

# DOS Project P1

## Team Members:

Layan Othman      Nagham Hanini

## Introduction

This project simulates an e-commerce platform using a **microservices** architecture, demonstrating how distributed systems collaborate through network communication.

## Technologies We Used

To bring our project to life, we relied on a modern tech stack:

- **Node.js** – for building all our backend services
- **SQLite** – for database.
- **Axios** – to enable HTTP requests between microservices
- **Docker & Docker Compose** – for containerizing and orchestrating our services
- **Visual Studio Code** – for efficient and organized development

## Our System Architecture

Our system is composed of **three** microservices

### Catalog Service

Handles the book catalog, storing and managing titles, topics, prices, and quantities  
the endpoints like:

**GET** /search/:topic – lists all books under a topic

**GET** /info/:itemNumber – returns book details by ID

**POST** /order – validates and processes

### Order Service

Endpoint: **POST** /purch

Client Service (Command-Line Interface & simple web design)

Allows users to:

Search for books by topic

Get details of a book by item number

Purchase a book by providing its ID and payment amount

## How we run our System

-Execute the command:

`docker-compose up --build`

```
C:\Users\hp\Desktop\DOSPt1>docker-compose up
time="2025-04-09T00:26:54+02:00" level=warning msg="C:\\Users\\hp\\Desktop\\DOSPt1\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 9/9
  ✔ Network dospt1_default          Created                                0.1s
  ✔ Container dospt1-catalog-server-2 Created                                0.3s
  ✔ Container dospt1-client-1       Created                                0.2s
  ✔ Container dospt1-redis-1        Created                                0.3s
  ✔ Container dospt1-order-server-2 Created                                0.3s
  ✔ Container dospt1-client-2       Created                                0.3s
  ✔ Container dospt1-order-server-1 Created                                0.3s
  ✔ Container dospt1-catalog-server-1 Created                              0.4s
  ✔ Container dospt1-nginx-1        Created                                0.2s
Attaching to catalog-server-1, catalog-server-2, client-1, client-2, nginx-1, order-server-1, order-server-2, redis-1
redis-1      | 1:C 08 Apr 2025 22:26:56.308 * o000o000o000o Redis is starting o000o000o000o
redis-1      | 1:C 08 Apr 2025 22:26:56.308 * Redis version=7.4.2, bits=64, commit=00000000, modified=0, pid=1, just started
redis-1      | 1:C 08 Apr 2025 22:26:56.308 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
redis-1      | 1:M 08 Apr 2025 22:26:56.308 * monotonic clock: POSIX clock_gettime
redis-1      | 1:M 08 Apr 2025 22:26:56.308 * Running mode=standalone, port=6379.
redis-1      | 1:M 08 Apr 2025 22:26:56.309 * Server initialized
redis-1      | 1:M 08 Apr 2025 22:26:56.309 * Ready to accept connections tcp
catalog-server-1 |
catalog-server-1 | > dos_project@1.0.0 start-catalog
order-server-1  |
client-2        |
```

-Once all services are up, access the CLI with:

`docker exec -it <client-container-name> node index.mjs`

```
PROBLEMS 1 TERMINAL PORTS

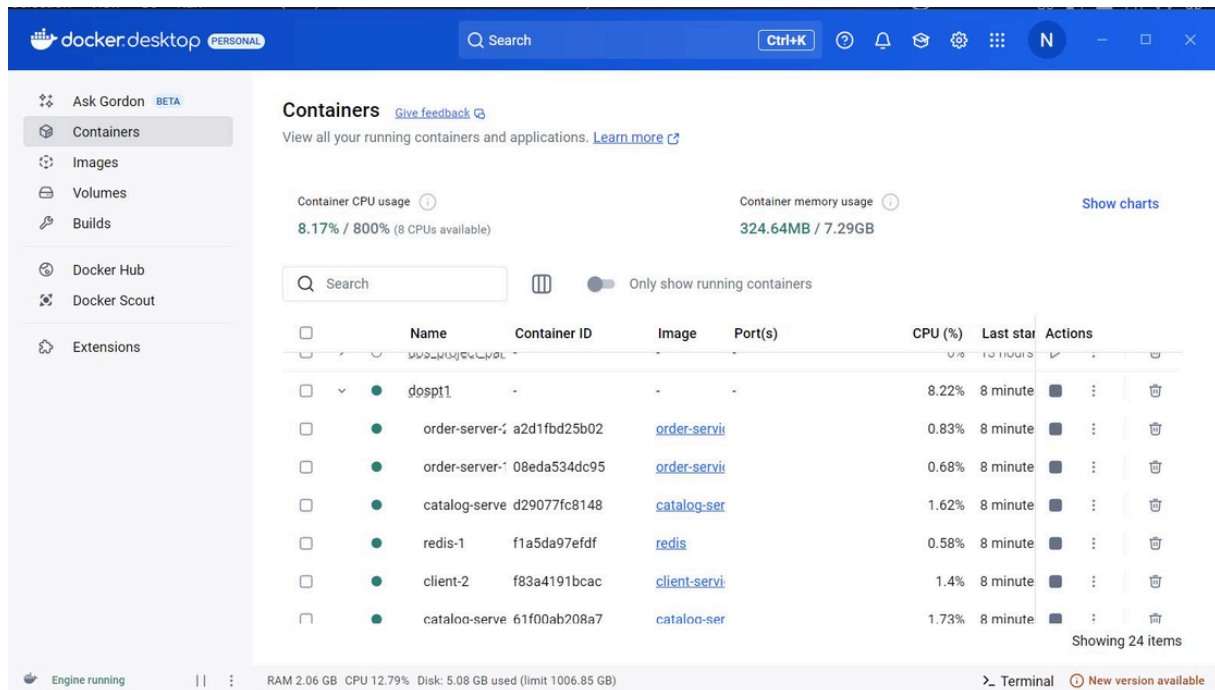
PS C:\Users\hp\Desktop\DOSPt1\src\client-service> node index.mjs
Usage: CLI [options] [command]

CLI for DOS Project

Options:
  -V, --version          output the version number
  -h, --help             display help for command

Commands:
  search-book-title|search  search about specific book using book topic
  info-book-item-number|info  info about specific book using item number
  purchase-book-by-item-number|order  purchase specific book using item number
  help [command]           display help for command
○ PS C:\Users\hp\Desktop\DOSPt1\src\client-service> 
```

## Docker Desktop showing running containers:



## Project structure in VS Code:

## Running a search by topic ( for example, distributed systems)

JS index.mjs JS index.js database.db X

src > catalog-service > database.db

Filter 4 rows...

TABLES

items

	id	topic	quantity	price	title
1	1	Distributed System	20	3000	How to get a good grade in DOS in 40 minutes a day
2	2	Distributed System	9	1500	RPCs for Noobs.
3	3	Undergraduate School	5	1000	Xen and the Art of Surviving Undergraduate School
4	4	Undergraduate School	13	900	Cooking for the Impatient Undergrad
5					

Page 1 / 1

Try Hydejack, a Jekyll theme for hackers, nerds, and academics

powerShell - client-service

```
}
PS C:\Users\hp\Desktop\DOSPt1\src\client-service> node index.mjs search
? please enter book topic to get details about it: Distributed System
Response Data {
  id: 1,
  topic: 'Distributed System',
  quantity: 20,
  price: 3000,
  title: 'How to get a good grade in DOS in 40 minutes a day'
}, {
  id: 2,
  topic: 'Distributed System',
  quantity: 9,
```

vulnerabilities found for all the providers combined

Go Live

## Viewing info of a book :

src > catalog-service > database.db

Filter 4 rows...

TABLES

items

	id	topic	quantity	price	title
1	1	Distributed System	20	3000	How to get a good grade in DOS in 40 minutes a day
2	2	Distributed System	9	1500	RPCs for Noobs.
3	3	Undergraduate School	5	1000	Xen and the Art of Surviving Undergraduate School
4	4	Undergraduate School	13	900	Cooking for the Impatient Undergrad
5					

Page 1 / 1

Try Hydejack, a Jekyll theme for hackers, nerds, and academics

powerShell

```
PS C:\Users\hp\Desktop\DOSPt1\src\client-service>
? please enter items number to get info about it: 2
Response Data: { item: [ { id: 2, quantity: 9, price: 1500 } ] }
PS C:\Users\hp\Desktop\DOSPt1\src\client-service>
PS C:\Users\hp\Desktop\DOSPt1\src\client-service>
```

src > catalog-service > database.db

Rows: 4

	id	topic	quantity	price	title
1	1	Distributed System	20	3000	How to get a good grade in DOS in 40 minutes a day
2	2	Distributed System	9	1500	RPCs for Noobs.
3	3	Undergraduate School	4	1000	Xen and the Art of Surviving Undergraduate School
4	4	Undergraduate School	13	900	Cooking for the Impatient Undergrad
5					

Page 1 / 1

PROBLEMS 1 TERMINAL PORTS

```
PS C:\Users\hp\Desktop\DOSPt1\src\client-service> node index.mjs info
? please enter items number to get info about it: 5
Response Data: { item: [] }
```

## Order before and after:

src > catalog-service > database.db

Rows: 4

	id	topic	quantity	price	title
1	1	Distributed System	20	3000	How to get a good grade in DOS in 40 minutes a day
2	2	Distributed System	9	1500	RPCs for Noobs.
3	3	Undergraduate School	4	1000	Xen and the Art of Surviving Undergraduate School
4	4	Undergraduate School	13	900	Cooking for the Impatient Undergrad
5					

Page 1 / 1

PROBLEMS 1 TERMINAL PORTS

```
PS C:\Users\hp\Desktop\DOSPt1\src\client-service> node index.mjs order
? please enter book item number to purchase it: 3
? Enter amount of money to pay: 1000
Response Data: { message: 'Send Request To Catalog' }
```

Body

200 OK 67 ms 304 B

Save as example

Pretty Raw Preview Visualize JSON

```
1
2  "message": "Send Request To Catalog"
3
```



src > catalog-service > database.db

Rows: 4

Filter 4 rows...

TABLES

items

	id	topic	quantity	price	title
	1	Distributed System	20	3000	How to get a good grade in DOS in 40 minutes a day
	2	Distributed System	9	1500	RPCs for Noobs.
	3	Undergraduate School	5	1000	Xen and the Art of Surviving Undergraduate School
	4	Undergraduate School	13	900	Cooking for the Impatient Undergrad
	5				

S... v0.10... 1 Page 1 / 1 Try Hydejack, a Jekyll theme for hackers, nerds, an

PROBLEMS 1

TERMINAL

PORTS

powershell + v

PS C:\Users\hp\Desktop\DOSPt1\src\client-service>  
PS C:\Users\hp\Desktop\DOSPt1\src\client-service> node index.mjs order  
? please enter book item number to purchase it: 3  
? Enter amount of money to pay: 1000  
Response Data: { message: 'Send Request To Catalog' }  
PS C:\Users\hp\Desktop\DOSPt1\src\client-service>

