

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>	GET	Search books by topic
Frontend	/info/<item_number>	GET	Get book details
Frontend	/purchase/<item_number>	POST	Purchase a book

GET localhost:5000/search/distributed%20systems

localhost:5000/search/distributed%20systems

pretty-print ☐

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

Name	Status	Type	Initiator	Size	Time
distributed%20systems	200	document	Other	253 B	107 ms
main.tsx-BwKX8QLm.js	200	script	main.tsx-loader-Dfs	11.2 kB	12 ms
main.tsx-BkNO3ALw.js	200	script	main.tsx-loader-V2	14.1 kB	13 ms
client-BmnyPwBA.js	200	script	main.tsx-BwKX8QLm	185 kB	48 ms
index-Cmg072g4.js	200	script	main.tsx-BwKX8QLm	11.7 kB	9 ms
webcomponents-bundle-CAO5Et...	200	script	main.tsx-BwKX8QLm	127 kB	18 ms
popup-blocking.js	200	script	doc-helpers.js:48	15.0 kB	34 ms

7 requests | 365 kB transferred | 365 kB resources | Finish: 428 ms | DOMContentLoaded: 189 ms | Lc

GET localhost:5000/info/2

localhost:5000/info/2

pretty-print ☐

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

Name	Status	Type	Initiator	Size	Time
2	200	document	Other	197 B	54 ms
main.tsx-BwKX8QLm.js	200	script	main.tsx-loader-Dfs	11.2 kB	4 ms
main.tsx-BkNO3ALw.js	200	script	main.tsx-loader-V2	14.1 kB	4 ms
client-BmnyPwBA.js	200	script	main.tsx-BwKX8QLm	185 kB	34 ms
index-Cmg072g4.js	200	script	main.tsx-BwKX8QLm	11.7 kB	7 ms
webcomponents-bundle-CAO5Et...	200	script	main.tsx-BwKX8QLm	127 kB	11 ms
popup-blocking.js	200	script	doc-helpers.js:48	15.0 kB	43 ms

7 requests | 365 kB transferred | 365 kB resources | Finish: 337 ms | DOMContentLoaded: 127 ms | Lc

POST localhost:5000/purchase/2

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

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results ↺ 200 OK • 36 ms • 210 B 🌐 📄 Save Response ⋮

{ } JSON ▼ ▶ Preview 🔗 Visualize ▼ 🔍 📄 🔗

```
1 {
2   |
3   |   "message": "Purchased 'RPCs for Noobs' for $50",
4   |   "success": true
5 }
```

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
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
=> => unpacking to docker.io/library/bazar_com-frontend:latest 0.0s
=> [frontend] resolving provenance for metadata file 0.0s
[+] Running 6/6
✓ catalog Built 0.0s
✓ frontend Built 0.0s
✓ order Built 0.0s
✓ Container bazar_com-frontend-1 Started 23.7s
✓ Container bazar_com-catalog-1 Started 12.3s
✓ Container bazar_com-order-1 Started 12.3s
* Terminal will be reused by tasks, press any key to close it.
```

4.3 Verification

Check running containers:

```
docker ps
```

```
PS C:\Users\hp\.vscode\extensions\ms-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-fronten
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
dad66ca7c74e   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
```

Returns:

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

5.3 INFO

```
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