**Week 5 Mandatory Hands on Exercises:**

**Skill:** **Microservices with Spring Boot 3 and Spring Cloud**

Hands-on:1:

**Creating Microservices for account and loan**

**Solution:**

1.In spring initializer create group as com.cognizant and artifact as account and add necessary dependency tools

2.For account microservices create a folder in drive and extract the zip file from spring initializer in to it.

And import the file in Eclipse Maven project

3.**Create a controller package and AccountController class:**

package com.cognizant.account.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Map<String, Object> getAccount(@PathVariable String number) {

return Map.of(

"number", number,

"type", "savings",

"balance", 234343

);

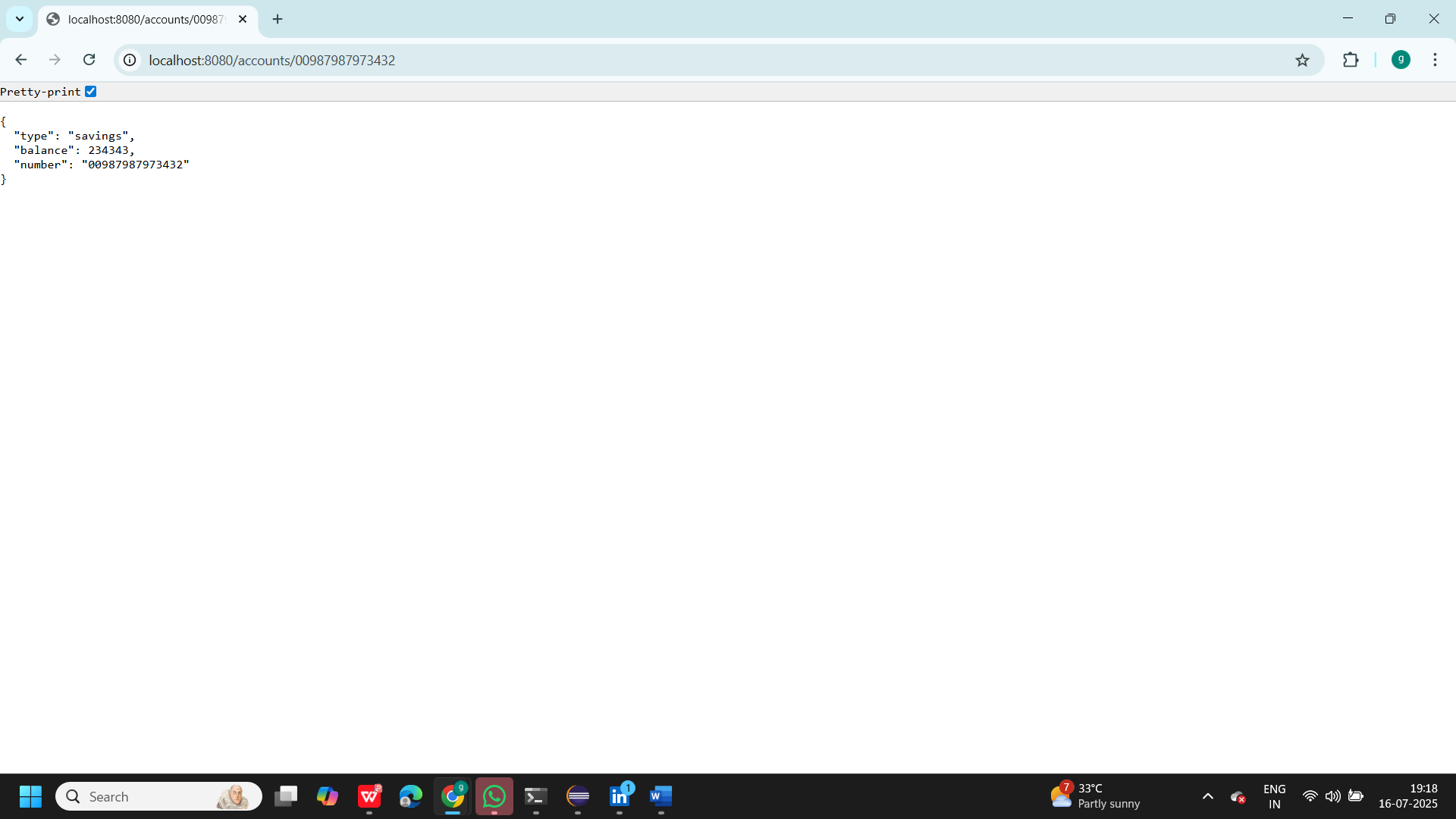
}

}

4.**Run the application and browse the following link**

<http://localhost:8080/accounts/00987987973432>

**Output:**



Part 2:Create a loan Microservices:

1.**In application properties create server:**

server.port=8081

2.**Create Loan Controller:**

package com.cognizant.loan.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Map<String, Object> getLoan(@PathVariable String number) {

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

"tenure", 18

);

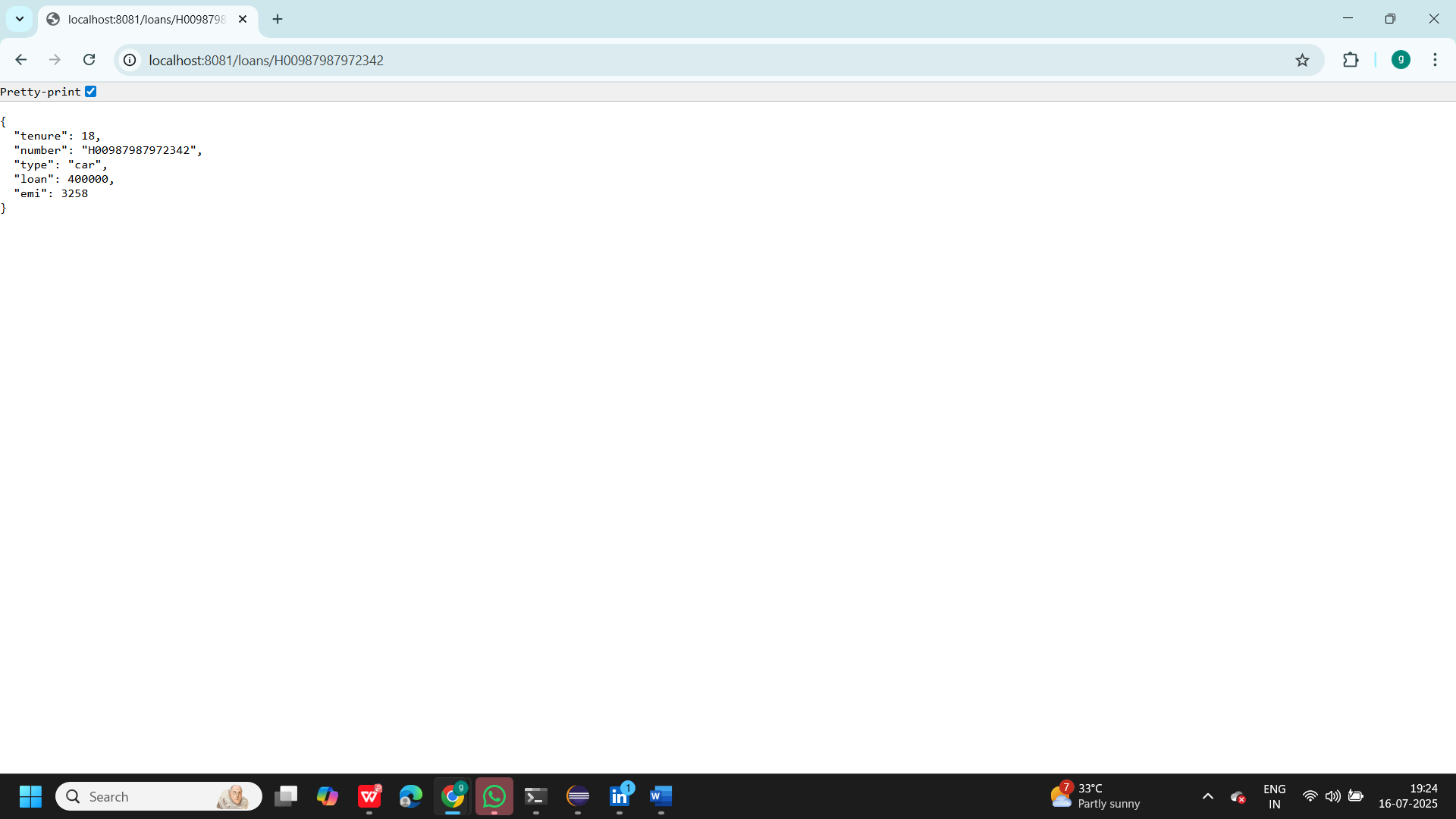
}

}

3**.Run and test the application**:

<http://localhost:8081/loans/H00987987972342>

**Output:**



**Additional Important Hands-on:**

**Create Eureka Discovery Server and register microservices**

**Solution:**

1.In Spring initializer keep Group name as com.cognizant and artifact as eureka\_discovery\_server

And add necessary dependencies and generate zip file

2. Extract zip file to microsevices folder which we created

3.**Modify application class as:**

@EnableEurekaServer

@SpringBootApplication

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

}

}

4**.Update application properties**:

server.port=8761

eureka.client.register-with-eureka=false

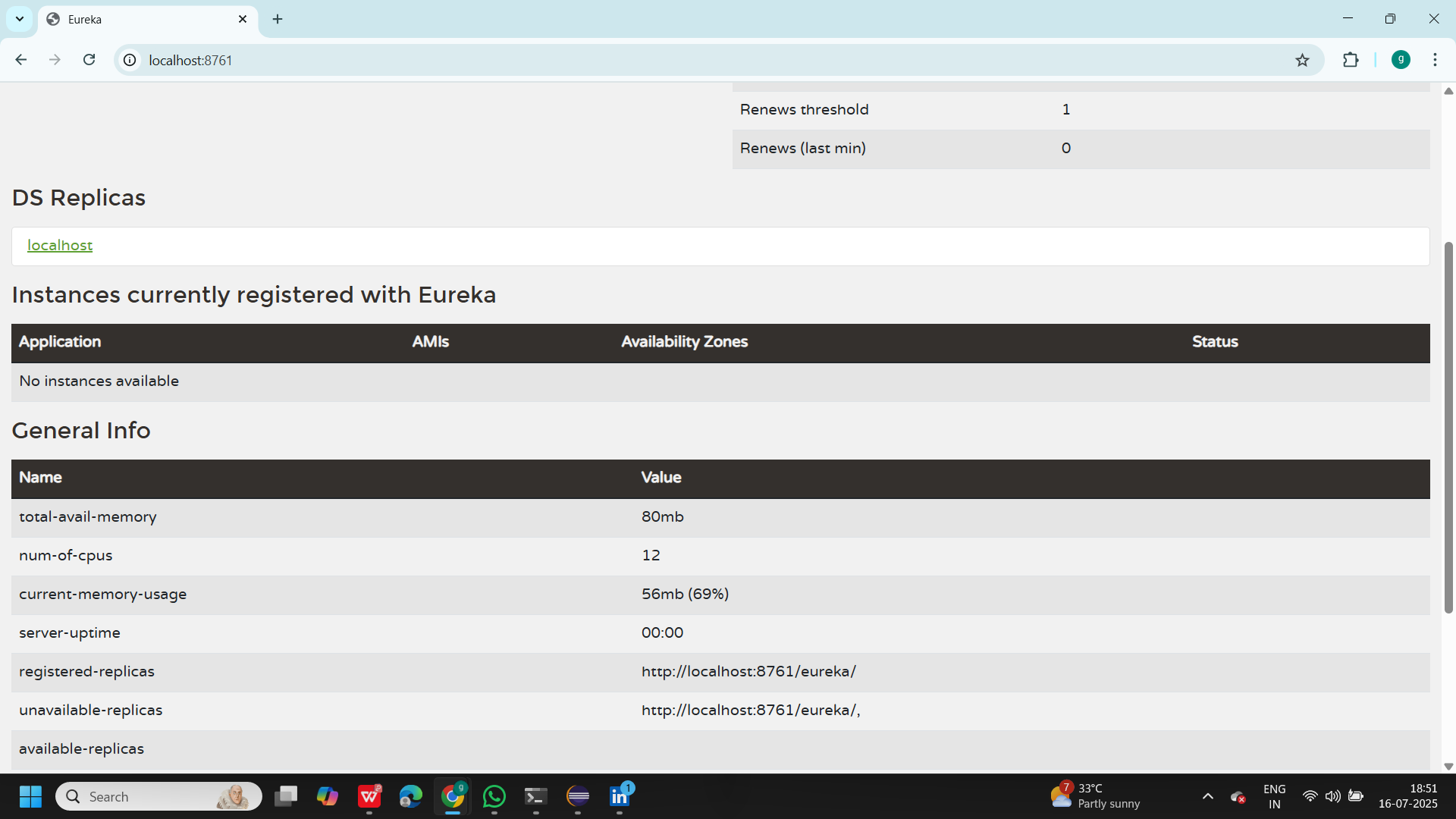
eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF

logging.level.com.netflix.discovery=OFF

5.Run the application and Visit: <http://localhost:8761>

We get empty instances as shown below



**6.Update account application that we created earlier:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

7.**update application properties in account:**

server.port=8080

spring.application.name=account-service

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

**8.Update Loan application :**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

}

}

**9.Update application properties in loan:**

server.port=8081

spring.application.name=loan-service

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

**10.Add dependencies in Account, loan and eureka:**

<dependencies>

<!-- Web for REST API -->

<dependency>

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

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

</dependency>

<!-- Eureka Server (for Discovery Server only) -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

<scope>compile</scope>

</dependency>

<!-- Eureka Client (for Account and Loan) -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

</dependencies>

<!-- Spring Cloud BOM for compatibility -->

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>2021.0.8</version> <!-- Use version matching your Spring Boot version -->

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

**11.refresh the eureka dashboard:**

