**COGNIZANT DIGITAL NURTURE PROGRAM 4.0**

**JAVA FSE DEEP SKILLING**

**SUPERSET ID: 6373880**

**NAME : SREE LAXMI V S**

**CLASS : ECE**

**REG NO : 727822TUEC219**

**WEEK 5HANDS ON EXERCISE – Microservices with Spring Boot 3 and Spring Cloud**

**Exercise : Creating Microservices for account and loan**

**Microservices with API gateway**

**PROBLEM STATEMENT:**

Design and implement two independent Spring Boot microservices for a bank—**Account Service** and **Loan Service**—each exposing a REST API to retrieve dummy account or loan details based on a given account number. Ensure both services run on different ports to simulate real microservice deployment environments.

**CODE:**

* **ACCOUNTS**

**AccountApplication.java**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

publicclassAccountApplication{

publicstaticvoidmain(String[]args){

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

}

}

**AccountController.java**

package com.cognizant.account;

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

*@RestController*

*@RequestMapping*("/accounts")

publicclassAccountController{

*@GetMapping*("/{number}")

publicAccountgetAccount(*@PathVariable*Stringnumber){

returnnewAccount(number,"savings",234343);

}

staticclassAccount{

privateStringnumber;

privateStringtype;

privatedoublebalance;

publicAccount(Stringnumber,Stringtype,doublebalance){

this.number=number;

this.type=type;

this.balance=balance;

}

// Getters

publicStringgetNumber(){returnnumber;}

publicStringgetType(){returntype;}

publicdoublegetBalance(){returnbalance;}

}

}

**application.properties**

spring.application.name=account

server.port=8081

spring.application.name=account-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

**pom.xml**

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

<projectxmlns="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>account</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>account</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>

* **LOANS**

**LoanApplication.java**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

publicclassLoanApplication {

    publicstaticvoidmain(String[] args) {

        SpringApplication.run(LoanApplication.class, args);

    }

}

**LoanController.java**

package com.cognizant.loan;

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

*@RestController*

*@RequestMapping*("/loans")

publicclassLoanController{

*@GetMapping*("/{number}")

publicLoangetLoan(*@PathVariable*Stringnumber){

returnnewLoan(number,"car",400000,3258,18);

}

staticclassLoan{

privateStringnumber;

privateStringtype;

privatedoubleloan;

privateintemi;

privateinttenure;

publicLoan(Stringnumber,Stringtype,doubleloan,intemi,inttenure){

this.number=number;

this.type=type;

this.loan=loan;

this.emi=emi;

this.tenure=tenure;

}

// Getters

publicStringgetNumber(){returnnumber;}

publicStringgetType(){returntype;}

publicdoublegetLoan(){returnloan;}

publicintgetEmi(){returnemi;}

publicintgetTenure(){returntenure;}

}

}

**application.properties**

spring.application.name=loan

server.port=8081

**pom.xml**

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

<projectxmlns="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>loan</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>loan</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>

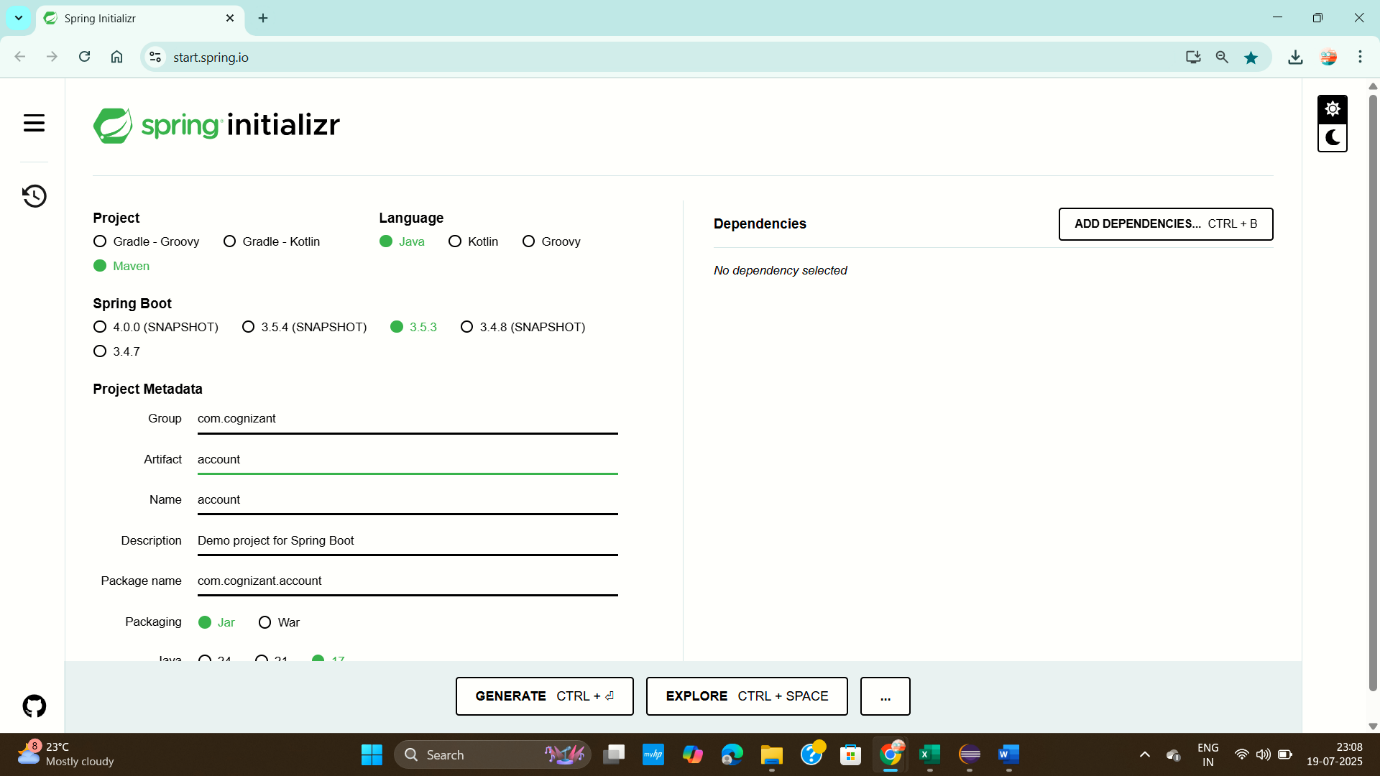
</plugins>

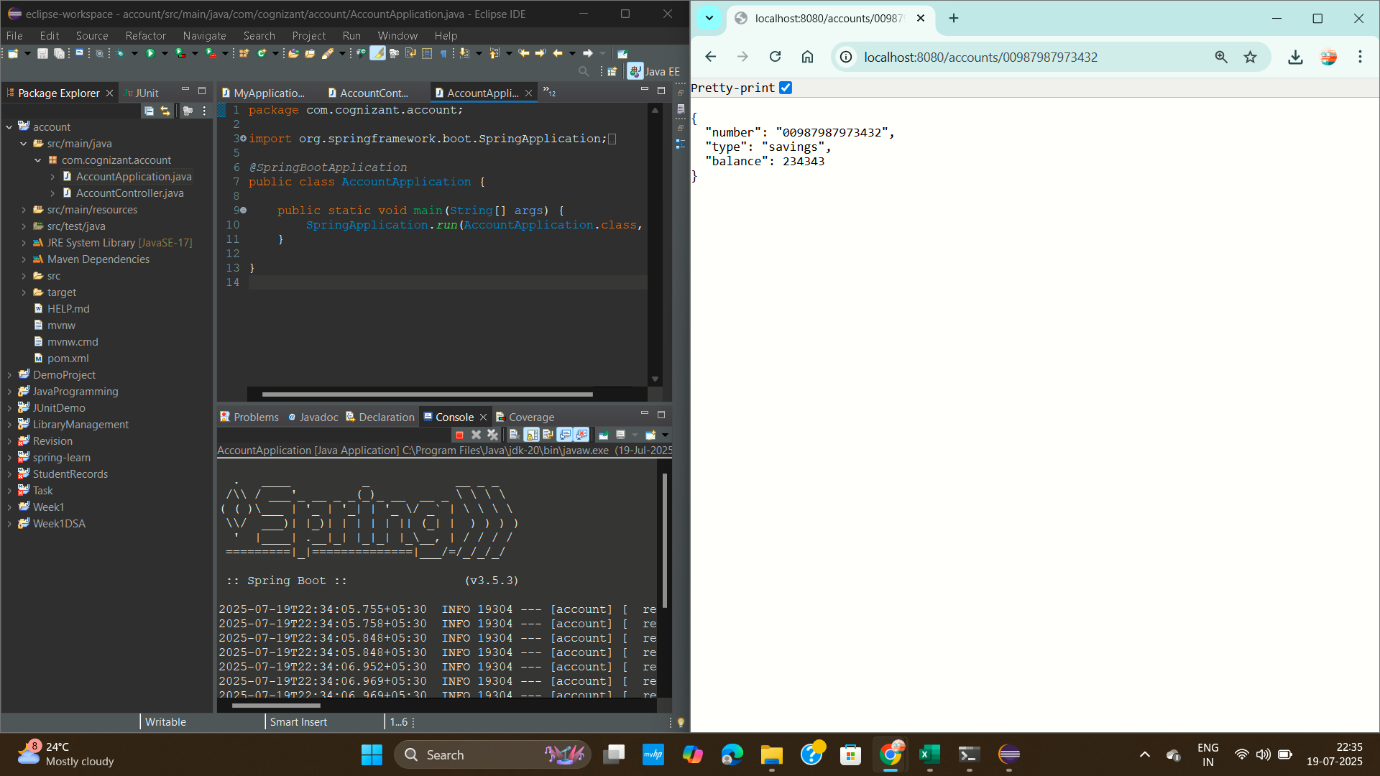
</build>

</project>

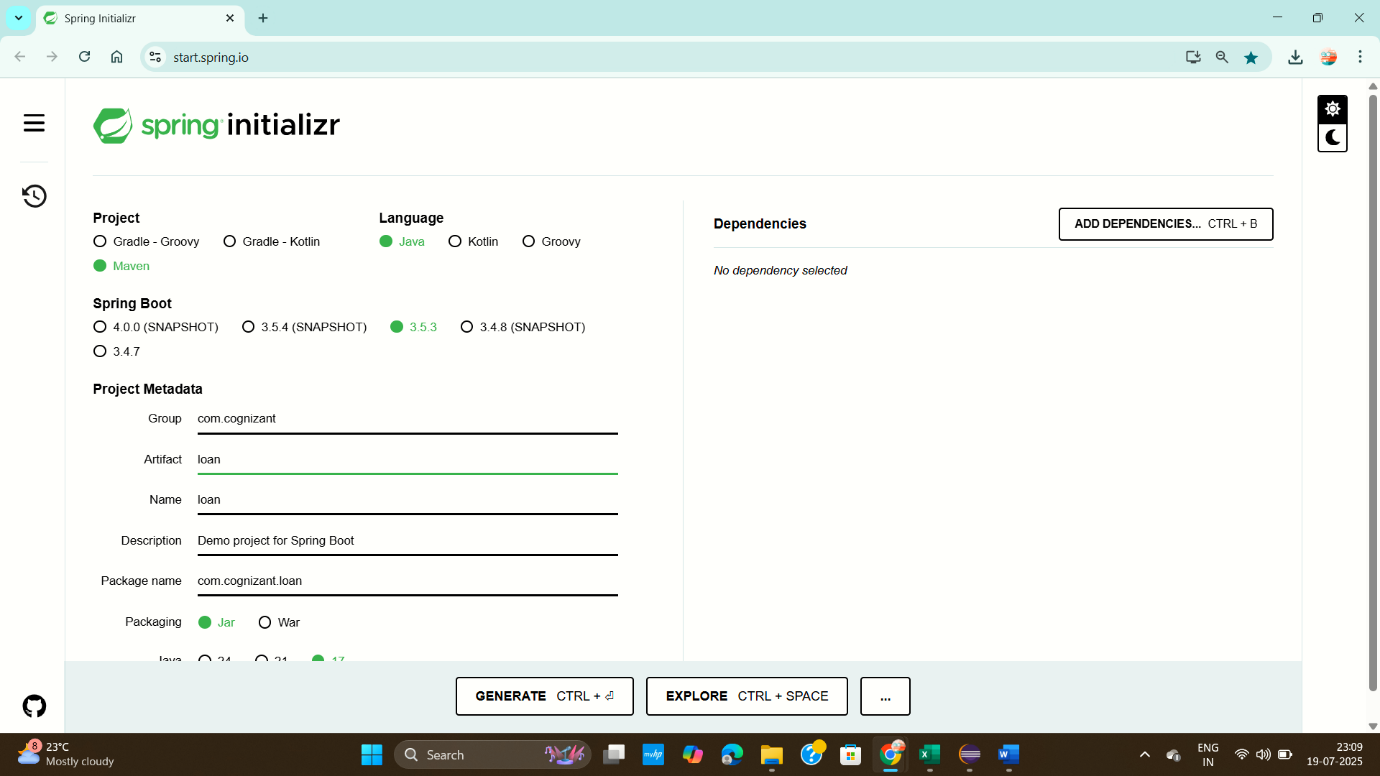
**OUTPUT :**

**accounts**





**loans**

****

