Bazar.com Online Book Store - Project Report

1. Introduction

Bazar.com is a multi-tier online book store application built with microservices architecture. The system consists of:

- Frontend Service: Flask microservice (Python) handling client requests
- Backend Services:
 - 1. Catalog service (manages book inventory)
 - 2. Order service (processes purchases)

Containerization: Docker with docker-compose for deployment

2. System Architecture

- The system follows a client-server model where:
 - 1. Clients interact with the Frontend service (Port 5000)
 - 2. Frontend routes requests to:
 - Catalog service (Port 5001) for search/info
 - Order service (Port 5002) for purchases

3. How the Application Works

3.1 Component Interactions

Frontend Service:

- 1. Accepts HTTP requests from clients
- 2. Routes to backend services via REST API calls

Catalog Service:

- 1. Maintains inventory in catalog_data.json
- 2. Handles:

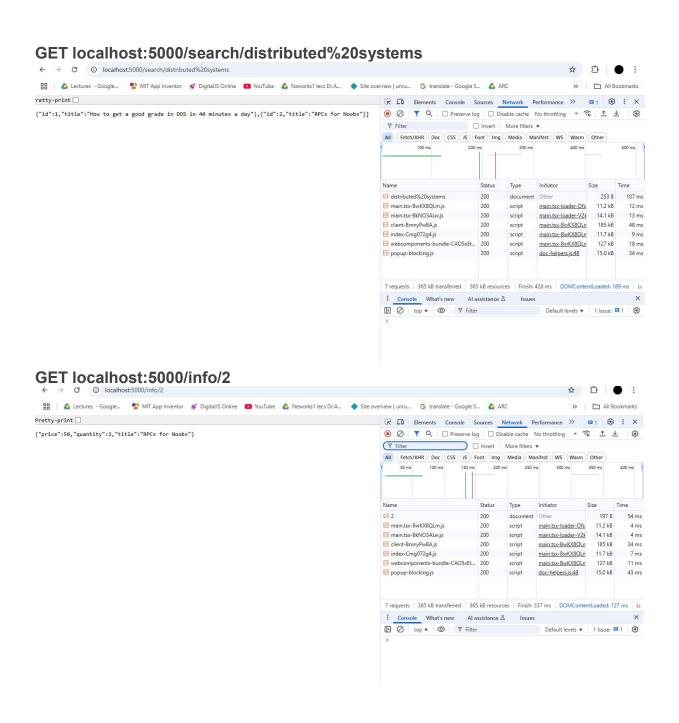
GET /search/<topic>: Find books by topic
GET /info/<item number>: Get book details

Order Service:

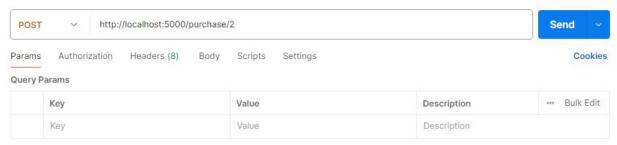
- 1. Processes POST /purchase/<item_number>
- 2. Logs transactions in order_log.txt

3.2 API Endpoints

Service	Endpoint	Method	Description
Frontend	/search/ <topic></topic>	GET	Search books by topic
Frontend	/info/ <item_number></item_number>	GET	Get book details
Frontend	/purchase/ <item_number></item_number>	POST	Purchase a book



POST localhost:5000/purchase/2



4. Setup Instructions

4.1 Prerequisites

- 1. Docker Desktop installed
- 2. Python 3.9+ (for local testing)
- 3. Git

4.2 Installation Steps

Clone repository:

git clone [my-repo-url] cd bazar com

Build and run containers:

docker-compose up -d --build

4.3 Verification

Check running containers:

```
docker ps

PS C:\Users\hp\.vscode\extensions\ms-python.python.python-2025.2.0-win32-x64\python_files\deactivate\powershell> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
597301a7a56f bazar_com-frontend "python app.py" 43 minutes ago Up 43 minutes 0.0.0.0:5000->5000/tcp bazar_com-frontend
d-1
67095b440e38 bazar_com-catalog "python app.py" 43 minutes ago Up 43 minutes 0.0.0.0:5001->5001/tcp bazar_com-catalog
-1
dadc6ca7c74e bazar_com-order "python app.py" 43 minutes ago Up 43 minutes 0.0.0.0:5002->5002/tcp bazar_com-order-1
```

Test endpoints:

curl http://localhost:5000/search/distributed%20systems

5. Usage Examples

5.1 Searching Books

GET http://localhost:5000/search/undergraduate%20school

Returns:

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

5.2 Making a Purchase

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

```
{
    "message": "Purchased 'RPCs for Noobs' for $50",
    "success": true
}
```

5.3 INFO

Returns:

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

Returns:

```
{"price":50,"quantity":3,"title":"RPCs for Noobs"}
```

6. Technical Highlights

- 1. Microservices architecture
- 2. REST API Design
- 3. Containerized Deployment (Docker)
- 4. Persistent data storage (JSON files)
- 5. Concurrent request handling

7. Future Enhancements

- 1. Add user authentication
- 2. Implement shopping cart functionality
- 3. Replace file storage with SQLite database
- 4. Add admin dashboard