



An-Najah National University

Faculty of Engineering

Computer Engineering Department

Distributed Operation Systems

Project part1

Student's Name:

AreejAmjad sawalha

Ibtisam kharrousheh

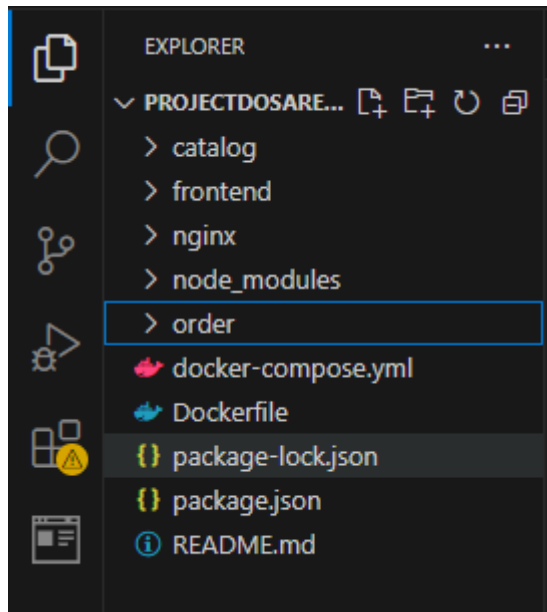
Student's ID:

12028958

12028305

Introduction:

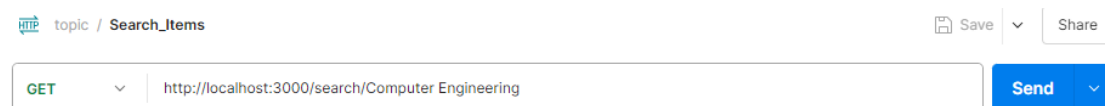
The files are divided in this way:



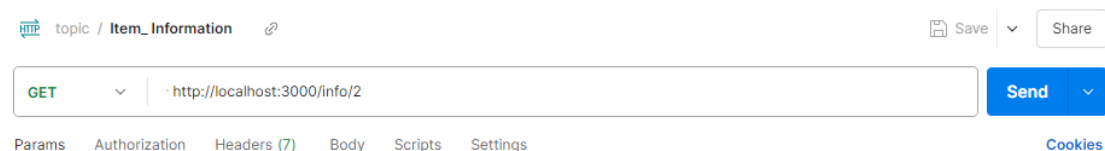
1) Fronted service:

This server has three operations:

1. Search: the catalog server receives the request and returns the item.



2. Information: the catalog server receives the request and returns the information.



3. Purchase:

HTTP topic / Purchase_Item Save Share

POST http://localhost:3000/purchase/1 Send

2) Order service:

```
POST createorder
GET getorderlist
GET GetOrderByld
PUT updatetheorder
DEL DeleteOrderByld
```

For example api order:

HTTP topic / createorder Save Share

POST http://localhost:5002/order/create Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "customer_name": "Areej",
3   "order_date": "2024-08-05",
4   "total": 166.75,
5   "items": "Item1, Item2, Item3"
6 }
7
```

HTTP topic / getorderlist Save Share

GET http://localhost:5002/order/list Send

HTTP topic / GetOrderByld Save Share

GET http://localhost:5002/order/7 Send

3)Catalog service:

```
GET All_Item
GET ByTopic
GET Byitem_id
PUT Item_UpdatebyID
DEL DeletebyID_item
```

For example api catalog:

HTTP topic / ByTopic Save Share

GET http://localhost:5001/catalog/query/topic/Computer Engineering Send

HTTP topic / Byitem_id Save Share

GET http://localhost:5001/catalog/query/item/3 Send

HTTP topic / Item_UpdatebyID Save Share

PUT http://localhost:5001/catalog/update/3 Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "stock": 5,
3   "price": 100
4 }
```

Running the project in Docker:

Building docker:

docker build -t my-app.

Creating common network to allow services to communicate:

docker network create projectPart1-nett

Running docker container for frontend server at port 3000:

docker run --name=frontend -p 3000:3000 --network=projectPart1-nett -it -v ./:/home my-app

npm install express

npm install axios

npm i sqlite3

node app.js

Running docker container for Catalog server at port 5001:

```
docker run --name=frontend -p 5001:5001 --  
network=projectPart1-nett -it -v ./home my-app  
npm install express  
npm install axios  
npm i sqlite3  
node app.js
```

Running docker container for Order server at port 5002:

```
docker run --name=frontend -p 5002:5002 --  
network=projectPart1-nett -it -v ./home my-app  
npm install express  
npm install axios  
npm i sqlite3  
node app.js
```

result:

<input type="checkbox"/>		catalog b1abb1fb5	my-app: <none>	Running	5001:5001	0%	5			
<input type="checkbox"/>		order de79bde65	my-app: <none>	Running	5002:5002	0%	5			
<input checked="" type="checkbox"/>		frontend 0a337839c	my-app: <none>	Running	3000:3000	0%	5			

```
root@de79bde6547d:/home# node app.js
Order service is running on http://localhost:5002
Connected to the SQLite database.
Orders table created or already exists.
Inserted order for: John Doe
Inserted order for: Jane Smith
Inserted order for: Mark Johnson
```

```
root@b1abb1fb575d:/home# node app.js
Server is running on http://localhost:5001
Connected to the SQLite database.
Catalog table created or already exists.
Inserted: Distributed Systems
Inserted: Database Management
Inserted: Artificial Intelligence
Inserted: Web Development
Inserted: Machine Learning
Inserted: Operating Systems
Inserted: Networking Basics
Inserted: Data Structures
Inserted: Cybersecurity
Inserted: Cloud Computing
Inserted: Deep Learning
```

```
root@0a337839cfc0:/home# node app.js
Frontend service running on port 3000
```

topic / Search_Items

GET http://localhost:3000/search/Computer Engineering Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK 111 ms 338 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "item_number": 6,
4     "title": "Operating Systems",
5     "stock": 7,
6     "price": 34.99,
7     "topic": "Computer Engineering"
8   }
9 ]
```

topic / All_Item

GET http://localhost:5001/catalog/items

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (7) Test Results 200 OK 12 ms 1.19 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "item_number": 1,
4     "title": "RPC for Noobs",
5     "stock": 10,
6     "price": 29.99,
7     "topic": "distributed systems"
8   },
9   {
10    "item_number": 2,
11    "title": "Cooking for the Impatient Undergraduate Student",
12    "stock": 6,
13    "price": 19.99,
14    "topic": "undergraduate school"
15  },
16 ]
```

GET http://localhost:5002/order/list

Params Auth Headers (7) Body Scripts Settings

Body 200 OK 57 ms 5.38 KB

Pretty Raw Preview Visualize JSON

```
2 {
3   "order_id": 2,
4   "customer_name": "Jane Smith",
5   "order_date": "2024-08-02",
6   "total": 50,
```

Conclusion:

We choose Node.js because it offers robust, lightweight, asynchronous micro services that are scalable, supported by a sizable community, and need specialized knowledge and use.