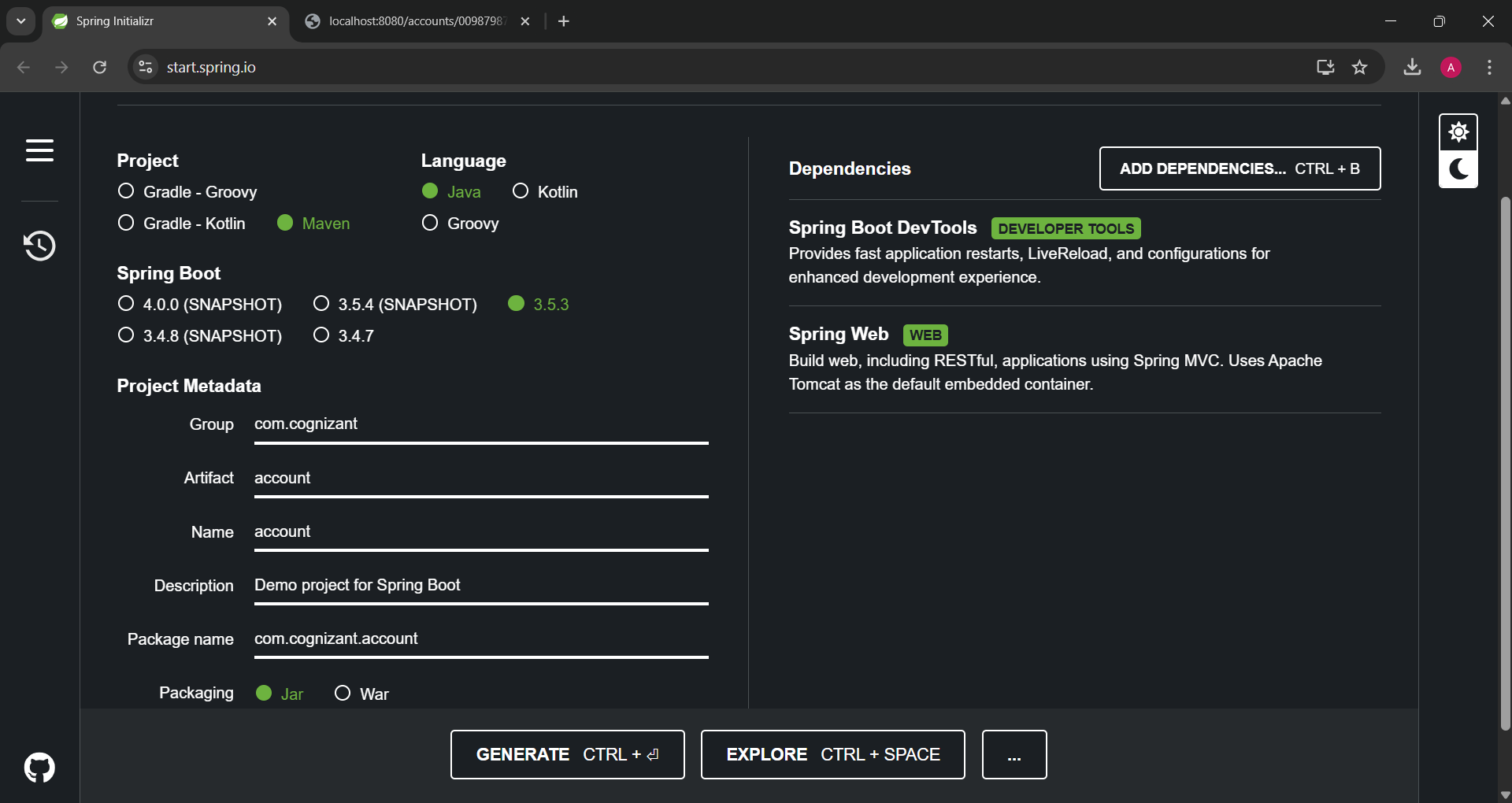
**WEEK 5 – MICROSERVCES WITH SPRINGBOOT3 AND SPRING CLOUD**

**EXERCISE 1: CREATING MICROSERVICES FOR ACCOUNT AND LOAN**

**STEP 1:** Create a folder in the system with the EmployeeID.

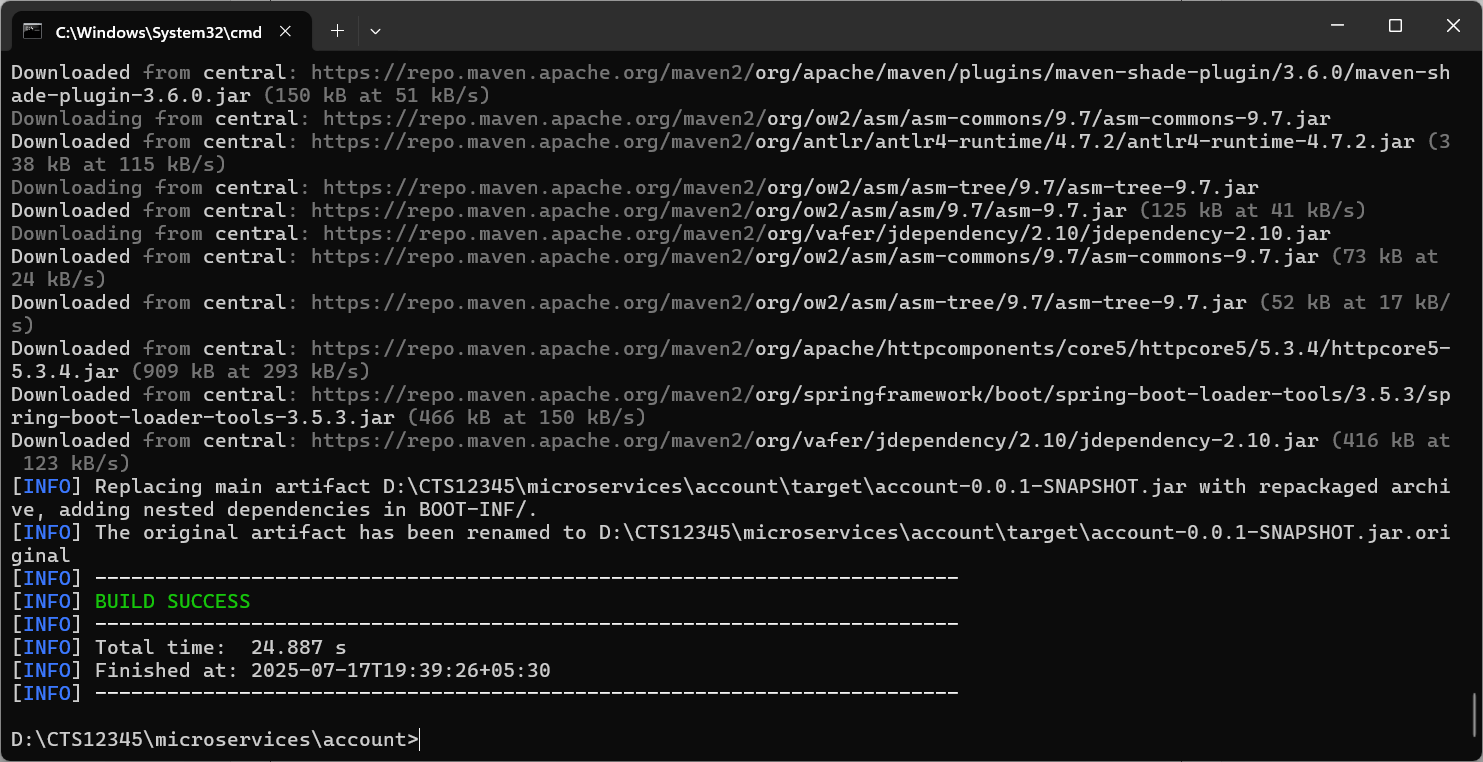
**STEP 2:** Again, Create a folder named microservies inside the same folder.

**STEP 3:** In Spring Initializer, create a project by adding dependenices and download the zip file.



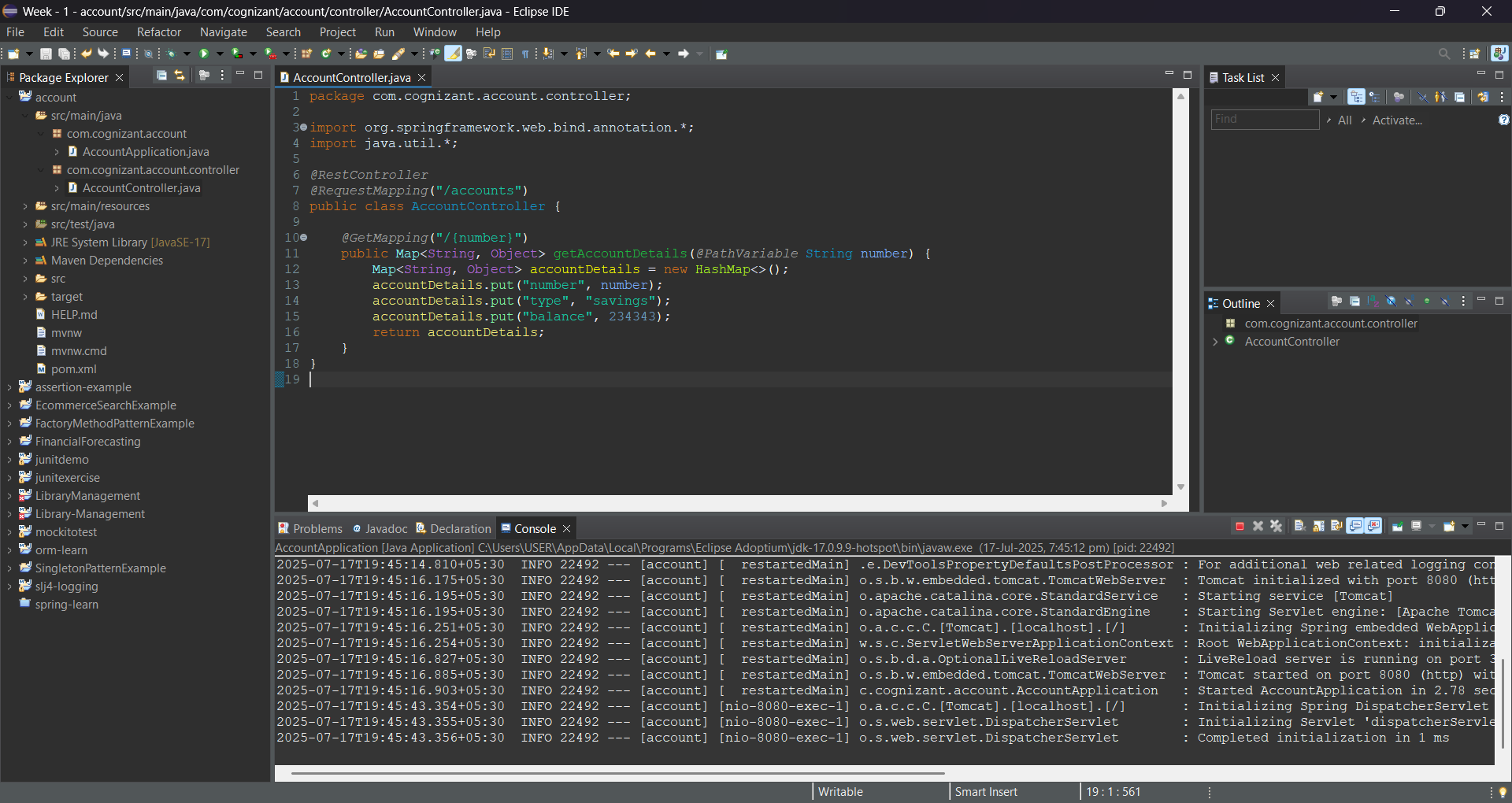
**STEP 4:** Extract the zip file and place it inside the microservies folder.

**STEP 5:** Run “mvn clean package” command in command prompt.



**STEP 6:** Import the folder to Eclipse IDE.

**STEP 7:** Create a controller package and create a class named AccountController.java inside the package.



package com.cognizant.account.controller;

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

import java.util.\*;

*@RestController*

*@RequestMapping*("/accounts")

public class AccountController {

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

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

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

accountDetails.put("number", number);

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

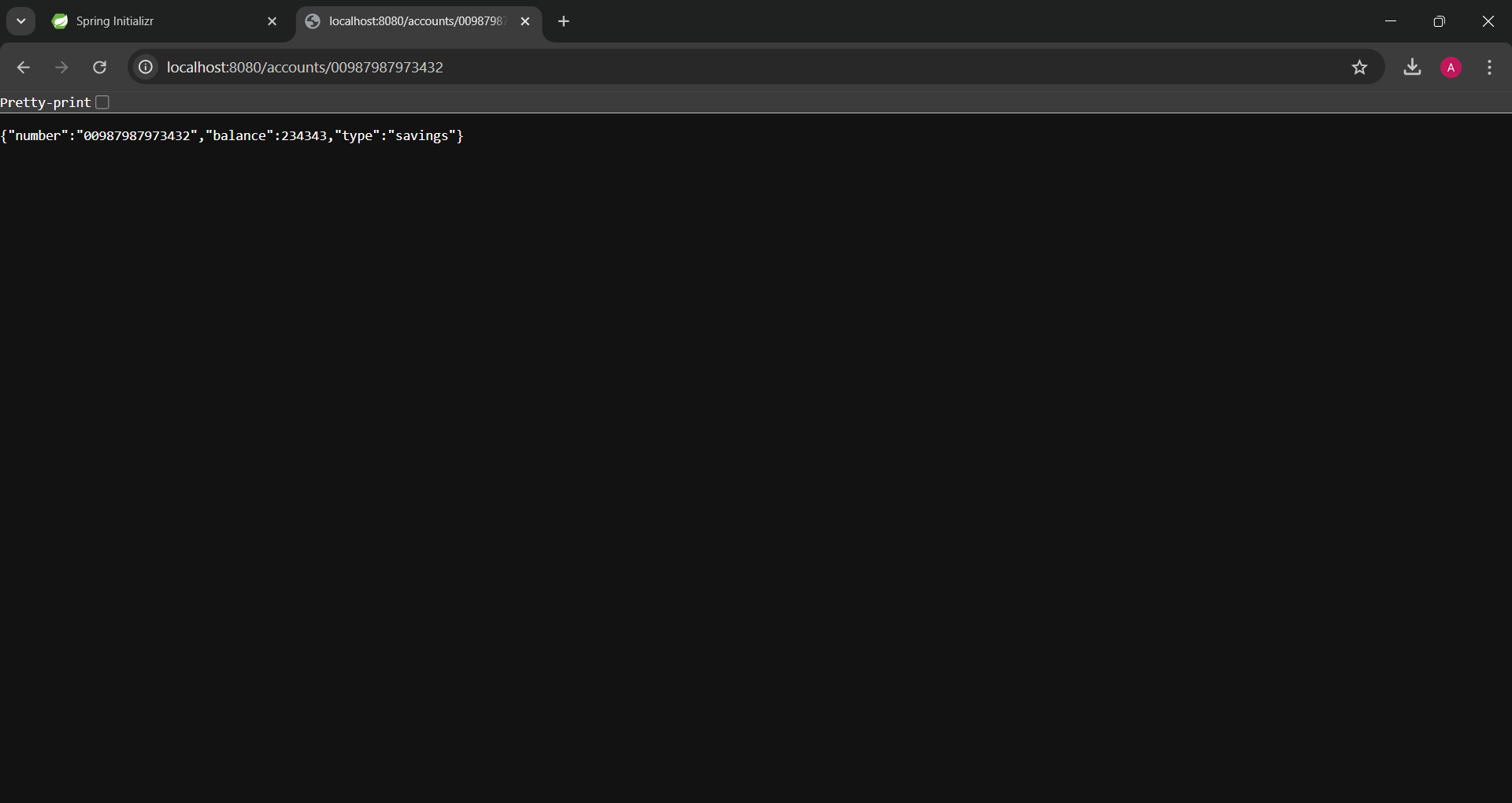
accountDetails.put("balance", 234343);

return accountDetails;

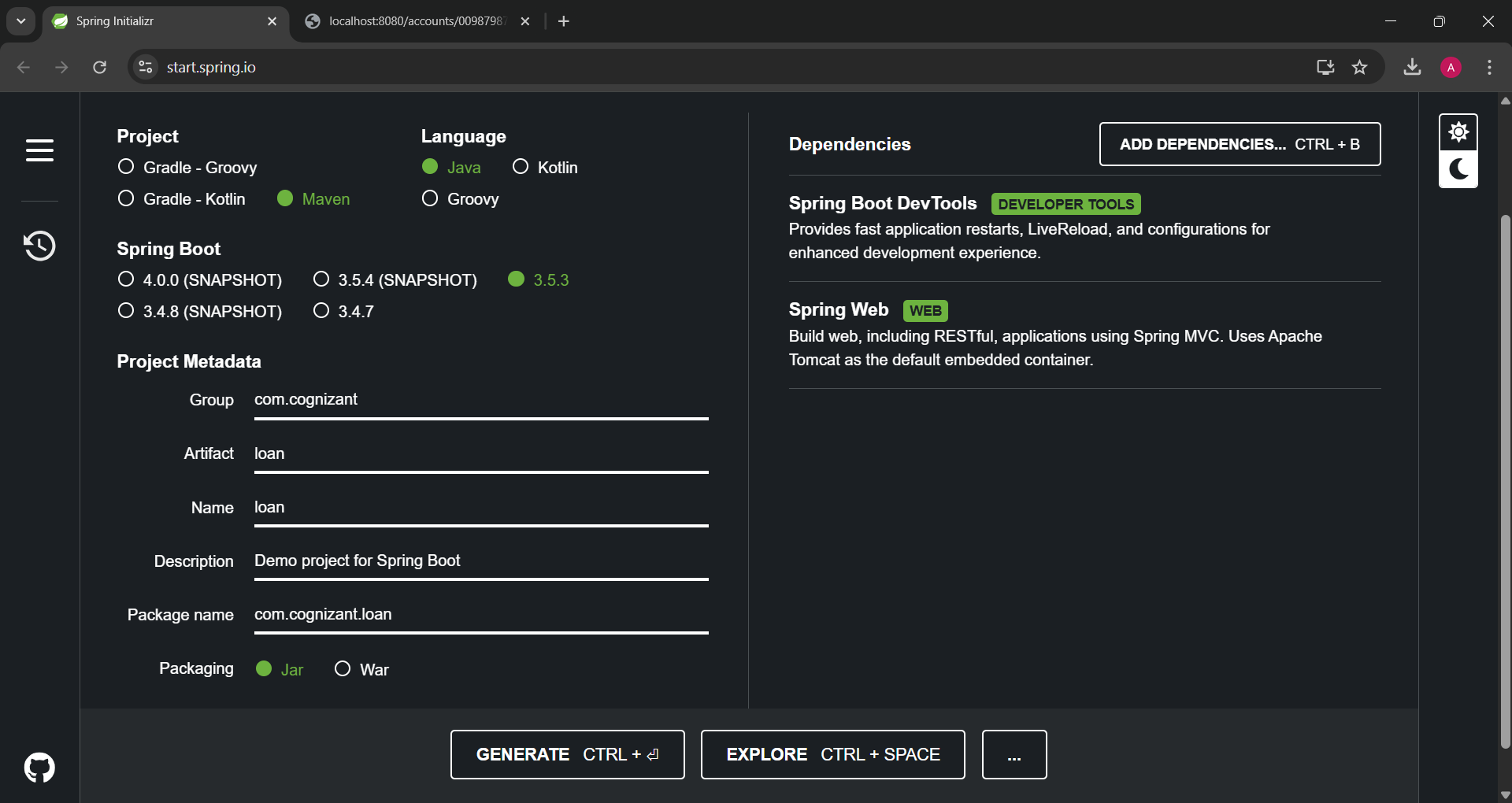
}

}

**STEP 8:** Run the application and view it in <http://localhost:8080/accounts/00987987973432>

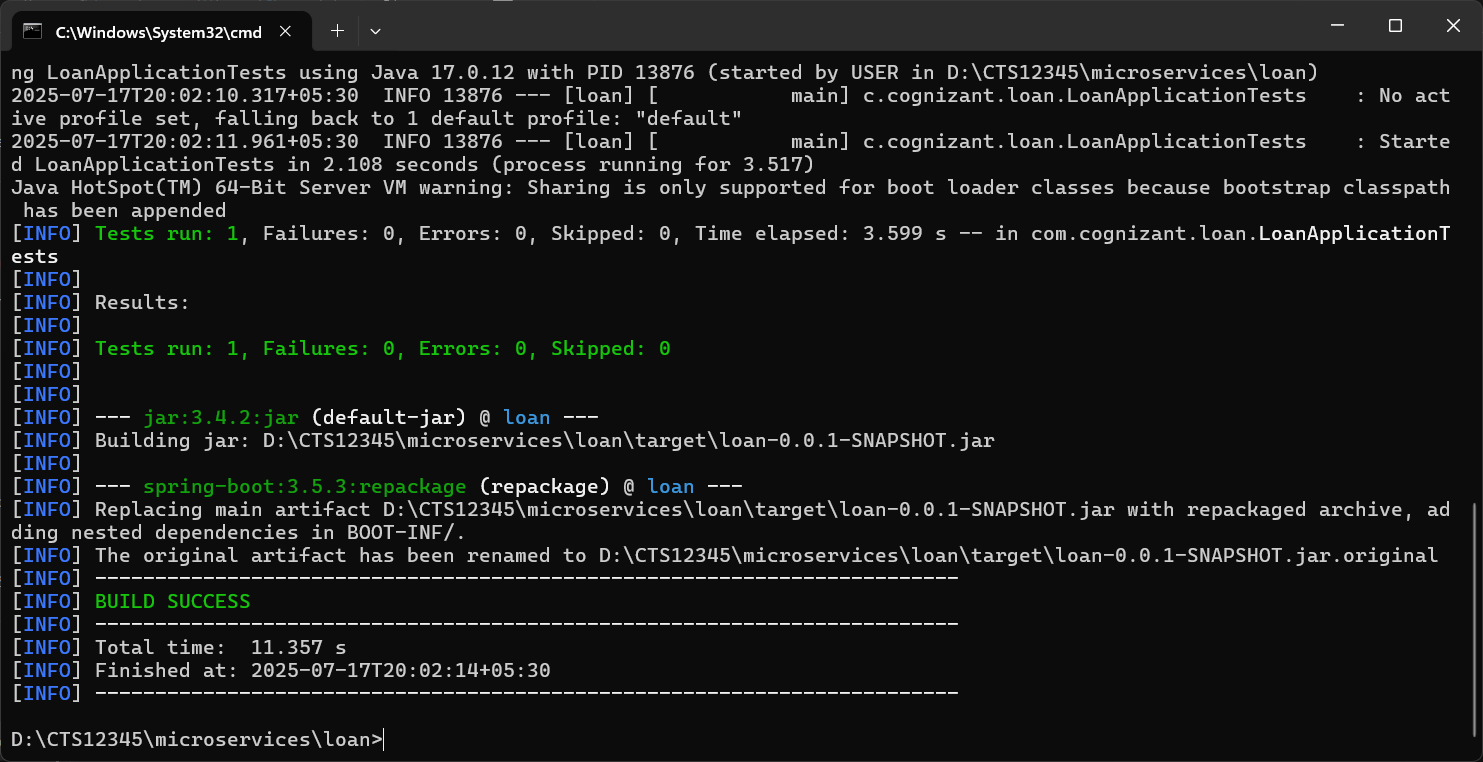


**STEP 9:** In Spring Initializer, create a project by adding dependenices and download the zip file.



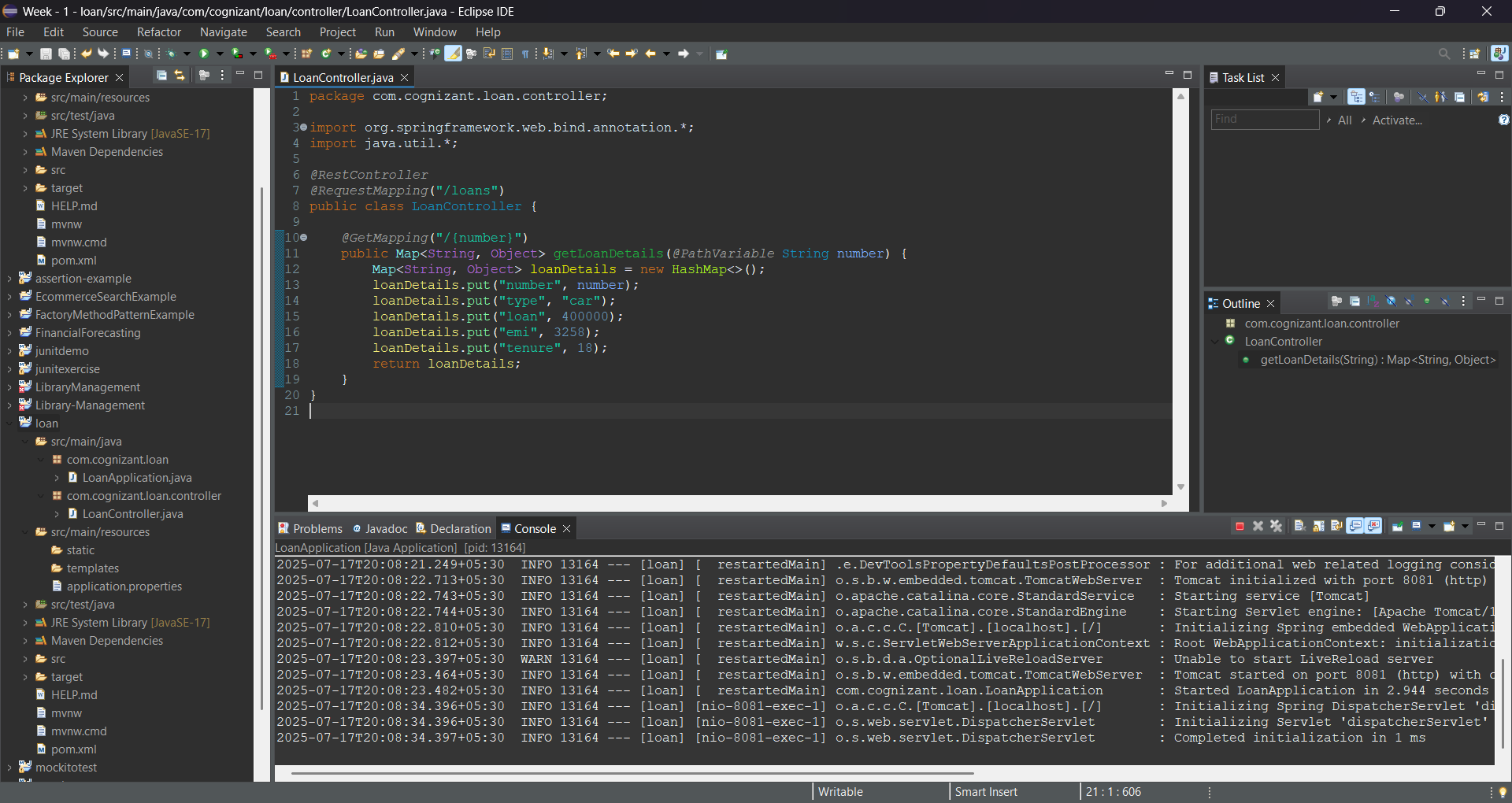
**STEP 10:** Extract the zip file and place it inside the microservies folder.

**STEP 11:** Run “mvn clean package” command in command prompt.



**STEP 12:** Import the folder to Eclipse IDE.

**STEP 13:** Create a controller package and create a class named LoanController.java inside the package.



package com.cognizant.loan.controller;

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

import java.util.\*;

*@RestController*

*@RequestMapping*("/loans")

public class LoanController {

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

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

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

loanDetails.put("number", number);

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

loanDetails.put("loan", 400000);

loanDetails.put("emi", 3258);

loanDetails.put("tenure", 18);

return loanDetails;

}

}

**STEP 14:** Run the application and view it in <http://localhost:8081/loans/H00987987972342>

