# Microservices (Mandatory hands-on)

Student Name: Yukta

## Q1. Creating Microservices for Account and Loan

Tools Used: Spring Initializr, Maven, Eclipse, Java 22

* ➤ Steps Followed:

1. 1. Created two Spring Boot projects using Spring Initializr: one named `account`, another `loan`.
2. 2. Added dependencies: Spring Web, Spring Boot DevTools
3. 3. Extracted into folders: C:\Users\YUKTA001\microservices\account and loan respectively
4. 4. Set ports in `application.properties` (8080 for account, 8081 for loan)
5. 5. Implemented REST controller in both services to return dummy JSON responses.

AccountController.java

@RestController  
public class AccountController {  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
 }  
}

LoanController.java

@RestController  
public class LoanController {  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "car",  
 "loan", 400000,  
 "emi", 3258,  
 "tenure", 18  
 );  
 }  
}

## Q2. Create Eureka Discovery Server and Register Microservices

Tools Used: Spring Initializr, Spring Cloud, Eclipse

* ➤ Steps Followed:

1. 1. Created a new Spring Boot project named `eureka-discovery-server` with dependency: Eureka Server
2. 2. Annotated main class with `@EnableEurekaServer`
3. 3. Configured application.properties with port 8761 and disabled self-registration
4. 4. Added Eureka Client dependencies to both account and loan microservices
5. 5. Annotated both services' main classes with `@EnableDiscoveryClient`
6. 6. Registered both services in Eureka by setting `spring.application.name` and Eureka URL

Sample application.properties for account-service

server.port=8080  
spring.application.name=account-service  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

✅ Both account and loan services appeared in Eureka dashboard successfully.

## Output Screens

➡️ Output from: http://localhost:8080/accounts/123

{  
 "number": "123",  
 "type": "savings",  
 "balance": 234343  
}

➡️ Output from: http://localhost:8081/loans/123

{  
 "number": "123",  
 "type": "car",  
 "loan": 400000,  
 "emi": 3258,  
 "tenure": 18  
}