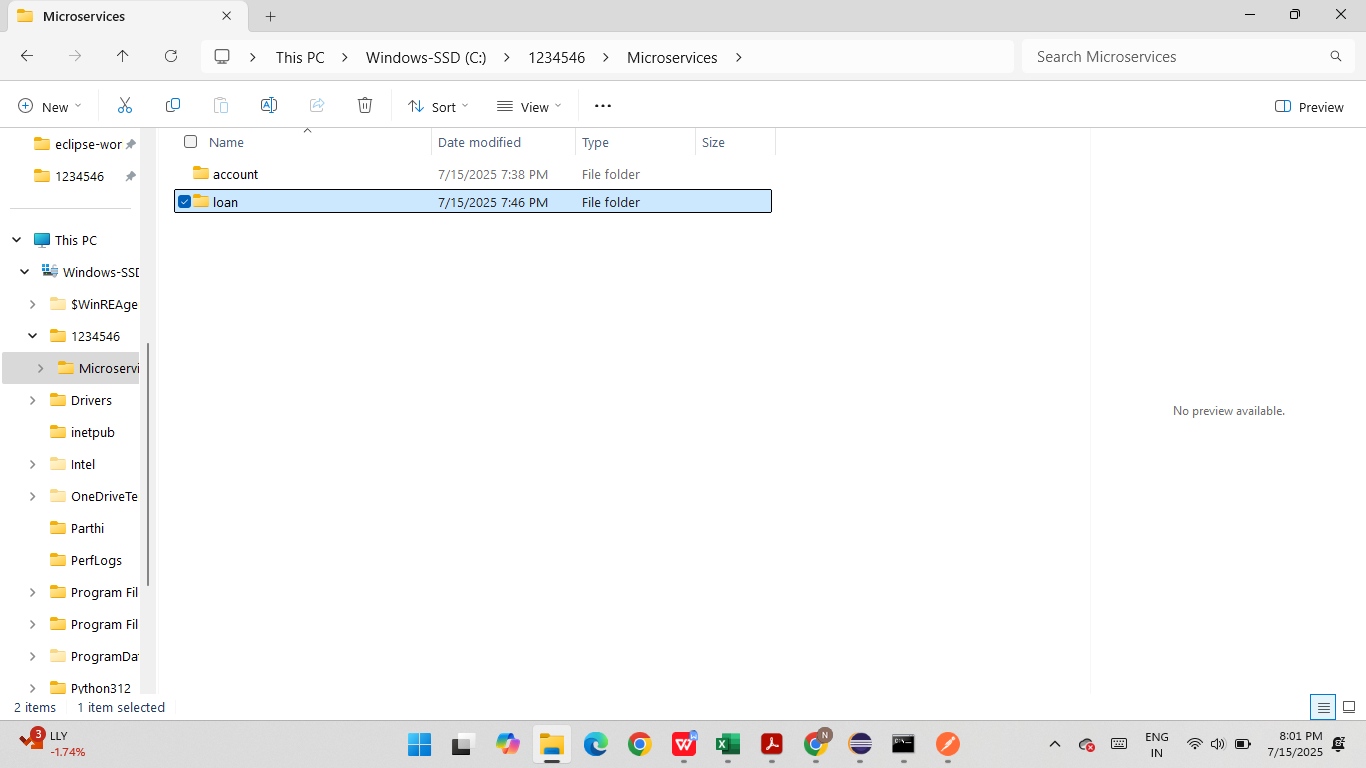
**Creating Microservices for account and loan**

In this hands on exercises, we will create two microservices for a bank. One microservice for handing accounts and one for handling loans.

**SOLUTION:**

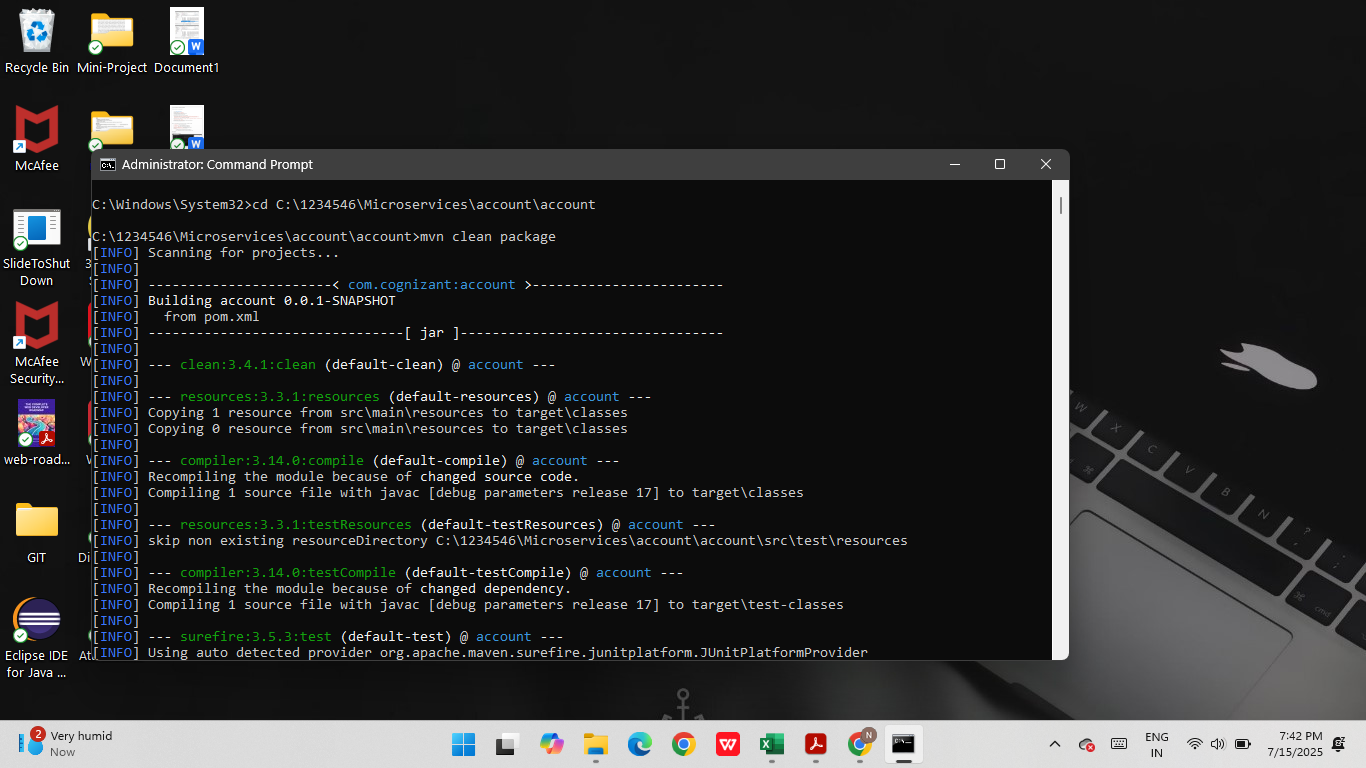
**ACCOUNT MICROSERVICE:**

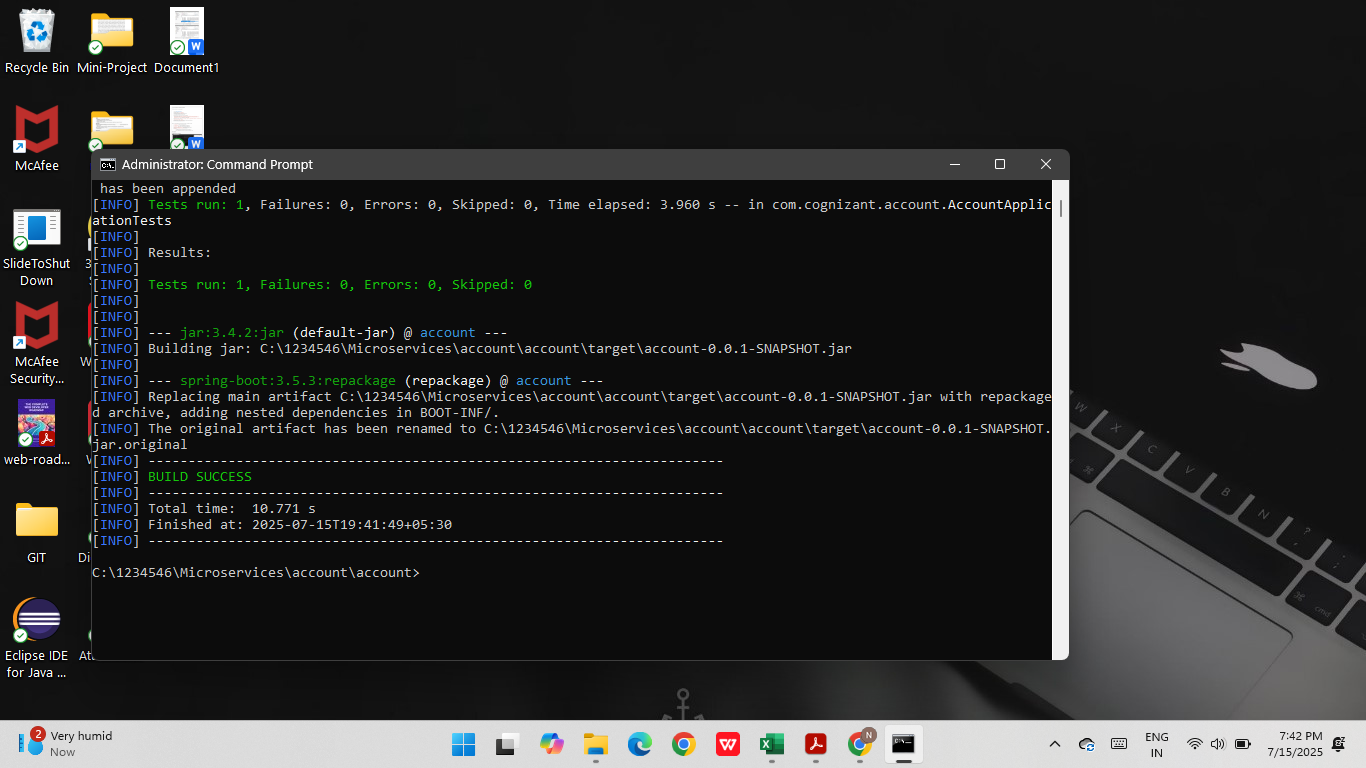
**STEP:1 FOLDER STRUCTURE SETUP**



STEP:2 Creating Account Microservice using Spring Initializr

* Extract the account.zip folder int the microservices folder.
* Run/ Setup the projec using command prompt





**Import Account Project into Eclipse**

**Package: com.cognizant.account**

**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; }

}

}

**AccountApplication.java**

**package** com.cognizant.account;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** AccountApplication {

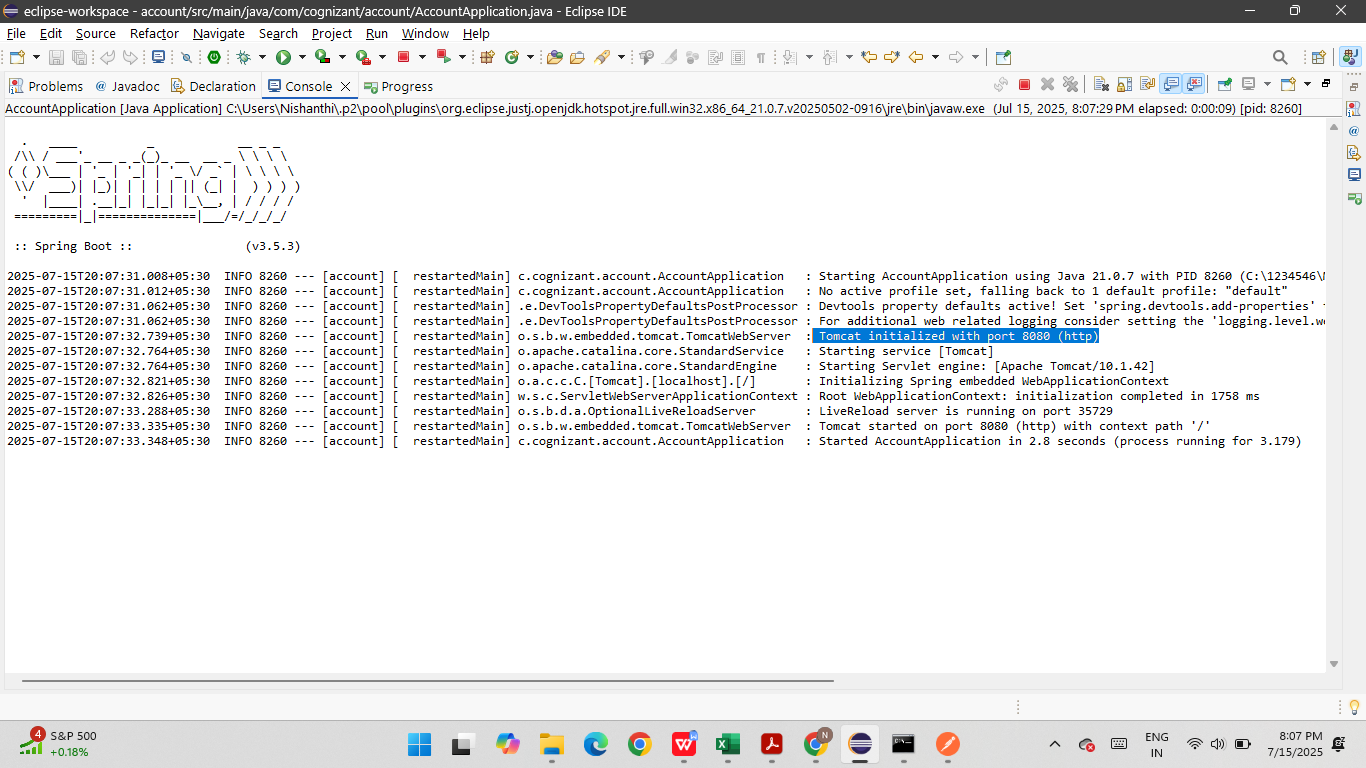
**public** **static** **void** main(String[] args) {

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

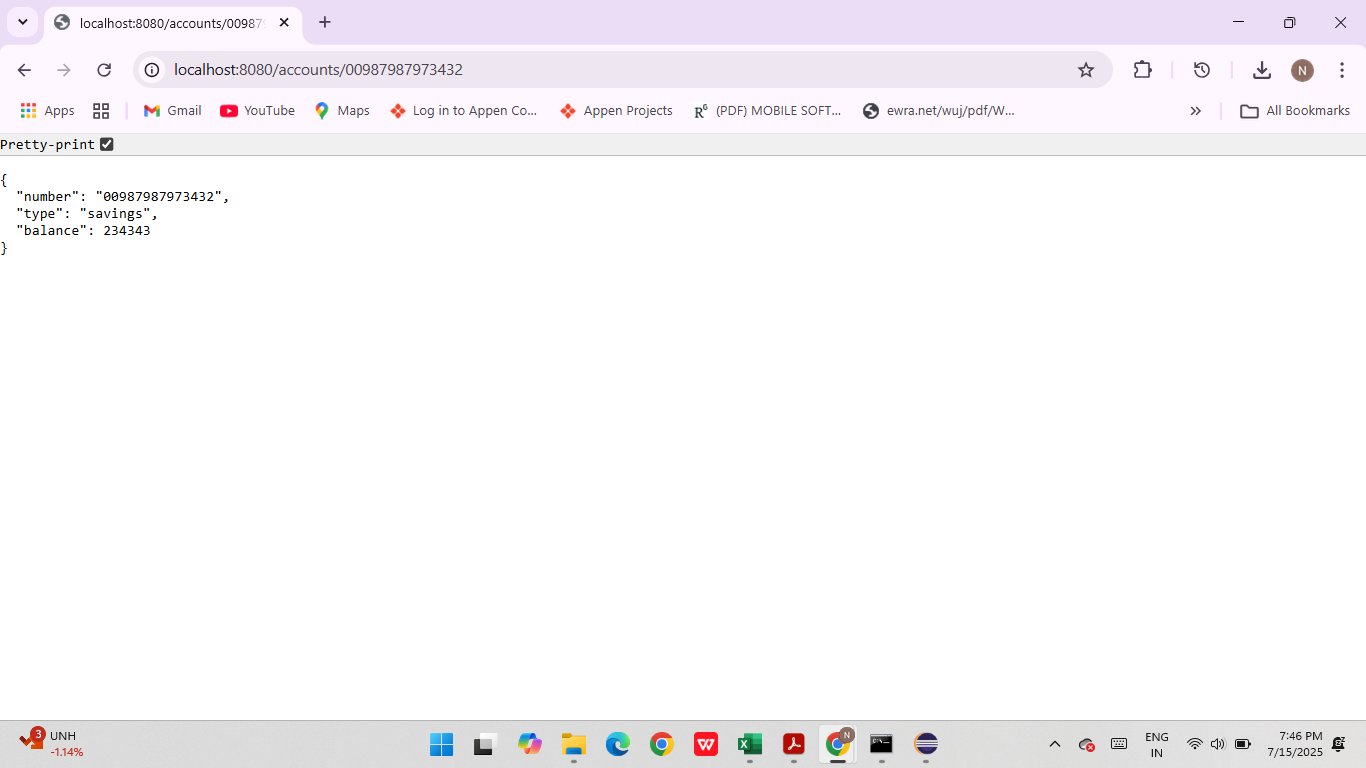
}

}

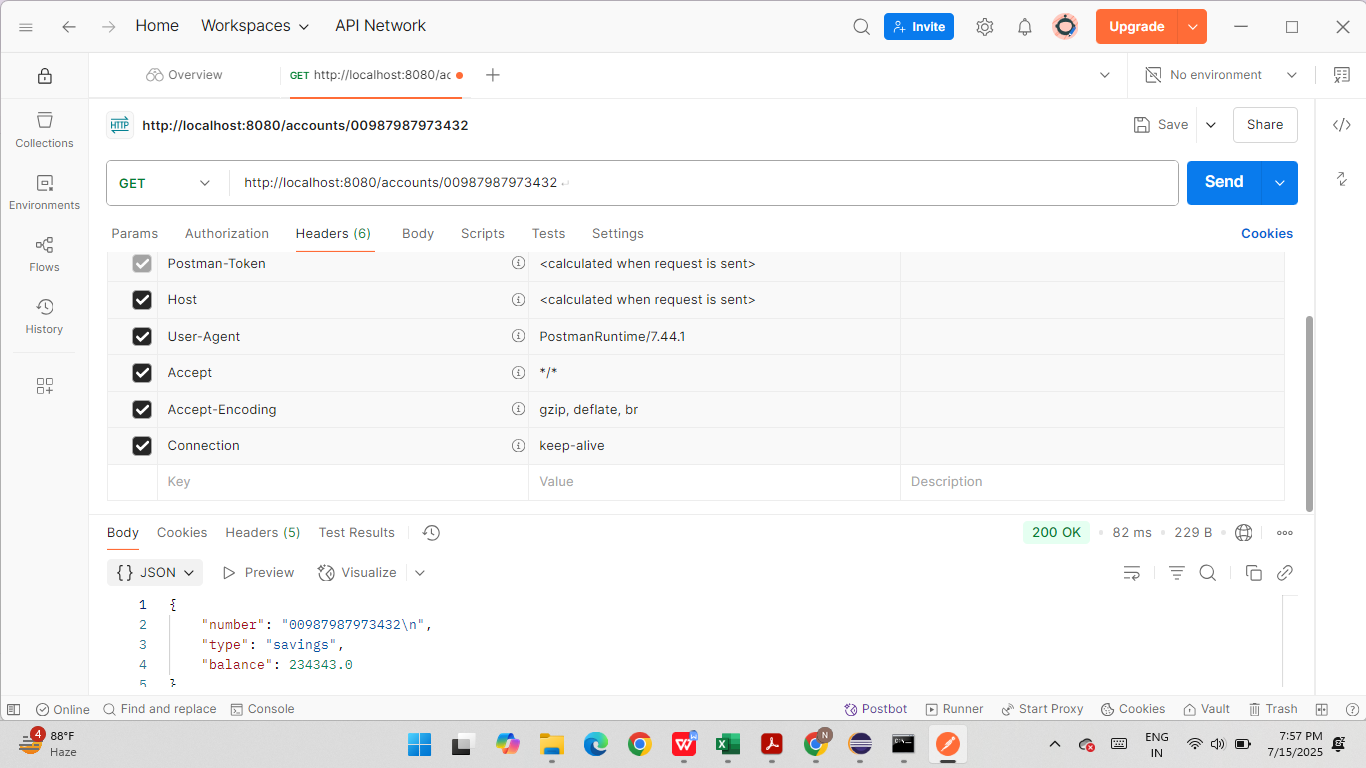
**OUTPUT:**



**CHROME OUTPUT**



**POSTMAN OUTPUT**



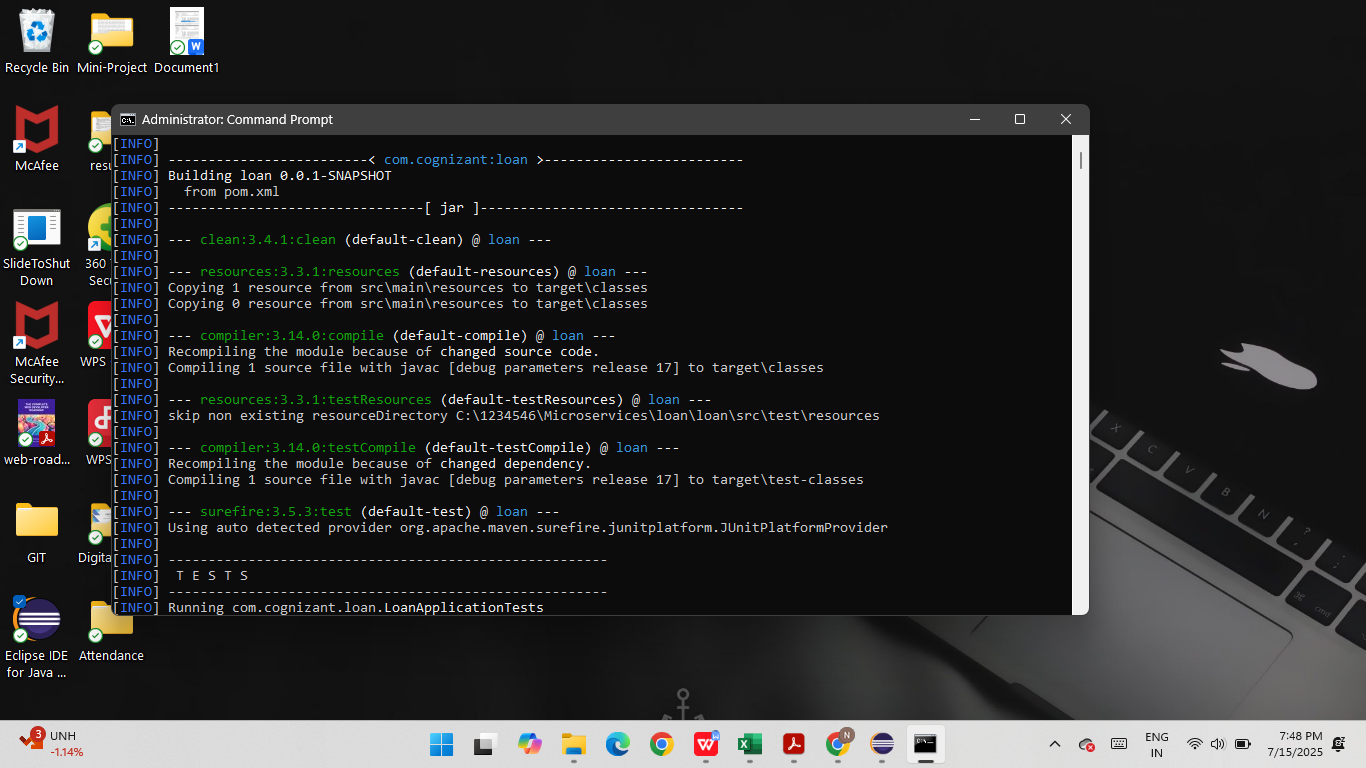
**LOAN MICROSERVICES**

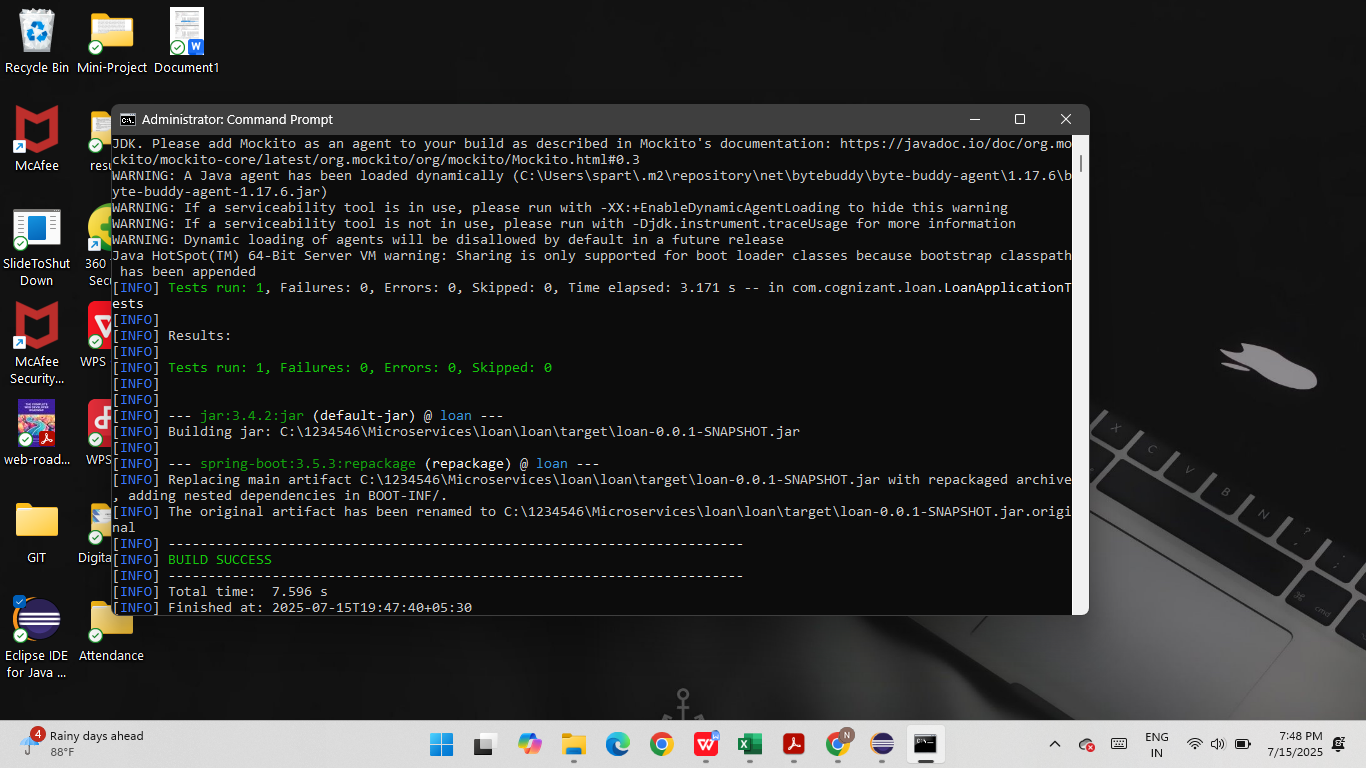
Step:1 Setup the folder structure.

Step:2 Follow the same Spring initializr steps and download the zip file

Step:3 Now move the extracted folder to the loan account.

Step:5 Setup the project using command prompt.





**Import loan project into Eclipse**

**Package: com.cognizant.loan**

**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; }

}

}

**LoanApplication.java**

**package** com.cognizant.loan;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** LoanApplication {

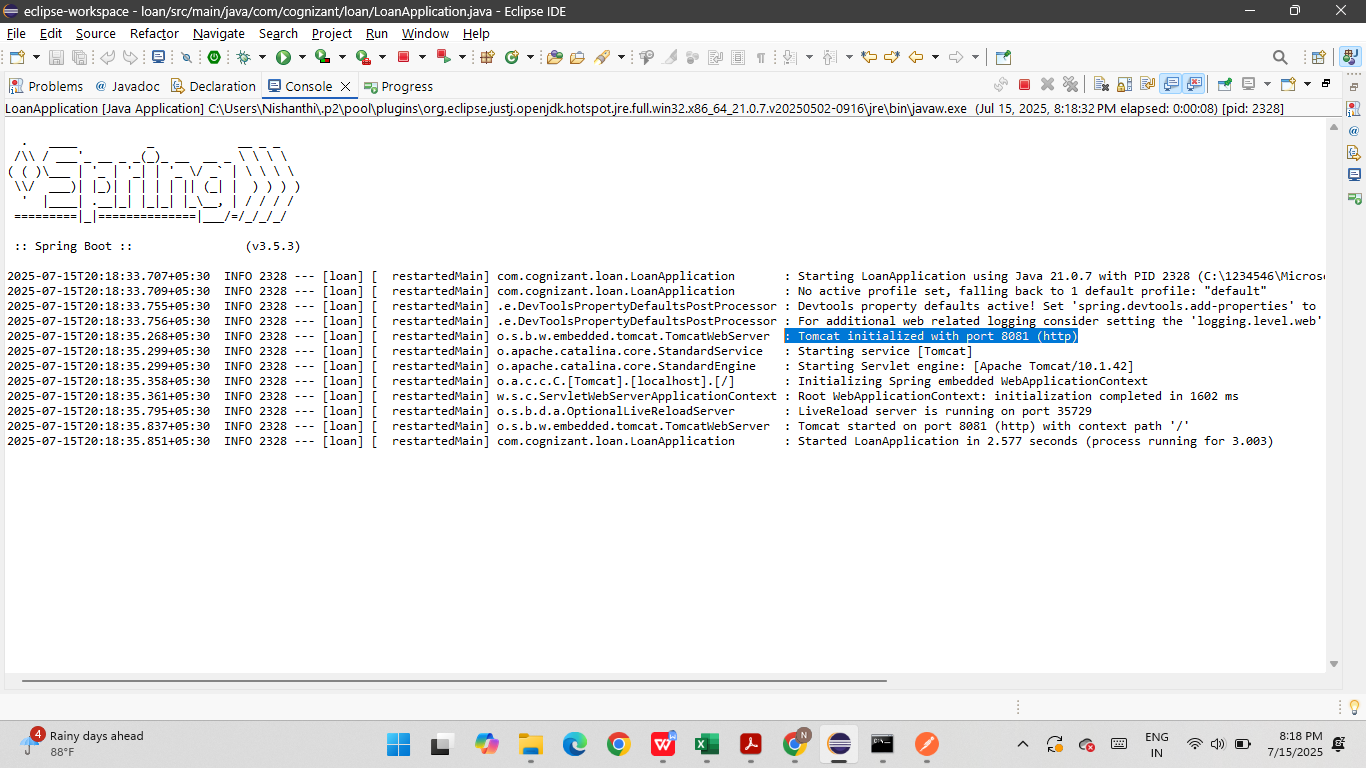
**public** **static** **void** main(String[] args) {

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

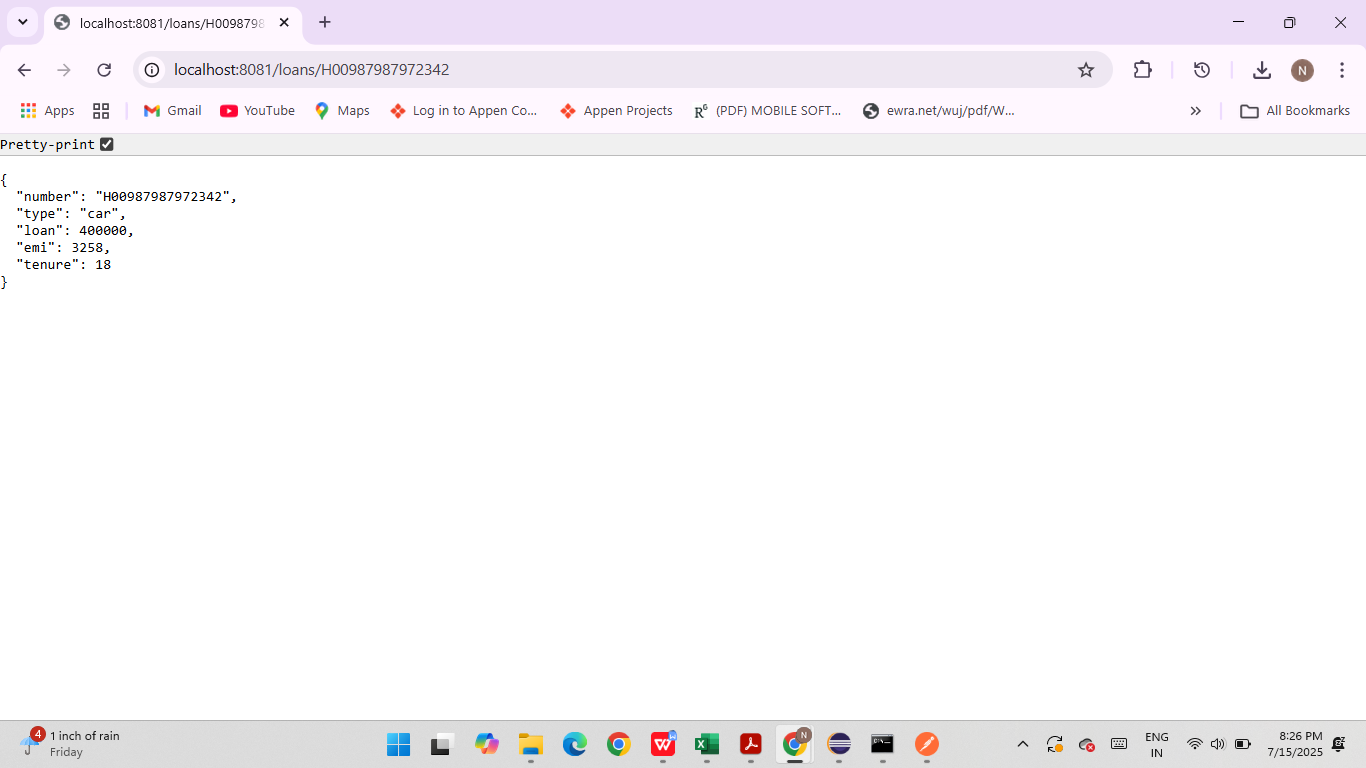
}

}

**OUTPUT**



**CHROME OUTPUT:**



**POSTMAN OUTPUT**

