**Technology and Concept Which Used in Assignment :->**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.Microservices

2.Database Migration

3.Dependency Injection

4.Api Gateway

5.LoadBalancer

6.Logging

7.Authentication and Authorization

8.Exception Handling

9.Docker support

10.Synchronice Communication between Microservices.

11.Eureka Server.

--------------------------------------------------------------------------------------------------------------------

**Assignment Overview**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I have Created one ProductService with database and one ProductDetailsService with database, In ProductService database model there are two entities "ProductId and ProductName", In ProductDetailService database model there are four entities "ProductId, Price, Size and Design".

ProductServiece is responsible for having product with their id and name and admin can add, remove and edit these entities in ProductServiece.

ProductDetailService is responsible for having the size, price and Design of Product and also Admin can Add,Remove and Edit these entities in ProductDetailService while Removing entity from ProductServcie The entity of ProductDetailService will also Remove using "Synchronous Communication" using ProductId Entity.

JWT Token based Authentication Is Used in which Identity Server is responsible for generating token and Policy(Role) based Authorization has applied in adding, Removing and Updating the Entities in Microservices Only Admin can Add, Remove and Update entities in both the Services.

ApiGateway -> It is Configure using Ocelot LoadBalancer roundrobin technique is used for both the services and also Fault Tolerance Circuit Breaker is used for both Services.

Exception Handling and logging is Used in both the services action method.

Docker support is given to every service and ApiGateway.

All Services is Registered in Eureka Server.

Identity Server is Responsible for generating Jwt Token to Validate Services.

------------------------------------------------------------------------------------------------------------------------------

**Postman Url to check the services :->**

**For ProductService :->**

http://localhost:43280/api/product -- To get all the Product from ProductServcie. (Get request)

[Admin Authorize] http://localhost:43280/api/product -- To Add all the Product In ProductServcie. (Post request)

[Admin Authorize] http://localhost:43280/api/product -- To Edit all the Product from ProductServcie. (Put request)

[Admin Authorize] http://localhost:43280/api/product/{id} -- To Remove all the Product from ProductServcie and ProductDetailService using Synchronous Communication. (Delete request)

**For ProductDetailService :->**

http://localhost:15586/api/ProductDetails -- To get all the product from ProductDetailService and Product. (Get request)

[Admin Authorize] http://localhost:15586/api/ProductDetails -- To Add all the product In ProductDetailService. (Post request)

[Admin Authorize] http://localhost:15586/api/ProductDetails -- To Edit all the product From ProductDetailService.(Put request)

http://localhost:15586/api/ProductDetails/AddCart -- To Add Product In Cart. (Post request)

http://localhost:15586/api/ProductDetails/GetAllCarts -- To Get All the Carts From Cart Inventory. (Get request)

**For ApiGateway :->**

http://localhost:33556/api/product (Get request

http://localhost:33556/api/product [Admin Authorize](Post request)

http://localhost:33556/api/product [Admin Authorize](Put request)

http://localhost:33556/api/product/{id} [Admin Authorize](Delete request)

http://localhost:33556/api/ProductDetails (Get request)

http://localhost:33556/api/ProductDetails [Admin Authorize](Post request)

http://localhost:33556/api/ProductDetails [Admin Authorize](Put request)

http://localhost:33556/api/ProductDetails/AddCart (Post request)

http://localhost:33556/api/ProductDetails/GetAllCarts (Get request)

**Identity Server Url To Get The Jwt Token For Authorization :->**

https://localhost:5001/.well-known/openid-configuration -- To Get the Token Url.

https://localhost:5001/connect/token -- To Get The Token to Validate Service with the Parameters given below for ProductService.

Key Value

grant\_type client\_credentials

scope ProductService

client\_id demosession

client\_secret training

https://localhost:5001/.well-known/openid-configuration -- To Get the Token Url.

https://localhost:5001/connect/token -- To Get The Token to Validate Service with the Parameters given below for ProductDetailService.

Key Value

grant\_type client\_credentials

scope ProductDetailService

client\_id demosession2

client\_secret training

Tried but not applied :--

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I tried to use Asynchronous Communication between my services and I configured both the services for asynchronous communication

using RabbitMq but it only created the queue in RabbitMq and didn't send the data and also not received the data so I comment all the configuration which I applied in both the services and used synchronous communication between services.

I give all the services to Docker Support and created container and image of it but in my machine Docker Software in not running properly.