

Output Document

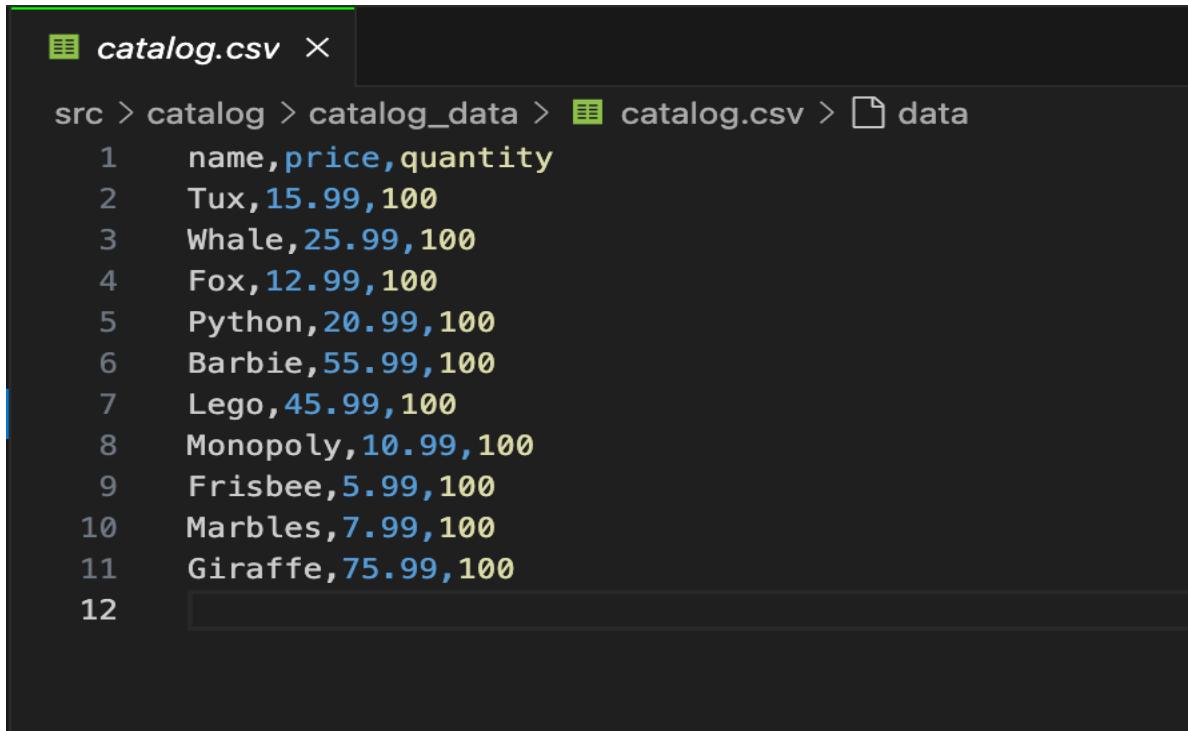
Lab 3

Shreya Birthare

Simran Malik

Running on same setup: microservices and client on local machine

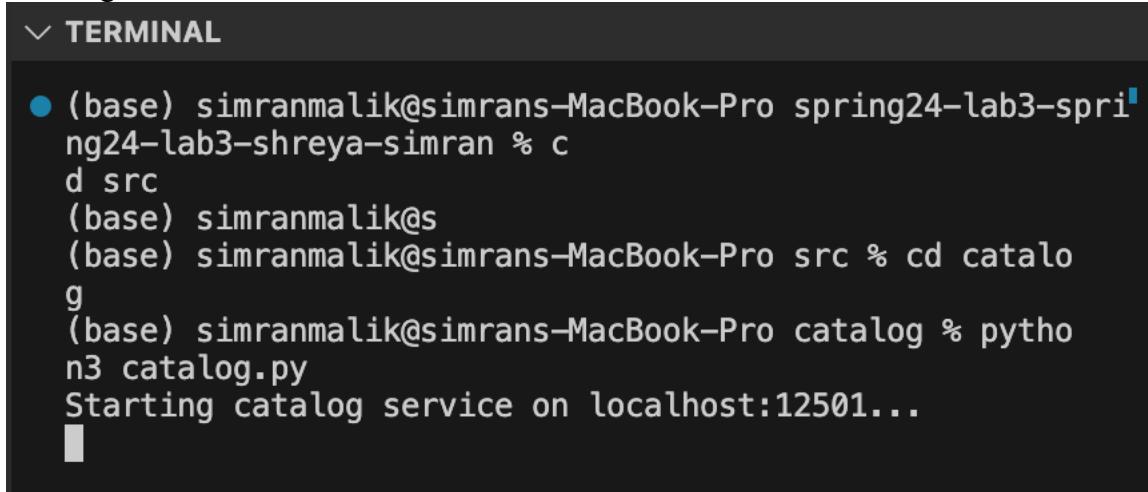
Initial catalog data



The screenshot shows a code editor window with a dark theme. The title bar says "catalog.csv". The file content is a CSV file with 12 rows, each containing three columns: name, price, and quantity. The data is as follows:

	name	price	quantity
1	Tux	15.99	100
2	Whale	25.99	100
3	Fox	12.99	100
4	Python	20.99	100
5	Barbie	55.99	100
6	Lego	45.99	100
7	Monopoly	10.99	100
8	Frisbee	5.99	100
9	Marbles	7.99	100
10	Giraffe	75.99	100
11			
12			

Running microservices on local terminal



The screenshot shows a terminal window with a dark theme. The title bar says "TERMINAL". The terminal output is as follows:

```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % cd src  
(base) simranmalik@simrans-MacBook-Pro src % cd catalog  
(base) simranmalik@simrans-MacBook-Pro catalog % python3 catalog.py  
Starting catalog service on localhost:12501...
```

```
(base) simranmalik@s  
(base) simranmalik@simrans-MacBook-Pro src % cd front_end_service  
(base) simranmalik@simrans-MacBook-Pro front_end_service  
% python3 front_end_service.py  
Starting front-end server on 0.0.0.0:12503...  
[]
```

```
(base) simranmalik@simrans-MacBook-Pro src % cd order  
(base) simranmalik@simrans-MacBook-Pro order % export Replica_id=3  
export ORDER_HOST=localhost  
export ORDER_LISTENING_PORT=12505  
python3 order.py  
Error requesting missed orders from replica 2: HTTPConnectionPool(host='localhost', port=12504): Max retries exceeded with url: /missed_order (Caused by NewConnectionError('<url lib3.connection.HTTPConnection object at 0x101d50400>: Failed to establish a new connection: [Errno 61] Connection refused'))  
Error requesting missed orders from replica 1: HTTPConnectionPool(host='localhost', port=12502): Max retries exceeded with url: /missed_order (Caused by NewConnectionError('<url lib3.connection.HTTPConnection object at 0x101d51660>: Failed to establish a new connection: [Errno 61] Connection refused'))  
Failed to receive missed orders from any replica  
Starting order service on localhost:12505...  
127.0.0.1 - - [06/May/2024 01:26:14] "POST /missed_order HTTP/1.1" 200 -  
127.0.0.1 - - [06/May/2024 01:26:33] "POST /missed_order HTTP/1.1" 200 -  
[]
```

```
(base) simranmalik@simrans-MacBook-Pro src % cd order
(base) simranmalik@simrans-MacBook-Pro order % export Replica_id=2
export ORDER_HOST=localhost
export ORDER_LISTENING_PORT=12504
python3 order.py
Nothing is missed
Starting order service on localhost:12504...
█
```

As it is visible in the screenshot, whenever an order service starts, it checks with the highest order replica id(other than itself-possible leader) if it has missed any order logs.

Output of catalog.py

Successfully handles query and buy requests. Also successfully sends cache invalidation requests to front_end_service when buy request is processed.

```
○ (base) simranmalik@simrans-MacBook-Pro catalog % python3 catalog.py
Starting catalog service on localhost:12501...
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Monopoly HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Whale HTTP/1.1" 200 -
product is in stock, updating catalog
{'data': 'Cache successfully invalidated for Whale'}
127.0.0.1 - - [06/May/2024 03:23:25] "POST /orders HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Barbie HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Python HTTP/1.1" 200 -
product is in stock, updating catalog
{'data': 'Cache successfully invalidated for Python'}
127.0.0.1 - - [06/May/2024 03:23:25] "POST /orders HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Marbles HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Tux HTTP/1.1" 200 -
product is in stock, updating catalog
{'data': 'Cache successfully invalidated for Tux'}
127.0.0.1 - - [06/May/2024 03:23:25] "POST /orders HTTP/1.1" 200 -
product is in stock, updating catalog
{'data': 'Cache successfully invalidated for Marbles'}
127.0.0.1 - - [06/May/2024 03:23:25] "POST /orders HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Whale HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Fox HTTP/1.1" 200 -
127.0.0.1 - - [06/May/2024 03:23:25] "GET /Frisbee HTTP/1.1" 200 -
product is in stock, updating catalog
{'data': 'Cache successfully invalidated for Frisbee'}
127.0.0.1 - - [06/May/2024 03:23:25] "POST /orders HTTP/1.1"
```

Output of front_end_service:

As it is visible in the screenshot, front end service successfully performs health checks and elects leader with highest id, notifies other replicas about leader, shows cache HIT or MISS whenever product query comes in, and invalidates cache for toy when catalog sends cache invalidation request whenever a successful buy operation happens.

```
Leader found: Order Service 3. {'status': 'healthy'}
Successfully notified http://localhost:12504/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
Successfully notified http://localhost:12502/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
127.0.0.1 -- [06/May/2024 02:47:23] "GET /orders/23 HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61286)
***** CACHE MISS *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Monopoly HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61288)
***** CACHE HIT *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Fox HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61289)
***** CACHE HIT *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Tux HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61290)
***** CACHE MISS *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Lego HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61292)
Leader found: Order Service 3. {'status': 'healthy'}
Successfully notified http://localhost:12504/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
Successfully notified http://localhost:12502/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
Thread ID 6201307136 handling request from ('127.0.0.1', 61298)
Cache successfully invalidated for Lego
127.0.0.1 -- [06/May/2024 02:47:23] "POST /invalidate/Lego HTTP/1.1" 200 -
127.0.0.1 -- [06/May/2024 02:47:23] "POST /orders/ HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61301)
Leader found: Order Service 3. {'status': 'healthy'}
Successfully notified http://localhost:12502/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
Successfully notified http://localhost:12504/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
127.0.0.1 -- [06/May/2024 02:47:23] "GET /orders/24 HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61306)
***** CACHE HIT *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Tux HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61307)
***** CACHE HIT *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Fox HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61308)
***** CACHE MISS *****
127.0.0.1 -- [06/May/2024 02:47:23] "GET /products/Barbie HTTP/1.1" 200 -
Thread ID 6183333888 handling request from ('127.0.0.1', 61310)
Leader found: Order Service 3. {'status': 'healthy'}
Successfully notified http://localhost:12502/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
Successfully notified http://localhost:12504/notify_leader_info_to_replica.
Leader ID: 3 accepted. {'status': 'Leader updated successfully'}
```

Order leader replica 3 output: Responds to health checks and shows when it is selected as leader, whenever any order request comes in. Processes both order query and purchase requests. Generates order number, updates its order log and replicates order log update to followers after successful buy operation.

```
-  
i am now leader  
Processing order forWhale, 1 quantity  
Successfully propagated to http://localhost:12502/replicate_order  
r  
Successfully propagated to http://localhost:12504/replicate_order  
r  
127.0.0.1 -- [06/May/2024 03:23:26] "POST /orders HTTP/1.1" 200  
-  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /orders/12 HTTP/1.1" 200 -  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
Processing order forPython, 8 quantity  
Successfully propagated to http://localhost:12504/replicate_order  
r  
Successfully propagated to http://localhost:12502/replicate_order  
r  
127.0.0.1 -- [06/May/2024 03:23:26] "POST /orders HTTP/1.1" 200  
-  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /orders/13 HTTP/1.1" 200 -  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
Processing order forMarbles, 6 quantity  
Successfully propagated to http://localhost:12502/replicate_order  
r  
Successfully propagated to http://localhost:12504/replicate_order  
r  
127.0.0.1 -- [06/May/2024 03:23:26] "POST /orders HTTP/1.1" 200  
-  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /orders/14 HTTP/1.1" 200 -  
127.0.0.1 -- [06/May/2024 03:23:26] "GET /health HTTP/1.1" 200  
-  
i am now leader  
Processing order forGiraffe, 4 quantity
```

Order replica 1, 2 output:

Successfully receives notification about leader from front end service and processes order log replication requests from leader replica 3

P/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /notify_leader_info_to_repli
ca HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /replicate_order HTTP/1.1"
200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /notify_leader_info_to_repli
ca HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /notify_leader_info_to_repli
ca HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /replicate_order HTTP/1.1"
200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /notify_leader_info_to_repli
ca HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29]
] "POST /replicate_order HTTP/1.1"
200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost', 'port': 12505}

P/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29] "POST /r
eplicate_order HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29] "POST /r
eplicate_order HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29] "POST /r
eplicate_order HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -
Order Committed Successfully
127.0.0.1 -- [06/May/2024 02:47:29] "POST /r
eplicate_order HTTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {
'host': 'localhost', 'port': 12505}
127.0.0.1 -- [06/May/2024 02:47:29] "POST /n
otify_leader_info_to_replica HTTP/1.1" 200 -

Client Output:

Successfully gets all its queries processed.

Also successfully verifies all its successful orders with order query requests

```
{'name': 'Fox', 'price': 12.99, 'quantity': 100}}
placing order for Fox, 7
Order result for Fox: {'data': {'order_number': 10}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 78}}
placing order for Marbles, 6
Order result for Marbles: {'data': {'order_number': 11}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 94}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 83}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 94}}
placing order for Whale, 1
Order result for Whale: {'data': {'order_number': 12}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 98}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 83}}
placing order for Python, 8
Order result for Python: {'data': {'order_number': 13}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 72}}
placing order for Marbles, 6
Order result for Marbles: {'data': {'order_number': 14}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 93}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 98}}
placing order for Giraffe, 4
Order result for Giraffe: {'data': {'order_number': 15}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 66}}
placing order for Marbles, 10
Order result for Marbles: {'data': {'order_number': 16}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 56}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 100}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 75}}
placing order for Python, 1
Order result for Python: {'data': {'order_number': 17}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 94}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 94}}
placing order for Giraffe, 8
Order result for Giraffe: {'data': {'order_number': 18}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 93}}
```

```
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 93}}
placing order for Fox, 3
Order result for Fox: {'data': {'order_number': 19}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 93}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 74}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 96}}
Verified order 0: {'data': {'data': {'number': '0', 'name': 'Whale', 'quantity': '2'}}}
Verified order 1: {'data': {'data': {'number': '1', 'name': 'Python', 'quantity': '9'}}}
Verified order 2: {'data': {'data': {'number': '2', 'name': 'Tux', 'quantity': '9'}}}
Verified order 3: {'data': {'data': {'number': '3', 'name': 'Marbles', 'quantity': '6'}}}
Verified order 4: {'data': {'data': {'number': '4', 'name': 'Frisbee', 'quantity': '4'}}}
Verified order 5: {'data': {'data': {'number': '5', 'name': 'Whale', 'quantity': '4'}}}
Verified order 6: {'data': {'data': {'number': '6', 'name': 'Marbles', 'quantity': '8'}}}
Verified order 7: {'data': {'data': {'number': '7', 'name': 'Python', 'quantity': '8'}}}
Verified order 8: {'data': {'data': {'number': '8', 'name': 'Marbles', 'quantity': '8'}}}
Verified order 9: {'data': {'data': {'number': '9', 'name': 'Giraffe', 'quantity': '2'}}}
Verified order 10: {'data': {'data': {'number': '10', 'name': 'Fox', 'quantity': '7'}}}
Verified order 11: {'data': {'data': {'number': '11', 'name': 'Marbles', 'quantity': '6'}}}
Verified order 12: {'data': {'data': {'number': '12', 'name': 'Whale', 'quantity': '1'}}}
Verified order 13: {'data': {'data': {'number': '13', 'name': 'Python', 'quantity': '8'}}}
Verified order 14: {'data': {'data': {'number': '14', 'name': 'Marbles', 'quantity': '6'}}}
Verified order 15: {'data': {'data': {'number': '15', 'name': 'Giraffe', 'quantity': '4'}}}
Verified order 16: {'data': {'data': {'number': '16', 'name': 'Marbles', 'quantity': '10'}}}
Verified order 17: {'data': {'data': {'number': '17', 'name': 'Python', 'quantity': '1'}}}
Verified order 18: {'data': {'data': {'number': '18', 'name': 'Giraffe', 'quantity': '8'}}}
Verified order 19: {'data': {'data': {'number': '19', 'name': 'Frisbee', 'quantity': '96'}}}
○ (base) simranmalik@simrans-MacBook-Pro src %
```

Updated catalog:

```
src > catalog > catalog_data > catalog.csv > data
1   name,price,quantity
2   Tux,15.99,91
3   Whale,25.99,93
4   Fox,12.99,90
5   Python,20.99,74
6   Barbie,55.99,100
7   Lego,45.99,100
8   Monopoly,10.99,100
9   Frisbee,5.99,96
10  Marbles,7.99,56
11  Giraffe,75.99,86
12  |
```

Order logs: same for all replicas

```
src > order > order_data > order_log_1.csv > data
1   0,Whale,2
2   1,Python,9
3   2,Tux,9
4   3,Marbles,6
5   4,Frisbee,4
6   5,Whale,4
7   6,Marbles,8
8   7,Python,8
9   8,Marbles,8
10  9,Giraffe,2
11  10,Fox,7
12  11,Marbles,6
13  12,Whale,1
14  13,Python,8
15  14,Marbles,6
16  15,Giraffe,4
17  16,Marbles,10
18  17,Python,1
19  18,Giraffe,8
20  19,Fox,3
```

src > order > order_data > order_log_2.csv > data

```
1  0,Whale,2
2  1,Python,9
3  2,Tux,9
4  3,Marbles,6
5  4,Frisbee,4
6  5,Whale,4
7  6,Marbles,8
8  7,Python,8
9  8,Marbles,8
10 9,Giraffe,2
11 10,Fox,7
12 11,Marbles,6
13 12,Whale,1
14 13,Python,8
15 14,Marbles,6
16 15,Giraffe,4
17 16,Marbles,10
18 17,Python,1
19 18,Giraffe,8
20 19,Fox,3
21
```

```

src > order > order_data > 📁 order_log_3.csv > 📄 data
1  0,Whale,2
2  1,Python,9
3  2,Tux,9
4  3,Marbles,6
5  4,Frisbee,4
6  5,Whale,4
7  6,Marbles,8
8  7,Python,8
9  8,Marbles,8
10 9,Giraffe,2
11 10,Fox,7
12 11,Marbles,6
13 12,Whale,1
14 13,Python,8
15 14,Marbles,6
16 15,Giraffe,4
17 16,Marbles,10
18 17,Python,1
19 18,Giraffe,8
20 19,Fox,3

```

Testing for restocking(separately in another run of application):

Successful restocking of Barbie and Monopoly:

```

127.0.0.1 - - [06/May/2024 04:59:24] "GET /Whale HTTP/1.1" 200 -
Restocking Barbie
{"data": "Cache successfully invalidated for Barbie"}
127.0.0.1 - - [06/May/2024 05:00:35] "POST /orders HTTP/1.1" 400 -
product is in stock, updating catalog
{"data": "Cache successfully invalidated for Monopoly"}
127.0.0.1 - - [06/May/2024 05:00:45] "POST /orders HTTP/1.1" 200 -
Restocking Monopoly
{"data": "Cache successfully invalidated for Monopoly"}
127.0.0.1 - - [06/May/2024 05:01:05] "POST /orders HTTP/1.1" 400 -

```

Catalog Before Restocking:

```
src > catalog > catalog_data > catalog.csv > data
1   name,price,quantity
2   Tux,15.99,2
3   Whale,25.99,5
4   Fox,12.99,73
5   Python,20.99,1
6   Barbie,55.99,0
7   Lego,45.99,2
8   Monopoly,10.99,0
9   Frisbee,5.99,1
10  Marbles,7.99,1
11  Giraffe,75.99,1
12
```

Catalog after Restocking:

```
src > catalog > catalog_data > catalog.csv > data
1   name,price,quantity
2   Tux,15.99,2
3   Whale,25.99,5
4   Fox,12.99,73
5   Python,20.99,1
6   Barbie,55.99,100
7   Lego,45.99,2
8   Monopoly,10.99,100
9   Frisbee,5.99,1
10  Marbles,7.99,1
11  Giraffe,75.99,1
12
```

Running on different setup: microservices on Edlab and client on local machine

Initial catalog:

PROBLEMS 3 OUTPUT PORTS TERMINAL

GNU nano 6.2 catalog.csv

name	price	quantity
Tux	15.99	100
Whale	25.99	100
Fox	12.99	100
Python	20.99	100
Barbie	55.99	100
Lego	45.99	100
Monopoly	10.99	100
Frisbee	5.99	100
Marbles	7.99	100
Giraffe	75.99	100

elinux2
src) >
cd fron
t_end_s
ervice
elinux2
front_e
nd_serv
ice) >
python3
front_
end_ser
vice.py
Startin
g front
-end se
rver on
0.0.0.
0:12503
:::

```
PROBLEMS ③ OUTPUT PORTS TERMINAL JavaSE-21 - src +  
  
elnux2 catalog) > elinux2 src > cd front_end_service  
> python3 catalog.py  
Starting catalog service on localhost:12501...  
[]  
  
elnux2 order) > python3 order.py  
Error requesting missed orders from replica 1: HTTPConnectionPool(host='localhost', port=12504): Max retries exceeded with url: /missed_order (Cause by NewConnectionError('urllib3.connection.HTTPConnection object at 0x7f644f5c50>: Failed to establish a new connection: [Errno 111] Connection refused'))  
Error requesting missed orders from replica 1: HTTPConnectionPool(host='localhost', port=12502): Max retries exceeded with url: /missed_order (Cause by NewConnectionError('urllib3.connection.HTTPConnection object at 0x7f644f5c50>: Failed to establish a new connection: [Errno 111] Connection refused'))  
Failed to receive missed orders from any replica  
Starting order service on localhost:12505...  
127.0.0.1 -- [06/May/2024 08:31:34]  
"POST /missed_order HTTP/1.1" 200 -  
127.0.0.1 -- [06/May/2024 08:31:35]  
"POST /missed_order HTTP/1.1" 200 -  
[]  
  
elnux2 order) > python3 order.py  
Nothing is missed  
Starting order service on localhost:12504...  
[]  
  
elnux2 order) > python3 order.py  
Nothing is missed  
Starting order service on localhost:12502...  
[]  
  
● (base) simranmalik@simrans-MacBook-Pro src % export FRONTEND_HOST=128.119.243.164  
○ (base) simranmalik@simrans-MacBook-Pro src %
```

Output for client:

```
● (base) simranmalik@simrans-MacBook-Pro src % export FRONTEND_HOST=128.119.243.164
● (base) simranmalik@simrans-MacBook-Pro src % python3 client.py
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 100}}
placing order for Python, 6
Order result for Python: {'data': {'order_number': 0}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 100}}
placing order for Frisbee, 10
Order result for Frisbee: {'data': {'order_number': 1}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 100}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 100}}
placing order for Monopoly, 3
Order result for Monopoly: {'data': {'order_number': 2}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 94}}
placing order for Python, 5
Order result for Python: {'data': {'order_number': 3}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 100}}
placing order for Fox, 10
Order result for Fox: {'data': {'order_number': 4}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 90}}
placing order for Frisbee, 8
Order result for Frisbee: {'data': {'order_number': 5}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 97}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 100}}
placing order for Giraffe, 8
Order result for Giraffe: {'data': {'order_number': 6}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 100}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 100}}
placing order for Whale, 9
Order result for Whale: {'data': {'order_number': 7}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 91}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 82}}
placing order for Frisbee, 3
Order result for Frisbee: {'data': {'order_number': 8}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
placing order for Lego, 3
Order result for Lego: {'data': {'order_number': 9}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 90}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 97}}
placing order for Monopoly, 6
Order result for Monopoly: {'data': {'order_number': 10}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 89}}
placing order for Python, 3
Order result for Python: {'data': {'order_number': 11}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 90}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 91}}
placing order for Monopoly, 8
Order result for Monopoly: {'data': {'order_number': 12}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 92}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 86}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
placing order for Tux, 5
```

```

Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 97}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 88}}
placing order for Giraffe, 10
Order result for Giraffe: {'data': {'order_number': 16}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 91}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 97}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 78}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 90}}
placing order for Fox, 10
Order result for Fox: {'data': {'order_number': 17}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 86}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 91}}
placing order for Tux, 6
Order result for Tux: {'data': {'order_number': 18}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 83}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 80}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 97}}
placing order for Lego, 9
Order result for Lego: {'data': {'order_number': 19}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 86}}
placing order for Python, 7
Order result for Python: {'data': {'order_number': 20}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
placing order for Barbie, 3
Order result for Barbie: {'data': {'order_number': 21}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 79}}
Query result for Whale: {'data': {'name': 'Whale', 'price': 25.99, 'quantity': 91}}
Verified order 0: {'data': {'data': {'number': '0', 'name': 'Python', 'quantity': '6'}}}
Verified order 1: {'data': {'data': {'number': '1', 'name': 'Frisbee', 'quantity': '10'}}}
Verified order 2: {'data': {'data': {'number': '2', 'name': 'Monopoly', 'quantity': '3'}}}
Verified order 3: {'data': {'data': {'number': '3', 'name': 'Python', 'quantity': '5'}}}
Verified order 4: {'data': {'data': {'number': '4', 'name': 'Fox', 'quantity': '10'}}}
Verified order 5: {'data': {'data': {'number': '5', 'name': 'Frisbee', 'quantity': '8'}}}
Verified order 6: {'data': {'data': {'number': '6', 'name': 'Giraffe', 'quantity': '8'}}}
Verified order 7: {'data': {'data': {'number': '7', 'name': 'Whale', 'quantity': '9'}}}
Verified order 8: {'data': {'data': {'number': '8', 'name': 'Frisbee', 'quantity': '3'}}}
Verified order 9: {'data': {'data': {'number': '9', 'name': 'Lego', 'quantity': '3'}}}
Verified order 10: {'data': {'data': {'number': '10', 'name': 'Monopoly', 'quantity': '6'}}}
Verified order 11: {'data': {'data': {'number': '11', 'name': 'Python', 'quantity': '3'}}}
Verified order 12: {'data': {'data': {'number': '12', 'name': 'Monopoly', 'quantity': '8'}}}
Verified order 13: {'data': {'data': {'number': '13', 'name': 'Tux', 'quantity': '5'}}}
Verified order 14: {'data': {'data': {'number': '14', 'name': 'Tux', 'quantity': '4'}}}
Verified order 15: {'data': {'data': {'number': '15', 'name': 'Giraffe', 'quantity': '4'}}}
Verified order 16: {'data': {'data': {'number': '16', 'name': 'Giraffe', 'quantity': '10'}}}
Verified order 17: {'data': {'data': {'number': '17', 'name': 'Fox', 'quantity': '10'}}}
Verified order 18: {'data': {'data': {'number': '18', 'name': 'Tux', 'quantity': '6'}}}
Verified order 19: {'data': {'data': {'number': '19', 'name': 'Lego', 'quantity': '9'}}}
Verified order 20: {'data': {'data': {'number': '20', 'name': 'Python', 'quantity': '7'}}}
Verified order 21: {'data': {'data': {'number': '21', 'name': 'Barbie', 'quantity': '3'}}}
Results for buy/purchase probability: 40.0%
***Average Response Time Product Query: 0.23322077751159667 seconds***
***Average Response Time Order Query: 0.3147640011527322 seconds***
***Average Response Time Buy Request: 0.5098049098795111 seconds***

```

Updated catalog

GNU nano 6.2			catalog.csv
	name	, price, quantity	
	Tux,	15.99,85	
	Whale,	25.99,91	
	Fox,	12.99,80	
	Python,	20.99,79	
	Barbie,	55.99,97	
	Lego,	45.99,88	
	Monopoly,	10.99,83	
	Frisbee,	5.99,79	
	Marbles,	7.99,100	
	Giraffe,	75.99,78	

New generated order logs(same for all order replicas)

```
elnux2 order) > cd order_data
elnux2 order_data) > ls
order_log_1.csv  order_log_2.csv  order_log_3.csv
elnux2 order_data) > []
```

```
GNU nano 6.2                                     order_log_1.csv
0,Python,6
1,Frisbee,10
2,Monopoly,3
3,Python,5
4,Fox,10
5,Frisbee,8
6,Giraffe,8
7,Whale,9
8,Frisbee,3
9,Lego,3
10,Monopoly,6
11,Python,3
12,Monopoly,8
13,Tux,5
14,Tux,4
15,Giraffe,4
16,Giraffe,10
17,Fox,10
18,Tux,6
19,Lego,9
20,Python,7
21,Barbie,3
```

```
GNU nano 6.2                                     order_log_2.csv
0,Python,6
1,Frisbee,10
2,Monopoly,3
3,Python,5
4,Fox,10
5,Frisbee,8
6,Giraffe,8
7,Whale,9
8,Frisbee,3
9,Lego,3
10,Monopoly,6
11,Python,3
12,Monopoly,8
13,Tux,5
14,Tux,4
15,Giraffe,4
16,Giraffe,10
17,Fox,10
18,Tux,6
19,Lego,9
20,Python,7
21,Barbie,3
```

GNU nano 6.2

order_log_3.csv

```
0,Python,6
1,Frisbee,10
2,Monopoly,3
3,Python,5
4,Fox,10
5,Frisbee,8
6,Giraffe,8
7,Whale,9
8,Frisbee,3
9,Lego,3
10,Monopoly,6
11,Python,3
12,Monopoly,8
13,Tux,5
14,Tux,4
15,Giraffe,4
16,Giraffe,10
17,Fox,10
18,Tux,6
19,Lego,9
20,Python,7
21,Barbie,3
```

Part 5 Raft output:

Raft logs:

Same for all replicas:

```
src > Part_5-RAFT > order_RAFT > raft_data > 📁 raft_log_3.csv > 📄 data
1  0,1,Whale,10
2  1,1,Frisbee,5
3  2,1,Frisbee,9
4  3,1,Monopoly,10
5  4,1,Lego,1
6  5,1,Python,1
7  6,1,Python,6
8  7,1,Whale,5
9  8,1,Whale,10
10 9,1,Fox,5
11 10,1,Marbles,8
12 11,1,Monopoly,3
13 12,1,Frisbee,10
14 13,1,Lego,5
15 14,1,Giraffe,5
16 15,1,Tux,5
17 16,1,Lego,9
18 17,1,Tux,7
19 18,1,Giraffe,7
20 19,1,Giraffe,3
21 20,1,Fox,5
22 21,1,Frisbee,3
23 22,1,Marbles,10
24
```

src > Part_5-RAFT > order_RAFT > raft_data >  raft_log_2.csv >  data

```
1  0,1,Whale,10
2  1,1,Frisbee,5
3  2,1,Frisbee,9
4  3,1,Monopoly,10
5  4,1,Lego,1
6  5,1,Python,1
7  6,1,Python,6
8  7,1,Whale,5
9  8,1,Whale,10
10 9,1,Fox,5
11 10,1,Marbles,8
12 11,1,Monopoly,3
13 12,1,Frisbee,10
14 13,1,Lego,5
15 14,1,Giraffe,5
16 15,1,Tux,5
17 16,1,Lego,9
18 17,1,Tux,7
19 18,1,Giraffe,7
20 19,1,Giraffe,3
21 20,1,Fox,5
22 21,1,Frisbee,3
23 22,1,Marbles,10
24
```

src > Part_5-RAFT > order_RAFT > raft_data > raft_log_3.csv > data

```
1 0,1,Whale,10
2 1,1,Frisbee,5
3 2,1,Frisbee,9
4 3,1,Monopoly,10
5 4,1,Lego,1
6 5,1,Python,1
7 6,1,Python,6
8 7,1,Whale,5
9 8,1,Whale,10
10 9,1,Fox,5
11 10,1,Marbles,8
12 11,1,Monopoly,3
13 12,1,Frisbee,10
14 13,1,Lego,5
15 14,1,Giraffe,5
16 15,1,Tux,5
17 16,1,Lego,9
18 17,1,Tux,7
19 18,1,Giraffe,7
20 19,1,Giraffe,3
21 20,1,Fox,5
22 21,1,Frisbee,3
23 22,1,Marbles,10
24
```

Order log- same for all replicas

src > Part_5-RAFT > order_RAFT > order_data > order_log_1.csv > data

```
1  0,Whale,10
2  1,Frisbee,5
3  2,Frisbee,9
4  3,Monopoly,10
5  4,Lego,1
6  5,Python,1
7  6,Python,6
8  7,Whale,5
9  8,Whale,10
10 9,Fox,5
11 10,Marbles,8
12 11,Monopoly,3
13 12,Frisbee,10
14 13,Lego,5
15 14,Giraffe,5
16 15,Tux,5
17 16,Lego,9
18 17,Tux,7
19 18,Giraffe,7
20 19,Giraffe,3
21 20,Fox,5
22 21,Frisbee,3
23 22,Marbles,10
24
```

```
src > Part_5-RAFT > order_RAFT > order_data > order_log_2.csv > data  
1 0,Whale,10  
2 1,Frisbee,5  
3 2,Frisbee,9  
4 3,Monopoly,10  
5 4,Lego,1  
6 5,Python,1  
7 6,Python,6  
8 7,Whale,5  
9 8,Whale,10  
10 9,Fox,5  
11 10,Marbles,8  
12 11,Monopoly,3  
13 12,Frisbee,10  
14 13,Lego,5  
15 14,Giraffe,5  
16 15,Tux,5  
17 16,Lego,9  
18 17,Tux,7  
19 18,Giraffe,7  
20 19,Giraffe,3  
21 20,Fox,5  
22 21,Frisbee,3  
23 22,Marbles,10  
24
```

src > Part_5-RAFT > order_RAFT > order_data >  order_log_3.csv >  data

```
1  0,Whale,10
2  1,Frisbee,5
3  2,Frisbee,9
4  3,Monopoly,10
5  4,Lego,1
6  5,Python,1
7  6,Python,6
8  7,Whale,5
9  8,Whale,10
10 9,Fox,5
11 10,Marbles,8
12 11,Monopoly,3
13 12,Frisbee,10
14 13,Lego,5
15 14,Giraffe,5
16 15,Tux,5
17 16,Lego,9
18 17,Tux,7
19 18,Giraffe,7
20 19,Giraffe,3
21 20,Fox,5
22 21,Frisbee,3
23 22,Marbles,10
24
```