**Microservices with API Gateway - Assignment Submission**

**Student Details:**

* **Name**: Pavithra P
* **Course**: Cognizant Java Microservices Track
* **Topic**: Microservices Architecture using Spring Boot, Eureka Server, and API Gateway

**1. Project Structure**

Cognizant\_Java/

├── loan-service/

├── greet-service/ (Optional)

├── eureka-server/

└── api-gateway/

**2. Eureka Server**

**application.properties**:

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

**Main Class**:

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

**3. Loan Service**

**application.properties**:

server.port=8888

spring.application.name=loan-service

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

**Main Class**:

@SpringBootApplication

@EnableDiscoveryClient

public class LoanServiceApplication {

public static void main(String[] args) {

SpringApplication.run(LoanServiceApplication.class, args);

}

}

**Sample REST Endpoint**:

@GetMapping("/loans/{id}")

public Loan getLoan(@PathVariable String id) {

return new Loan(id, "Personal", 50000);

}

**4. API Gateway**

**application.properties**:

server.port=8080

spring.application.name=api-gateway

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

**application.yml**:

spring:

cloud:

gateway:

routes:

- id: loan-service

uri: http://localhost:8888

predicates:

- Path=/loan/\*\*

**Main Class**:

@SpringBootApplication

@EnableDiscoveryClient

public class ApiGatewayApplication {

public static void main(String[] args) {

SpringApplication.run(ApiGatewayApplication.class, args);

}

}

**5. pom.xml Summary**

Each module contains relevant dependencies:

**loan-service / api-gateway:**

* spring-boot-starter-web
* spring-cloud-starter-netflix-eureka-client
* spring-cloud-starter-gateway (in gateway)
* spring-boot-starter-actuator (optional)

**eureka-server:**

* spring-cloud-starter-netflix-eureka-server

**6. Screenshots / Output**

