Bazar.com - Distributed Online Bookstore Project Report

Distributed Operating Systems-Dr Samer Arandi

By Zain Abubaker & Abdulkareem Almasri

Bazar.com is a simple online bookstore system developed using a microservices-based architecture

Technologies & Tools

- ➔ Programming Language: Python 3.11
- ➔ Framework: Flask (Microservices framework)
- ➔ Containerization: Docker
- ➔ Testing Tool: Postman
- ➔ Database: CSV files(to keep the project lightweight)

Microservices Architecture

Service	Responsibilities	Endpoints

Catalog	Manages book catalog and stock data	`GET /search/<topic>`
		`GET /info/<item_id>`
		`POST /update/<item_id>`

Order	Handles purchases and updates stock	`POST /purchase/<item_id>`Logs orders to `order_log.csv`

Frontend	Acts as the client gateway for users	`GET /search/<topic>`
		`GET /info/<item_id>`
		`POST /purchase/<item_id>`

CSV Files Used

- ➔ catalog_data.csv (located in catalog service): Stores book ID, title, topic, quantity, and price
- ➔ order_log.csv (located in order service): Logs purchases with timestamp, item_id, and title

Docker Setup & Networking

Each service runs in its own container:

- ☐ `catalog` on port 5001
- ☐ `order` on port 5002
- ☐ `frontend` on port 5000

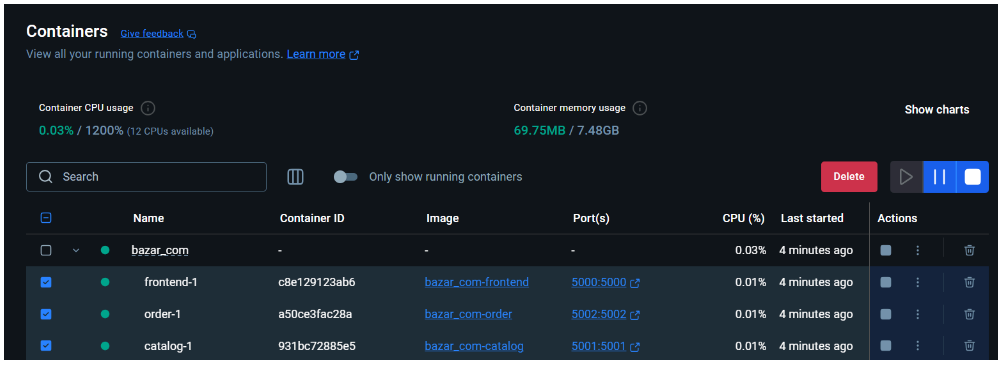
Dockerfiles: Each service has a `Dockerfile` that installs Flask and dependencies.

docker-compose.yml: Coordinates the build and run of all containers.

Installation & Commands Used

Environment Setup	**Docker Commands**	**WSL Setup (Windows Only)**
python -m venv venv venv\Scripts\activate pip install flask flask-cors requests pip freeze > requirements.txt	docker-compose build docker-compose up # or in background docker-compose up -d docker ps docker exec -it bazar_com-catalog-1 cat catalog_data.csv	wsl --shutdown # BIOS enabled virtualization for WSL2

Green lights for all 3 containers (instances) in Docker Desktop



Containers Give feedback

View all your running containers and applications. Learn more

Container CPU usage
0.03% / 1200% (12 CPUs available)

Container memory usage
69.75MB / 7.48GB

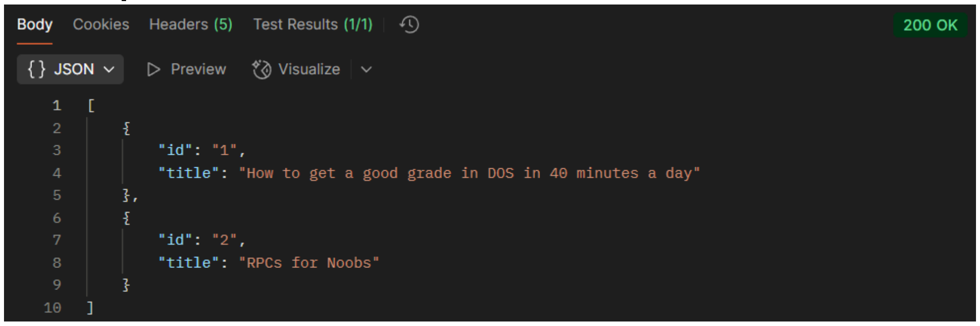
Show charts

Search   Only show running containers   Delete

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
bazar_com	-	-	-	0.03%	4 minutes ago	
frontend-1	c8e129123ab6	bazar_com-frontend	5000:5000	0.01%	4 minutes ago	
order-1	a50ce3fac28a	bazar_com-order	5002:5002	0.01%	4 minutes ago	
catalog-1	931bc72885e5	bazar_com-catalog	5001:5001	0.01%	4 minutes ago	

Postman test results for search, info, purchase, and out-of-stock case

GET http://localhost:5000/search/distributed

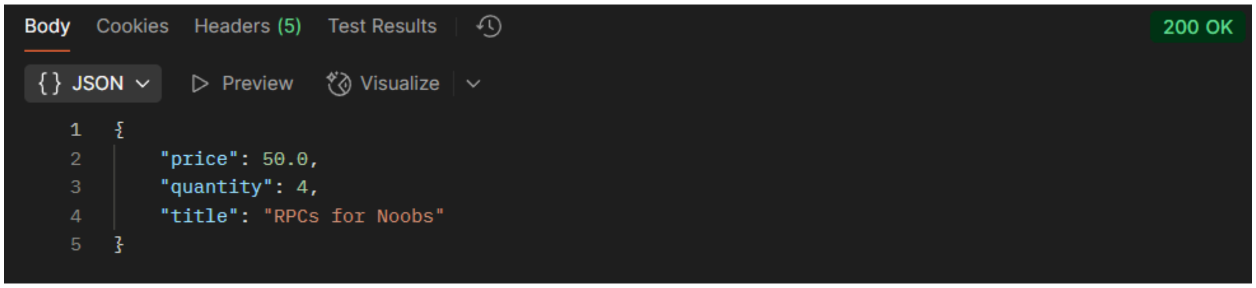


Body Cookies Headers (5) Test Results (1/1) 200 OK

{} JSON Preview Visualize

```
[
    {
        "id": "1",
        "title": "How to get a good grade in DOS in 40 minutes a day"
    },
    {
        "id": "2",
        "title": "RPCs for Noobs"
    }
]
```

GET http://localhost:5000/info/2

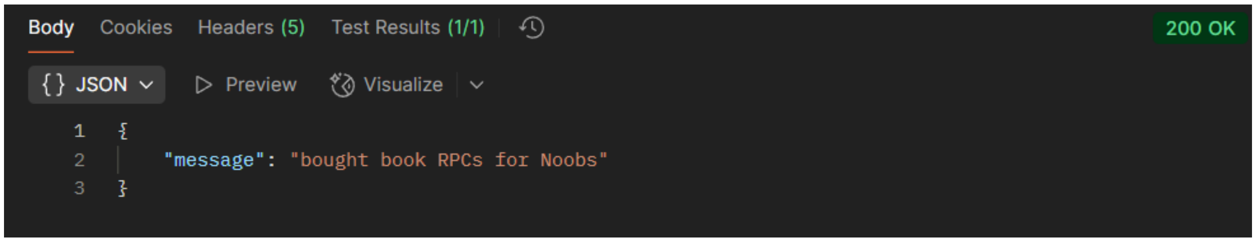


Body Cookies Headers (5) Test Results 200 OK

{} JSON Preview Visualize

```
{
    "price": 50.0,
    "quantity": 4,
    "title": "RPCs for Noobs"
}
```

POST http://localhost:5000/purchase/2

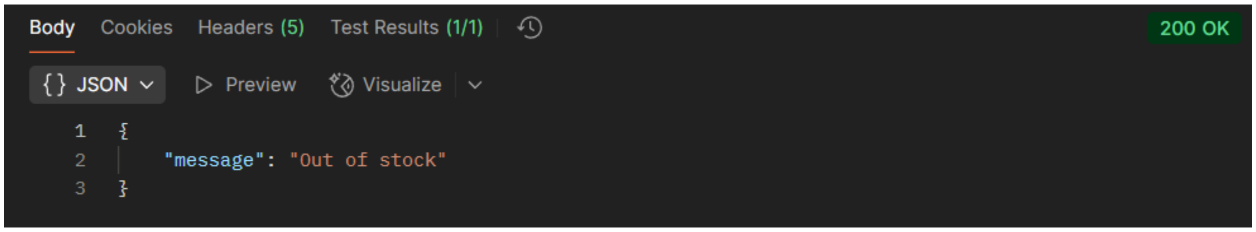


Body Cookies Headers (5) Test Results (1/1) 200 OK

{} JSON Preview Visualize

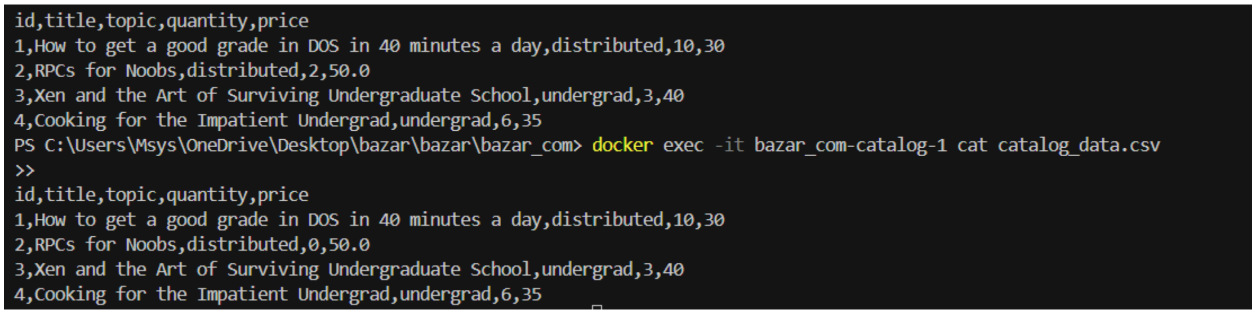
```
{
    "message": "bought book RPCs for Noobs"
}
```

After multiple purchases for the same book, until the quantity was 0



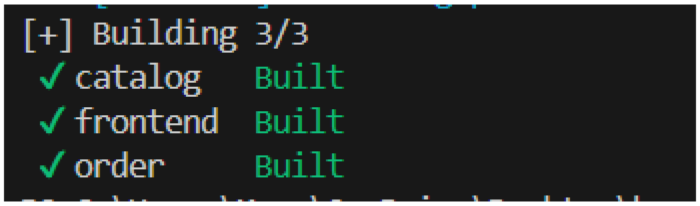
```
{
    "message": "Out of stock"
}
```

After purchases, the quantity field in `catalog_data.csv` is reduced



```
id,title,topic,quantity,price
1,How to get a good grade in DOS in 40 minutes a day,distributed,10,30
2,RPCs for Noobs,distributed,2,50.0
3,Xen and the Art of Surviving Undergraduate School,undergrad,3,40
4,Cooking for the Impatient Undergrad,undergrad,6,35
PS C:\Users\Msys\OneDrive\Desktop\bazar\bazar\bazar_com> docker exec -it bazar_com-catalog-1 cat catalog_data.csv
>>
id,title,topic,quantity,price
1,How to get a good grade in DOS in 40 minutes a day,distributed,10,30
2,RPCs for Noobs,distributed,0,50.0
3,Xen and the Art of Surviving Undergraduate School,undergrad,3,40
4,Cooking for the Impatient Undergrad,undergrad,6,35
```

VS Code command line after composing up the three microservices instances on the docker



```
[+] Building 3/3
 ✓ catalog   Built
 ✓ frontend  Built
 ✓ order     Built
```