

An-Najah National University

Computer Engineering

Distributed Operation System

BAZAR.COM part2

Results report and screen shots

Students

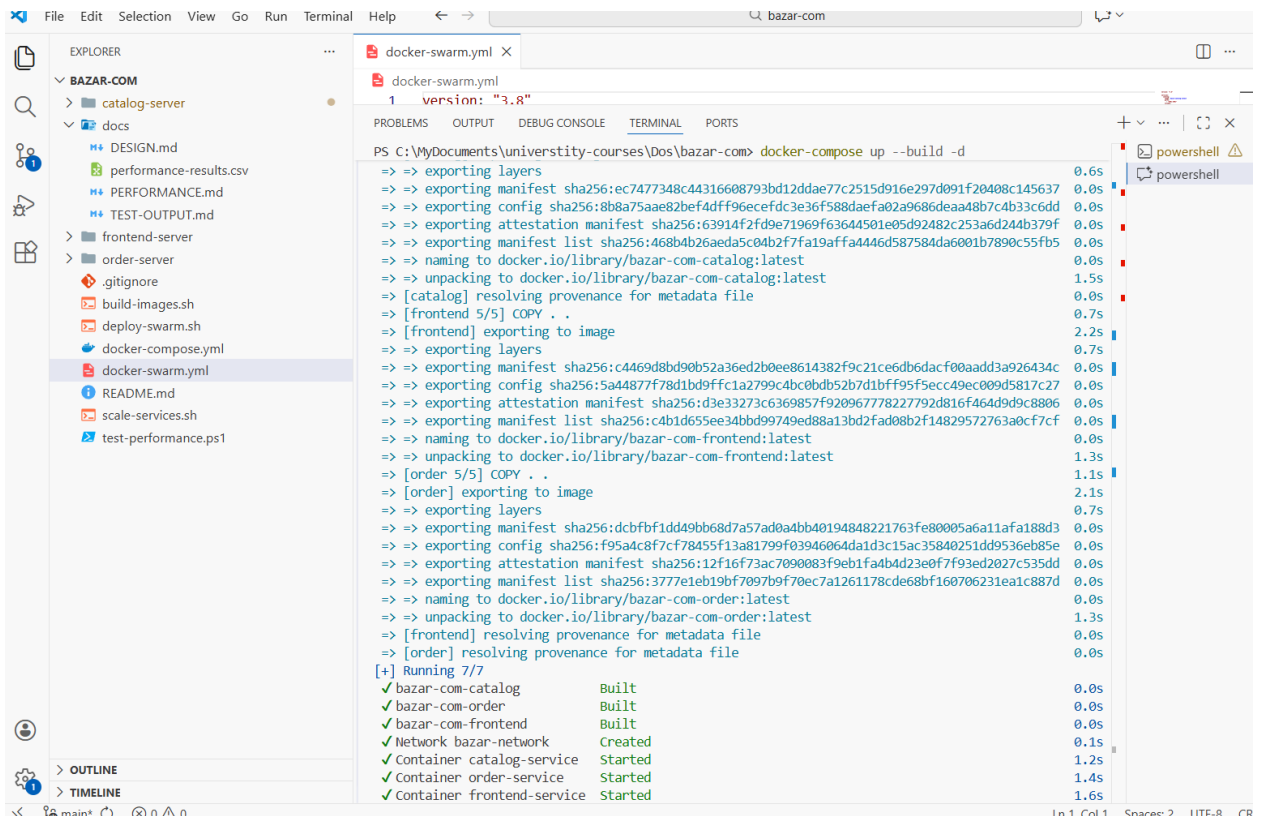
Malik Ali 12114096

Nassar Harashi 12144079

Dr. Samer Arandi

1. Start services:

Services started using `docker-compose up --build -d`. Shown: container names, state, ports (frontend:3002, catalog:3001, order:3000).



The screenshot shows the Visual Studio Code interface with the 'docker-swarm.yml' file open in the editor. The file content is as follows:

```
1 version: "3.8"
```

The terminal output shows the command `docker-compose up --build -d` being executed. The output displays the progress of building and starting the services, including the export of layers, manifests, and configurations, and the resolution of provenance for metadata files. The final output shows the status of the services:

```
[+] Running 7/7
✓ bazar-com-catalog      Built      0.0s
✓ bazar-com-order        Built      0.0s
✓ bazar-com-frontend     Built      0.0s
✓ Network bazar-network  Created    0.1s
✓ Container catalog-service Started    1.2s
✓ Container order-service Started    1.4s
✓ Container frontend-service Started    1.6s
```

2. Frontend Health:

Health check: reports frontend status and dependency checks for catalog and order.

Note [status](#) (healthy/degraded) and service URLs."

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> Invoke-RestMethod -Uri "http://localhost:3002/health"
```

```
status      : healthy
timestamp   : 2025-12-21T18:27:19.487Z
uptime      : 337.218445984
environment : production
serviceId    : frontend-1766341300935
services     : @{{catalog=; order=}}
cache        : @{{keys=0; stats=}}
```

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```

3. Cache stats before tests:

Cache initial state (keys, hits/misses). Use as baseline for cache performance comparison.

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> Invoke-RestMethod -Uri "http://localhost:3002/cache-stats"
```

keys	stats	count	serviceId
{}	@{hits=0; misses=0; keys=0; ksize=0; vsize=0}	0	frontend-1766341300935

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```

4. Search endpoint — show MISS then HIT:

Search test: compare MISS vs HIT. Record counts and response source fields (e.g., [source: catalog-service](#) vs [source: cache](#)). Note time differences if you measured."

Invalidate cache:

Command :

```
Invoke-RestMethod `
  -Uri "http://localhost:3002/invalidate-cache" `
  -Method POST `
  -Headers @{ "Content-Type" = "application/json" } `
  -Body (@{ bookId = "1" } | ConvertTo-Json)
```

```

PS C:\MyDocuments\universtity-courses\Dos\bazar-com> Invoke-RestMethod `
>> -Uri "http://localhost:3002/invalidate-cache" `
>> -Method POST `
>> -Headers @{ "Content-Type" = "application/json" } `
● >> -Body (@{ bookId = "1" } | ConvertTo-Json)
>>

status  message                serviceId
-----  -
success Cache invalidated frontend-1766341300935

PS C:\MyDocuments\universtity-courses\Dos\bazar-com>

```

Request 1(cache miss):

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#).

RestClient New Import Overview GET info PUT update POST purchase GET orders GET info POST purchase GET search No environment

Collections + Search collections

- api-rest
- BAZAR-COM
 - catalog-service
 - order-server
 - frontend-server
 - GET search
 - GET info
 - POST purchase
- Courses-Project
- Doslab1
- learnNodejsAPIs
- simpleProject

Environments History Flows

BAZAR-COM / frontend-server / search Save Share

GET http://localhost:3002/search/undergraduate%20school Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results 200 OK 32 ms 458 B Save Response

JSON Preview Visualize

```

1 {
2   "success": true,
3   "source": "catalog-service",
4   "count": 2,
5   "data": [
6     {
7       "id": "3",
8       "title": "Xen and the Art of Surviving Undergraduate School"
9     },
10    {
11      "id": "4",
12      "title": "Cooking for the Impatient Undergrad"
13    }
14  ]
15 }

```

Cloud View Find and replace Console Runner Start Proxy Cookies Vault Trash 8:52 PM 12/21/2025

Request 2(cache hit):

The screenshot shows the Postman interface with a GET request to `http://localhost:3002/search/undergraduate%20school`. The response is a 200 OK status with a 5 ms response time. The JSON body is as follows:

```
{
  "success": true,
  "source": "cache",
  "count": 2,
  "data": [
    {
      "id": "3",
      "title": "Xen and the Art of Surviving Undergraduate School"
    },
    {
      "id": "4",
      "title": "Cooking for the Impatient Undergrad"
    }
  ]
}
```

5. Book info /info/1 — MISS then HIT

Invalidate cache :

```
>> -Uri "http://localhost:3002/invalidate-cache" `get a good book`
>> -Method POST `
>> -Headers @{ "Content-Type" = "application/json" } `
● >> -Body (@{ bookId = "1" } | ConvertTo-Json)-com>

status  message                                serviceId
-----  -
success Cache invalidated frontend-1766341300935

PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```

Request 1(miss):

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#).

BAZAR-COM / frontend-server / info

GET http://localhost:3002/info/1

Send

Params

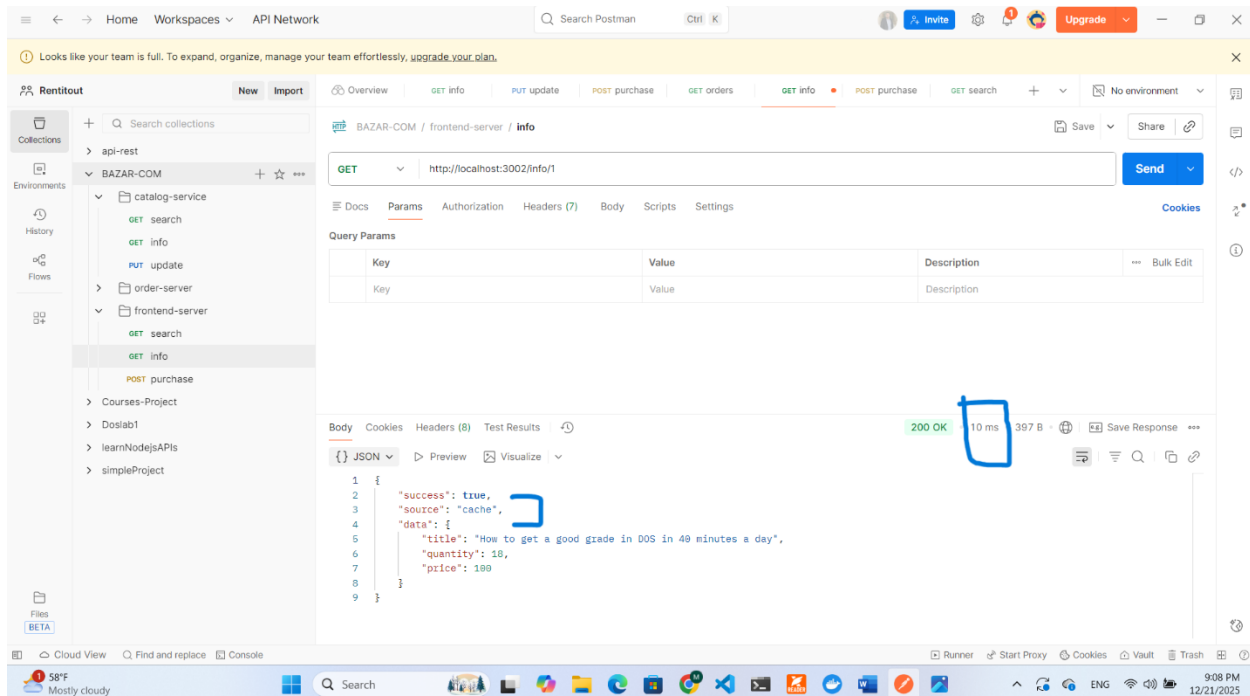
Key	Value	Description
Key	Value	Description

Body

200 OK 20 ms 407 B

```
1 {
2   "success": true,
3   "source": "catalog-service",
4   "data": {
5     "title": "How to get a good grade in DOS in 40 minutes a day",
6     "quantity": 10,
7     "price": 100
8   }
9 }
```

Request 2(HIT)



6. Purchase flow (end-to-end) and confirm [orders.csv](#):

End-to-end purchase: frontend -> order -> catalog stock decrement. Include purchase response ([orderId](#), [price](#)) and verify [orders.csv](#) contains the generated order. Note any changes in catalog stock.

Purchase:

```

PS C:\MyDocuments\university-courses\Dos\bazar-com> Invoke-RestMethod -Method POST -Uri "http://localhost:3002/purchase/1" |
>> ConvertTo-Json -Depth 10
{
  "success": true,
  "source": "order-service",
  "data": {
    "message": "Purchase successful",
    "book": "How to get a good grade in DOS in 40 minutes a day",
    "orderId": "1766344524241",
    "price": 100
  }
}

```

Show from host:

```
Windows PowerShell
}
  "price": 100
}
}
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> type .\order-server\orders.csv
ORDER_ID,BOOK_ID,TITLE,QUANTITY,TOTAL_PRICE,TIMESTAMP
1762992825675,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:13:45
1762992826985,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:13:46
1762993208881,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:08
1762993239276,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:39
1762993240530,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:40
1762993718017,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:28:38
1762994246614,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:37:26
1762994472206,2,RPCs for Noobs,1,30,2025-11-13 00:41:12
1763823620531,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-22 15:00:20
1763824201412,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-22 15:10:01
1764341576145,2,RPCs for Noobs,1,100,2025-11-28 14:52:56
1764341714504,2,RPCs for Noobs,1,100,2025-11-28 14:55:14
1764341718714,2,RPCs for Noobs,1,100,2025-11-28 14:55:18
1766237349735,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-20 13:29:09
1766237362708,2,RPCs for Noobs,1,100,2025-12-20 13:29:22
1766237387359,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:29:47
1766237914633,2,RPCs for Noobs,1,100,2025-12-20 13:38:34
1766237915171,3,Xen and the Art of Surviving Undergraduate School,1,25,2025-12-20 13:38:35
1766237915698,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:38:35
1766237917886,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-20 13:38:37
1766237918417,2,RPCs for Noobs,1,100,2025-12-20 13:38:38
1766237918970,3,Xen and the Art of Surviving Undergraduate School,1,25,2025-12-20 13:38:38
1766237919547,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:38:39
1766344524241,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-21 19:15:24
PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```

Show from docker container:

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> docker-compose exec order cat /app/orders.csv
ORDER_ID,BOOK_ID,TITLE,QUANTITY,TOTAL_PRICE,TIMESTAMP
1762992825675,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:13:45
1762992826985,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:13:46
1762993208881,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:08
1762993239276,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:39
1762993240530,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:20:40
1762993718017,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:28:38
1762994246614,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-13 00:37:26
1762994472206,2,RPCs for Noobs,1,30,2025-11-13 00:41:12
1763823620531,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-22 15:00:20
1763824201412,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-11-22 15:10:01
1764341576145,2,RPCs for Noobs,1,100,2025-11-28 14:52:56
1764341714504,2,RPCs for Noobs,1,100,2025-11-28 14:55:14
1764341718714,2,RPCs for Noobs,1,100,2025-11-28 14:55:18
1766237349735,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-20 13:29:09
1766237362708,2,RPCs for Noobs,1,100,2025-12-20 13:29:22
1766237387359,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:29:47
1766237914633,2,RPCs for Noobs,1,100,2025-12-20 13:38:34
1766237915171,3,Xen and the Art of Surviving Undergraduate School,1,25,2025-12-20 13:38:35
1766237915698,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:38:35
1766237917886,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-20 13:38:37
1766237918417,2,RPCs for Noobs,1,100,2025-12-20 13:38:38
1766237918970,3,Xen and the Art of Surviving Undergraduate School,1,25,2025-12-20 13:38:38
1766237919547,4,Cooking for the Impatient Undergrad,1,20,2025-12-20 13:38:39
1766344524241,1,How to get a good grade in DOS in 40 minutes a day,1,100,2025-12-21 19:15:24
PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```


The stock in catalog.csv:

docker-swarm.yml catalog.csv X

```
catalog-server > cat catalog.csv
1 ID,TOPIC,TITLE,PRICE,STOCK
2 1,distributed systems,How to get a good grade in DOS in 40 minutes a day,100,18
3 2,distributed systems,RPCs for Noobs,100,19
4 3,undergraduate school,Xen and the Art of Surviving Undergraduate School,25,10
5 4,undergraduate school,Cooking for the Impatient Undergrad,20,9
6 5,project management,How to finish Project 3 on time,150,15
7 6,theory,Why theory classes are so hard,120,20
8 7,spring,Spring in the Pioneer Valley,80,25
9
```

before=19

7. Run performance script and collect CSV:

Command (PowerShell): `powershell -ExecutionPolicy Bypass -File test-performance.ps1`

Performance benchmark results: include averages, medians, min/max for Search (with/without cache), Info (with/without cache), Purchase, and Cache Invalidation. Attach the [performance-results.csv](#) and add short commentary about observed cache improvement percentages.

```
Windows PowerShell
PS C:\MyDocuments\university-courses\Dos\bazar-com> powershell -ExecutionPolicy Bypass -File test-performance.ps1
=== Bazar.com Performance Testing ===

[Test 1] Measuring Search Performance WITHOUT Cache...
Clearing cache first...

status  message                                serviceId
-----
success Cache invalidated frontend-1766341300935
success Cache invalidated frontend-1766341300935
Request 1 : 17ms (MISS)
success Cache invalidated frontend-1766341300935
Request 2 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 3 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 4 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 5 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 6 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 7 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 8 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 9 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 10 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 11 : 5ms (MISS)
success Cache invalidated frontend-1766341300935
Request 12 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 13 : 3ms (MISS)
success Cache invalidated frontend-1766341300935
Request 14 : 6ms (MISS)
success Cache invalidated frontend-1766341300935
Request 15 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 16 : 2ms (MISS)
```

```
Windows PowerShell
success Cache invalidated frontend-1766341300935
Request 17 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 18 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 19 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 20 : 2ms (MISS)

[Test 2] Measuring Search Performance WITH Cache...
Request 1 : 2ms (HIT)
Request 2 : 3ms (HIT)
Request 3 : 2ms (HIT)
Request 4 : 2ms (HIT)
Request 5 : 3ms (HIT)
Request 6 : 2ms (HIT)
Request 7 : 2ms (HIT)
Request 8 : 11ms (HIT)
Request 9 : 2ms (HIT)
Request 10 : 2ms (HIT)
Request 11 : 2ms (HIT)
Request 12 : 2ms (HIT)
Request 13 : 2ms (HIT)
Request 14 : 2ms (HIT)
Request 15 : 3ms (HIT)
Request 16 : 2ms (HIT)
Request 17 : 2ms (HIT)
Request 18 : 2ms (HIT)
Request 19 : 4ms (HIT)
Request 20 : 2ms (HIT)

[Test 3] Measuring Info Performance WITHOUT Cache...
success Cache invalidated frontend-1766341300935
Request 1 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 2 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 3 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 4 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 5 : 7ms (MISS)
```

```
Windows PowerShell
success Cache invalidated frontend-1766341300935
Request 5 : 8ms (MISS)
success Cache invalidated frontend-1766341300935
Request 6 : 6ms (MISS)
success Cache invalidated frontend-1766341300935
Request 7 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 8 : 11ms (MISS)
success Cache invalidated frontend-1766341300935
Request 9 : 8ms (MISS)
success Cache invalidated frontend-1766341300935
Request 10 : 6ms (MISS)
success Cache invalidated frontend-1766341300935
Request 11 : 5ms (MISS)
success Cache invalidated frontend-1766341300935
Request 12 : 6ms (MISS)
success Cache invalidated frontend-1766341300935
Request 13 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 14 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 15 : 9ms (MISS)
success Cache invalidated frontend-1766341300935
Request 16 : 6ms (MISS)
success Cache invalidated frontend-1766341300935
Request 17 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 18 : 7ms (MISS)
success Cache invalidated frontend-1766341300935
Request 19 : 8ms (MISS)
success Cache invalidated frontend-1766341300935
Request 20 : 7ms (MISS)

[Test 4] Measuring Info Performance WITH Cache...
Request 1 : 3ms (HIT)
Request 2 : 3ms (HIT)
Request 3 : 2ms (HIT)
Request 4 : 2ms (HIT)
Request 5 : 3ms (HIT)
Request 6 : 2ms (HIT)
Request 7 : 2ms (HIT)
```

```
Windows PowerShell

Request 8 : 3ms (HIT)
Request 9 : 2ms (HIT)
Request 10 : 2ms (HIT)
Request 11 : 2ms (HIT)
Request 12 : 2ms (HIT)
Request 13 : 2ms (HIT)
Request 14 : 2ms (HIT)
Request 15 : 2ms (HIT)
Request 16 : 2ms (HIT)
Request 17 : 2ms (HIT)
Request 18 : 2ms (HIT)
Request 19 : 2ms (HIT)
Request 20 : 2ms (HIT)

[Test 5] Measuring Purchase + Cache Invalidation Overhead...
Purchase 1 (Book 2): 56ms
Purchase 2 (Book 3): 22ms
Purchase 3 (Book 4): 26ms
Purchase 4 (Book 5): 27ms
Purchase 5 (Book 6): 27ms
Purchase 6 (Book 7): 22ms
Purchase 7 (Book 1): 23ms
Purchase 8 (Book 2): 22ms
Purchase 9 (Book 3): 24ms
Purchase 10 (Book 4): 20ms

[Test 6] Measuring Cache Miss Latency After Invalidation...
success Cache invalidated frontend-1766341300935
Test 1 - Invalidation: 4ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 2 - Invalidation: 3ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 3 - Invalidation: 4ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 4 - Invalidation: 4ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 5 - Invalidation: 3ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 6 - Invalidation: 4ms, Next Request: 6ms
success Cache invalidated frontend-1766341300935
Test 7 - Invalidation: 3ms, Next Request: 6ms
```

```
Windows PowerShell

success Cache invalidated frontend-1766341300935
Test 8 - Invalidation: 3ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 9 - Invalidation: 3ms, Next Request: 5ms
success Cache invalidated frontend-1766341300935
Test 10 - Invalidation: 4ms, Next Request: 7ms

=== Performance Test Results ===

Search Requests (without cache):
Average: 3.4ms
Median: 2ms
Min: 2ms
Max: 17ms

Search Requests (with cache):
Average: 2.7ms
Median: 2ms
Min: 2ms
Max: 11ms

Search Cache Improvement: 20.59%

Info Requests (without cache):
Average: 7.15ms
Median: 7ms
Min: 5ms
Max: 11ms

Info Requests (with cache):
Average: 2.2ms
Median: 2ms
Min: 2ms
Max: 3ms

Info Cache Improvement: 69.23%

Purchase Operations:
Average: 26.9ms
Median: 23.5ms
Min: 20ms
```

```
Windows PowerShell
Median: 7ms
Min: 5ms
Max: 11ms

Info Requests (with cache):
Average: 2.2ms
Median: 2ms
Min: 2ms
Max: 3ms

Info Cache Improvement: 69.23%

Purchase Operations:
Average: 26.9ms
Median: 23.5ms
Min: 20ms
Max: 56ms

Cache Invalidation:
Average: 3.5ms
Median: 3.5ms
Min: 3ms
Max: 4ms

Results exported to: docs\performance-results.csv

=== Test Complete ===

PS C:\MyDocuments\universtity-courses\Dos\bazar-com> |
```

8. Final status and shutdown

Commands:

```
docker-compose ps
```

```
docker-compose down
```

```
docker-compose ps
```

```
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> docker-compose ps
NAME                IMAGE                COMMAND                SERVICE    CREATED        STATUS
catalog-service    bazar-com-catalog  "docker-entrpoint.s..." catalog    About an hour ago Up About an hour
0.0.0.0:3001->3001/tcp, [::]:3001->3001/tcp
frontend-service   bazar-com-frontend  "docker-entrpoint.s..." frontend  About an hour ago Up About an hour
0.0.0.0:3002->3002/tcp, [::]:3002->3002/tcp
order-service      bazar-com-order     "docker-entrpoint.s..." order     About an hour ago Up About an hour
0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> docker-compose down
[+] Running 4/4
  ✓ Container frontend-service    Removed
  ✓ Container order-service       Removed
  ✓ Container catalog-service     Removed
  ✓ Network bazar-network         Removed
PS C:\MyDocuments\universtity-courses\Dos\bazar-com> docker-compose ps
NAME                IMAGE                COMMAND                SERVICE    CREATED        STATUS    PORTS
PS C:\MyDocuments\universtity-courses\Dos\bazar-com>
```

THANK YOU 🙏