An-Najah National University

Computer Engineering

Distributed Operation System

BAZAR.COM part2

Results report and screen shots

Students

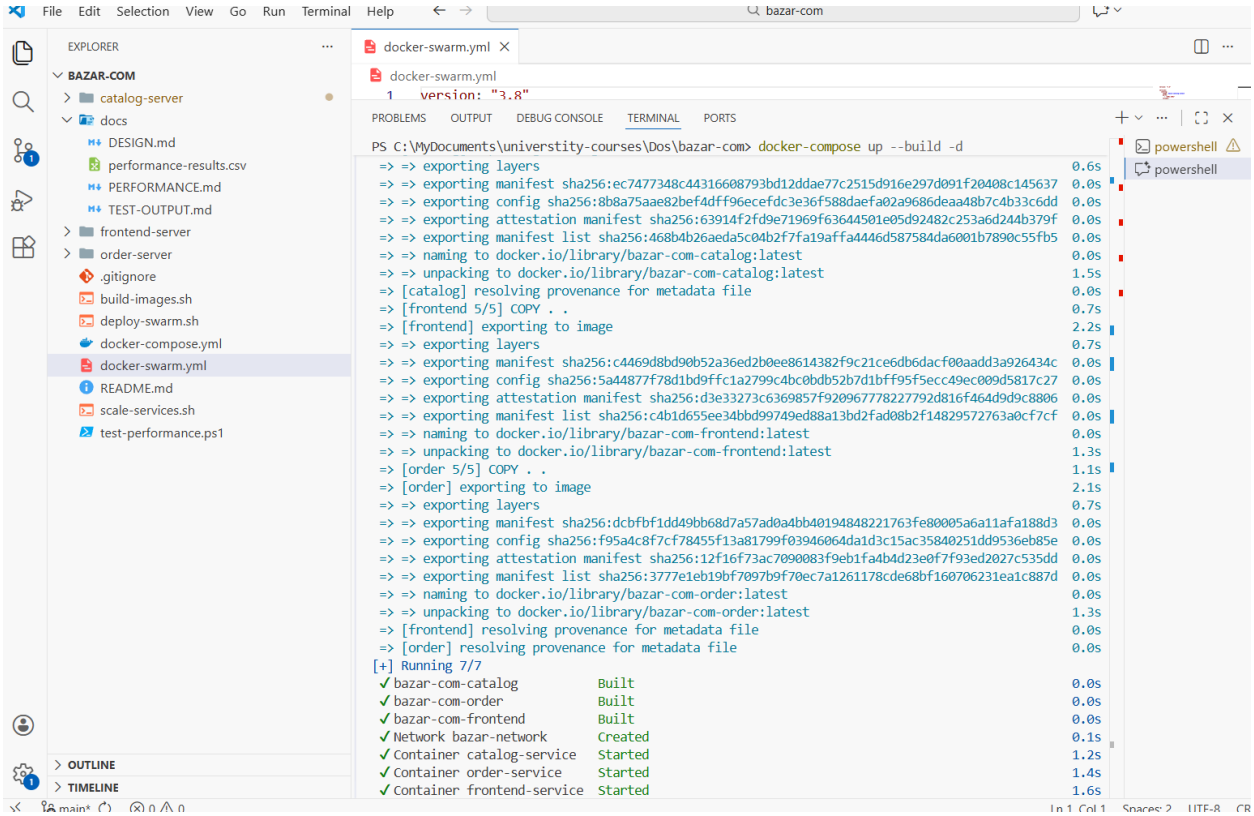
Malik Ali 12114096

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).



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."

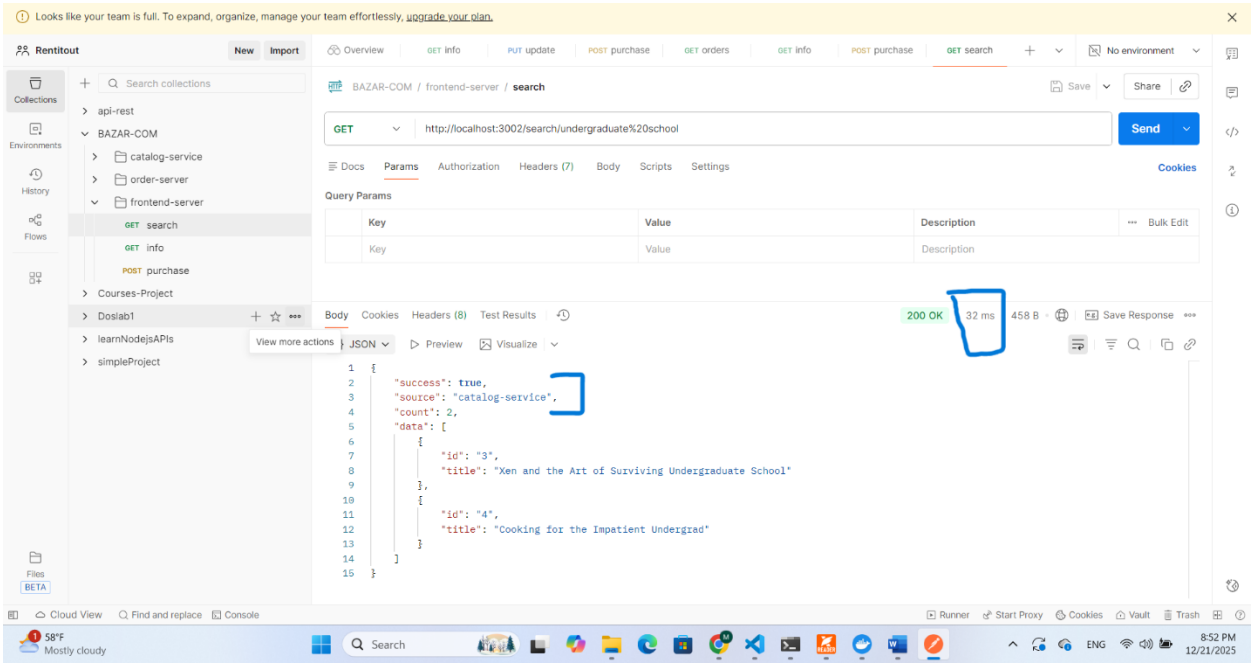
   Invalidate cache:

   Command :

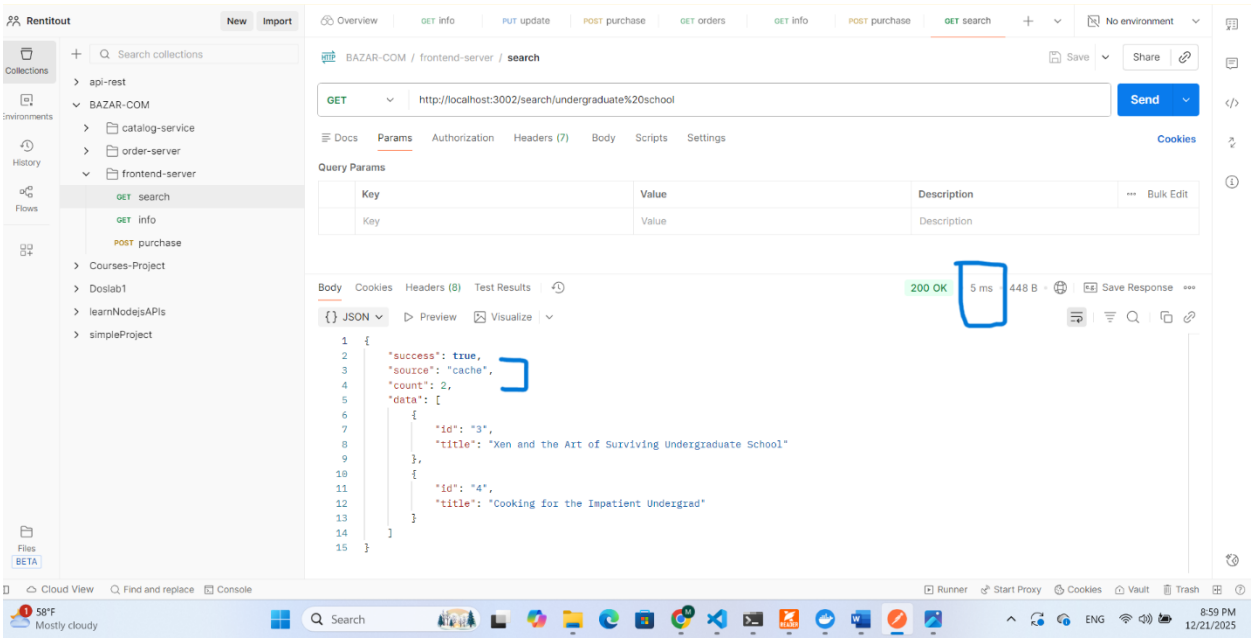
```
Invoke-RestMethod `
  -Uri "http://localhost:3002/invalidate-cache" `
  -Method POST `
  -Headers @{ "Content-Type" = "application/json" } `
  -Body (@{ bookId = "1" } | ConvertTo-Json)
```

```
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):



Request 2(cache hit):

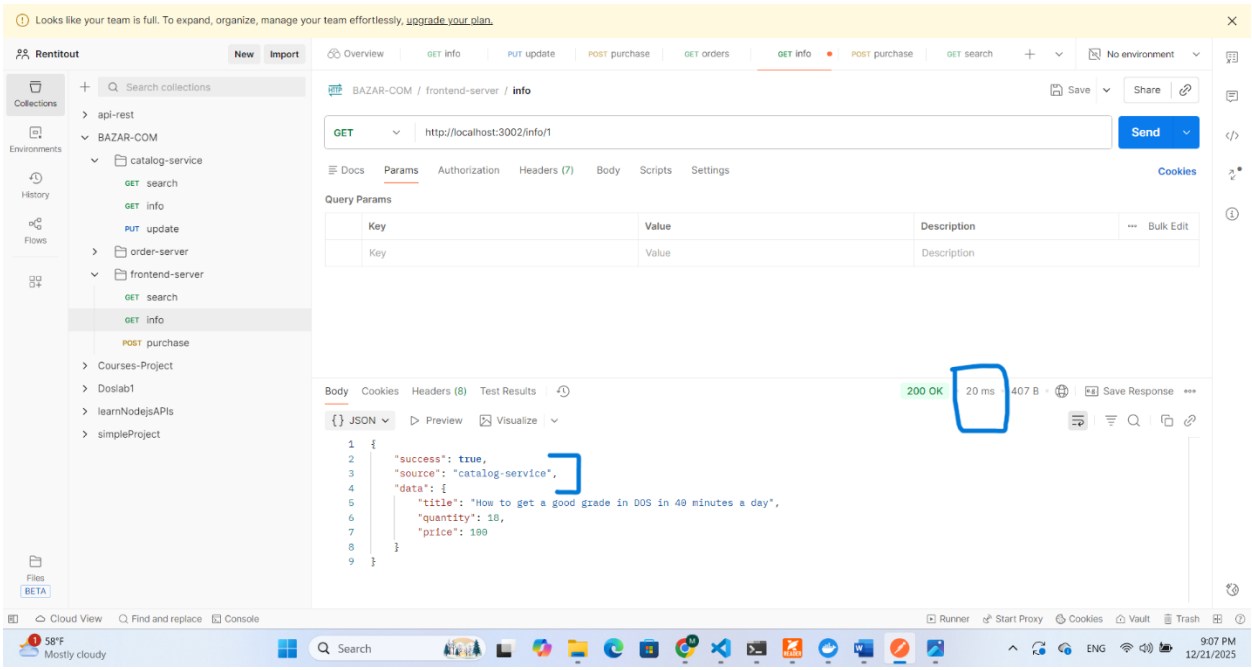


5. **Book info /info/1 — MISS then HIT**

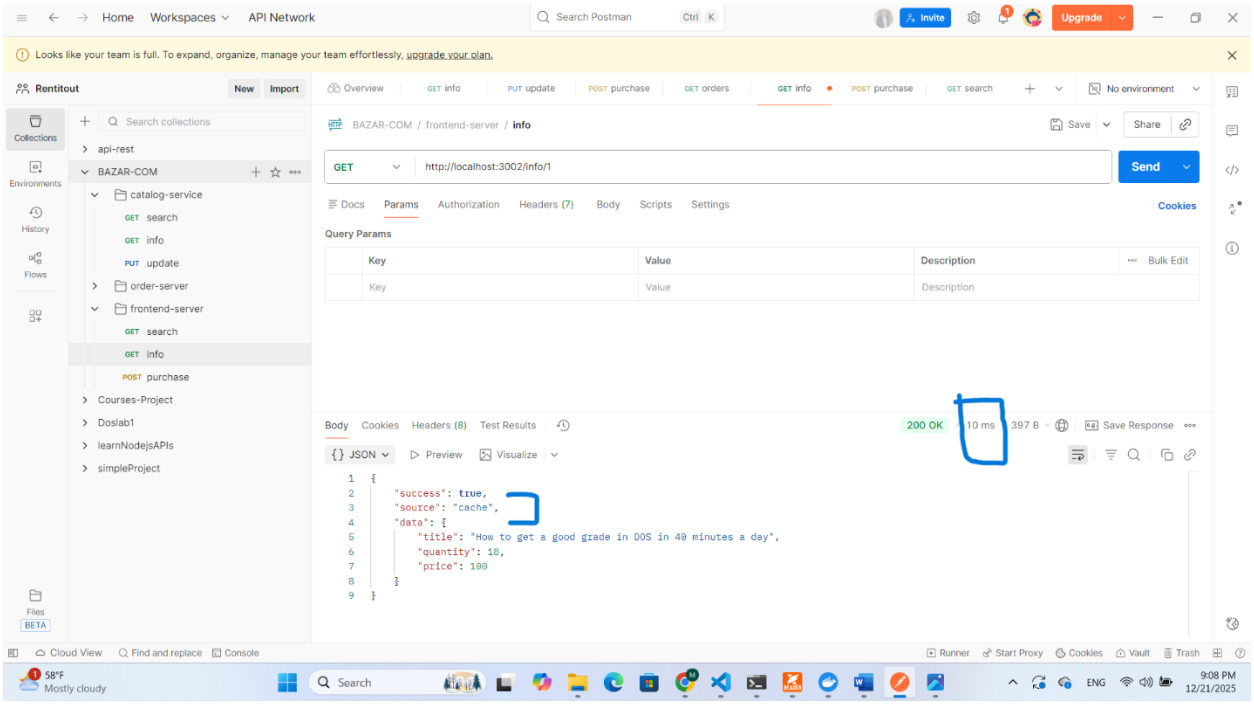
Invalidate cache :

```
>>    -Uri "http://localhost:3002/invalidate-cache" `get a goo
>>    -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):



Request 2(HIT)



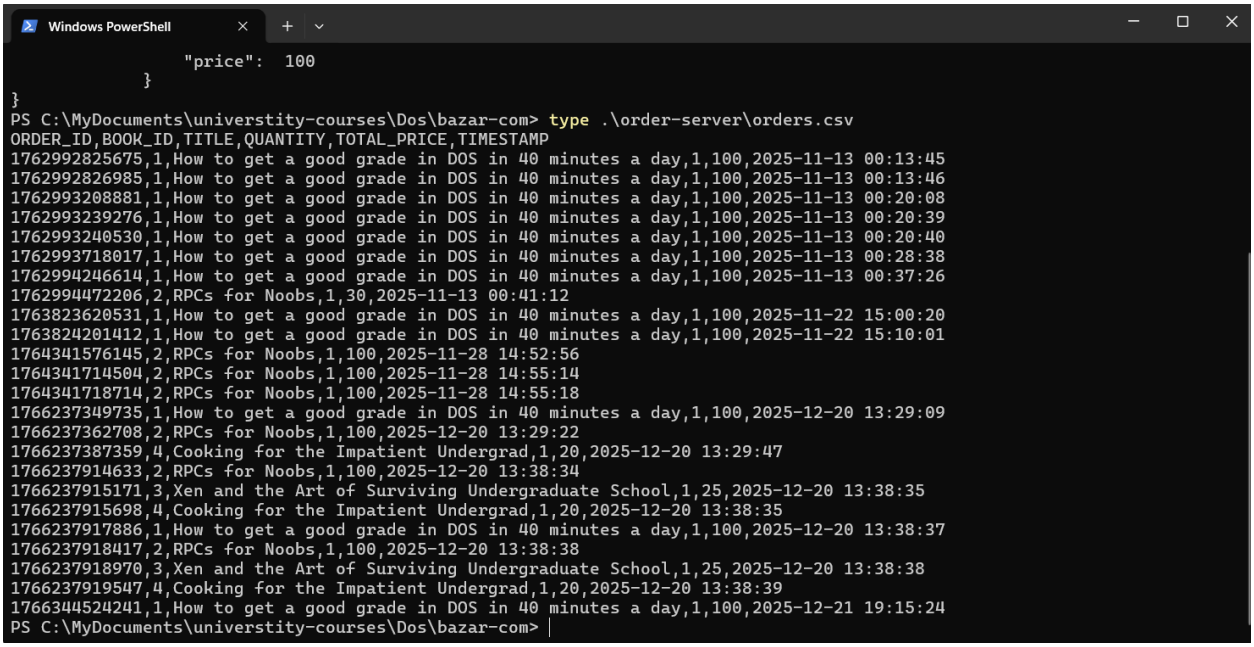
6. **Purchase flow (end-to-end) and confirm orders.csv:**

   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.

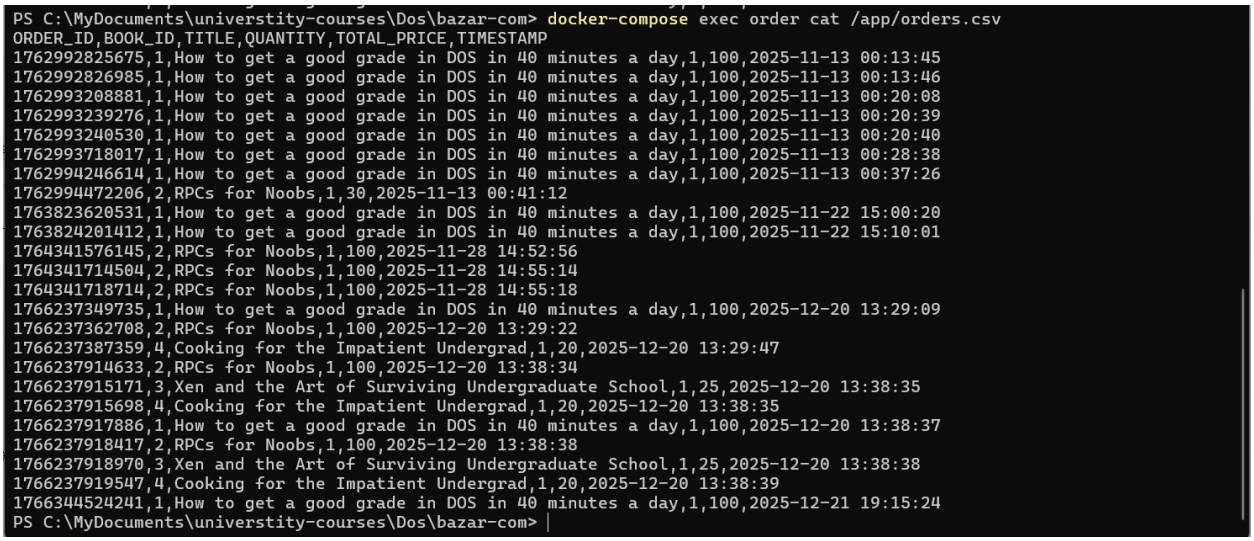
   Purchase:

```
PS C:\MyDocuments\universtity-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:



Show from docker container:



The stock in catalog.csv:

docker-swarm.yml | *catalog.csv* ×

catalog-server > catalog.csv

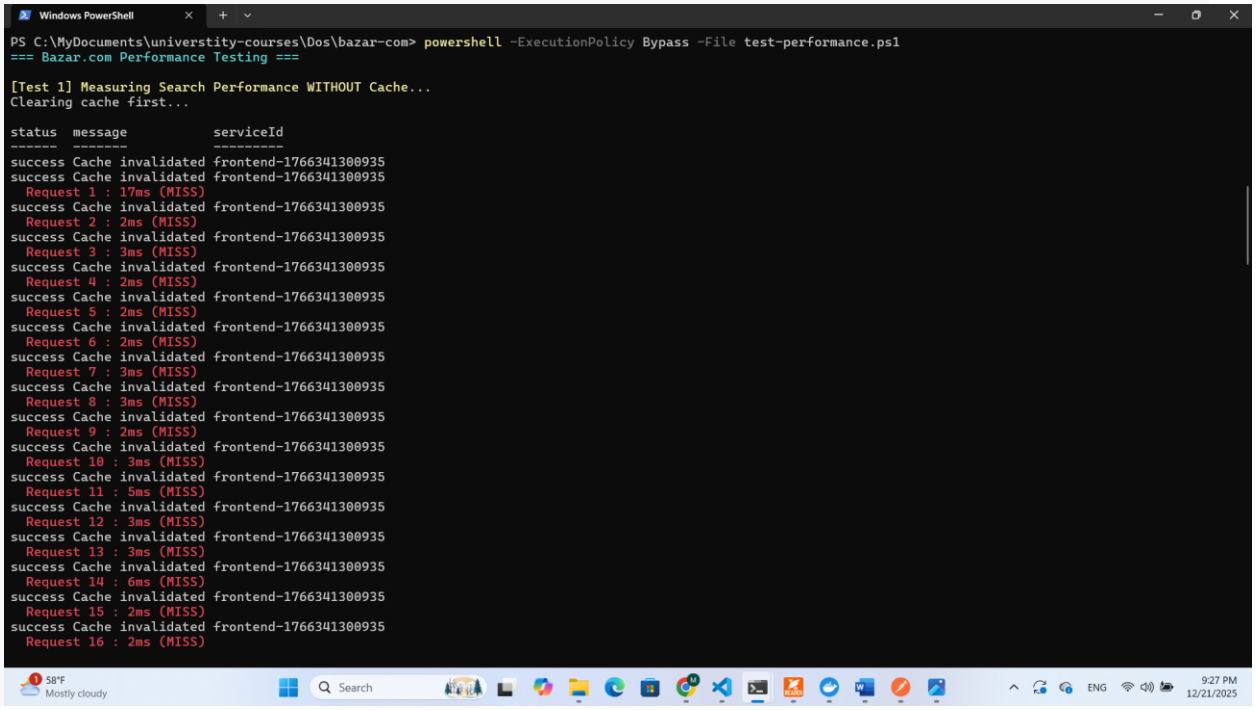
befor=19

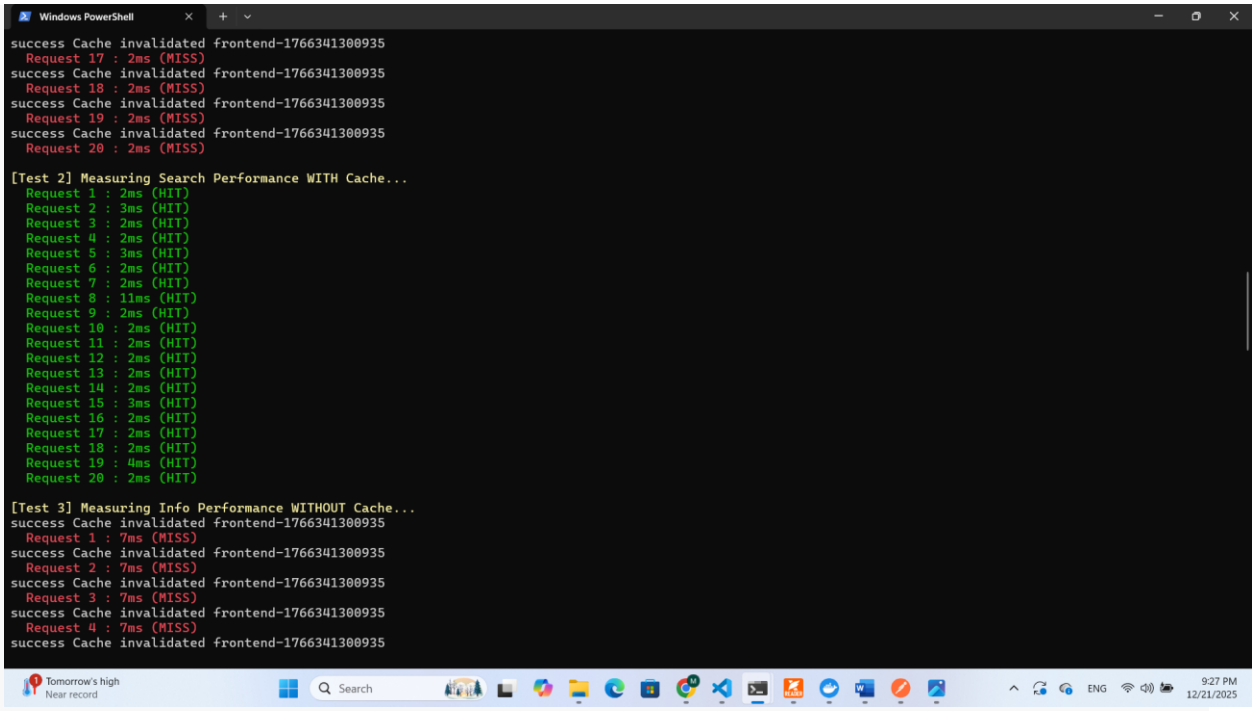
```
1   ID,TOPIC,TITLE,PRICE,STOCK
2   1,distributed systems,How to get a good grade in DOS in 40 minutes a day,100,18
3   2,distributed systems,RPCs for Noobs,100,19
4   3,undergraduate school,Xen and the Art of Surviving Undergraduate School,25,10
5   4,undergraduate school,Cooking for the Impatient 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
```

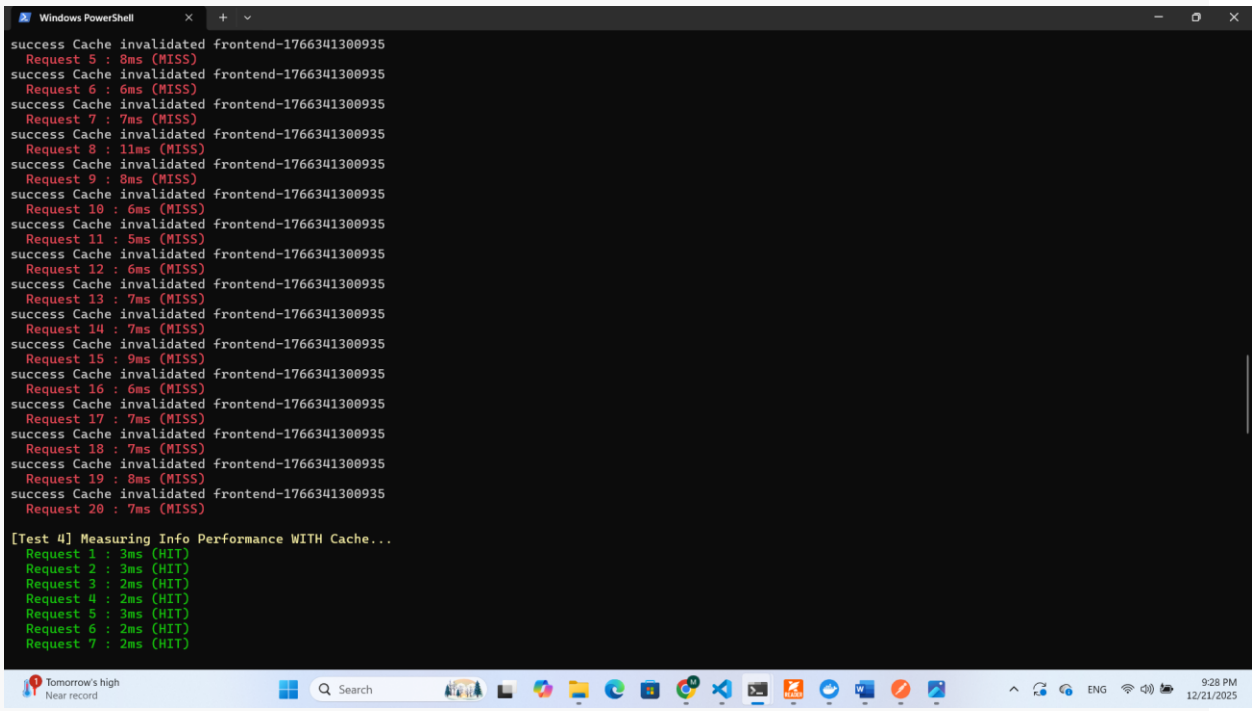
7. Run performance script and collect CSV:

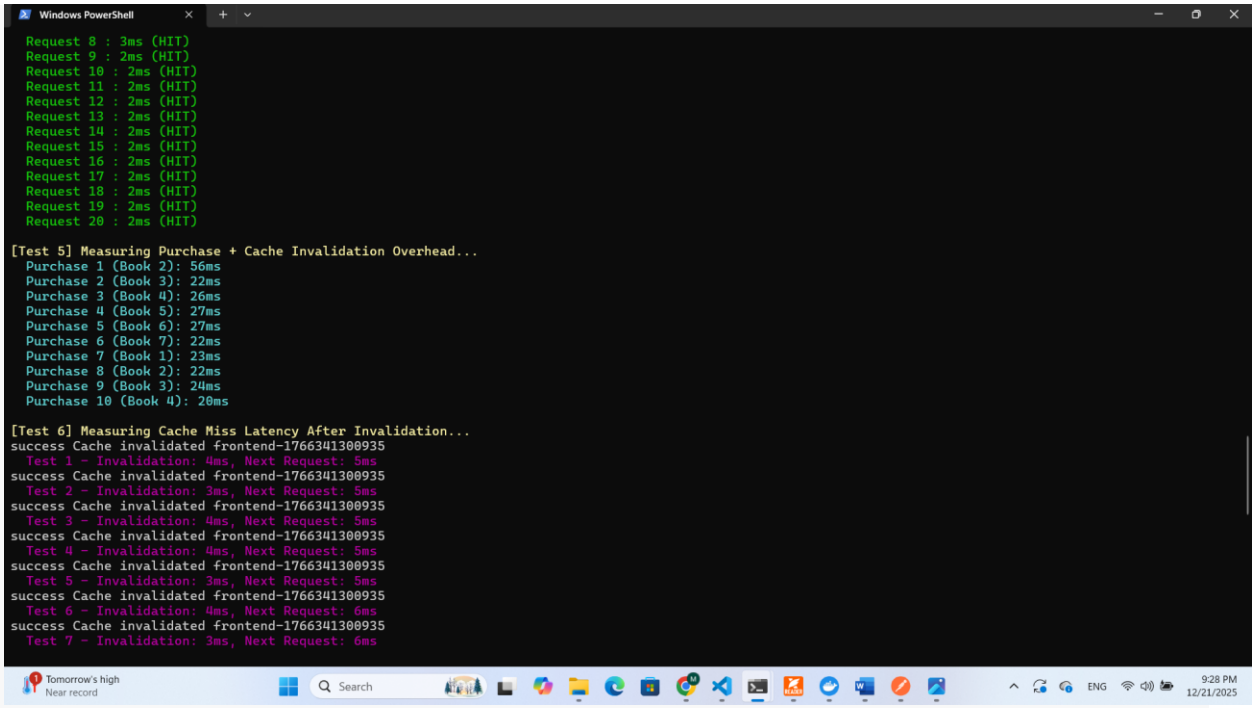
Command (PowerShell): powershell -ExecutionPolicy Bypass -File test-performance.ps1

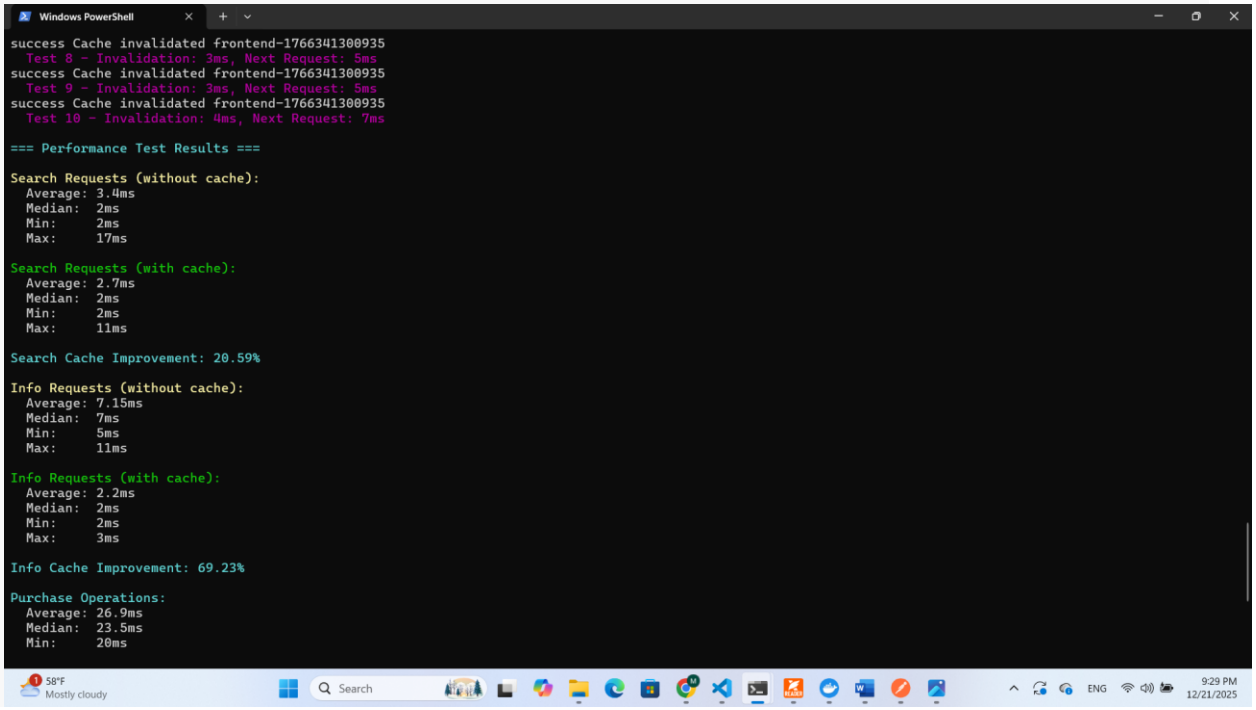
Performance benchmark results: include averages, medians, min/max for Search (with/without cache), Info (with/without cache), Purchase, and Cache Invalidation. Attach the performance-results.csv and add short commentary about observed cache improvement percentages.

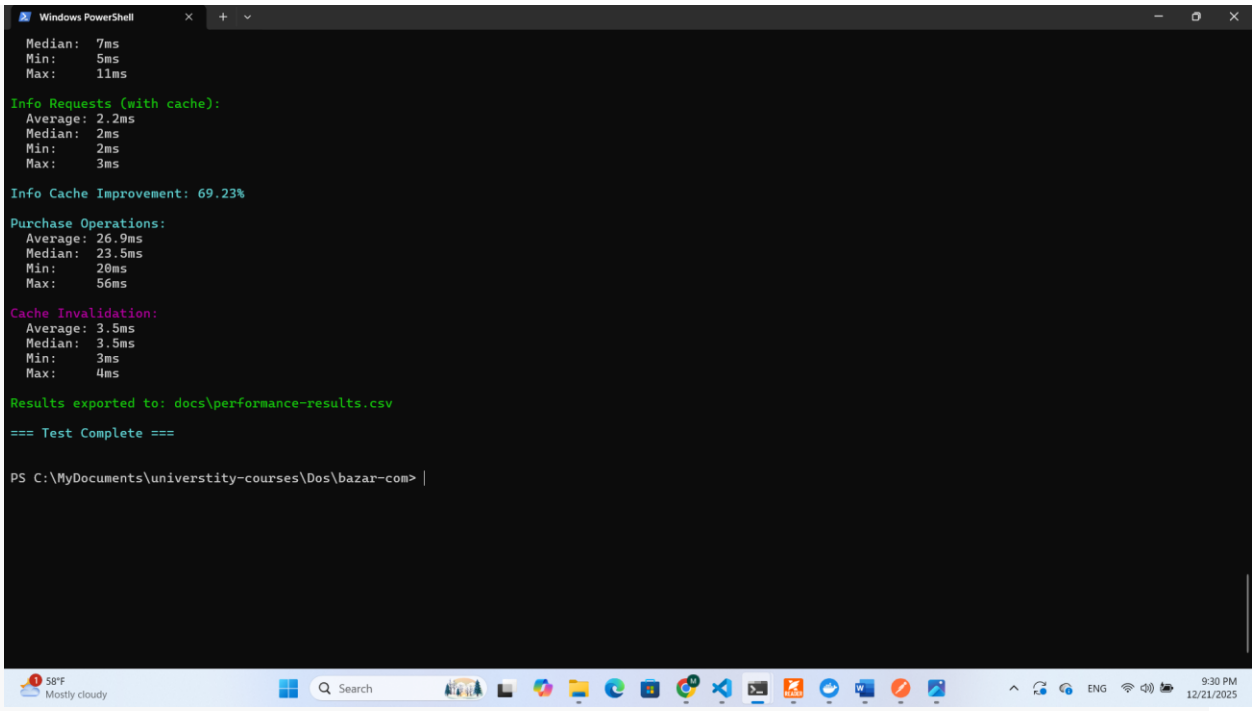



Windows PowerShell
success Cache invalidated frontend-1766341300935
Request 17 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 18 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 19 : 2ms (MISS)
success Cache invalidated frontend-1766341300935
Request 20 : 2ms (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
Tomorrow's high
Near record
Search
ENG
9:27 PM
12/21/2025


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)
Tomorrow's high
Near record
Search
ENG
9:28 PM
12/21/2025


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
Tomorrow's high
Near record
Search
ENG
9:28 PM
12/21/2025


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
58°F
Mostly cloudy
Search
ENG
9:29 PM
12/21/2025



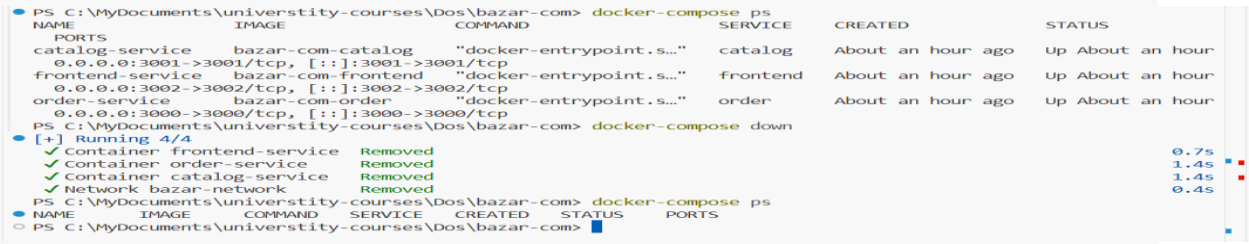
8. Final status and shutdown

Commands:

docker-compose ps

docker-compose down

docker-compose ps



THANK YOU 😊