**Week 5 Microservices Madatory Hands-On**

**Creating Microservices for account and loan**

* **Account Microservice**

Code:

AccountApplication.java

package com.cognizant.account;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AccountApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(AccountApplication.class, args);  
 }  
  
}

AccountController.java

**package** com.cognizant.account.controller;

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

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

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

**import** java.util.HashMap;

**import** java.util.Map;

@RestController

**public** **class** AccountController {

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

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

Map<String, Object> response = **new** HashMap<>();

response.put("number", number);

response.put("type", "savings");

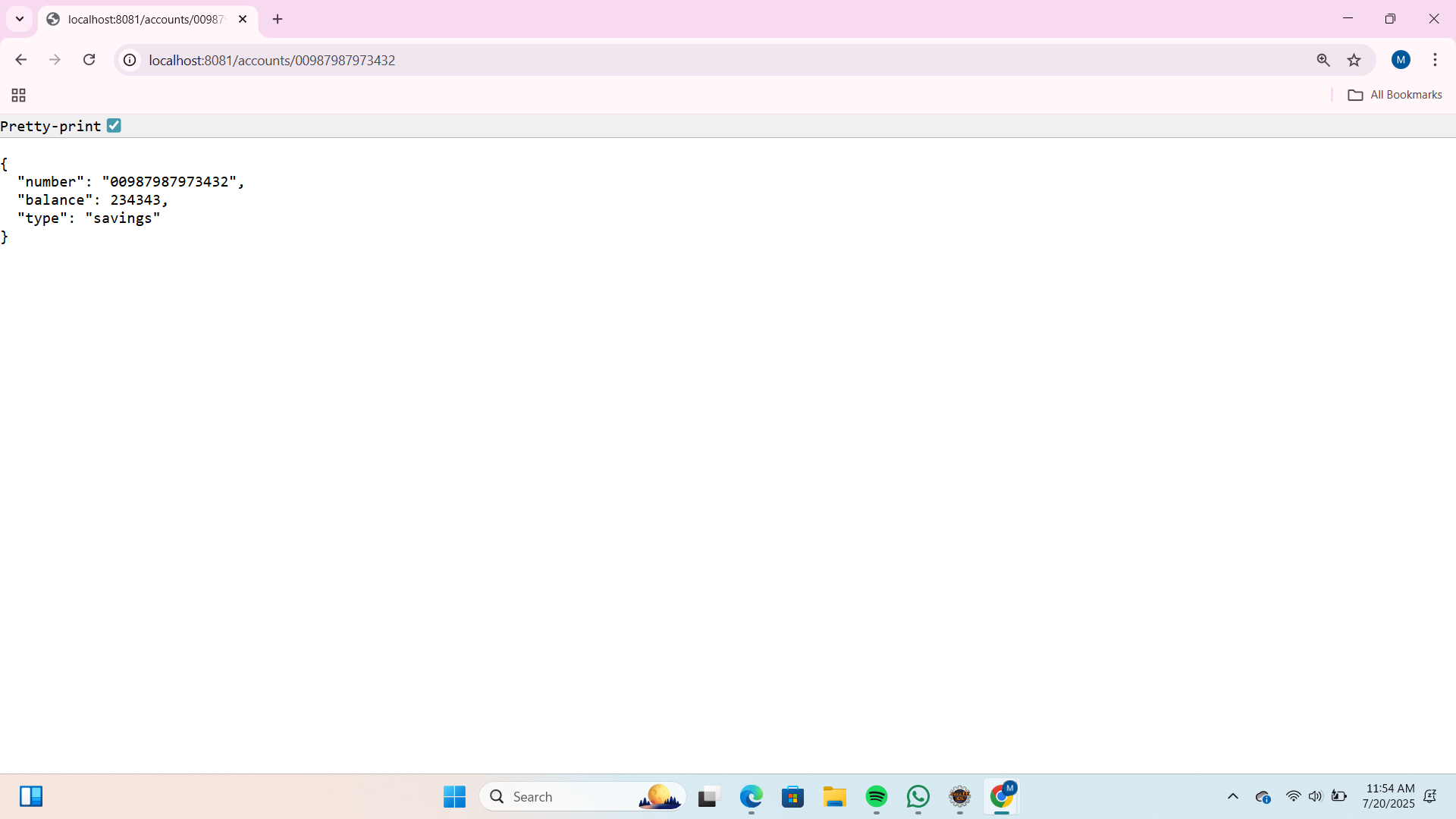
response.put("balance", 234343);

**return** response;

}

}

Output:



* **Loan Microservice**

Code:

LoanApplication.java

package com.cognizant.loan;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class LoanApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(LoanApplication.class, args);  
 }  
  
}

LoanController.java

**package** com.cognizant.loan.controller;

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

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

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

**import** java.util.HashMap;

**import** java.util.Map;

@RestController

**public** **class** LoanController {

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

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

Map<String, Object> response = **new** HashMap<>();

response.put("number", number);

response.put("type", "car");

response.put("loan", 400000);

response.put("emi", 3258);

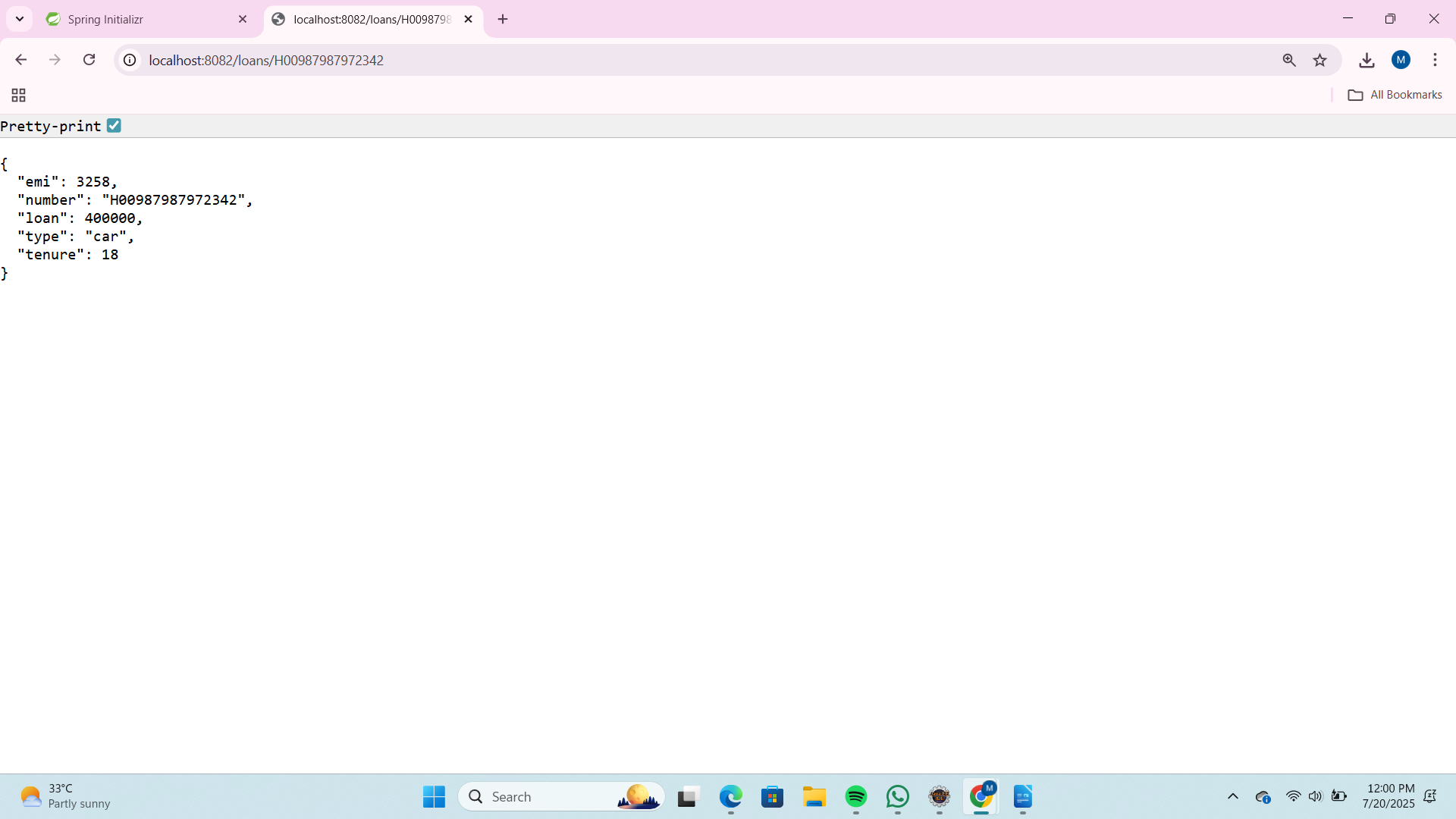
response.put("tenure", 18);

**return** response;

}

}

Output:



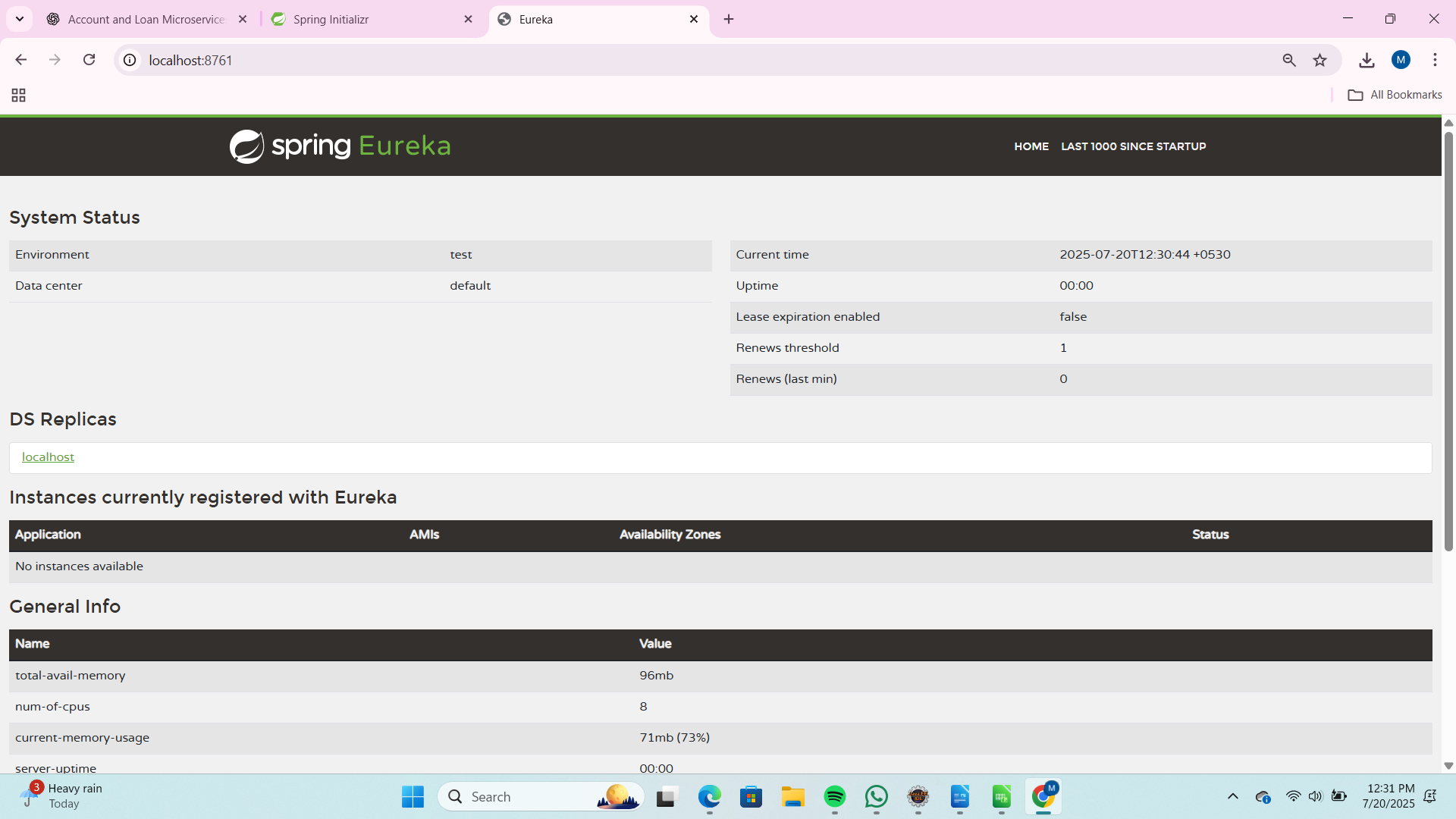
**Additional Hands-On**

**Create Eureka Discovery Server and register microservices**

Code:

EurekaDiscoveryServerApplication.java

package com.cognizant.eureka\_discovery\_server;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  
  
@SpringBootApplication  
@EnableEurekaServer  
public class EurekaDiscoveryServerApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(EurekaDiscoveryServerApplication.class, args);  
 }  
}

Output:  
  
  
  
Code:

AccountApplication.java

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);  
 }  
  
}

Output:

