**Creating Microservices for account and loan**

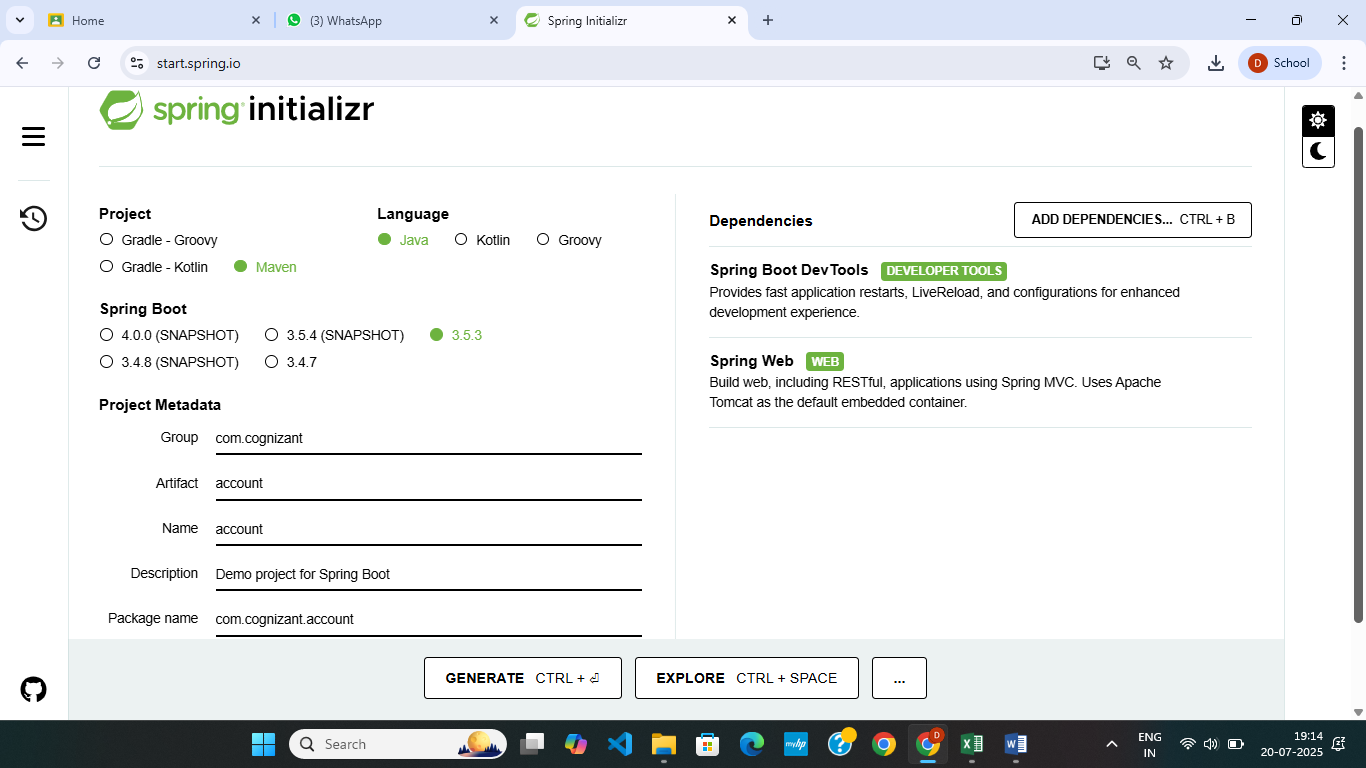
**Objective:**

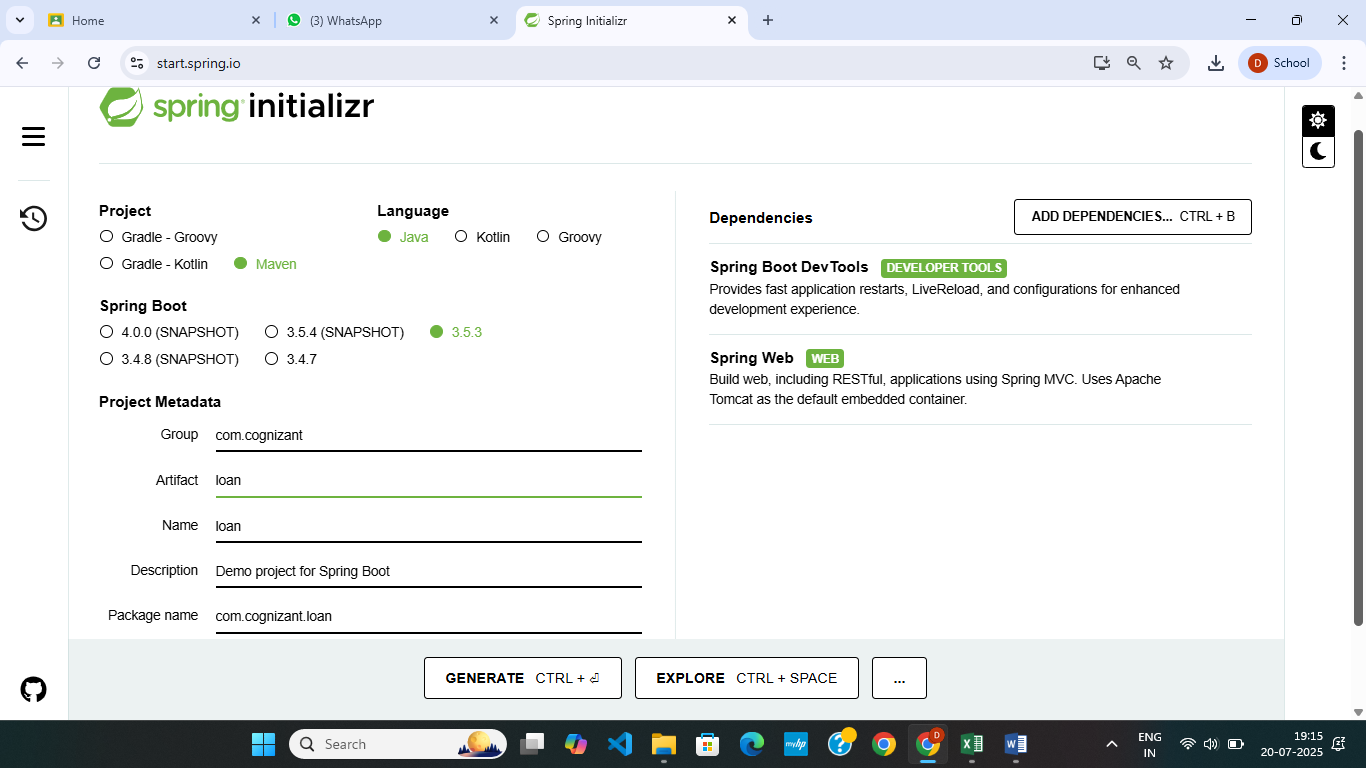
To create and run two independent Spring Boot microservices using “https://start.spring.io/”:

* AccountService – returns basic account details.
* LoanService – returns basic loan details.
* Include dependencies:

i)Spring Boot Dev Tools

ii)Spring Web





Each service should:

* Have its own @RestController.
* Run on a different port.(account ->port 8080, loan ->8081)

**Coding:**

**i)Project : account**

package com.cognizant.account.controller;

import java.util.HashMap;

import java.util.Map;

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

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

import org.springframework.web.bind.annotation.RequestMapping;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

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

account.put("number", number);

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

account.put("balance", 344763);

return account;

}

}

spring.application.name=account

server.port=8080

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);

}

}

**ii) Project: loan**

package com.cognizant.loan.controller;

import java.util.HashMap;

import java.util.Map;

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

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

import org.springframework.web.bind.annotation.RequestMapping;

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

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

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

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

loan.put("number", number);

loan.put("type", "gold");

loan.put("loan", 125000);

loan.put("emi", 2560);

loan.put("tenure", 8);

return loan;

}

}

spring.application.name=loan

server.port=8081

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LoanApplication {

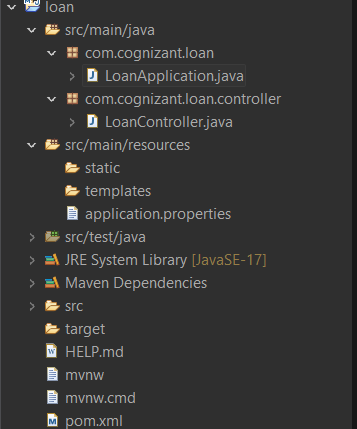
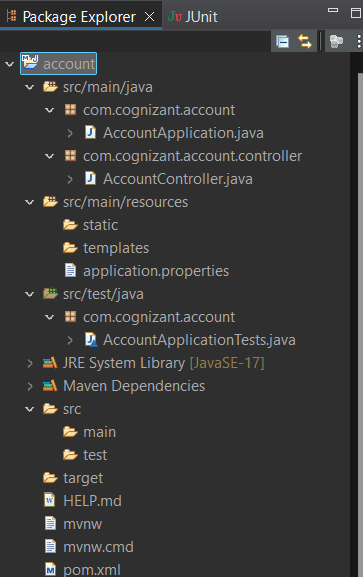
public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

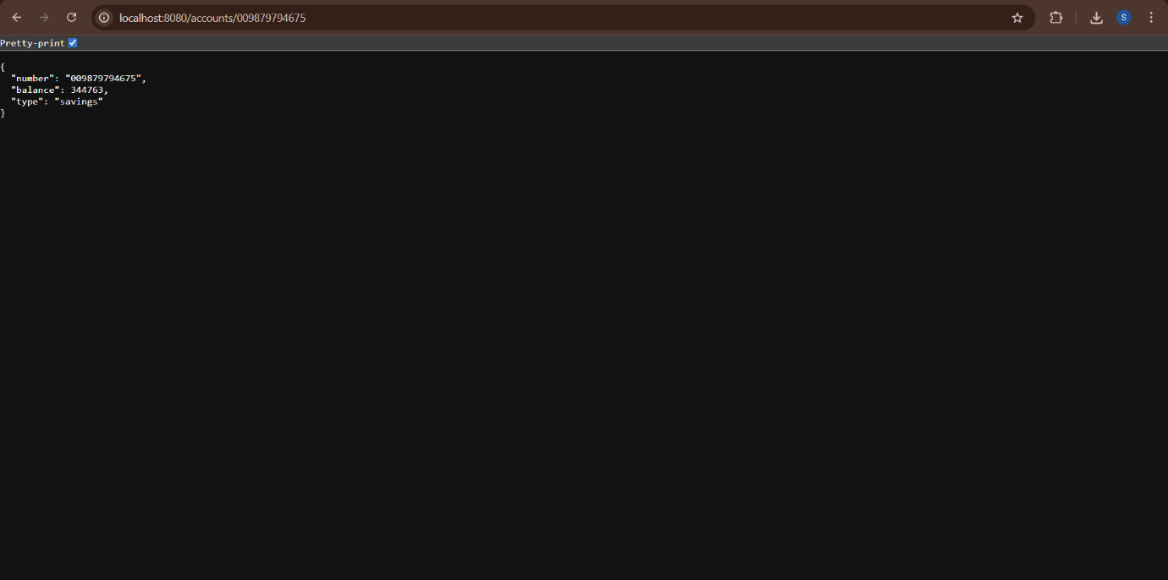
}

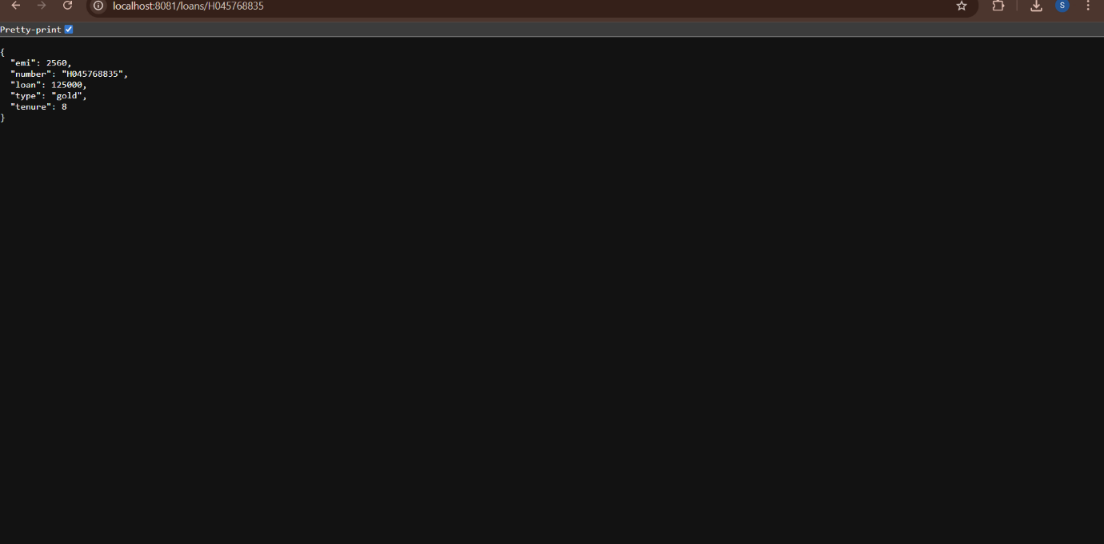
}

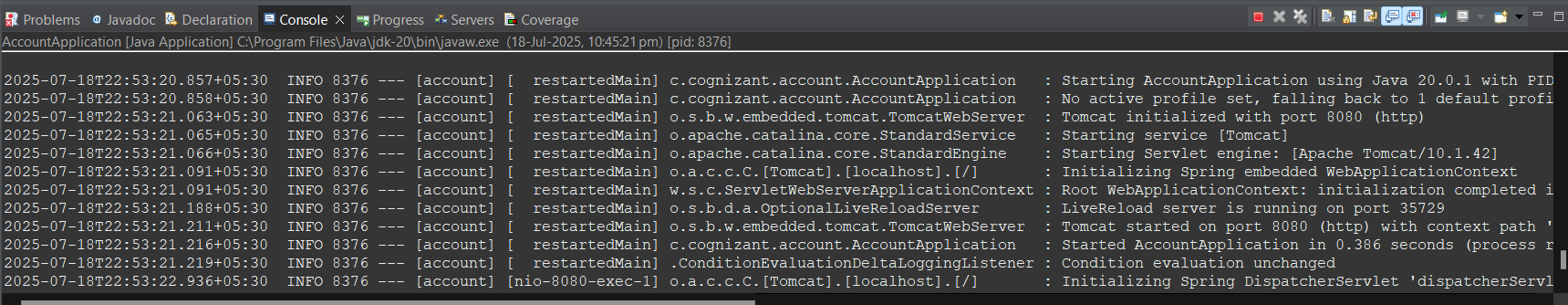
**File Structure:**



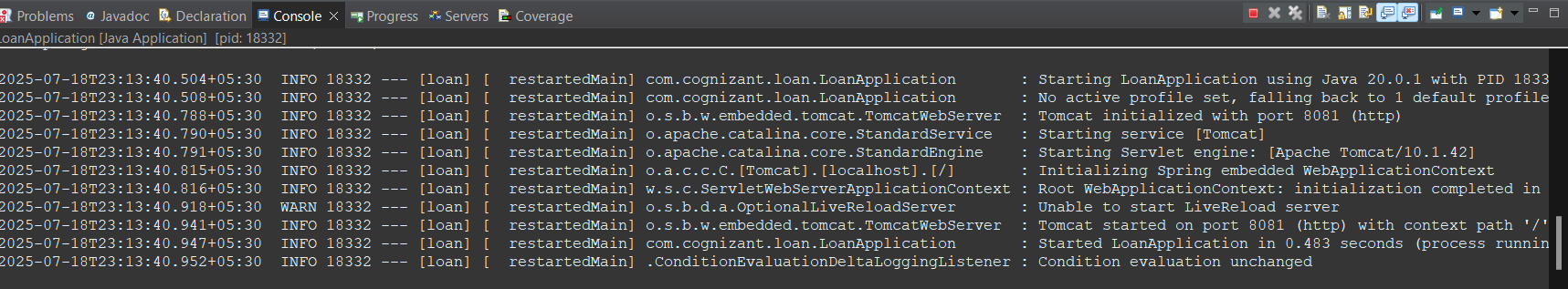
**Output:**













**ADDITIONAL EXERCISE:**

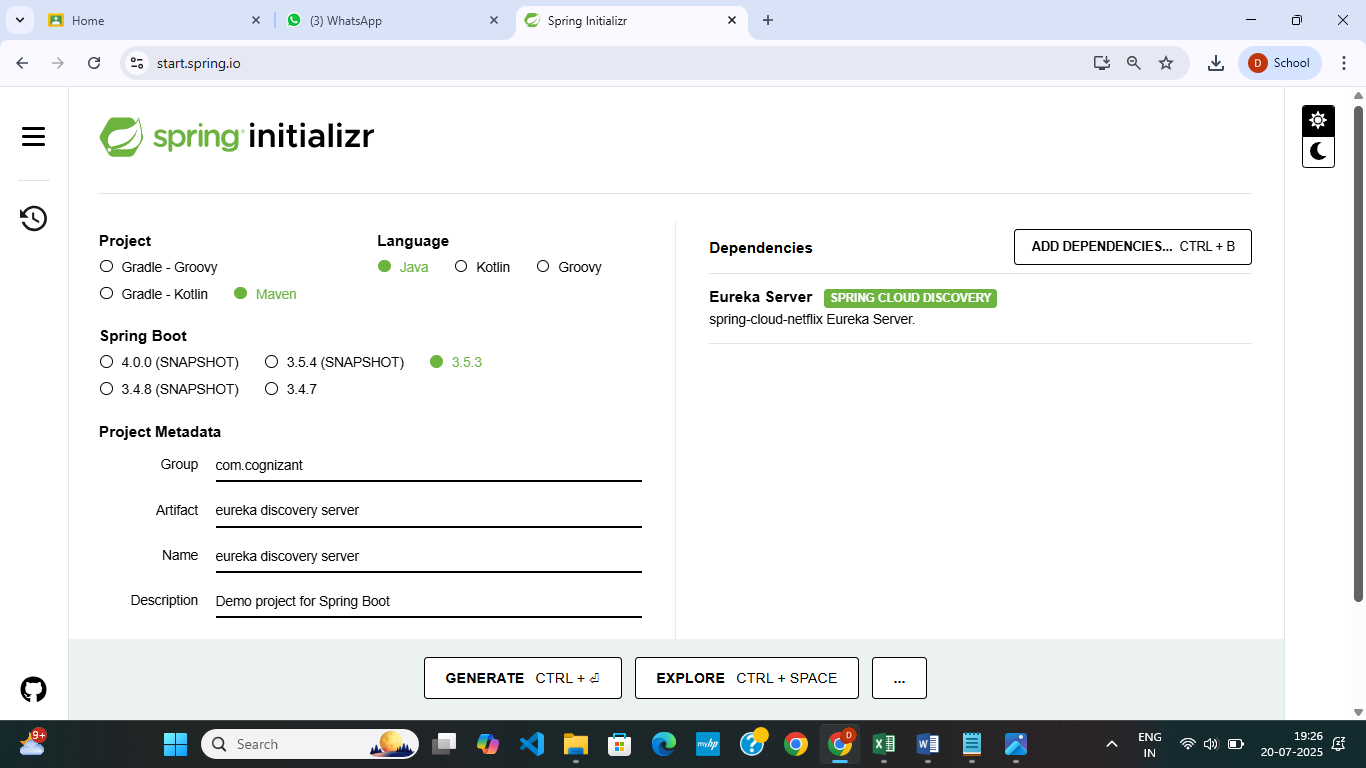
**Create Eureka Discovery Server and register microservices**

**Objective:**

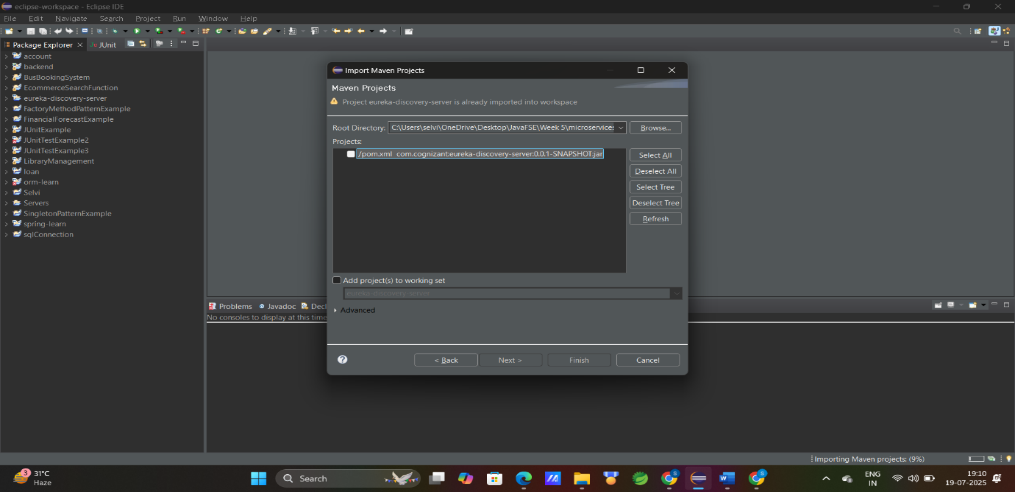
To create a **Eureka Discovery Server** for service registration and discovery, enabling microservices (like AccountService and LoanService) to dynamically register and communicate without hardcoding URLs.

**Steps involved:**

i)Create a Spring Project for Eureka-Discovery-Service using <https://start.spring.io/> with dependency EurekaServer



ii) Import project in Eclipse



**iii)Setup the Eruka-Discovery-Server**

**application.properties**

spring.application.name=eureka-discovery-server

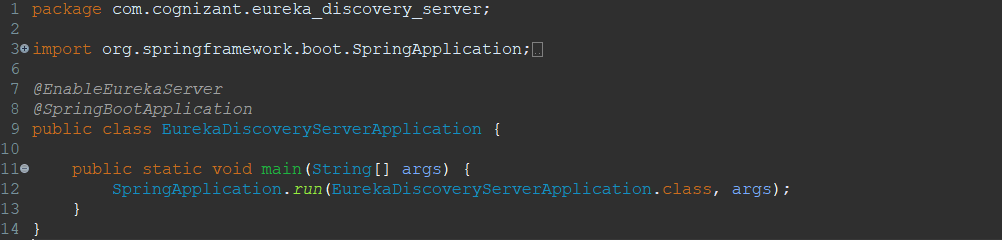
server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

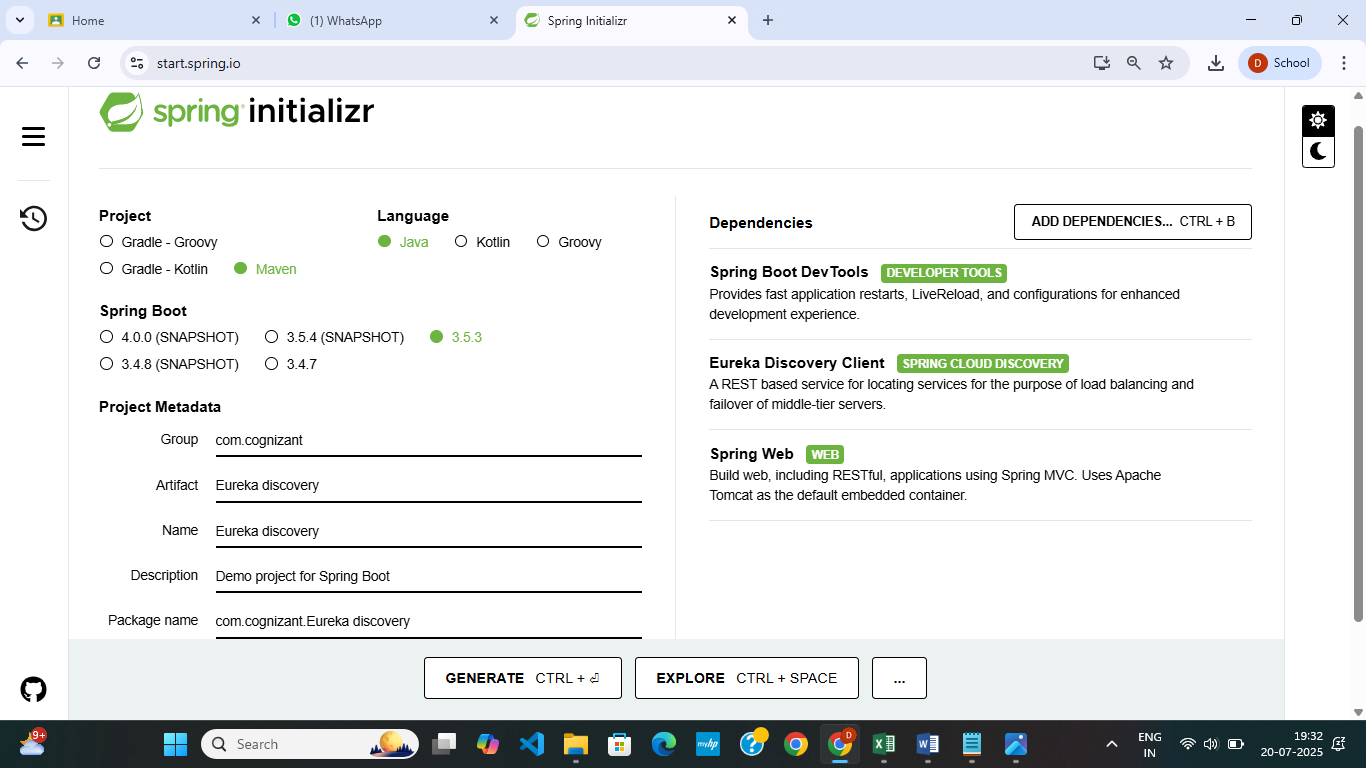
logging.level.com.netflix.eureka=OFF

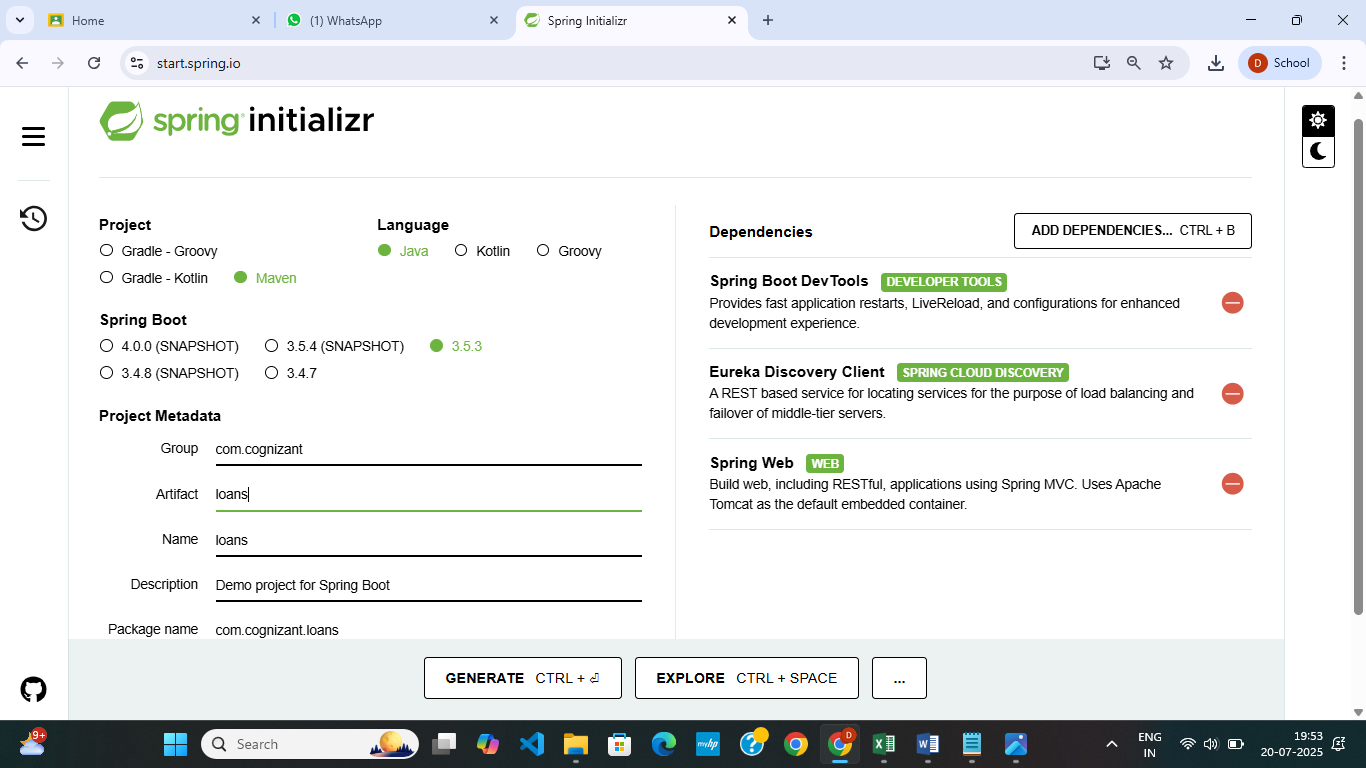
logging.level.com.netflix.discovery=OFF



iv)Create two microservices for accout and loan:

dependencies:Spring Boot DevTools, Eureka DiscoveryClient,SpringWeb





v)Setup accounts and loans

MicroService : Accounts

package com.cognizant.accounts;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountsApplication {

public static void main(String[] args) {

SpringApplication.run(AccountsApplication.class, args);

}

}

package com.cognizant.accounts.controller;

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

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

import org.springframework.web.bind.annotation.RequestMapping;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{accountNumber}")

public String getAccount(@PathVariable String accountNumber) {

return "Account details for " + accountNumber;

}

}

**Application.properties : accounts**

spring.application.name=accounts

server.port=8081

spring.application.name=account-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

MicroService: Loans

package com.cognizant.loans;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class LoansApplication {

public static void main(String[] args) {

SpringApplication.run(LoansApplication.class, args);

}

}

package com.cognizant.loans.controller;

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

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

import org.springframework.web.bind.annotation.RequestMapping;

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

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping

("/{loanNumber}")

public String getLoan(@PathVariable String loanNumber) {

return "Loan details for " + loanNumber;

}

}

**Application.properties : loans**

spring.application.name=loans

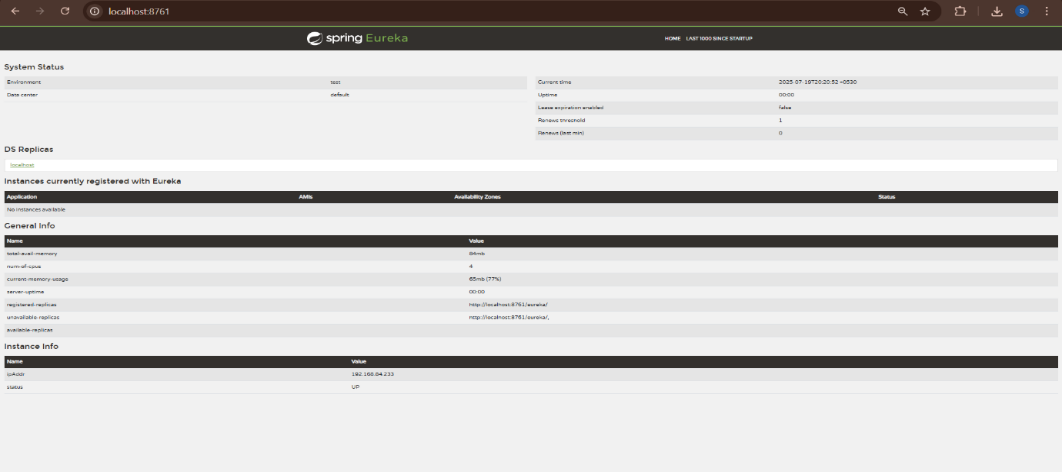
server.port=8082

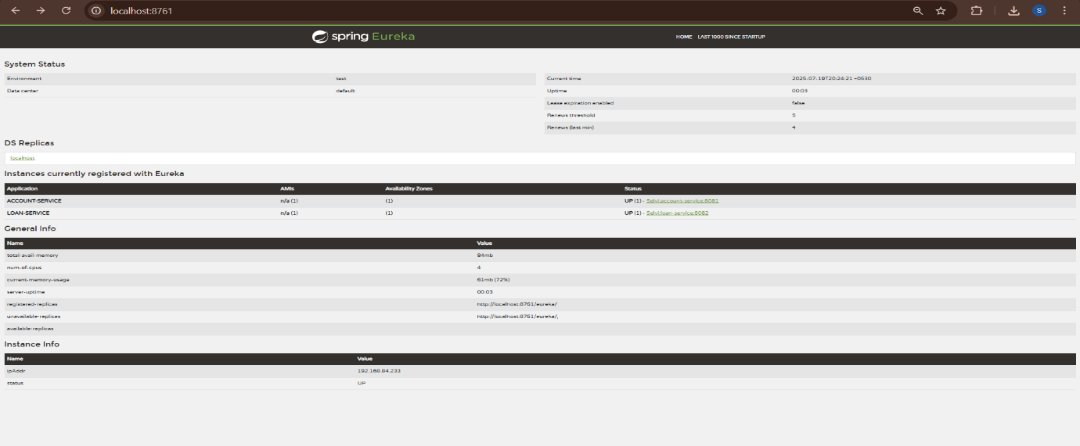
spring.application.name=loan-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

**OUTPUT:**

Port : <http://localhost:8761/>

**-** 



Port : <http://localhost:8081/accounts/123456>



Port : <http://localhost:8082/loans/123456>

