**WEEK-5**

**Creating Microservices for Account and Loan**

Create two separate Spring Boot microservices:

1. One to handle account details
2. Another to handle loan details

Each service should:

* Be an independent Maven project
* Have its own controller and endpoint
* Return a dummy JSON response (without any database)

Both microservices must run at the same time on different ports (e.g., 8080 and 8081).

**Steps :**

* Account Microservice

1. I went to start.spring.io
2. Then selected:
   * Group: com.cognizant
   * Artifact: account
   * Dependencies: Spring Web, Spring Boot DevTools
3. Downloaded the ZIP and opened it in IntelliJ
4. Created a model class Account.java to hold account info
5. Created a controller AccountController.java with endpoint /accounts/{number}
6. Ran the app and tested using bash

http://localhost:8080/accounts/00987987973432

* Loan Microservice

1. Again used start.spring.io
2. Selected:
   * Group: com.cognizant
   * Artifact: loan
   * Dependencies: Spring Web, Spring Boot DevTools
3. Extracted and opened the project in IntelliJ
4. Created a model class Loan.java
5. Created a controller LoanController.java with endpoint /loans/{number}
6. Added the line below in application.properties to avoid port conflict:

server.port=8081

It’s important to assign the port number.

1. Ran the app and tested using: bash

http://localhost:8081/loans/H00987987972342

**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);  
 }  
}

**Account.java**

package com.cognizant.account;  
  
public class Account {  
 private String number;  
 private String type;  
 private double balance;  
  
 public Account(String number, String type, double balance) {  
 this.number = number;  
 this.type = type;  
 this.balance = balance;  
 }  
  
 public String getNumber() {  
 return number;  
 }  
  
 public String getType() {  
 return type;  
 }  
  
 public double getBalance() {  
 return balance;  
 }  
}

**AccountController.java**

package com.cognizant.account;  
  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Account getAccount(@PathVariable String number) {  
 return new Account(number, "savings",40000.00);  
 }  
}

**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);  
 }  
  
}

**Loan.java**

package com.cognizant.loan;  
  
public class Loan {  
 private String number;  
 private String type;  
 private double loan;  
 private double emi;  
 private int tenure;  
  
 public Loan(String number, String type, double loan, double emi, int tenure) {  
 this.number = number;  
 this.type = type;  
 this.loan = loan;  
 this.emi = emi;  
 this.tenure = tenure;  
 }  
  
 public String getNumber() { return number; }  
 public String getType() { return type; }  
 public double getLoan() { return loan; }  
 public double getEmi() { return emi; }  
 public int getTenure() { return tenure; }  
}

**LoanController.java**

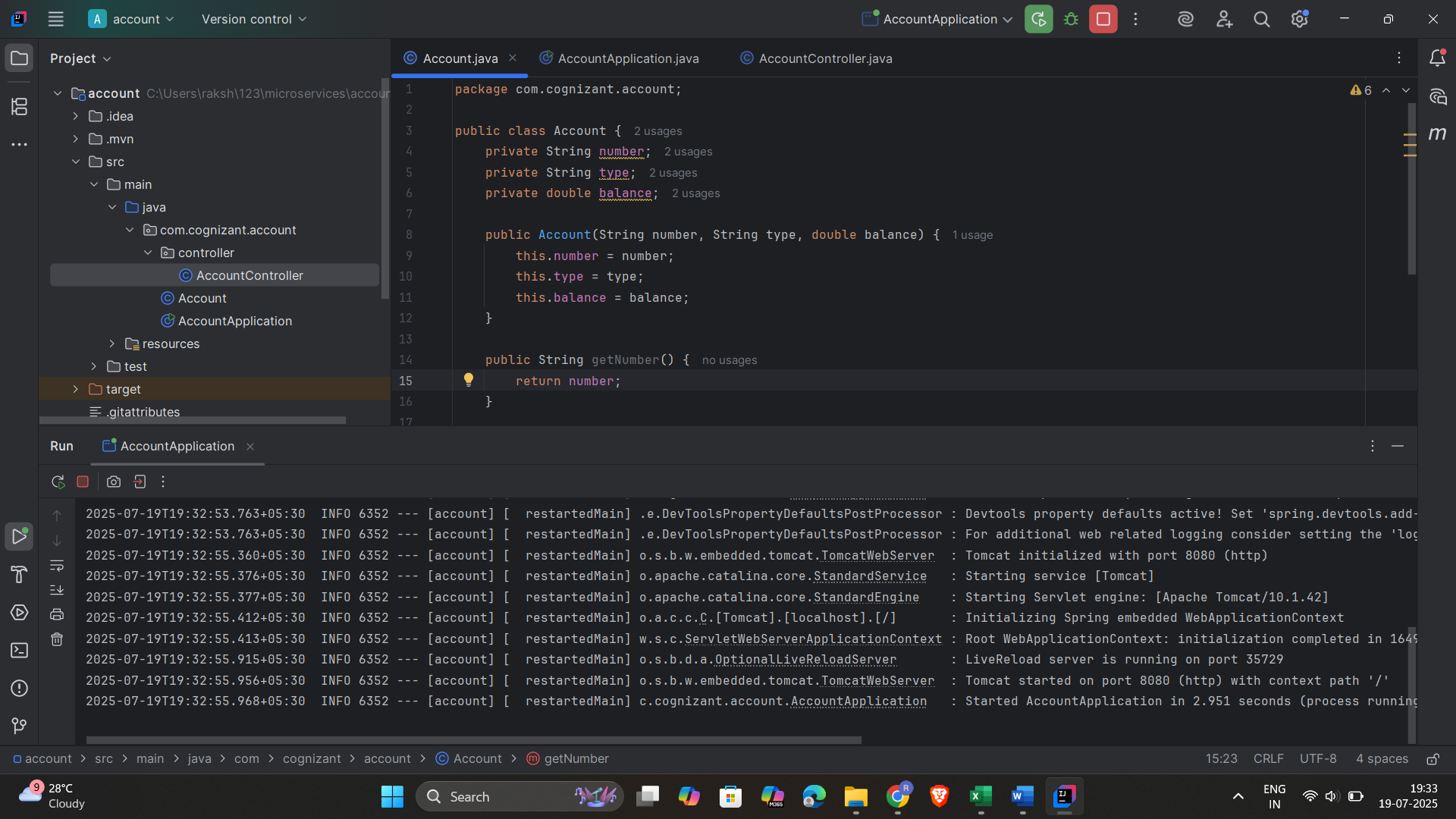
package com.cognizant.loan;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Loan getLoanDetails(@PathVariable String number) {  
 return new Loan(number, "car", 400000, 3258, 18);  
 }  
}

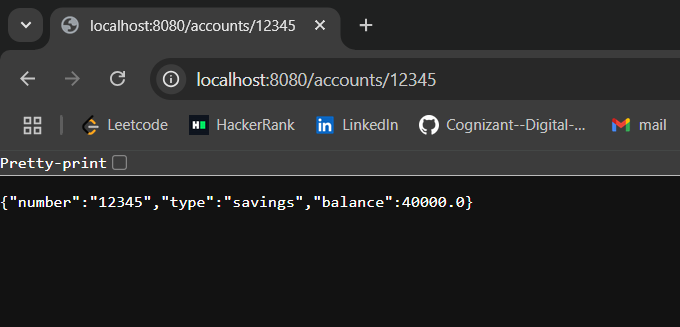
**application.properties**

spring.application.name=loan  
server.port=8081

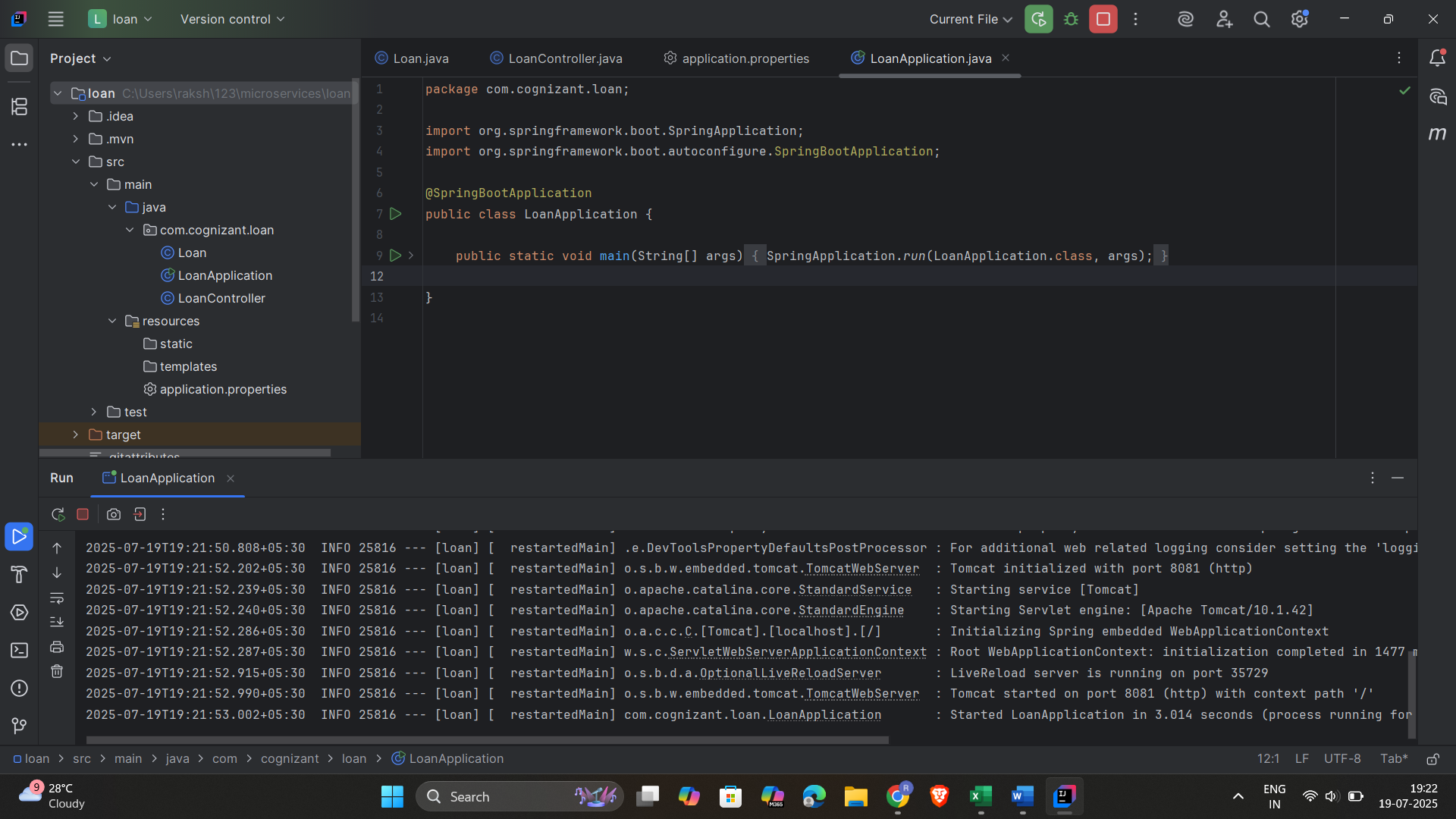
**Output:**

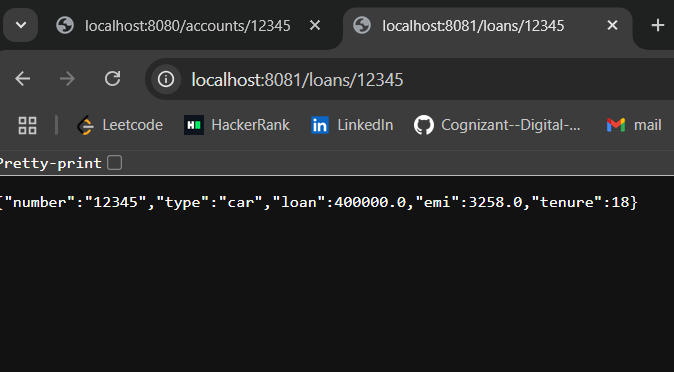
**For Account Microservice: http://localhost:8080/accounts/12345**



****

**For Loan Microservice: http://localhost:8081/loans/H00987987972342**





**Explanation:**

1. I created two different microservices: one for account and one for loan info.
2. Each project is independent and uses Spring Boot to build REST APIs.
3. The controller sends dummy data in JSON format when the endpoint is called.
4. I made sure the services don’t run on the same port to avoid errors.
5. I tested both in the browser and saw correct results on port 8080 and 8081.