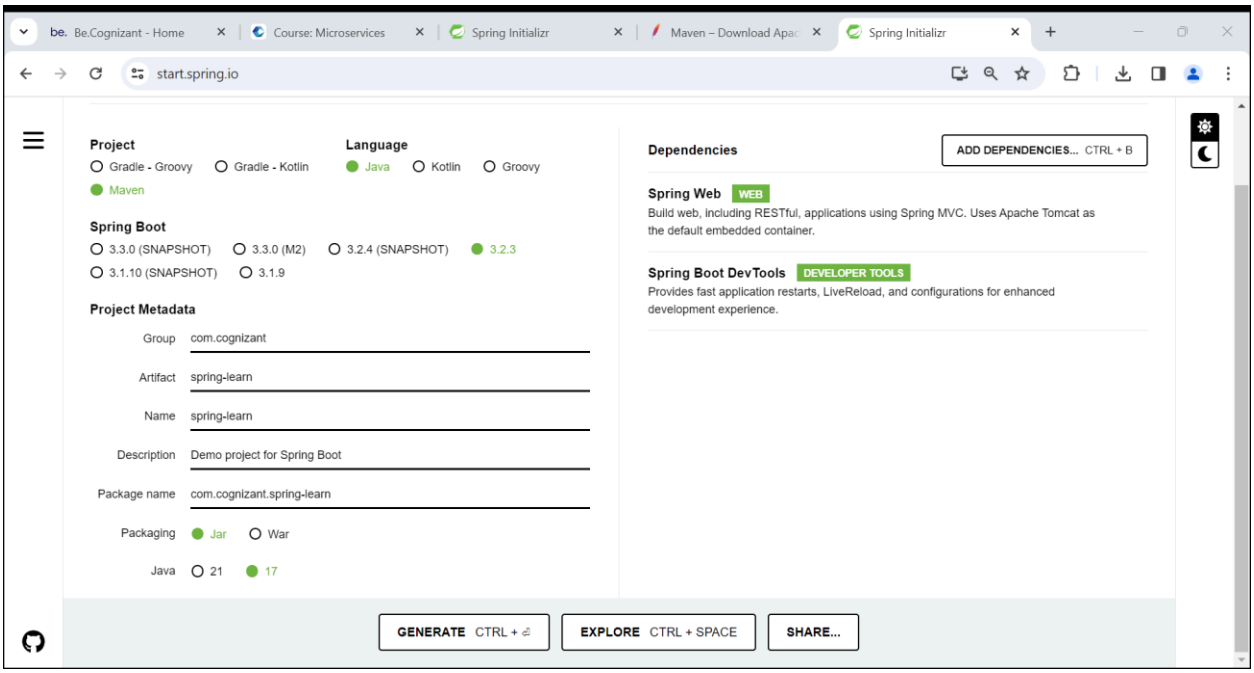
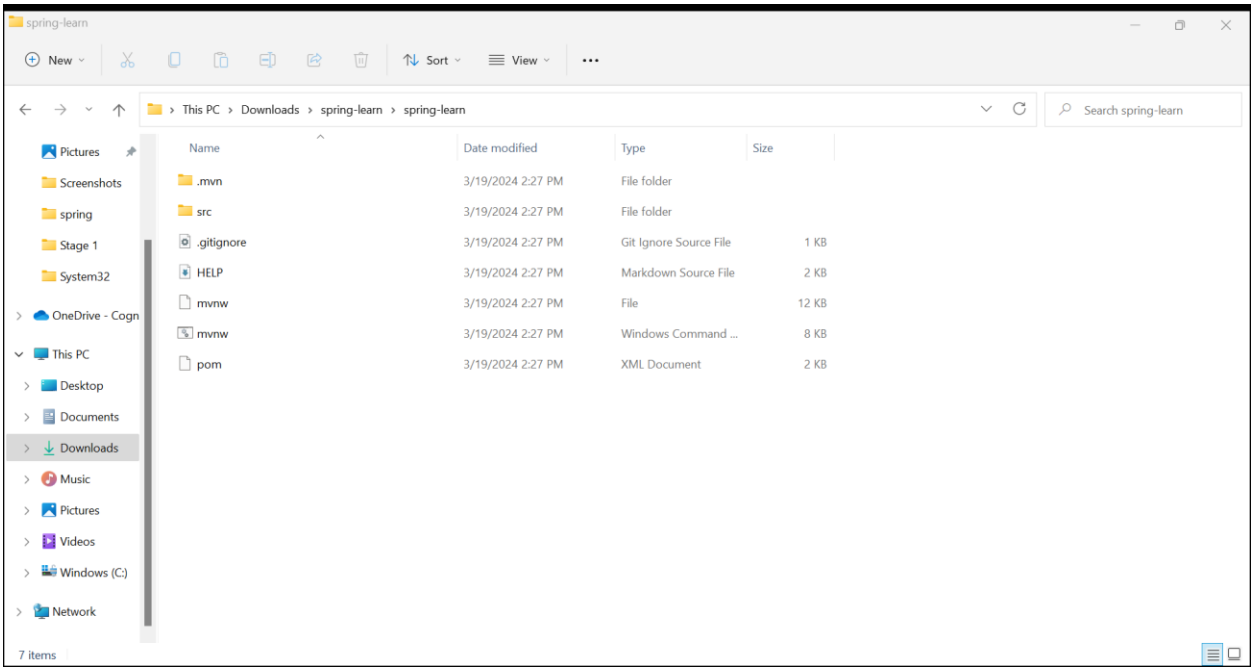
Hands on 1

Create a Spring Web Project using Maven

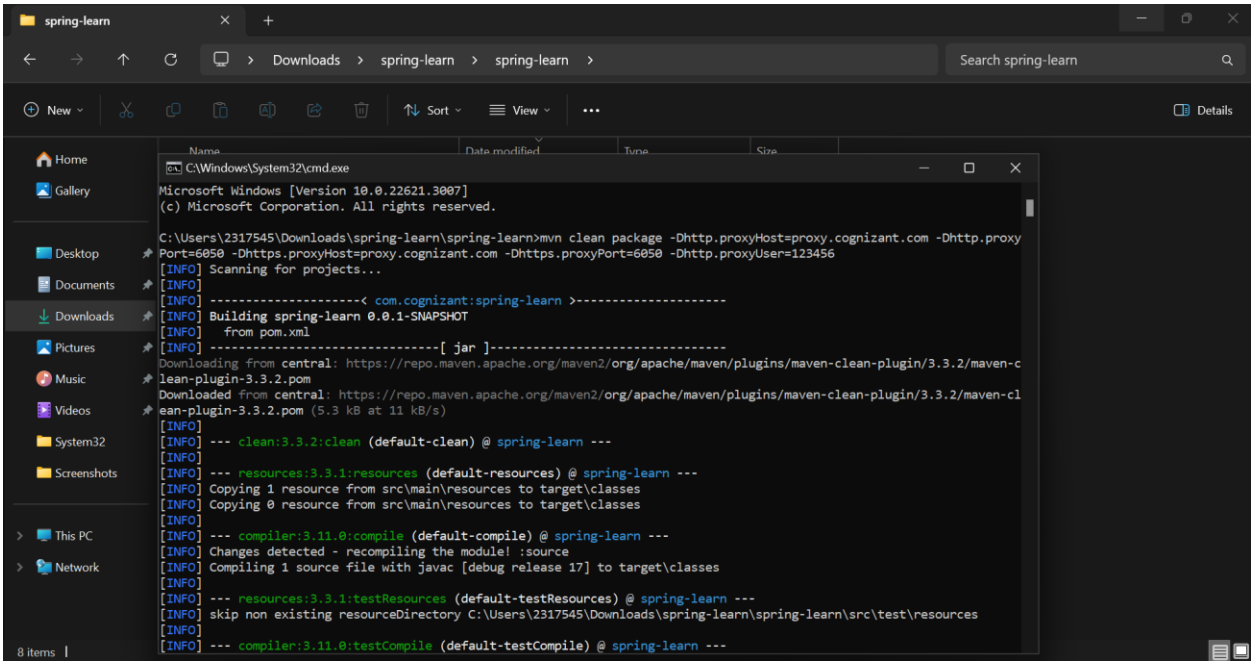
1. Created one spring learn maven project with the help of Spring Initializer



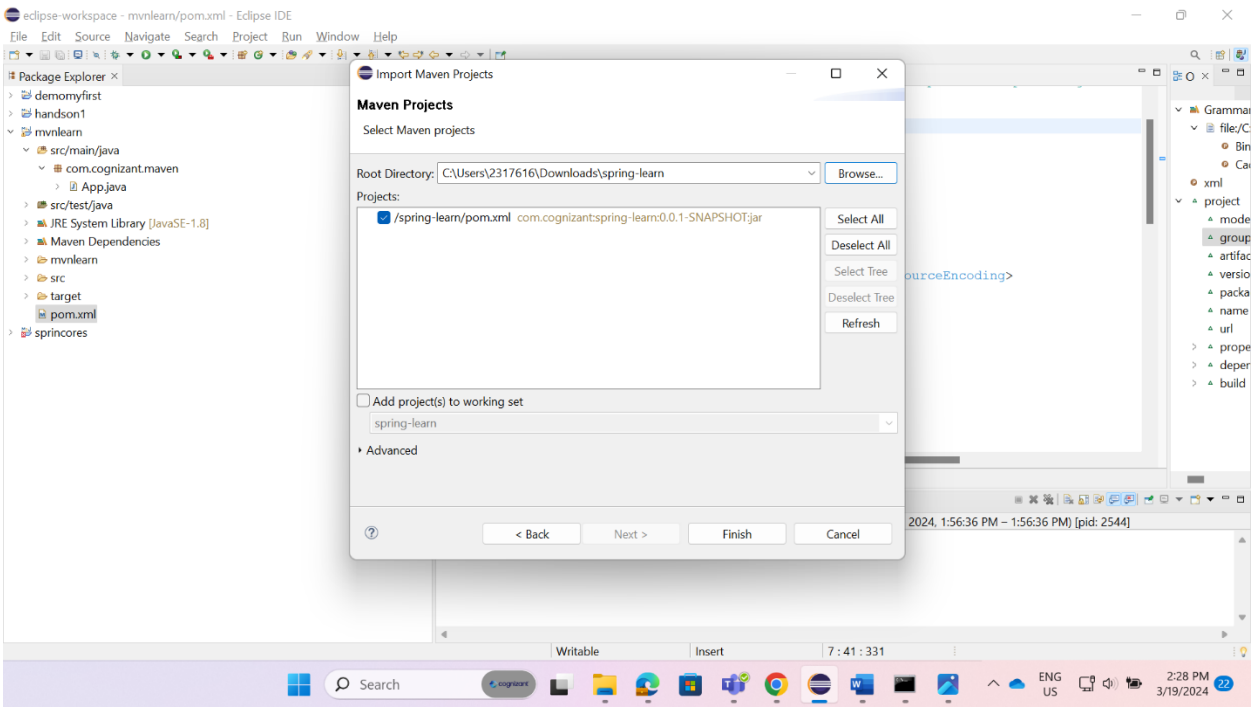
2. Downloaded the project as zip and extract the zip in root folder to Eclipse Workspace



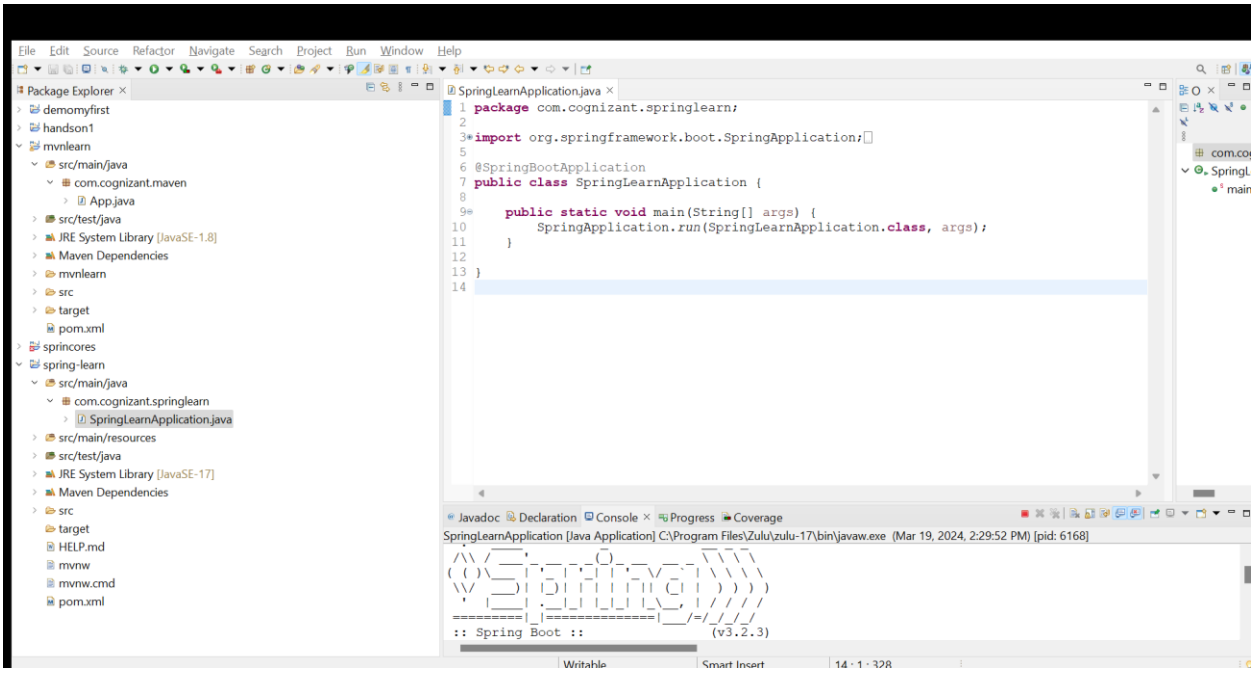
3. Build the project using given command on command line



4. Imported the project in Eclipse IDE



5. Run the SpringLearnApplication class.


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
demomyfirst
handson1
mvnlearn
src/main/java
com.cognizant.maven
App.java
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
mvnlearn
src
target
pom.xml
sprincores
spring-learn
src/main/java
com.cognizant.springlearn
SpringLearnApplication.java
src/main/resources
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
SpringLearnApplication.java
1 package com.cognizant.springlearn;
3 import org.springframework.boot.SpringApplication;
6 @SpringBootApplication
7 public class SpringLearnApplication {
9 public static void main(String[] args) {
10 SpringApplication.run(SpringLearnApplication.class, args);
11 }
13 }
Javadoc Declaration Console Progress Coverage
SpringLearnApplication [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 19, 2024, 2:29:52 PM) [pid: 6168]
:: Spring Boot :: (v3.2.3)
Writable Smart Insert 14 : 1 : 328

Hands on 2

Spring Core – Load SimpleDateFormat from Spring Configuration XML

1. Created the date-format.xml file.

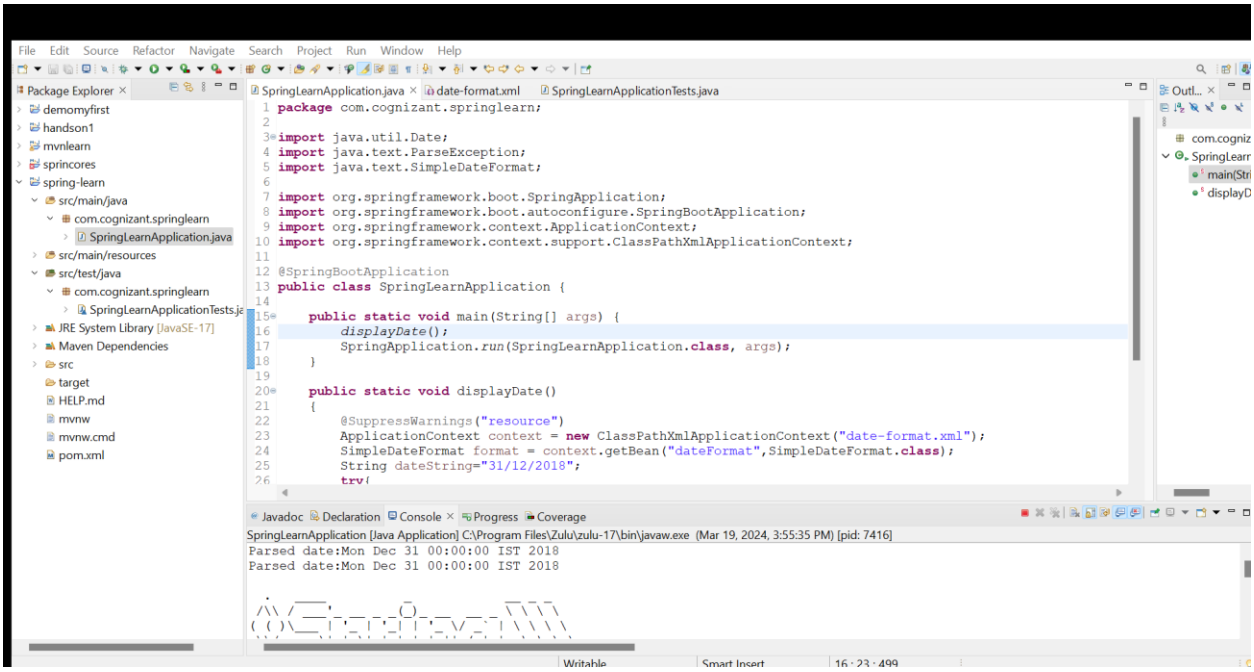
SpringLearnApplication.java | *date-format.xml

```
<?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="dateFormat" class="java.text.SimpleDateFormat">
        <constructor-arg value="dd/MM/yyyy" />
    </bean>

</beans>
```

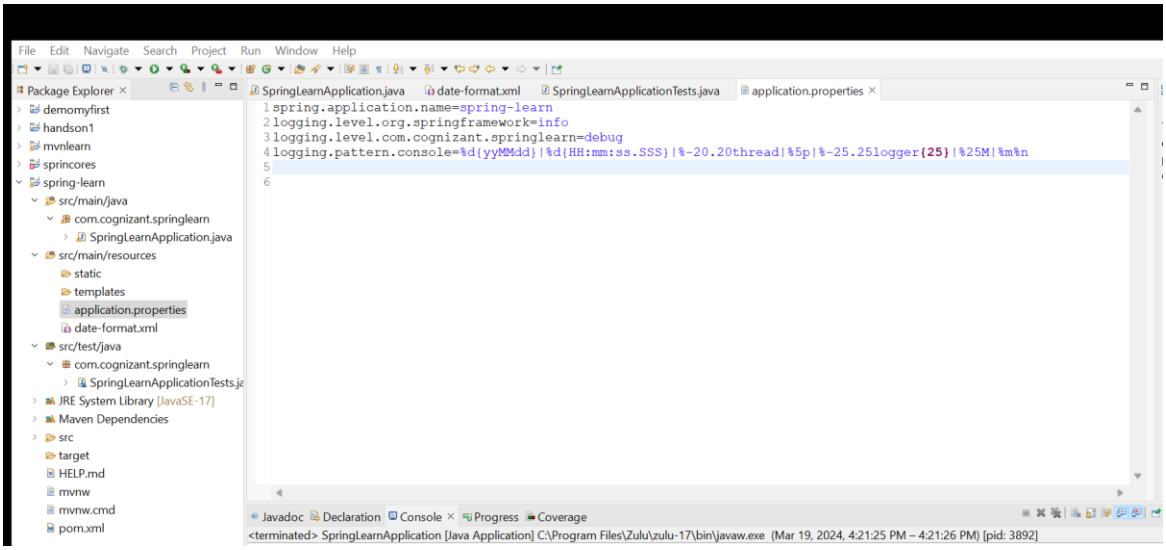
2. Created new method displayDate() in SpringLearnApplication.java In displayDate() method create the ApplicationContext.
3. Get the dateFormat using getBean() method. Using the format variable try to parse string '31/12/2018' to Date class and display the result using System.out.println.
4. Run the application as 'Java Application' and checked the result in console log output.



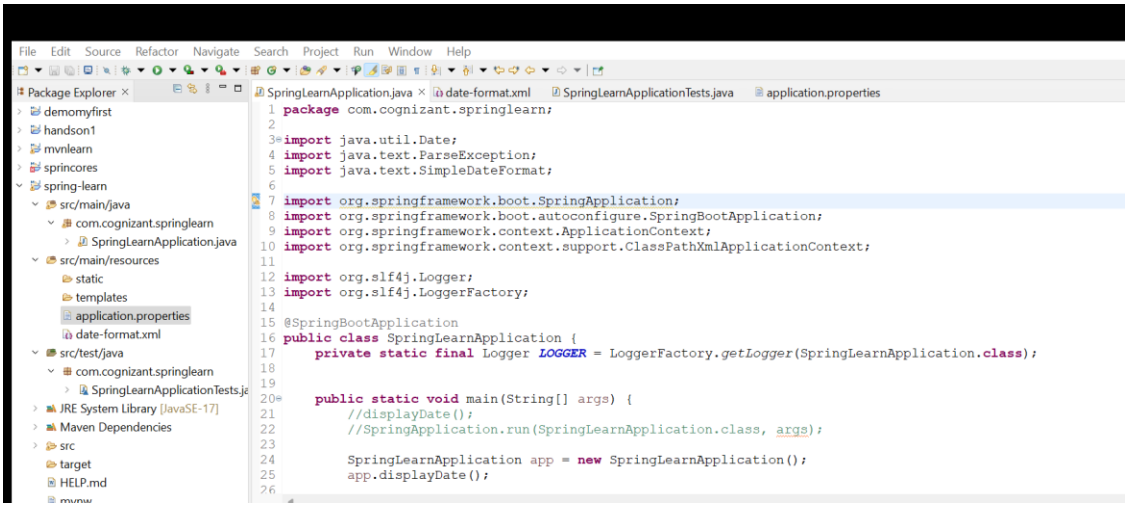
Hands on 3

Spring Core - Incorporate Logging

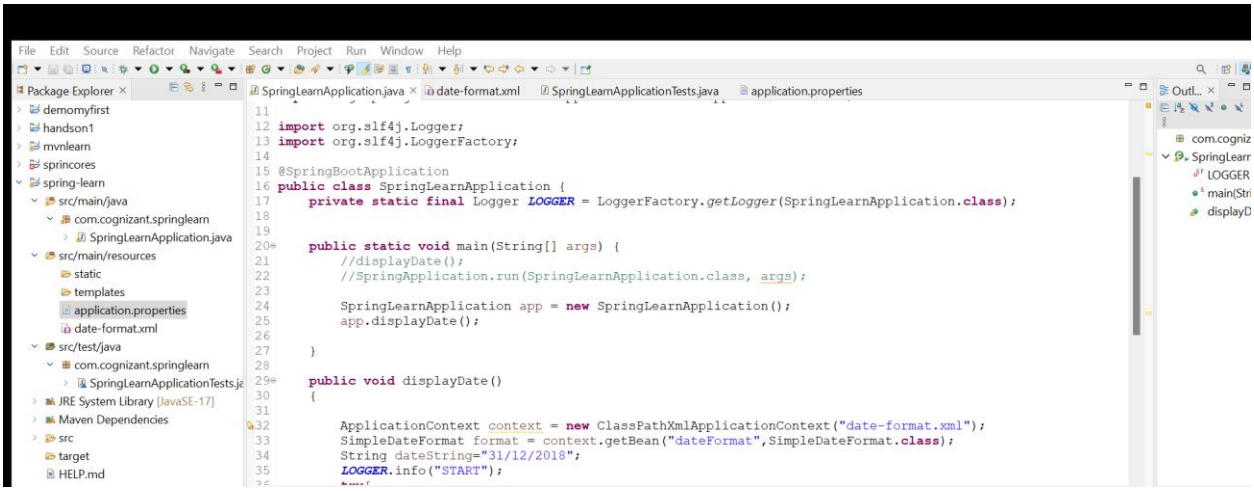
1. Created application.properties in src/main/resources folder



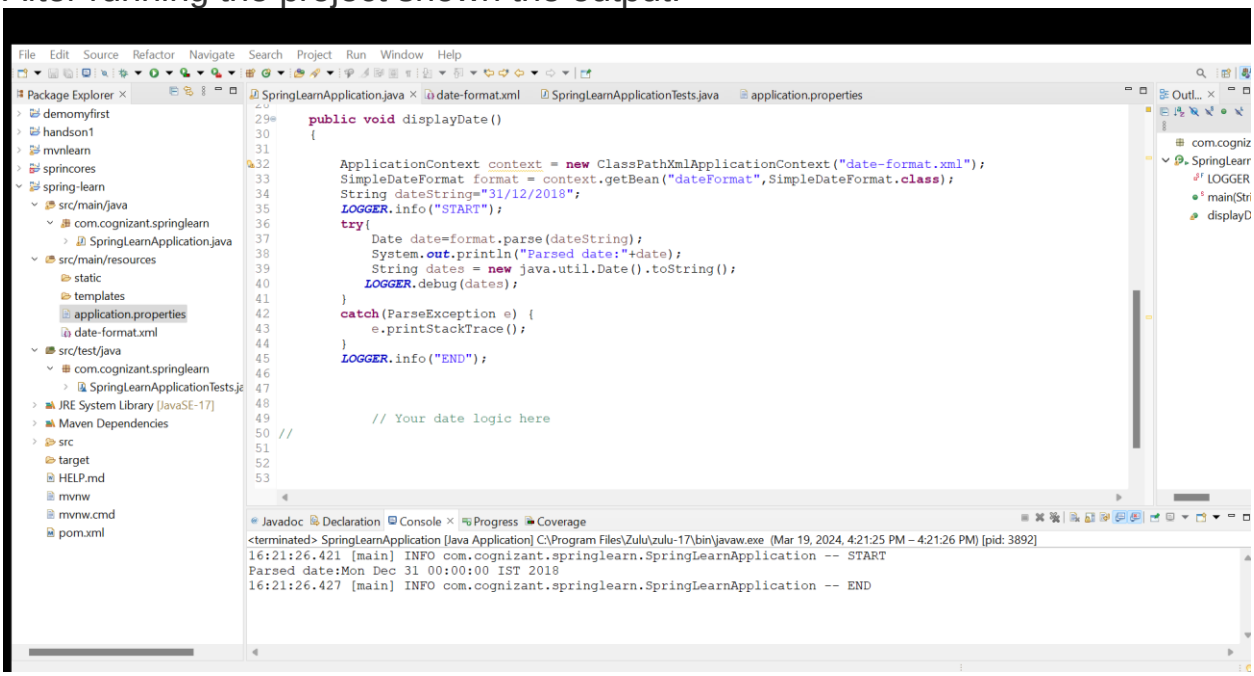
2. In SpringLearnApplication.java include the following imports



3. Included the static variable in SpringLearnApplication.java:



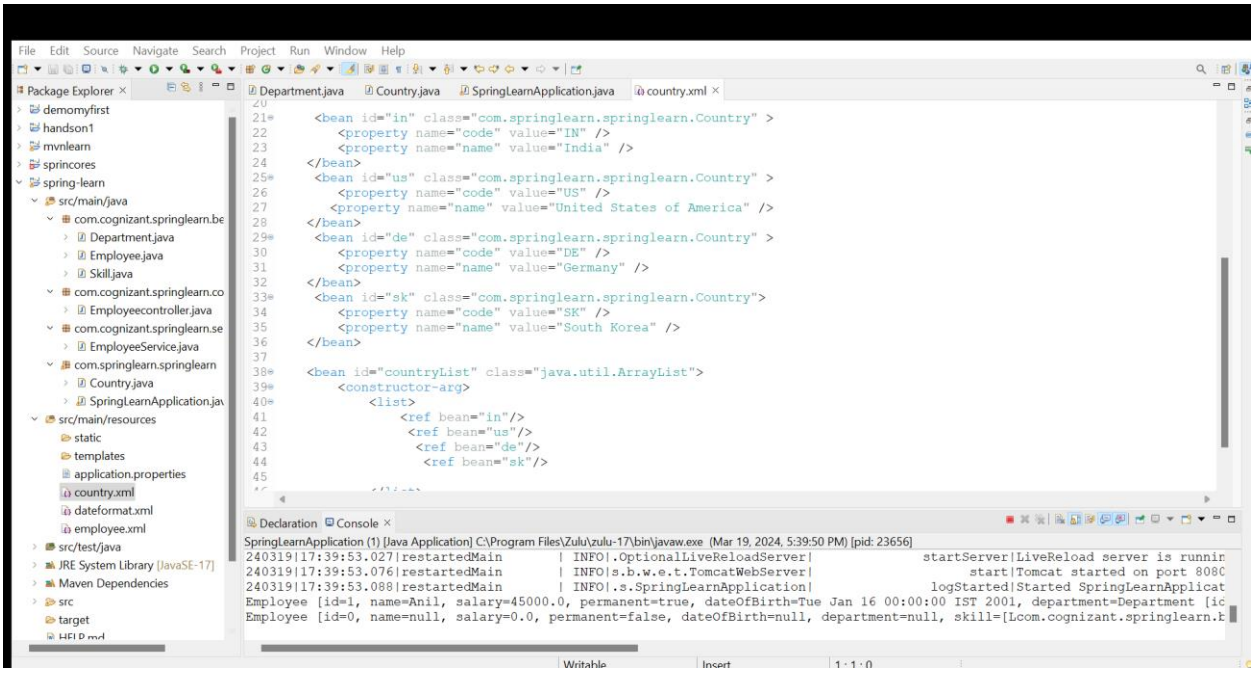
4. After running the project shown the output.



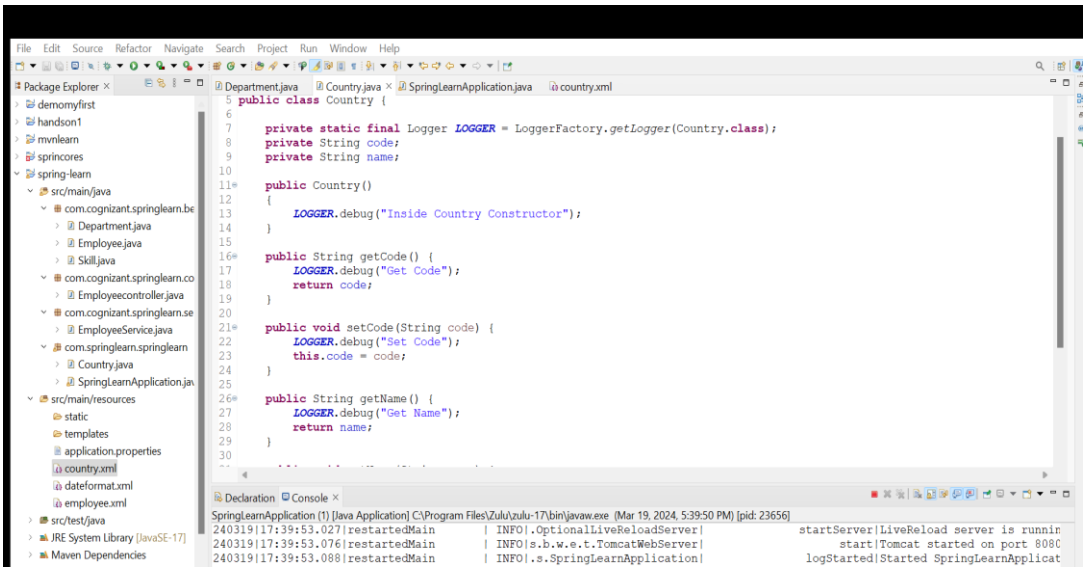
Hands on 4

Spring Core – Load Country from Spring Configuration XML

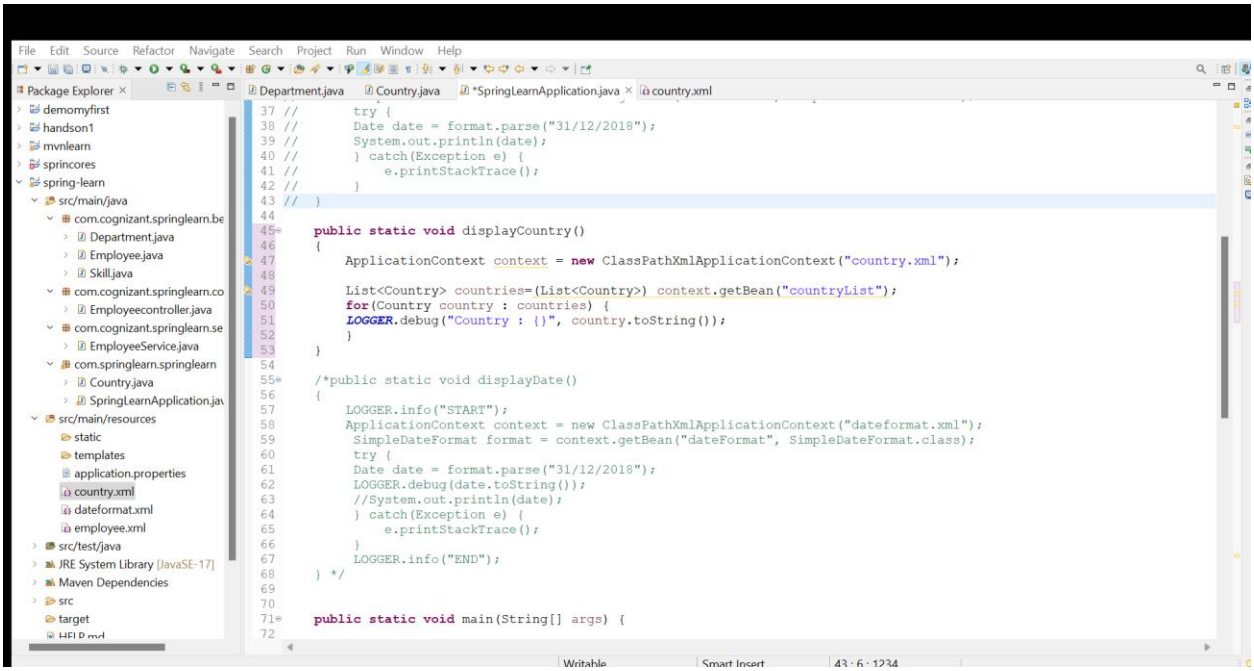
1. Created the country.xml file and included the requirement code in that.



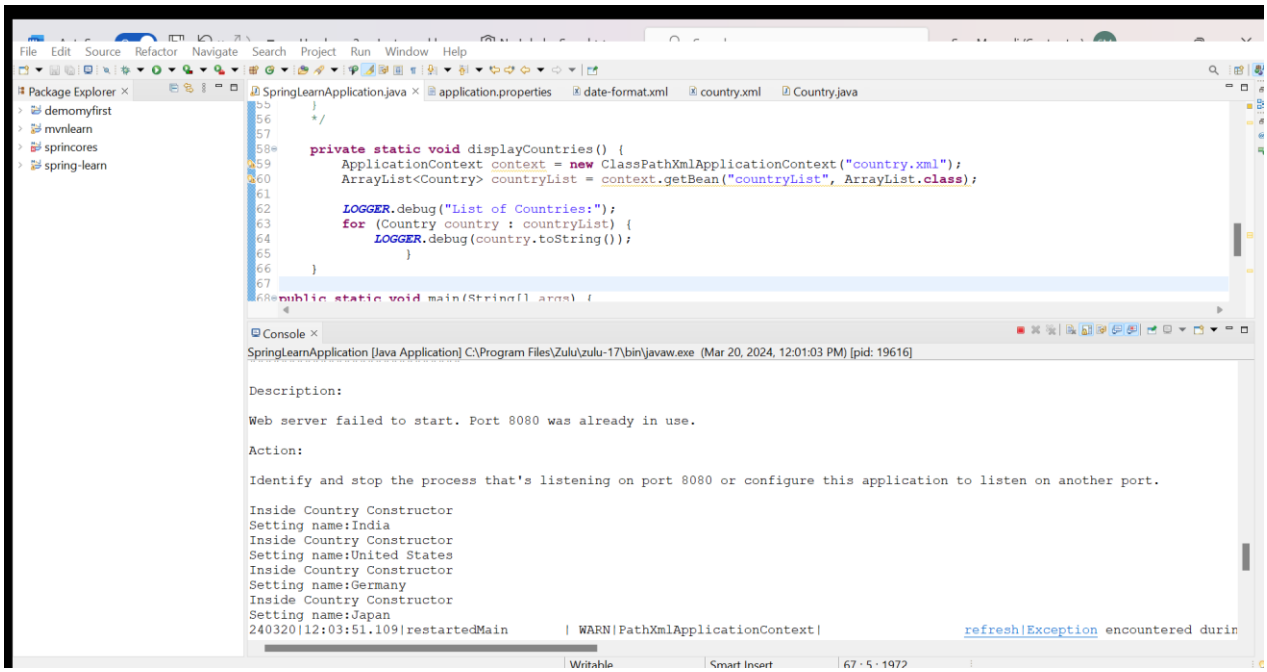
2. Create Country class with following aspects:



3. Created a method displayCountry() in SpringLearnApplication.java, which will read the country bean from spring configuration file and display the country details.
4. Invoked displayCountry() method in main() method of SpringLearnApplication.java.



5. Run the application and shown the output.

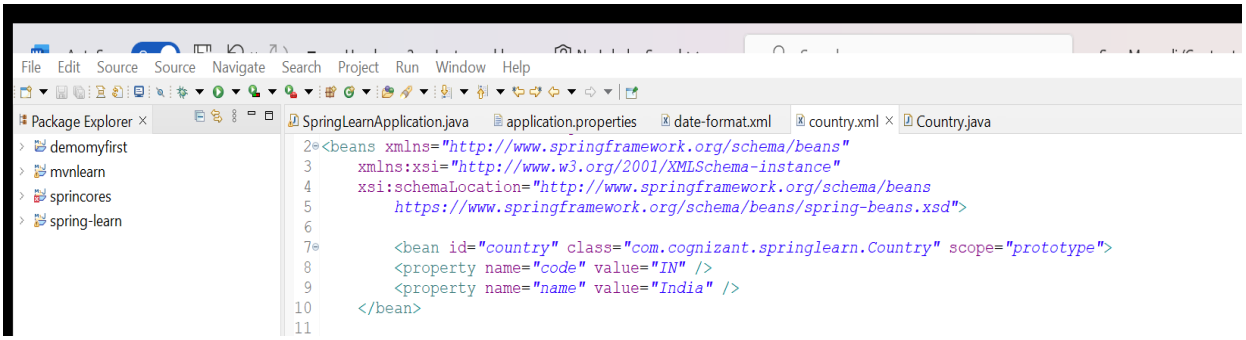


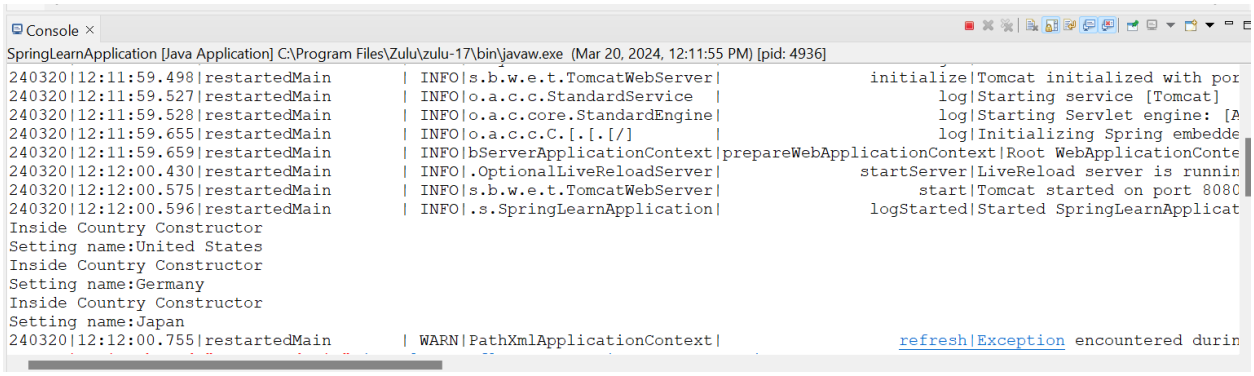
Hands on 5

Spring Core – Demonstration of Singleton Scope and Prototype Scope

1. Run the application

2. Constructor will be called twice, which means that two instances of country is created.





Hands on 6

Spring Core – Load list of countries from Spring Configuration XML

1. Created a separate bean for each of the four country in country.xml.
2. Created an ArrayList of Country in country.xml.