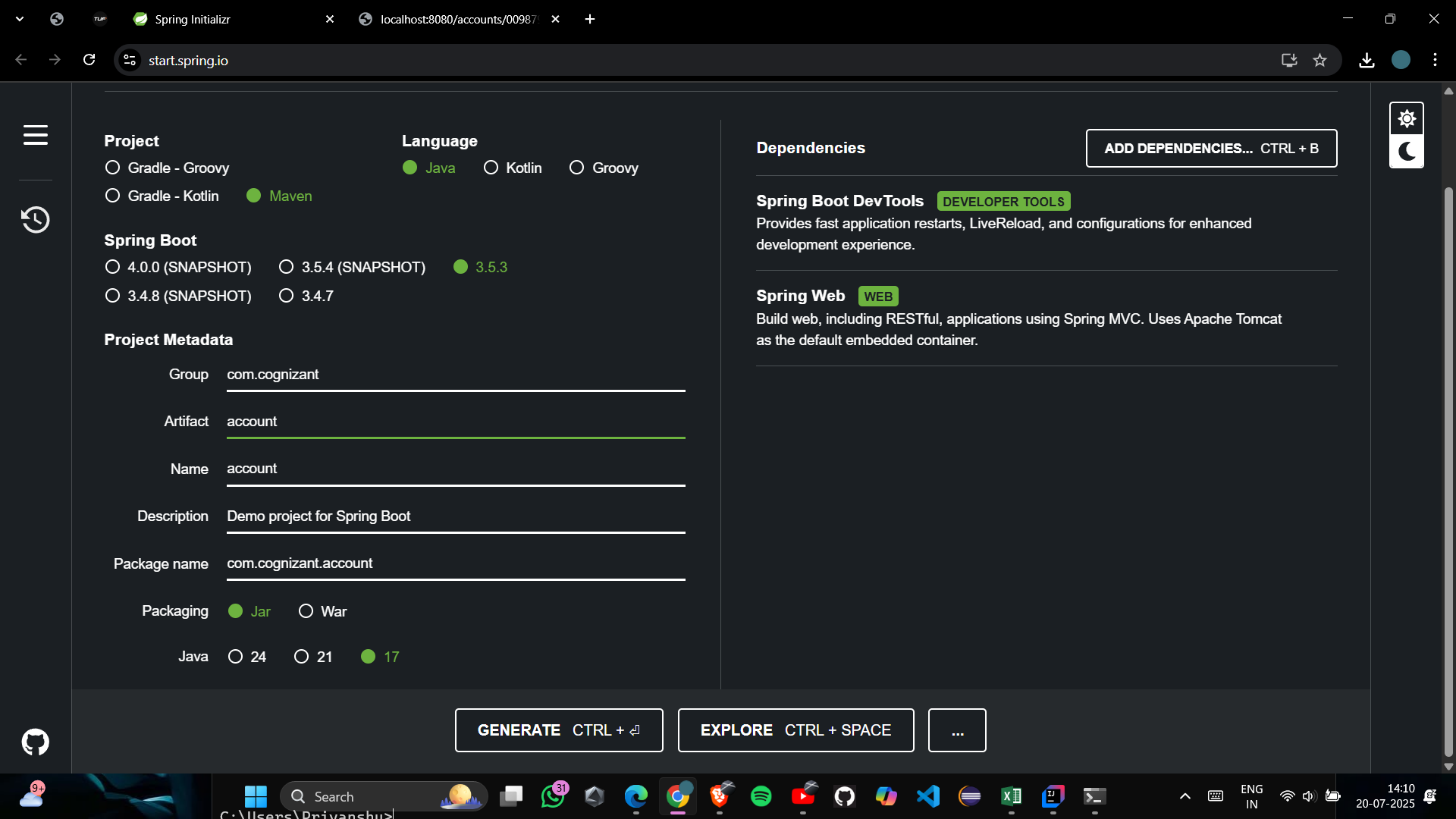
# Week 5 Hands-on : Microservices

Creating Microservices for Account and Loan

## 1. Spring Initializr Configuration – Account Service

Group: com.cognizant

Artifact: account

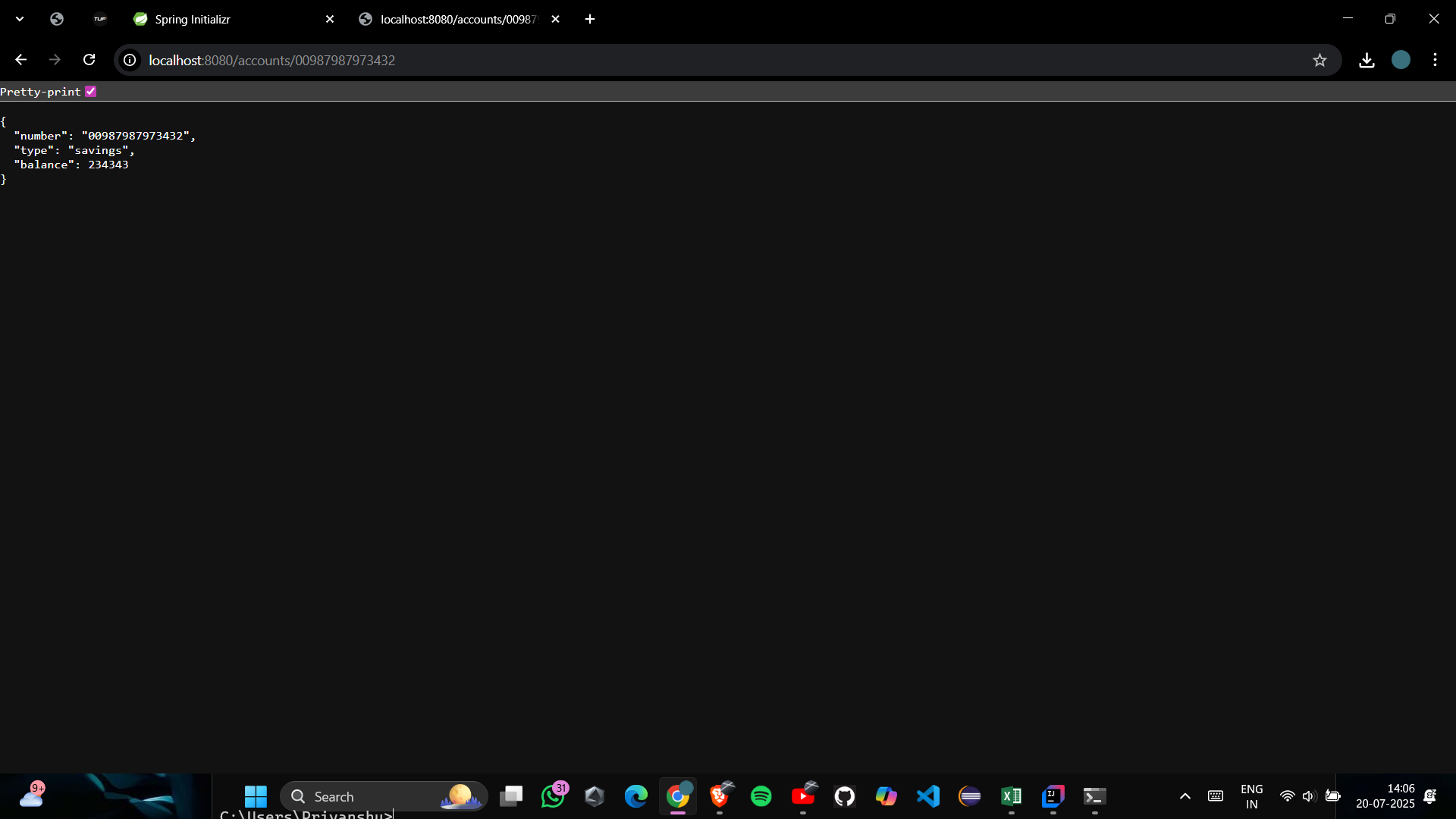
Dependencies: Spring Web, Spring Boot DevTools

## File: AccountController.java

package com.cognizant.account;  
  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Account getAccountDetails(@PathVariable String number) {  
 return new Account(number, "savings", 234343);  
 }  
  
 static 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; }  
 }  
}

## . IntelliJ Console Showing Running Services

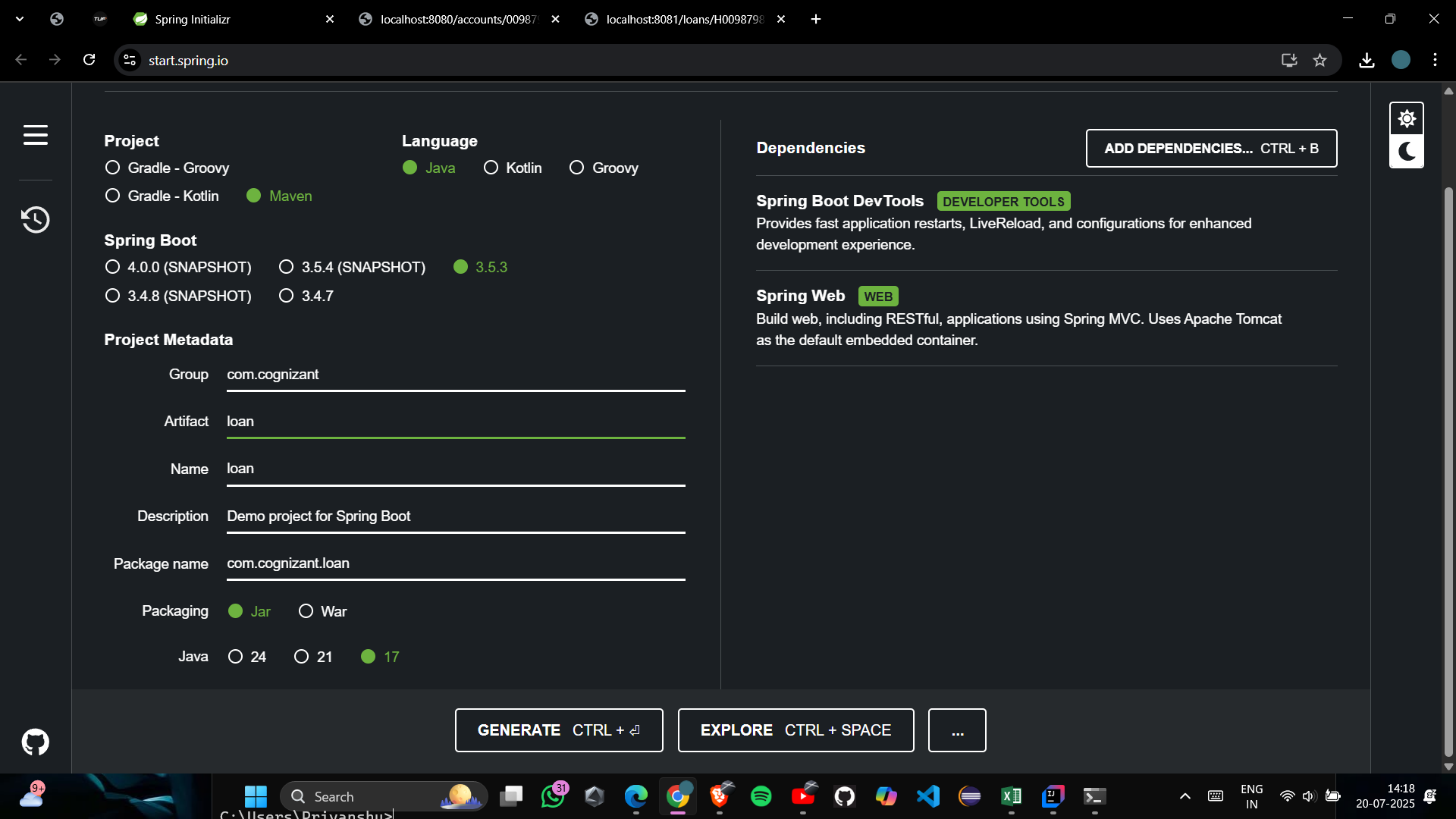
## . Account Service Output

Endpoint: http://localhost:8080/accounts/00987987973432

## 2. Spring Initializr Configuration – Loan Service

Group: com.cognizant

Artifact: loan

Dependencies: Spring Web, Spring Boot DevTools

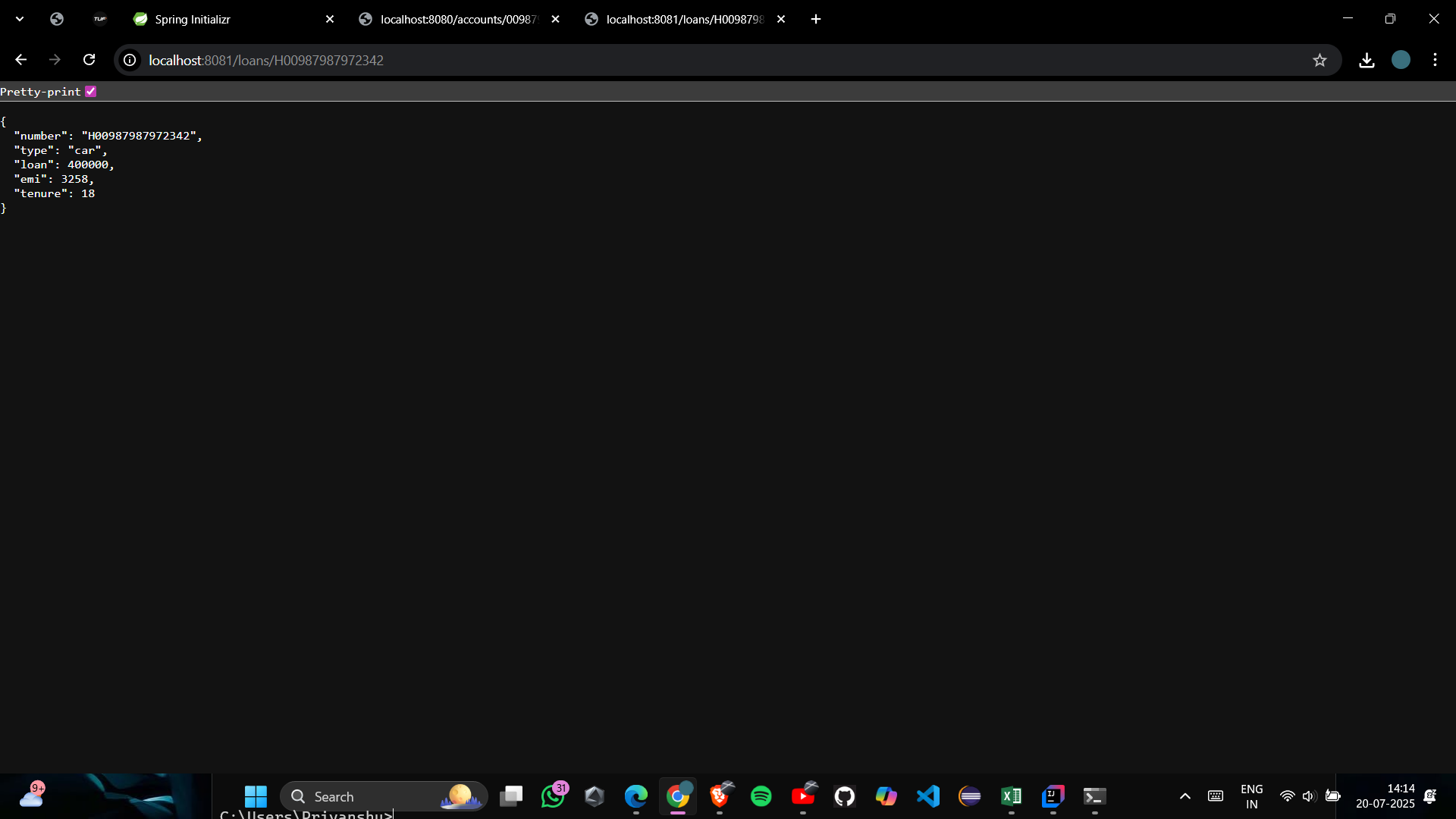
## File: LoanController.java

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

## 6. Loan Service Output

Endpoint: http://localhost:8081/loans/H00987987972342

{  
 "number": "H00987987972342",  
 "type": "car",  
 "loan": 400000,  
 "emi": 3258,  
 "tenure": 18  
}



## 7. Conclusion

Both microservices were successfully:

- Created using Spring Boot

- Built with Maven

- Run on independent ports

- Tested via browser using dummy responses