"An-Najah National University"

"Faculty of Engineering"

"Computer Engineering Department"

DOS Project Part-1

Preparing By

Aisha Ishtayeh "12028269"

- **The services in our project:**

1) Front end service

2) Order service

3) Catalog service

1)Front-end service:

There are three operations in this server

- **1-Search**: the request is sent to catalog server then catalog return the item **http. get(http://catalog:4000/search/ Distributed systems)**
- **2-Info**: the request is sent to catalog server then catalog return the info **http. get(http://catalog:4000/info/1)**
- **3-Purchase**: The purchase order is sent to the order server

  app.post(/purchase/:item_number/1)

-

2)Order service:

Receives requests from the front-end and transfers them to catalog server When it is sent to the catalog, the quantity of stock checked and modified.

http.get(http://catalog:4000/info/1) axios.put('http://catalog:4000/update/1)

3)Catalog service:

It receives requests from the order server, adjusts the quantity, sends the response to the order, and also sends the response to search and info requests.

How to run the program?

1. clone the project using the following command: `git clone`

   `https://github.com/RihamKatout/Bazar-Project.git`

2. You should have "`docker`" installed and run on your device.
3. Using maven commands ❼ maven install
4. Now in terminal write the following commands:
   a. `docker-compose build` ❼ to build docker image.
   b. `docker-compose up` ❼ to start and run the entire application.

2. Start sending requests from the tables above.

This is what you will see when you run the application, there are three servers running, but you can reach the Front server only.

Noticing that sometimes containers don't see the libraries (sqlite3, express, axios) even they are installed when the image was built, so we reinstalled them inside the containers. In next part of the project, we will use docker compose to fix this problem

We use Node.js because it provides powerful microservices that are asynchronous and lightweight also it is scalable they are backed by a large community and require specific utilization and expertise.