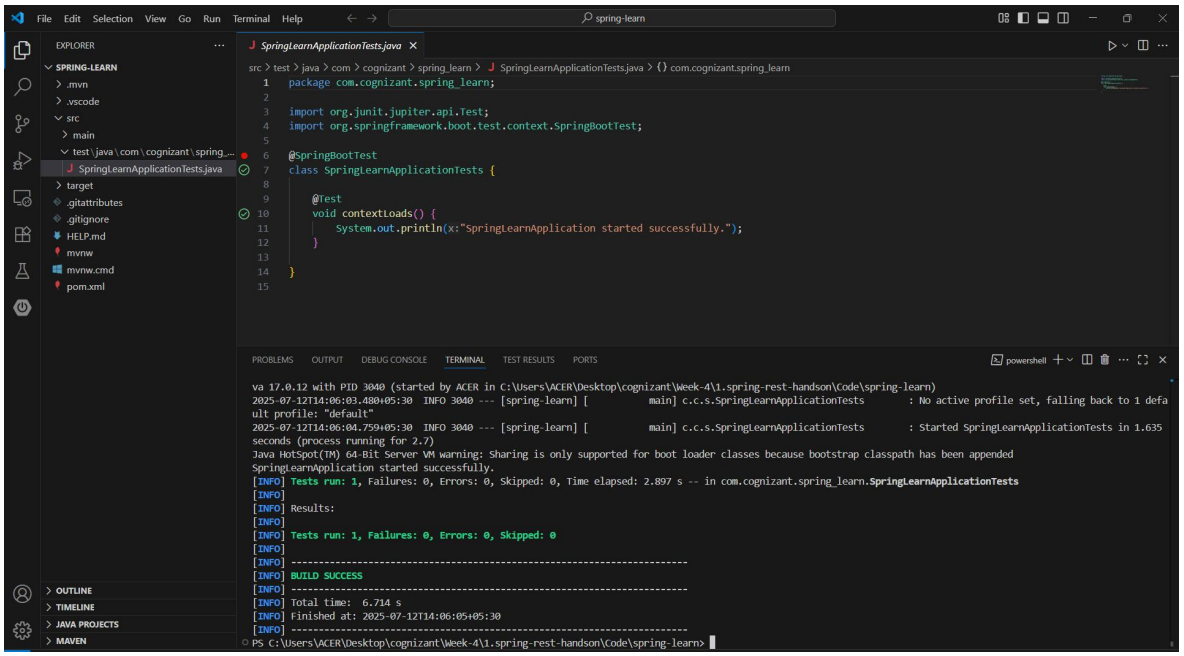
Pom.xl

```
<?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>
```

Output:



EXERCISE 2 : SPRING CORE LOAD COUNTRY FROM SPRING CONFIGURATION

Country. java

```
package com.cognizant.spring_learn.model;
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 + "]";
    }
}
```

SpringLearnApplication.java

```
package com.cognizant.spring_learn;

import com.cognizant.spring_learn.model.Country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication
public class SpringLearnApplication {

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

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
        displayCountry();
    }

    public static void displayCountry() {
        LOGGER.debug("START displayCountry()");

        try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("country.xml")) {
            Country country = context.getBean("country", Country.class);
```

```
        LOGGER.debug("Country : {}", country);
    }
    LOGGER.debug("END displayCountry()");
  }
}
```

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.model.Country">
    <property name="code" value="IN" />
    <property name="name" value="India" />
  </bean>
</beans>
```

SpringLearnApplicationTest.java

```
package com.cognizant.spring_learn;
import com.cognizant.spring_learn.model.Country;
import org.junit.jupiter.api.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import static org.junit.jupiter.api.Assertions.assertEquals;
public class SpringLearnApplicationTests {
  @Test
  void testCountryBean() {
    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
```

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


        assertEquals("IN", country.getCode());
        assertEquals("India", country.getName());
        ((ClassPathXmlApplicationContext) context).close();
    }
}
```

EXERCISE 3: HELLO WORLD RESTFUL WEB SERVICE

HelloController.java

```
package com.cognizant.springlearn.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 of sayHello() method");
        String response = "Hello World!!";
        LOGGER.info("End of sayHello() method");
        return response;
    }
}
```

Digital Nuture Program

Week 4

SpringApplication.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);

    }

}
```

Application.properties

```
 Server port configuration

server.port=8083


# Logging configuration

logging.level.com.cognizant.springlearn=DEBUG
```

OUTPUT:

Digital Nuture Program

Week 4

EXERCISE 4: **REST - COUNTRY WEB SERVICE**

CountryController.java

```
package com.cognizant.springlearn.controller;

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

import com.cognizant.springlearn.model.Country;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@RestController
@RequestMapping("/country")
public class CountryController {

    @GetMapping
    public Country getCountryIndia() {
        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");
        Country country = (Country) context.getBean("in");
        context.close();
        return country;
    }
}
```

Country.java

```
package com.cognizant.springlearn.model;

public class Country {
    private String code;
    private String name;
```

```
    // Constructors
    public Country() {
    }

    public Country(String code, String name) {
        this.code = code;
        this.name = 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;
    }

    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}
```

Digital Nuture Program

Week 4

SpringApplication.java

```
package com.cognizant.springlearn;


import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;


@SpringBootApplication
@ImportResource({"classpath:beans.xml"})
public class SpringLearnApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

Beans.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.springlearn.model.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>
</beans>
```

Week 4

OUTPUT:

EXERCISE 5:REST - GET COUNTRY BASED ON COUNTRY CODE

CountryController.java

```
package com.cognizant.springlearn.controller;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.service.CountryService;
@RestController
@RequestMapping("/countries")
public class CountryController {
    @Autowired
    private CountryService countryService;
    @GetMapping("/{code}")
    public Country getCountry(@PathVariable String code) {
        return countryService.getCountry(code);
    }
}
```

Digital Nuture Program

Week 4

Country.java

```
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; }
}
```

CountryService.java

```
package com.cognizant.springlearn.service;
import java.util.List;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Service;
import com.cognizant.springlearn.model.Country;
@Service
public class CountryService {
    public Country getCountry(String code) {
        ClassPathXmlApplicationContext context =
            new ClassPathXmlApplicationContext("country.xml");
        List<Country> countries =
            (List<Country>) context.getBean("countryList");
        context.close();
```

```
        return countries.stream()
                .filter(c -> c.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElseThrow(() -> new RuntimeException("Country not found"));
    }
}
```

SpringApplication.java

```
package com.cognizant.springlearn;
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);
    }
}
```

country.xml

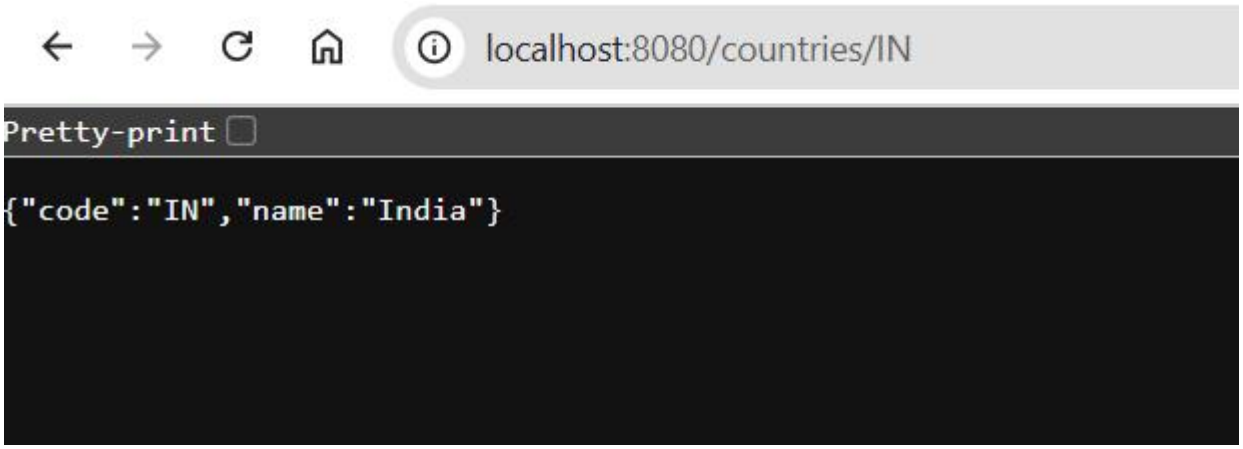
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
       http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="in" class="com.cognizant.springlearn.model.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>
```

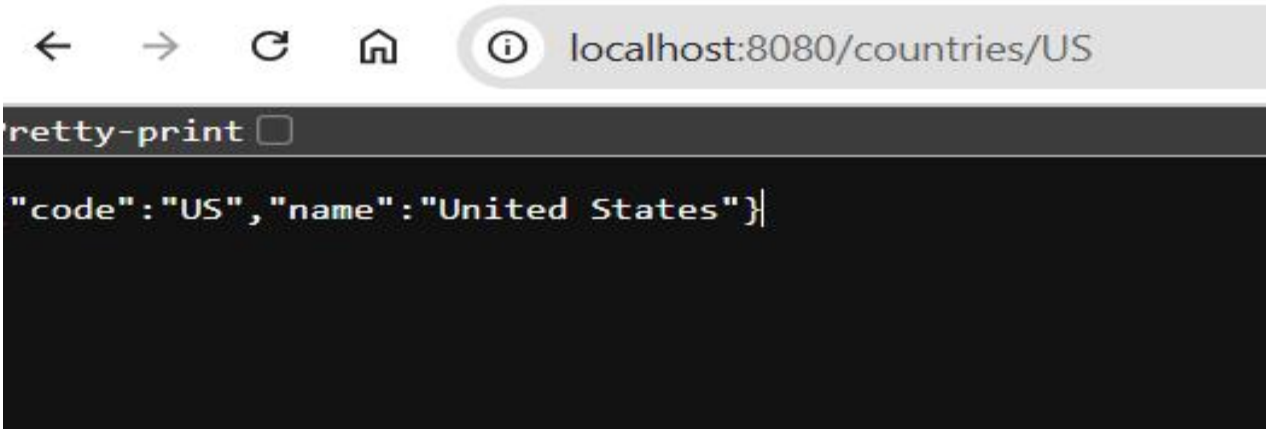
Week 4

```
<bean id="us" class="com.cognizant.springlearn.model.Country">
    <property name="code" value="US"/>
    <property name="name" value="United States"/>
</bean>
<bean id="countryList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="in"/>
            <ref bean="us"/>
            <!-- Add more countries as needed -->
        </list>
    </constructor-arg>
</bean>
```

OUTPUT:



Week 4



EXERCISE 6: CREATE AUTHENTICATION SERVICE THAT RETURNS JWT

AuthenticationController.java

```
package com.cognizant.springlearn.controller;
import com.cognizant.springlearn.config.JwtTokenUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.web.bind.annotation.GetMapping;


import org.springframework.web.bind.annotation.RestController;
import javax.servlet.http.HttpServletRequest;
import java.util.Base64;
import java.util.HashMap;
import java.util.Map;
@RestController
public class AuthenticationController {
    @Autowired
    private JwtTokenUtil jwtTokenUtil;
    @Autowired
    private InMemoryUserDetailsManager inMemoryUserDetailsManager;
    @Autowired
```

```
    private PasswordEncoder passwordEncoder;
    @GetMapping("/authenticate")
    public ResponseEntity<?> createAuthenticationToken(HttpServletRequest request) {
        final String authorization = request.getHeader("Authorization");
        if (authorization != null && authorization.startsWith("Basic ")) {
            String base64Credentials = authorization.substring("Basic ".length());
            String credentials = new String(Base64.getDecoder().decode(base64Credentials));
            String[] values = credentials.split(":", 2);

            if (values.length == 2) {
                String username = values[0];
                String password = values[1];
                try {
                    UserDetails userDetails = inMemoryUserDetailsManager.loadUserByUsername(username);
                    if (passwordEncoder.matches(password, userDetails.getPassword())) {
                        String token = jwtTokenUtil.generateToken(username);
                        return ResponseEntity.ok(new HashMap<>() {{
                            put("token", token);
                        }});
                    }



                } catch (Exception e) {
                    return ResponseEntity.badRequest().body("Invalid credentials");
                }
            }
        }
        return ResponseEntity.badRequest().body("Authentication failed");
    }
}
```

SecurityConfig.java

```
package com.cognizant.springlearn.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
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.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable()
            .authorizeRequests()
            .antMatchers("/authenticate").permitAll()
            .anyRequest().authenticated();
    }




    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
    @Bean
    public InMemoryUserDetailsManager inMemoryUserDetailsManager() {
        UserDetails user = User.builder()
```

```
            .username("user")
            .password(passwordEncoder().encode("pwd"))
            .roles("USER")
            .build();
        return new InMemoryUserDetailsManager(user);
    }
}
```

JwtTokenUtil.java

```
package com.cognizant.springlearn.config;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import org.springframework.stereotype.Component;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
@Component
public class JwtTokenUtil {
    private static final String SECRET_KEY = "cognizant-secret-key-123"; // Stronger secret key
    private static final long EXPIRATION_TIME = 3600000; // 1 hour
    public String generateToken(String username) {
        Map<String, Object> claims = new HashMap<>();
        return Jwts.builder()
                .setClaims(claims)
                .setSubject(username)
                .setIssuedAt(new Date(System.currentTimeMillis()))
                .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
                .signWith(SignatureAlgorithm.HS256, SECRET_KEY)
                .compact();
    }
}
```

Digital Nuture Program

Week 4

Pom.XML

```
<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-api</artifactId>
      <version>0.11.5</version>
   </dependency>
   <dependency>
      <groupId>io.jsonwebtoken</groupId>
      <artifactId>jjwt-impl</artifactId>
      <version>0.11.5</version>
      <scope>runtime</scope>
   </dependency>
</dependencies>
```

OUTPUT: