BAZAR

Made by:

Ahmad Tubaila 12028317
Abdulkareem Abu-Zahra

Design:

1. Catalog backend file: it’s the core of the backend where it has 4 files in it which are:

   A: catalog.py, the source code for flask where it has three main functions (info, search, update), the info is for searching the full information of a book using its own id, search is done by searching for topic and return all titles of the same matching topic, then the update for a purchase when it’s made.

   B: catalog.csv, it’s a file where it stores the main books information

   C: Dockerfile, it’s the main docker file of catalog, which should run a container of catalog on port 5000 (for listening).

   D: requirements.txt, it has the main feature to download for using my project such as flask, which is downloaded using Dockerfile.

2. Order backend file: its main purpose is to get the purchase body from frontend and give it to the catalog and it has 3 files.

   A: order.py, it does ask for info to see if book exists in catalog and if it does, it makes the second move for the purchase which updates the catalog api for decreasing the quantity of the id for book asked for purchase by 1.

B: Dockerfile, it's the main docker file of order, which should run a container of order on port 5001 (also for listening).

C: requirements.txt, it has the main feature to download for using my project such as flask and requirements, which is downloaded using Dockerfile.

3. Frontend file: it's the api that should act as the UI for the user which communicate with order and catalog and wait for response from them, it has 3 files also:

   A: frontend.py, it has info, search, purchase. The info and search communicate with catalog and purchase speaks with order.

   B: Dockerfile, it's the main docker file of order, which should run a container of order on port 5002 (also for listening).

   C: requirements.txt, it has the main feature to download for using my project such as flask and requirements, which is downloaded using Dockerfile.

4. docker-compose.yml, which handle the 3 Dockerfile in the system to run them all at once.

Design tradeoffs:

1- we could use a real data base such as mysql in order for better manipulation of data and more secure.
2- Using mono design instead of micro service because it's a small project and using micro design will add just some complexity.

our project could behave unexpectedly when any server has an error or down for any reason since they count on each other, such as front count on order and order count on catalog.

How to run our project:

1- Make sure docker is running on your device.

2- Open CMD and head to our project folder (ex: cd bazar)

3- After you are in our folder type docker compose up --build

4- Use and app like postman to send the data on the ports (5000,5001,5002), the frontend is on 5002 so if it works then all the project works.

5- After opening postman you can try these:
http://localhost:5002/info/3 (method: GET)
http://localhost:5002/search/distributed systems (method: GET)

6- For purchase type: http://localhost:5002/purshase then go to body section and press on raw and type:

```
{
      "book_id": 3
}
```

7- You can see the result in the response bellow and that's it 😊.