Outputs and Tests

How to run tests?

Instructions for running the code-

1. Ssh into elnux1 and login and clone repository in home directory. Enter into lab-3-lab-3-gupta-kapadia

   cd ~

   git clone https://github.com/ds-umass/lab-3-lab-3-gupta-kapadia.git

   cd lab-3-lab-3-gupta-kapadia

2. Change permissions of starter script: chmod +x src/starter.sh
3. Execute starter script: ./src/starter.sh
4. Enter password when prompted
5. To view logs in the same src/ folder there are files- order/order_Server1.log, order/order_Server1.log and catalog/catalog_server1.log, catalog/catalog_server2.log
6. To view the output of the servers it is saved in each of the folders inside src/

catalog/catalog_server_output1.txt

catalog/catalog_server_output2.txt

catalog/restart_catalog_output.txt

catalog/resync.output.txt

order/order_server1_output.txt

order/order_server2_output.txt

front_end/front_end_output.txt

This script installs the pre-requisites, starts servers, runs tests, kills catalog server, resyncs it and finally runs the tests again.

Starter.sh:

- The starter script does the following -
- installs required dependencies,
- Kills previous processes on same ports
- ssh's into remote edlab machines and
- executes all 3 servers and their instances in the background
- Executes client.py
- Kills catalog server 2
- Resyncs the catalog server 2 and restarts it making it fault tolerance
- It then executes the client test script

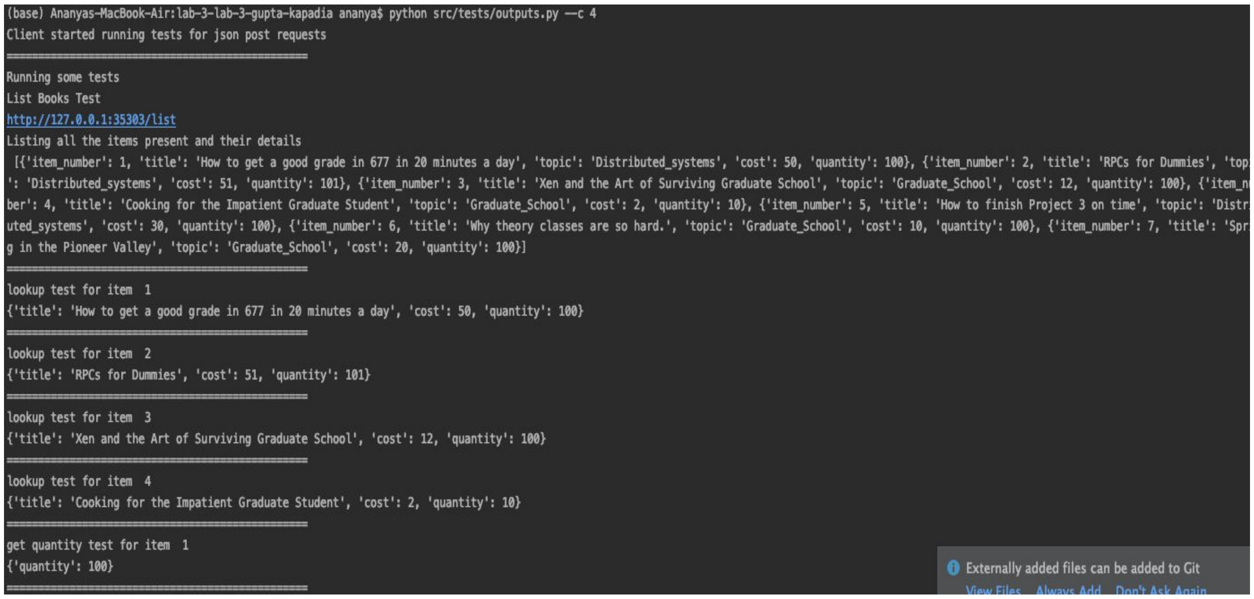
OUTPUTS

Part 1: Replication and caching

Load Balancing

I sent lookup requests to the front end server. The front end server is directing the requests to the alternate servers and is doing load balancing which avoids one server from getting overloaded.
Lookup requests for items 1,2,3,4 being served by catalog 1 and catalog 2 as shown in the output-

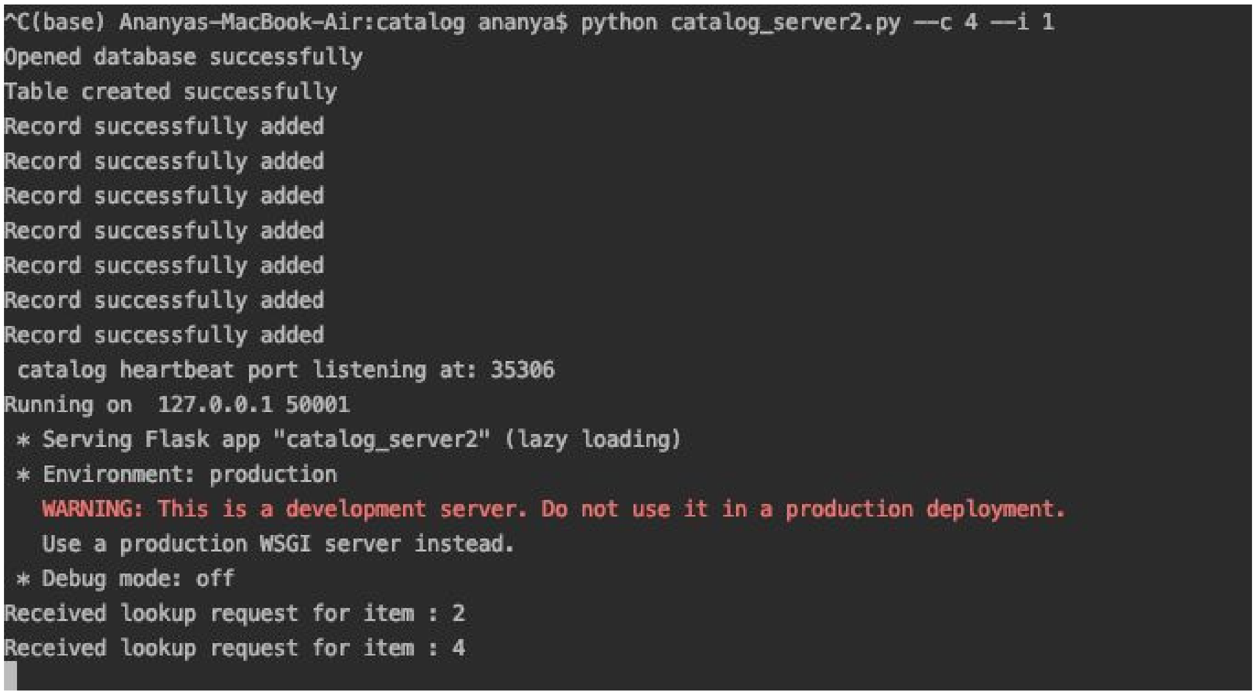
Making requests list, lookup for 1,2,3,4



```
(base) Ananyas-MacBook-Air:lab-3-lab-3-gupta-kapadia ananya$ python src/tests/outputs.py —c 4
Client started running tests for json post requests
==============================================
Running some tests
List Books Test
http://127.0.0.1:35303/list
Listing all the items present and their details
 [{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 minutes a day', 'topic': 'Distributed_systems', 'cost': 50, 'quantity': 100}, {'item_number': 2, 'title': 'RPCs for Dummies', 'top
': 'Distributed_systems', 'cost': 51, 'quantity': 101}, {'item_number': 3, 'title': 'Xen and the Art of Surviving Graduate School', 'topic': 'Graduate_School', 'cost': 12, 'quantity': 100}, {'item_n
ber': 4, 'title': 'Cooking for the Impatient Graduate Student', 'topic': 'Graduate_School', 'cost': 2, 'quantity': 10}, {'item_number': 5, 'title': 'How to finish Project 3 on time', 'topic': 'Distr
uted_systems', 'cost': 30, 'quantity': 100}, {'item_number': 6, 'title': 'Why theory classes are so hard.', 'topic': 'Graduate_School', 'cost': 10, 'quantity': 100}, {'item_number': 7, 'title': 'Spr
g in the Pioneer Valley', 'topic': 'Graduate_School', 'cost': 20, 'quantity': 100}]
==============================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 50, 'quantity': 100}
==============================================
lookup test for item  2
{'title': 'RPCs for Dummies', 'cost': 51, 'quantity': 101}
==============================================
lookup test for item  3
{'title': 'Xen and the Art of Surviving Graduate School', 'cost': 12, 'quantity': 100}
==============================================
lookup test for item  4
{'title': 'Cooking for the Impatient Graduate Student', 'cost': 2, 'quantity': 10}
==============================================
get quantity test for item  1
{'quantity': 100}
==============================================
```

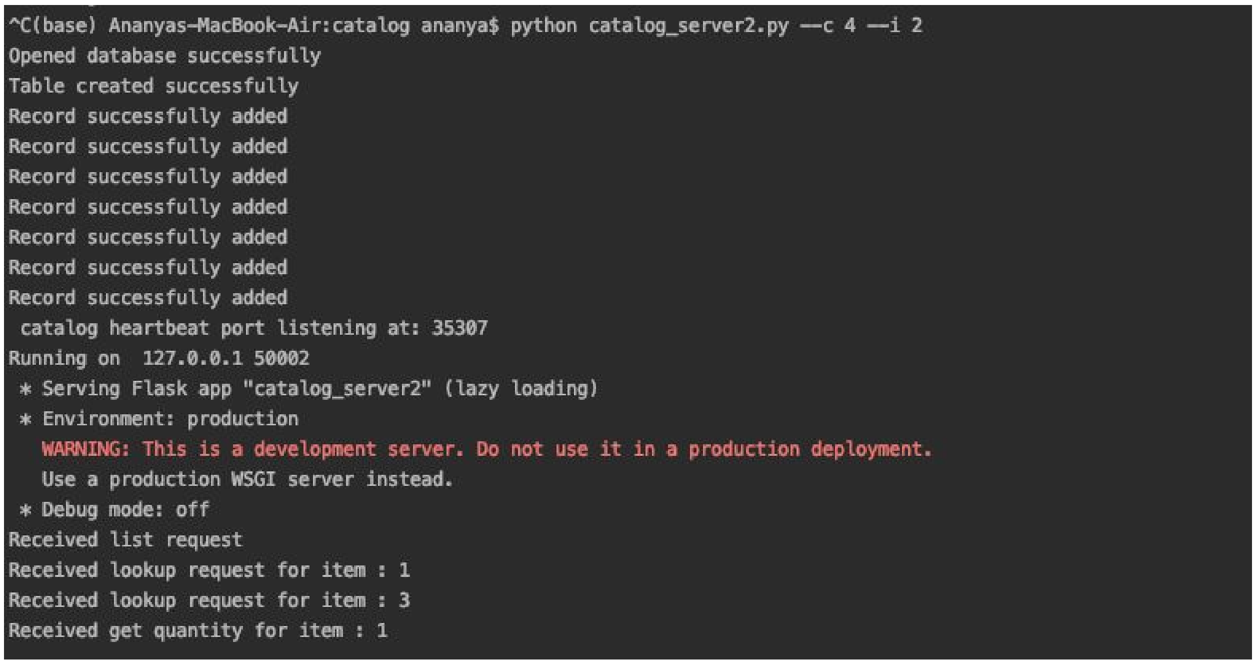
Externally added files can be added to Git
View Files   Always Add   Don't Ask Again

Catalog1 gets lookup for 2 and 4 items



```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py —c 4 —i 1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received lookup request for item : 2
Received lookup request for item : 4
```

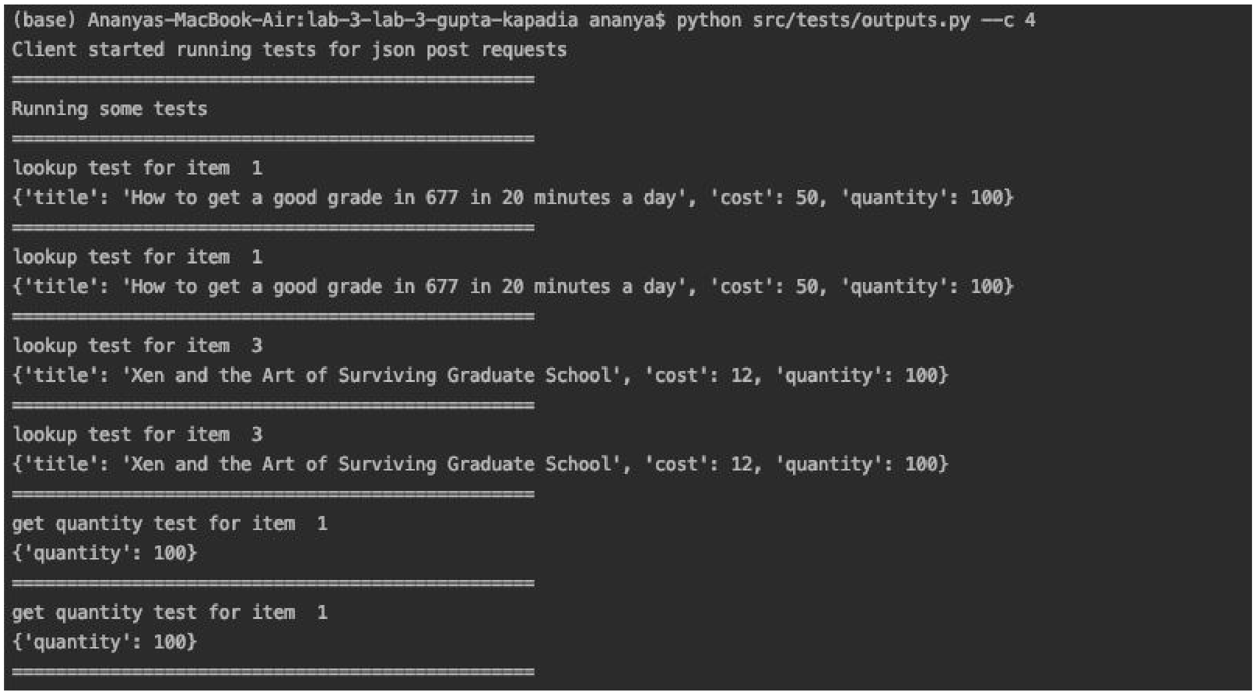
Catalog 2 gets lookup requests for items 1 and 3



```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 2
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received list request
Received lookup request for item : 1
Received lookup request for item : 3
Received get quantity for item : 1
```

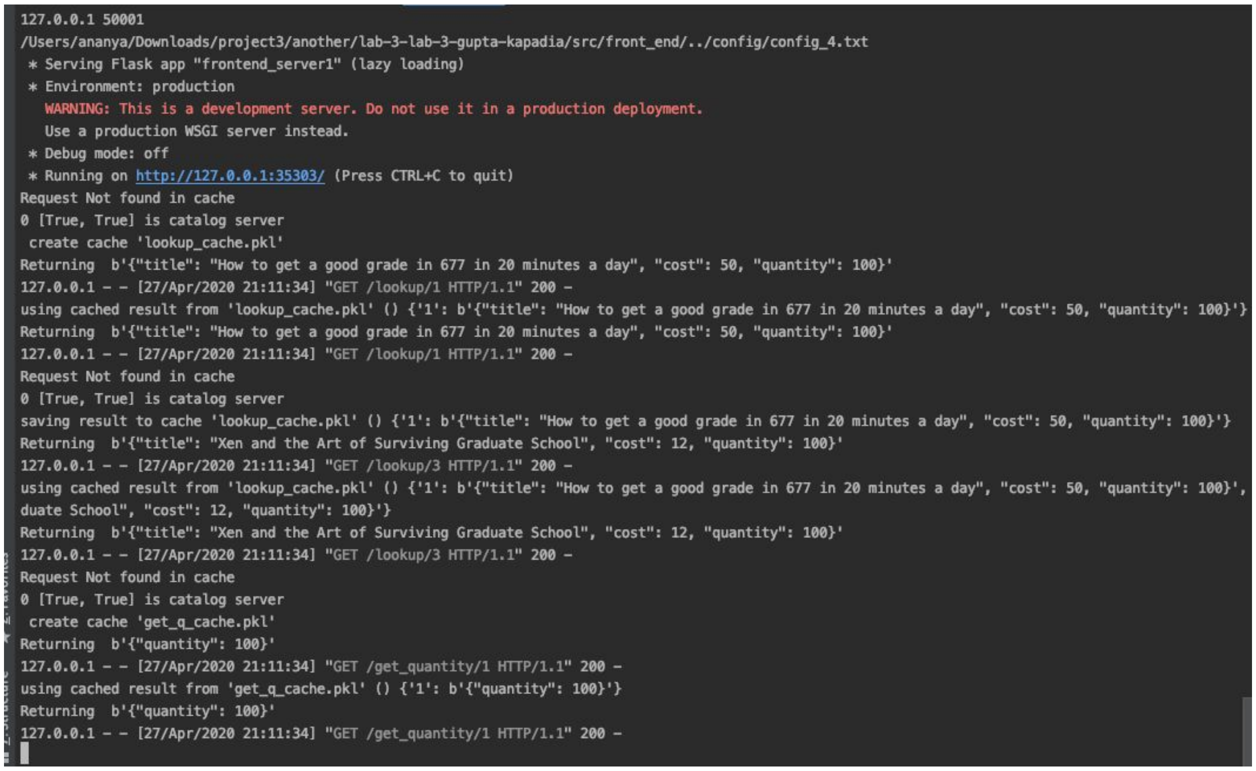
Caching

Repeated Lookup requests made- first goes to the catalog servers and second one is served from cache



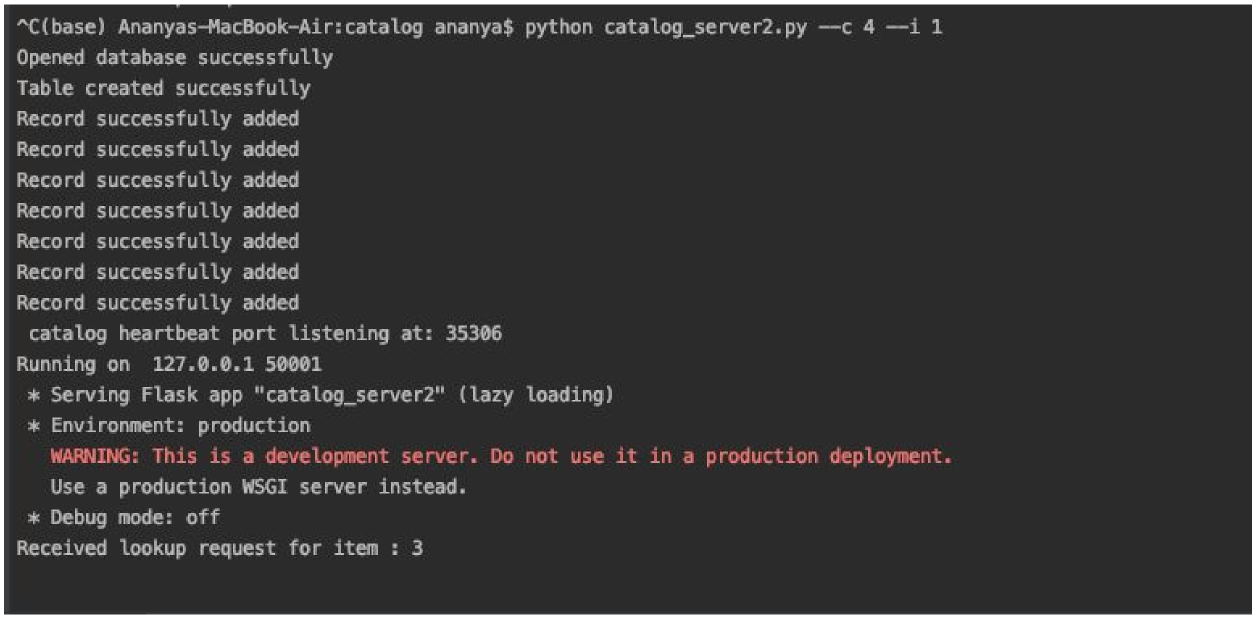
```
(base) Ananyas-MacBook-Air:lab-3-lab-3-gupta-kapadia ananya$ python src/tests/outputs.py --c 4
Client started running tests for json post requests
================================================
Running some tests
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 50, 'quantity': 100}
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 50, 'quantity': 100}
================================================
lookup test for item  3
{'title': 'Xen and the Art of Surviving Graduate School', 'cost': 12, 'quantity': 100}
================================================
lookup test for item  3
{'title': 'Xen and the Art of Surviving Graduate School', 'cost': 12, 'quantity': 100}
================================================
get quantity test for item  1
{'quantity': 100}
================================================
get quantity test for item  1
{'quantity': 100}
================================================
```

Frontend server which sends calls catalog for first request and then caches it and uses cached result when it gets the same request again



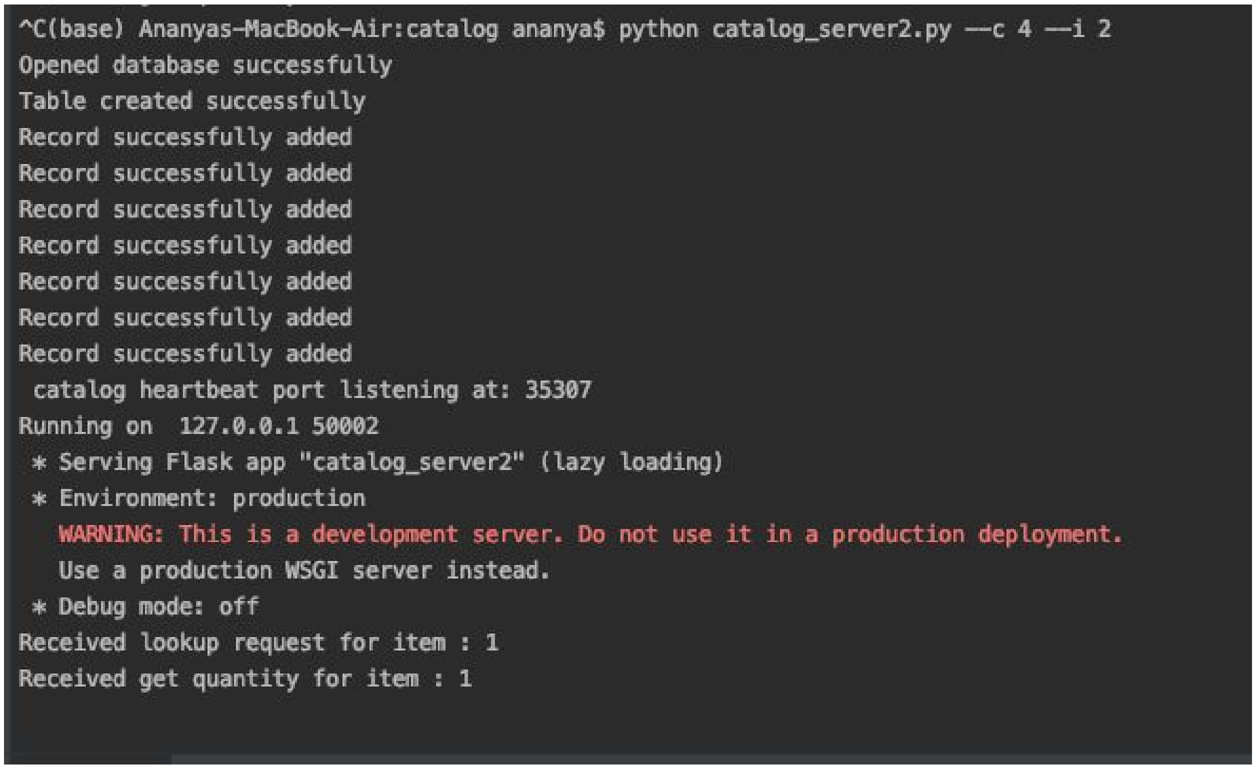
```
127.0.0.1 50001
/Users/ananya/Downloads/project3/another/lab-3-lab-3-gupta-kapadia/src/front_end/../config/config_4.txt
 * Serving Flask app "frontend_server1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:35303/ (Press CTRL+C to quit)
Request Not found in cache
0 [True, True] is catalog server
 create cache 'lookup_cache.pkl'
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /lookup/1 HTTP/1.1" 200 -
using cached result from 'lookup_cache.pkl' () {'1': b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'}
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /lookup/1 HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
saving result to cache 'lookup_cache.pkl' () {'1': b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'}
Returning  b'{"title": "Xen and the Art of Surviving Graduate School", "cost": 12, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /lookup/3 HTTP/1.1" 200 -
using cached result from 'lookup_cache.pkl' () {'1': b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}',
duate School", "cost": 12, "quantity": 100}'}
Returning  b'{"title": "Xen and the Art of Surviving Graduate School", "cost": 12, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /lookup/3 HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
 create cache 'get_q_cache.pkl'
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /get_quantity/1 HTTP/1.1" 200 -
using cached result from 'get_q_cache.pkl' () {'1': b'{"quantity": 100}'}
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 21:11:34] "GET /get_quantity/1 HTTP/1.1" 200 -
```

Catalog server instance 1 only gets request once for item 3



```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received lookup request for item : 3
```

Catalog server instance 2 gets request once for lookup and once for get quantity

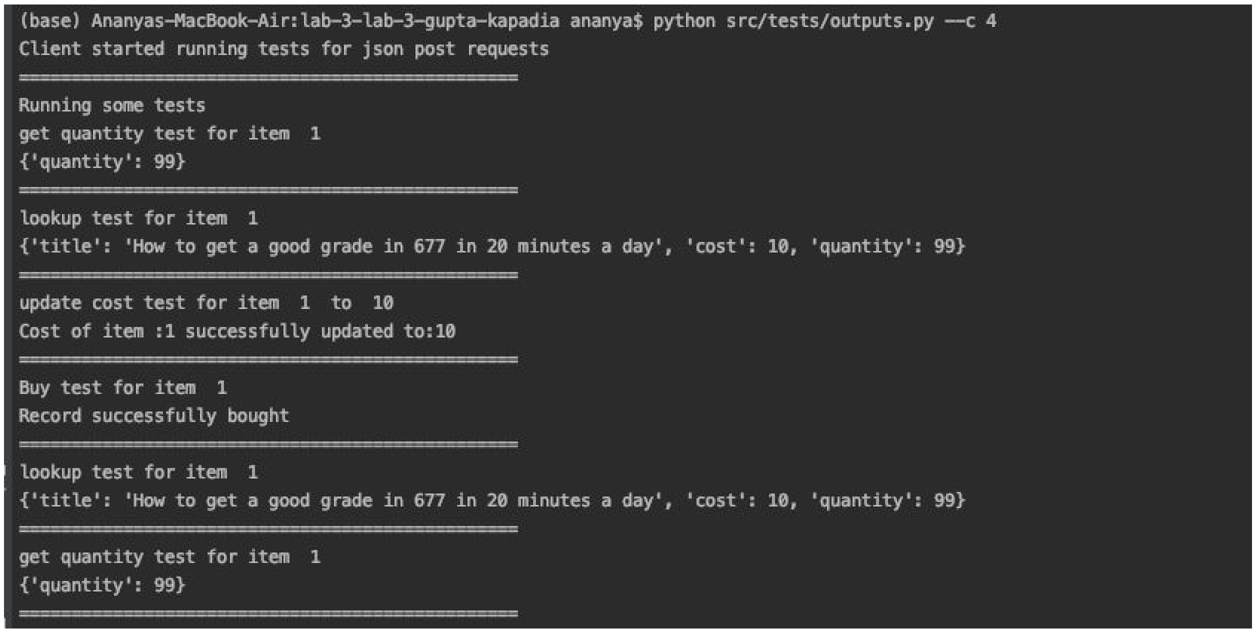


```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 2
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received lookup request for item : 1
Received get quantity for item : 1
```

Cache consistency

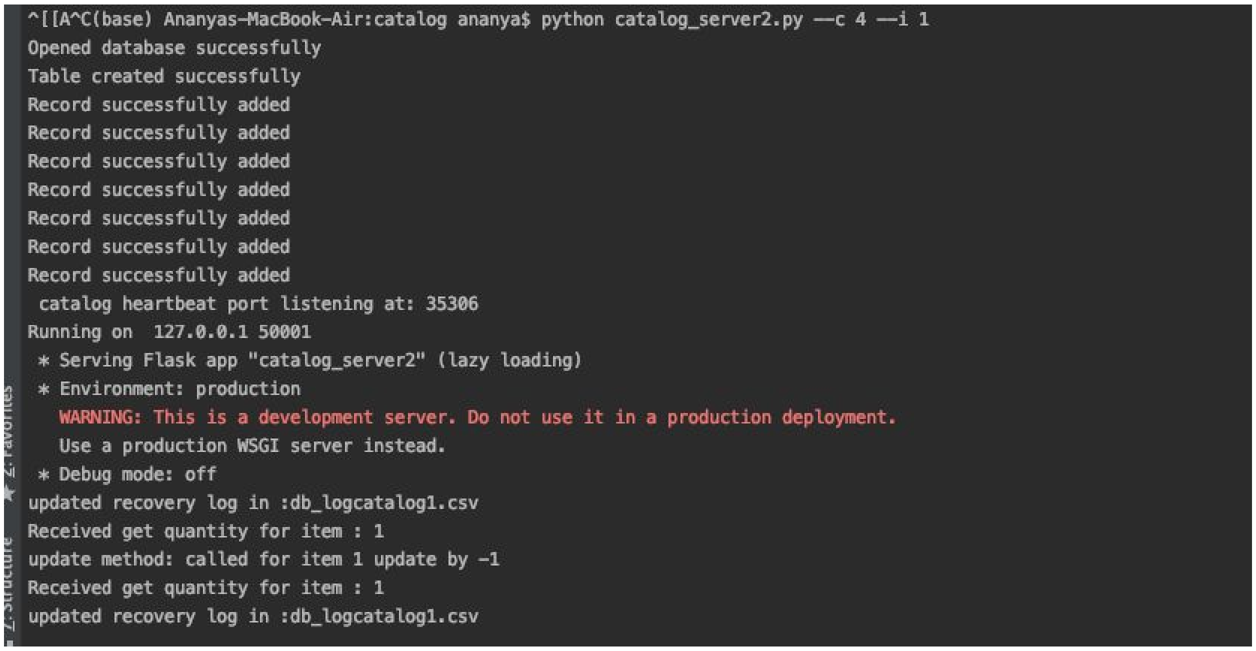
Making lookup requests with buy and update cost, invalidate cache is forced by the server push update requests

The next lookup is again sent to the catalog server since the cache gets invalidated by the update and buy requests



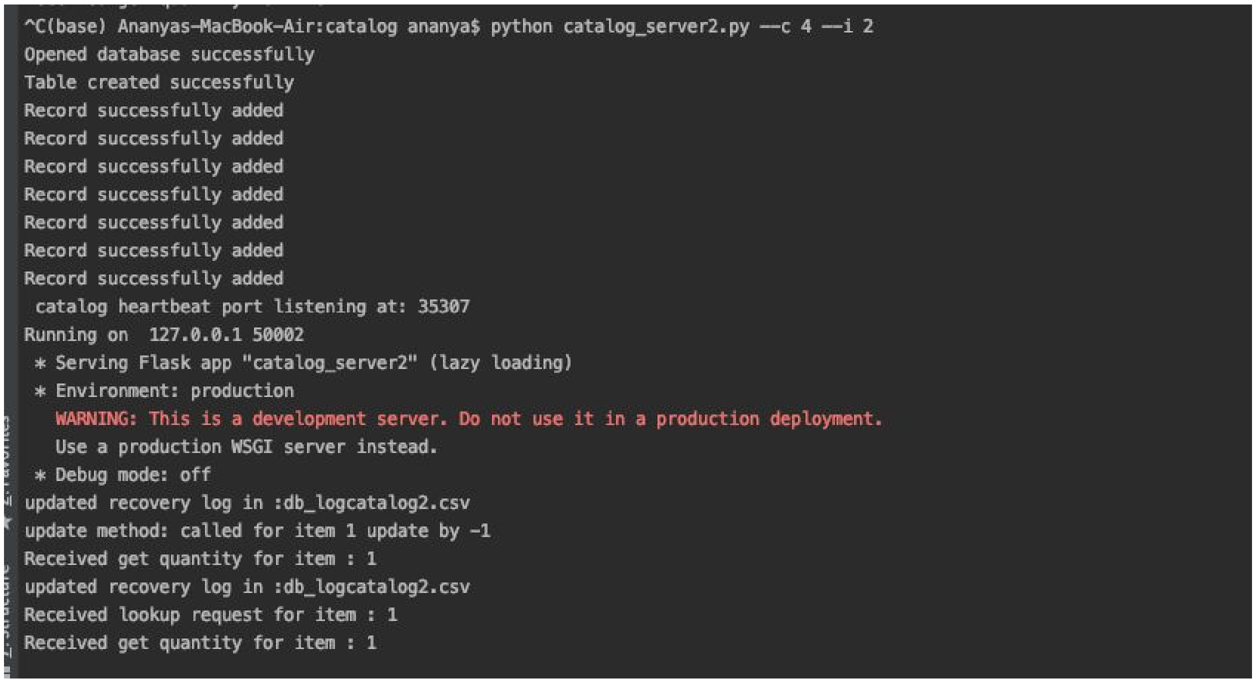
```
(base) Ananyas-MacBook-Air:lab-3-lab-3-gupta-kapadia ananya$ python src/tests/outputs.py --c 4
Client started running tests for json post requests
================================================
Running some tests
get quantity test for item  1
{'quantity': 99}
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 10, 'quantity': 99}
================================================
update cost test for item  1  to  10
Cost of item :1 successfully updated to:10
================================================
Buy test for item  1
Record successfully bought
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 10, 'quantity': 99}
================================================
get quantity test for item  1
{'quantity': 99}
================================================
```

Catalog server 1 gets updates and get requests



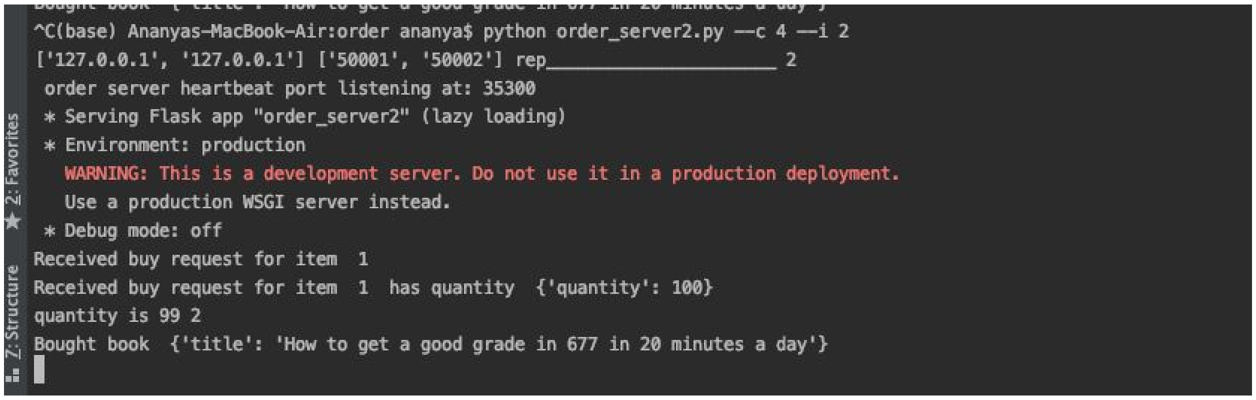
```
^[[A^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
updated recovery log in :db_logcatalog1.csv
Received get quantity for item : 1
update method: called for item 1 update by -1
Received get quantity for item : 1
updated recovery log in :db_logcatalog1.csv
```

Catalog server 2 gets update and get requests



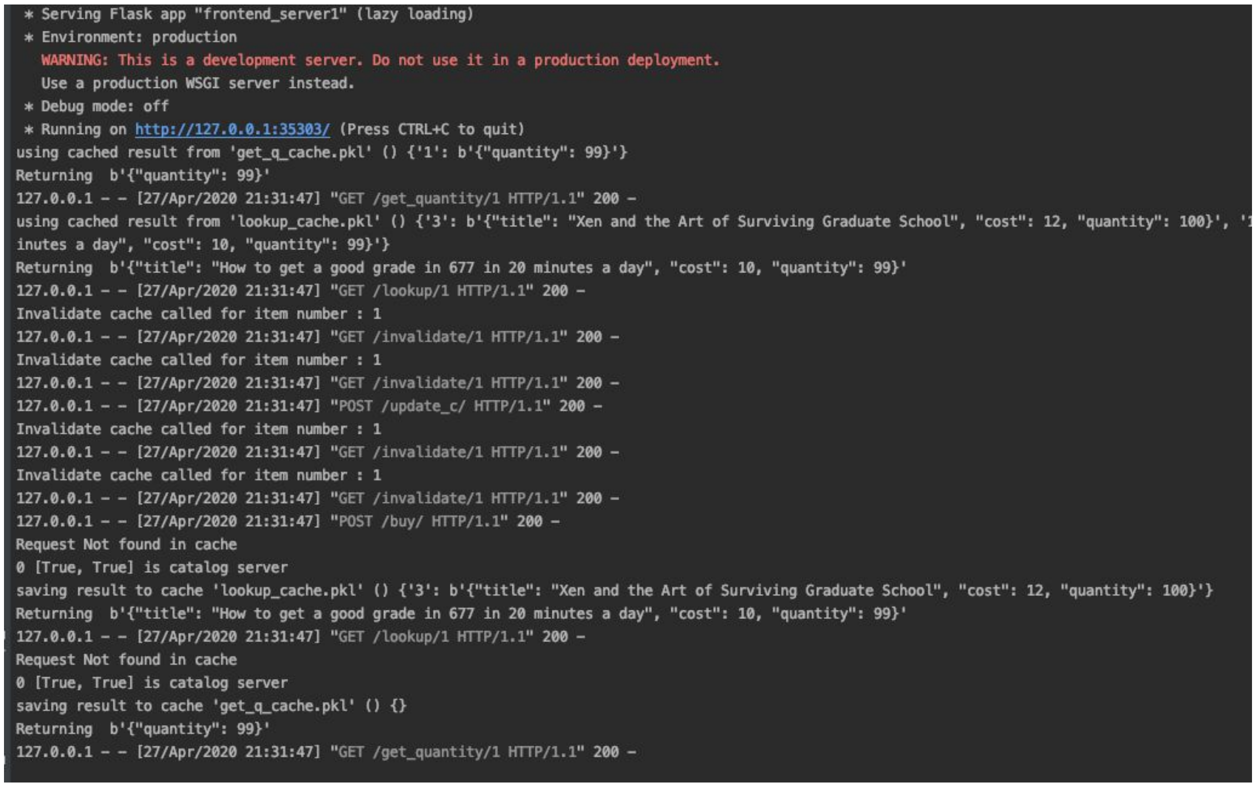
```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 2
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
updated recovery log in :db_logcatalog2.csv
update method: called for item 1 update by -1
Received get quantity for item : 1
updated recovery log in :db_logcatalog2.csv
Received lookup request for item : 1
Received get quantity for item : 1
```

Order server 2 receives buy request



```
^C(base) Ananyas-MacBook-Air:order ananya$ python order_server2.py --c 4 --i 2
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 2
 order server heartbeat port listening at: 35300
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received buy request for item  1
Received buy request for item  1  has quantity  {'quantity': 100}
quantity is 99 2
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
```

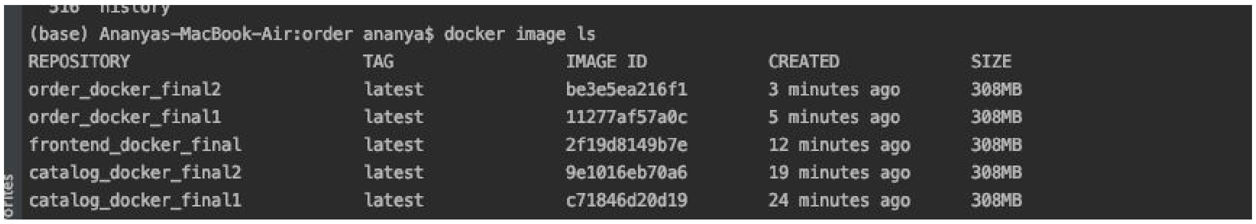
Front end receives invalidate cache requests after update queries from the catalog server



```
 * Serving Flask app "frontend_server1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:35303/ (Press CTRL+C to quit)
using cached result from 'get_q_cache.pkl' () {'1': b'{"quantity": 99}'}
Returning  b'{"quantity": 99}'
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /get_quantity/1 HTTP/1.1" 200 -
using cached result from 'lookup_cache.pkl' () {'3': b'{"title": "Xen and the Art of Surviving Graduate School", "cost": 12, "quantity": 100}', '1
inutes a day", "cost": 10, "quantity": 99}'}
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 10, "quantity": 99}'
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /lookup/1 HTTP/1.1" 200 -
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /invalidate/1 HTTP/1.1" 200 -
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /invalidate/1 HTTP/1.1" 200 -
127.0.0.1 - - [27/Apr/2020 21:31:47] "POST /update_c/ HTTP/1.1" 200 -
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /invalidate/1 HTTP/1.1" 200 -
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /invalidate/1 HTTP/1.1" 200 -
127.0.0.1 - - [27/Apr/2020 21:31:47] "POST /buy/ HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
saving result to cache 'lookup_cache.pkl' () {'3': b'{"title": "Xen and the Art of Surviving Graduate School", "cost": 12, "quantity": 100}'}
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 10, "quantity": 99}'
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /lookup/1 HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
saving result to cache 'get_q_cache.pkl' () {}
Returning  b'{"quantity": 99}'
127.0.0.1 - - [27/Apr/2020 21:31:47] "GET /get_quantity/1 HTTP/1.1" 200 -
```

Part 2: Dockerize your application

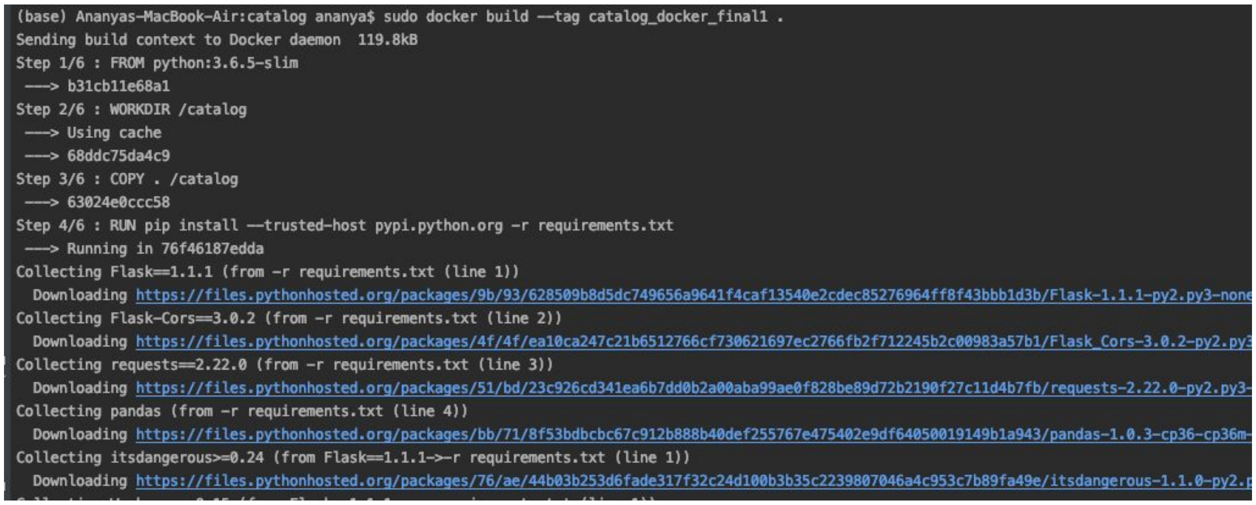
The docker images of the server instances have been uploaded on
https://drive.google.com/drive/folders/1tdUwYpCmRxQqVV5EjF55L0cywIebub0f?usp=sharing



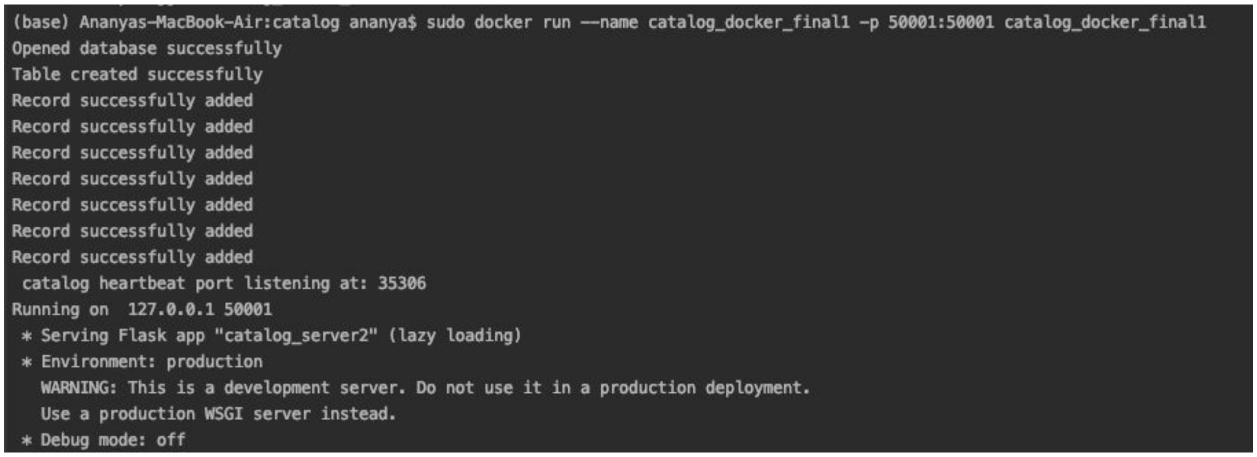
```
  516  history
(base) Ananyas-MacBook-Air:order ananya$ docker image ls
REPOSITORY                 TAG                 IMAGE ID            CREATED             SIZE
order_docker_final2        latest              be3e5ea216f1        3 minutes ago       308MB
order_docker_final1        latest              11277af57a0c        5 minutes ago       308MB
frontend_docker_final      latest              2f19d8149b7e        12 minutes ago      308MB
catalog_docker_final2      latest              9e1016eb70a6        19 minutes ago      308MB
catalog_docker_final1      latest              c71846d20d19        24 minutes ago      308MB
```

The dockerfile and the requirements.txt files are in each folder of the apps order, catalog and the frontend folders.
Build docker image-
  sudo docker build --tag catalog_docker_final1 .
Run the docker image-
   sudo docker run --name catalog_docker_final1 -p 50001:50001 catalog_docker_final1

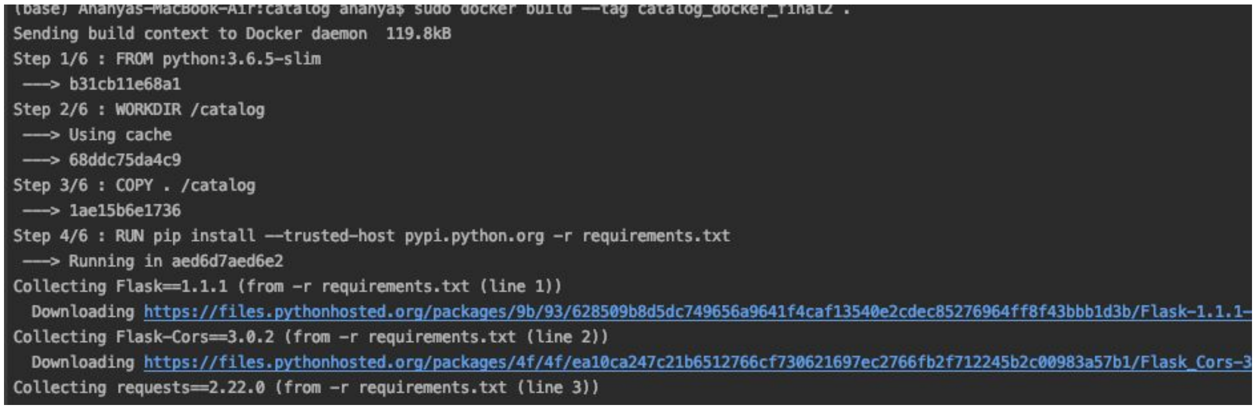
Build docker image-
  sudo docker build --tag catalog_docker_final2 .
Run the docker image-
   sudo docker run --name catalog_docker_final2 -p 50002:50002 catalog_docker_final2
The following screenshots show the same commands executed-



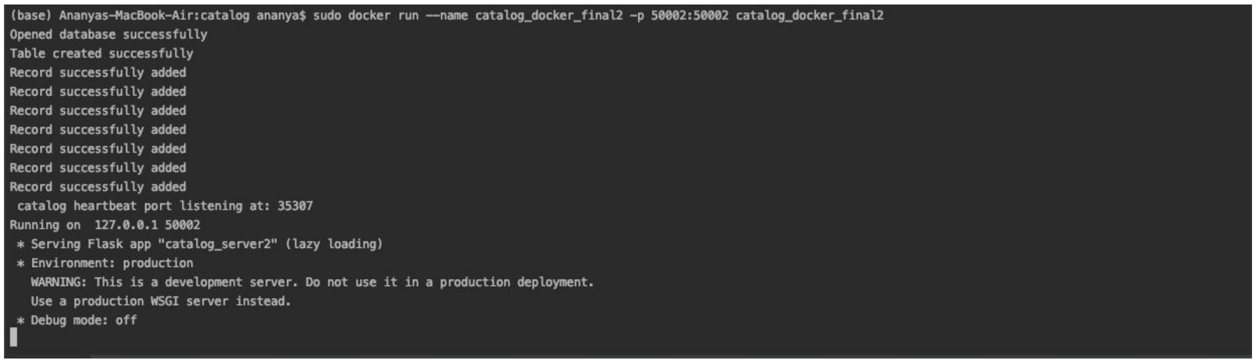
```
(base) Ananyas-MacBook-Air:catalog ananya$ sudo docker build --tag catalog_docker_final1 .
Sending build context to Docker daemon  119.8kB
Step 1/6 : FROM python:3.6.5-slim
 ---> b31cb11e68a1
Step 2/6 : WORKDIR /catalog
 ---> Using cache
 ---> 68ddc75da4c9
Step 3/6 : COPY . /catalog
 ---> 63024e0ccc58
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in 76f46187edda
Collecting Flask==1.1.1 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none
Collecting Flask-Cors==3.0.2 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/4f/4f/ea10ca247c21b6512766cf730621697ec2766fb2f712245b2c00983a57b1/Flask_Cors-3.0.2-py2.py3
Collecting requests==2.22.0 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0b2a00aba99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-
Collecting pandas (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/bb/71/8f53bdbcbc67c912b888b40def255767e475402e9df64050019149b1a943/pandas-1.0.3-cp36-cp36m-
Collecting itsdangerous>=0.24 (from Flask==1.1.1->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.p
```



```
(base) Ananyas-MacBook-Air:catalog ananya$ sudo docker run --name catalog_docker_final1 -p 50001:50001 catalog_docker_final1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```



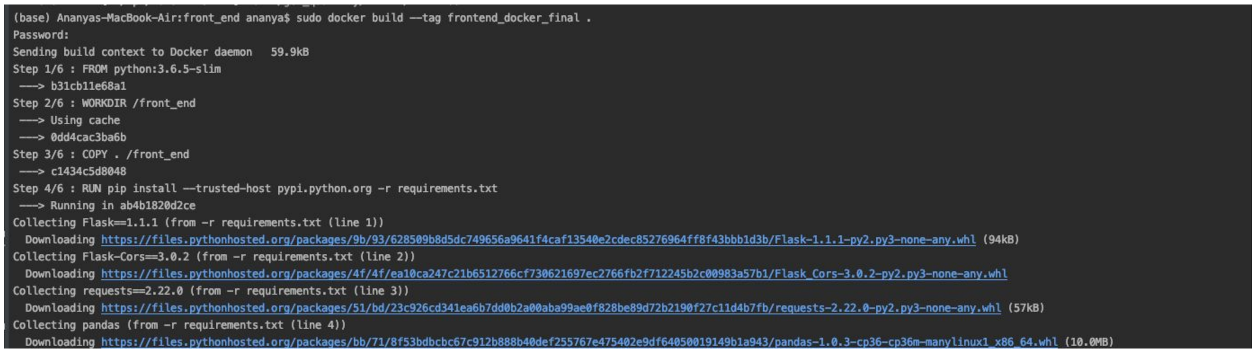
```
(base) Ananyas-MacBook-Air:catalog ananya$ sudo docker build --tag catalog_docker_final2 .
Sending build context to Docker daemon  119.8kB
Step 1/6 : FROM python:3.6.5-slim
 ---> b31cb11e68a1
Step 2/6 : WORKDIR /catalog
 ---> Using cache
 ---> 68ddc75da4c9
Step 3/6 : COPY . /catalog
 ---> 1ae15b6e1736
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in aed6d7aed6e2
Collecting Flask==1.1.1 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-
Collecting Flask-Cors==3.0.2 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/4f/4f/ea10ca247c21b6512766cf730621697ec2766fb2f712245b2c00983a57b1/Flask_Cors-3
Collecting requests==2.22.0 (from -r requirements.txt (line 3))
```



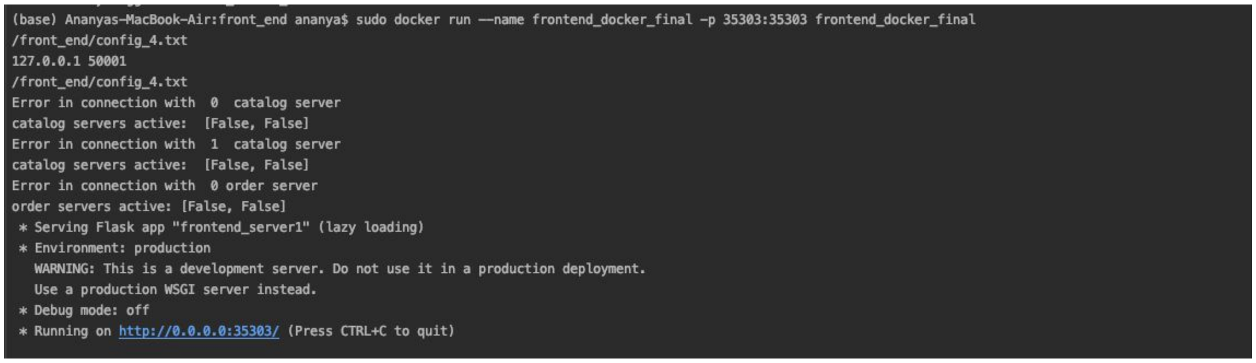
```
(base) Ananyas-MacBook-Air:catalog ananya$ sudo docker run --name catalog_docker_final2 -p 50002:50002 catalog_docker_final2
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

sudo docker build --tag frontend_docker_final .

sudo docker run --name frontend_docker_final -p 35303:35303 frontend_docker_final



```
(base) Ananyas-MacBook-Air:front_end ananya$ sudo docker build --tag frontend_docker_final .
Password:
Sending build context to Docker daemon   59.9kB
Step 1/6 : FROM python:3.6.5-slim
 ---> b31cb11e68a1
Step 2/6 : WORKDIR /front_end
 ---> Using cache
 ---> 0dd4cac3ba6b
Step 3/6 : COPY . /front_end
 ---> c1434c5d8048
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in ab4b1820d2ce
Collecting Flask==1.1.1 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
Collecting Flask-Cors==3.0.2 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/4f/4f/ea10ca247c21b6512766cf730621697ec2766fb2f712245b2c00983a57b1/Flask_Cors-3.0.2-py2.py3-none-any.whl
Collecting requests==2.22.0 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0b2a00aba99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-none-any.whl (57kB)
Collecting pandas (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/bb/71/8f53bdbcbc67c912b888b40def255767e475402e9df64050019149b1a943/pandas-1.0.3-cp36-cp36m-manylinux1_x86_64.whl (10.0MB)
```



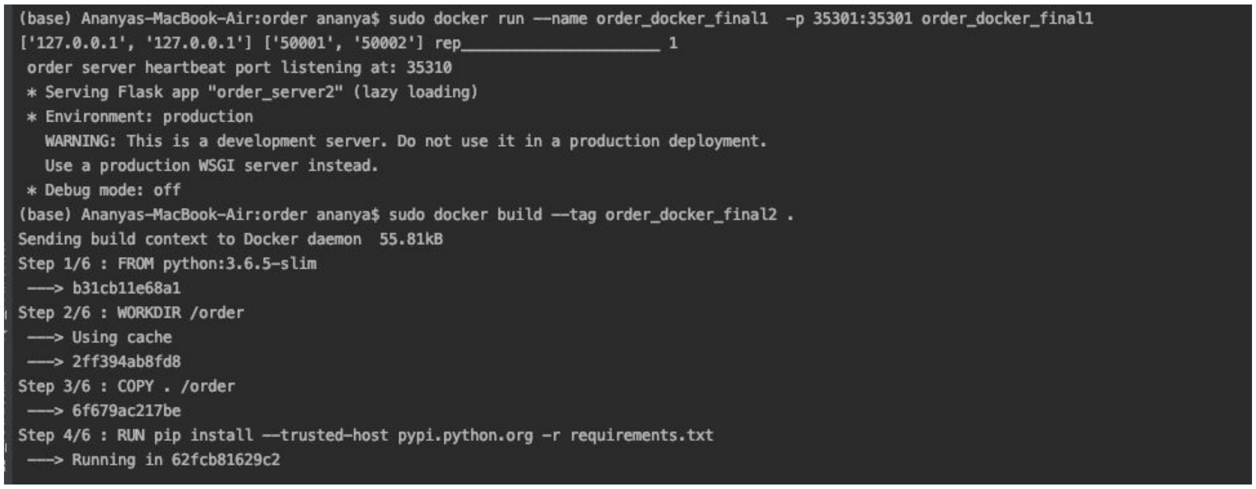
```
(base) Ananyas-MacBook-Air:front_end ananya$ sudo docker run --name frontend_docker_final -p 35303:35303 frontend_docker_final
/front_end/config_4.txt
127.0.0.1 50001
/front_end/config_4.txt
Error in connection with  0  catalog server
catalog servers active:  [False, False]
Error in connection with  1  catalog server
catalog servers active:  [False, False]
Error in connection with  0 order server
order servers active: [False, False]
 * Serving Flask app "frontend_server1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:35303/ (Press CTRL+C to quit)
```

sudo docker build --tag order_docker_final1 .

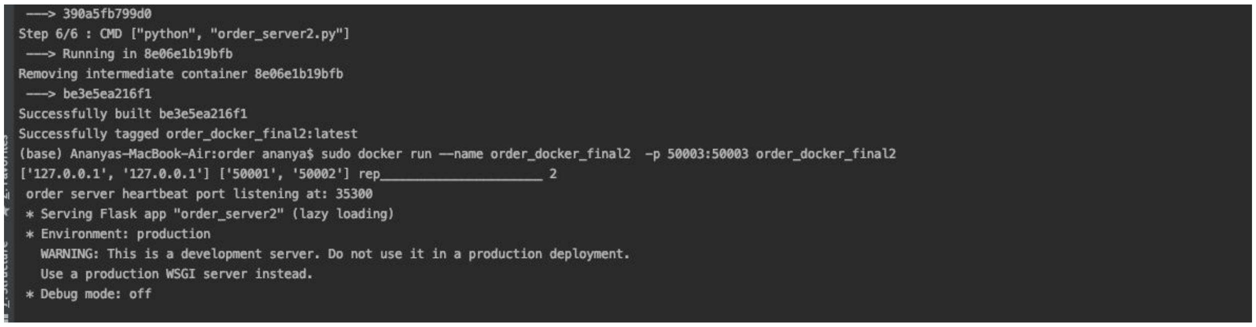
sudo docker run --name order_docker_final1 -p 35301:35301 order_docker_final1

sudo docker build --tag order_docker_final2 .

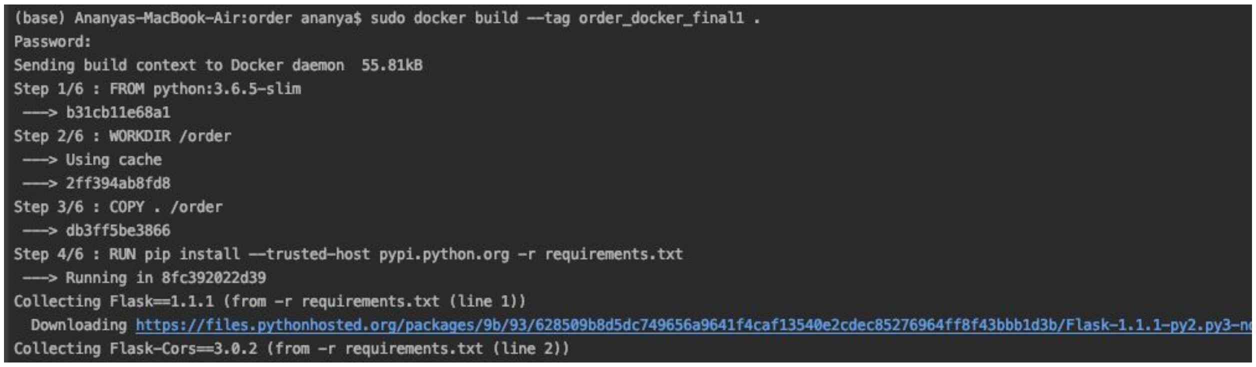
sudo docker run --name order_docker_final2 -p 50003:50003 order_docker_final2



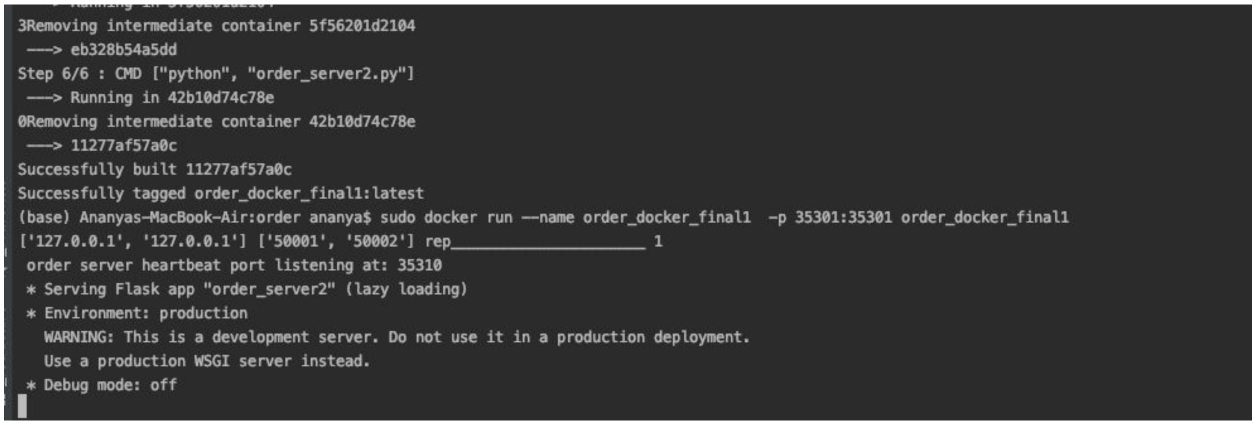
```
(base) Ananyas-MacBook-Air:order ananya$ sudo docker run --name order_docker_final1  -p 35301:35301 order_docker_final1
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 1
 order server heartbeat port listening at: 35310
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
(base) Ananyas-MacBook-Air:order ananya$ sudo docker build --tag order_docker_final2 .
Sending build context to Docker daemon  55.81kB
Step 1/6 : FROM python:3.6.5-slim
 ---> b31cb11e68a1
Step 2/6 : WORKDIR /order
 ---> Using cache
 ---> 2ff394ab8fd8
Step 3/6 : COPY . /order
 ---> 6f679ac217be
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in 62fcb81629c2
```



```
 ---> 390a5fb799d0
Step 6/6 : CMD ["python", "order_server2.py"]
 ---> Running in 8e06e1b19bfb
Removing intermediate container 8e06e1b19bfb
 ---> be3e5ea216f1
Successfully built be3e5ea216f1
Successfully tagged order_docker_final2:latest
(base) Ananyas-MacBook-Air:order ananya$ sudo docker run --name order_docker_final2  -p 50003:50003 order_docker_final2
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 2
 order server heartbeat port listening at: 35300
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```



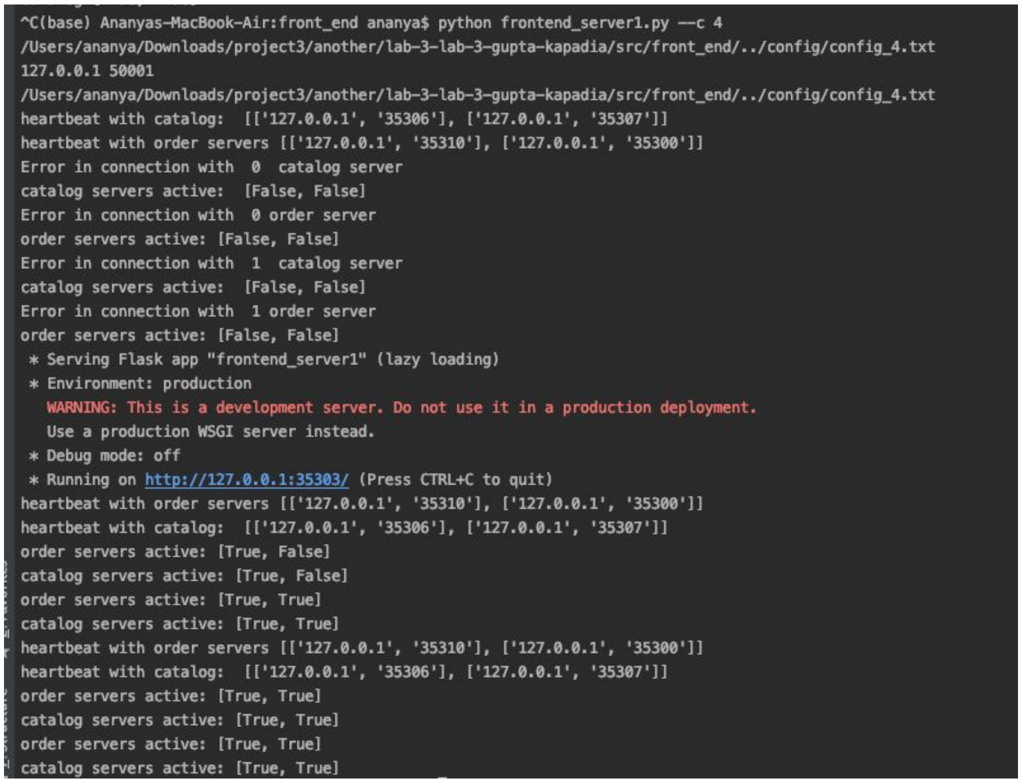
```
(base) Ananyas-MacBook-Air:order ananya$ sudo docker build --tag order_docker_final1 .
Password:
Sending build context to Docker daemon  55.81kB
Step 1/6 : FROM python:3.6.5-slim
 ---> b31cb11e68a1
Step 2/6 : WORKDIR /order
 ---> Using cache
 ---> 2ff394ab8fd8
Step 3/6 : COPY . /order
 ---> db3ff5be3866
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in 8fc392022d39
Collecting Flask==1.1.1 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-n
Collecting Flask-Cors==3.0.2 (from -r requirements.txt (line 2))
```



```
3Removing intermediate container 5f56201d2104
 ---> eb328b54a5dd
Step 6/6 : CMD ["python", "order_server2.py"]
 ---> Running in 42b10d74c78e
0Removing intermediate container 42b10d74c78e
 ---> 11277af57a0c
Successfully built 11277af57a0c
Successfully tagged order_docker_final1:latest
(base) Ananyas-MacBook-Air:order ananya$ sudo docker run --name order_docker_final1  -p 35301:35301 order_docker_final1
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 1
 order server heartbeat port listening at: 35310
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

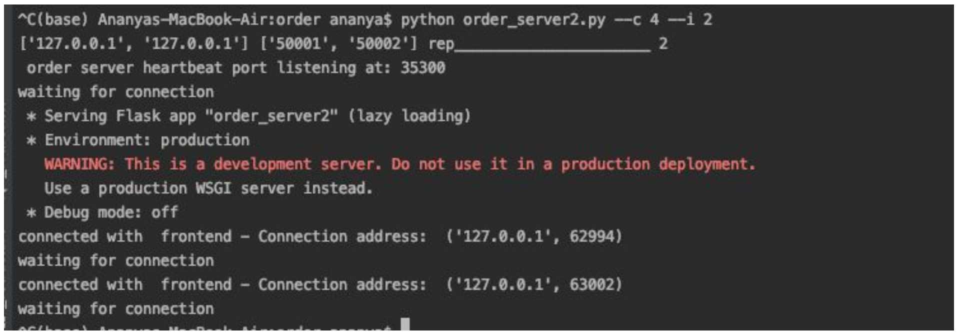
Part 3: Fault tolerance

Heartbeat connection outputs- I first started catalog1, order1 and frontend followed by catalog2, order2.
Frontend Makes heartbeat connections with the 2 catalog servers and 2 order servers
Initially when the catalog 2 and order 2 is not active, shows false status for them but when all the instances are active then all the servers status becomes active.



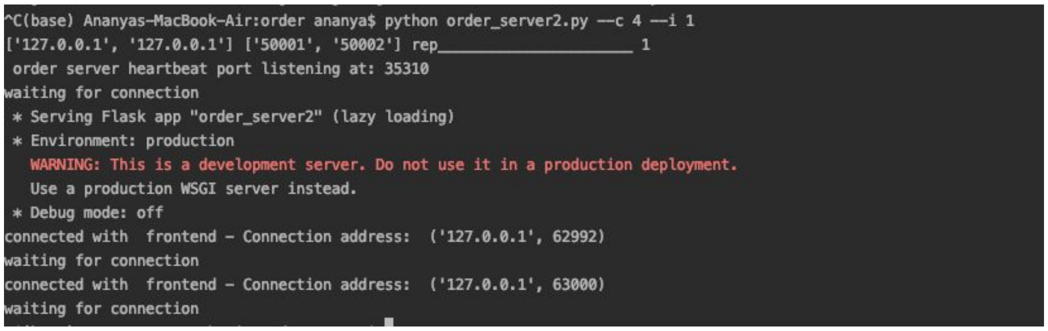
```
^C(base) Ananyas-MacBook-Air:front_end ananya$ python frontend_server1.py --c 4
/Users/ananya/Downloads/project3/another/lab-3-lab-3-gupta-kapadia/src/front_end/../config/config_4.txt
127.0.0.1 50001
/Users/ananya/Downloads/project3/another/lab-3-lab-3-gupta-kapadia/src/front_end/../config/config_4.txt
heartbeat with catalog:  [['127.0.0.1', '35306'], ['127.0.0.1', '35307']]
heartbeat with order servers [['127.0.0.1', '35310'], ['127.0.0.1', '35300']]
Error in connection with  0  catalog server
catalog servers active:  [False, False]
Error in connection with  0 order server
order servers active: [False, False]
Error in connection with  1  catalog server
catalog servers active:  [False, False]
Error in connection with  1 order server
order servers active: [False, False]
 * Serving Flask app "frontend_server1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:35303/ (Press CTRL+C to quit)
heartbeat with order servers [['127.0.0.1', '35310'], ['127.0.0.1', '35300']]
heartbeat with catalog:  [['127.0.0.1', '35306'], ['127.0.0.1', '35307']]
order servers active: [True, False]
catalog servers active: [True, False]
order servers active: [True, True]
catalog servers active: [True, True]
heartbeat with order servers [['127.0.0.1', '35310'], ['127.0.0.1', '35300']]
heartbeat with catalog:  [['127.0.0.1', '35306'], ['127.0.0.1', '35307']]
order servers active: [True, True]
catalog servers active: [True, True]
order servers active: [True, True]
catalog servers active: [True, True]
```

Second order instance on and maintains heartbeat with frontend



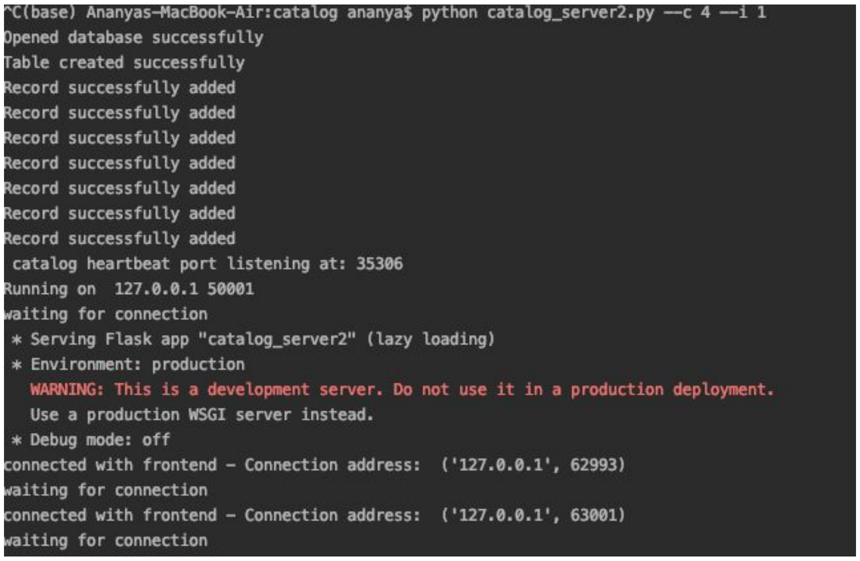
```
^C(base) Ananyas-MacBook-Air:order ananya$ python order_server2.py --c 4 --i 2
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 2
 order server heartbeat port listening at: 35300
waiting for connection
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
connected with  frontend - Connection address:  ('127.0.0.1', 62994)
waiting for connection
connected with  frontend - Connection address:  ('127.0.0.1', 63002)
waiting for connection
```

First order server maintaining heartbeat with frontend



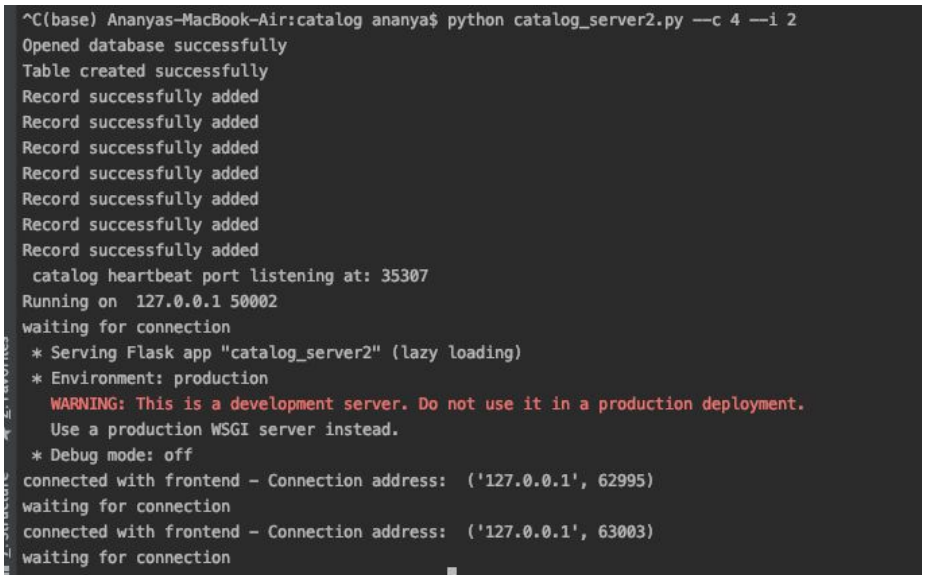
```
^C(base) Ananyas-MacBook-Air:order ananya$ python order_server2.py --c 4 --i 1
['127.0.0.1', '127.0.0.1'] ['50001', '50002'] rep______________________ 1
 order server heartbeat port listening at: 35310
waiting for connection
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
connected with  frontend - Connection address:  ('127.0.0.1', 62992)
waiting for connection
connected with  frontend - Connection address:  ('127.0.0.1', 63000)
waiting for connection
```

First catalog instance maintaining heartbeat with frontend



```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
waiting for connection
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
connected with frontend - Connection address:  ('127.0.0.1', 62993)
waiting for connection
connected with frontend - Connection address:  ('127.0.0.1', 63001)
waiting for connection
```

Second catalog instance maintaining heartbeat with frontend



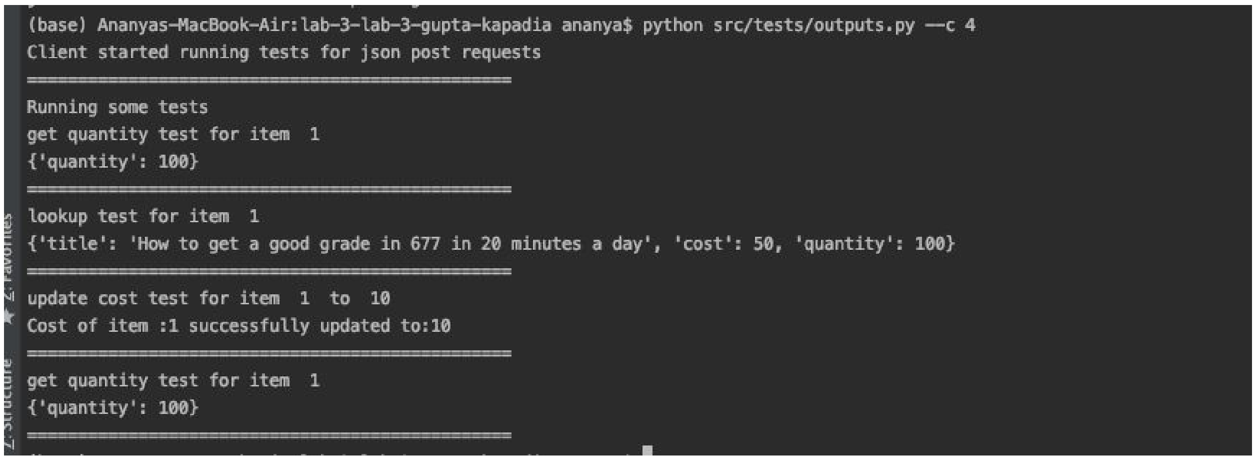
```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 2
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
waiting for connection
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
connected with frontend - Connection address:  ('127.0.0.1', 62995)
waiting for connection
connected with frontend - Connection address:  ('127.0.0.1', 63003)
waiting for connection
```

Failure of catalog server and load balancing

ONLY catalog server 1 is active

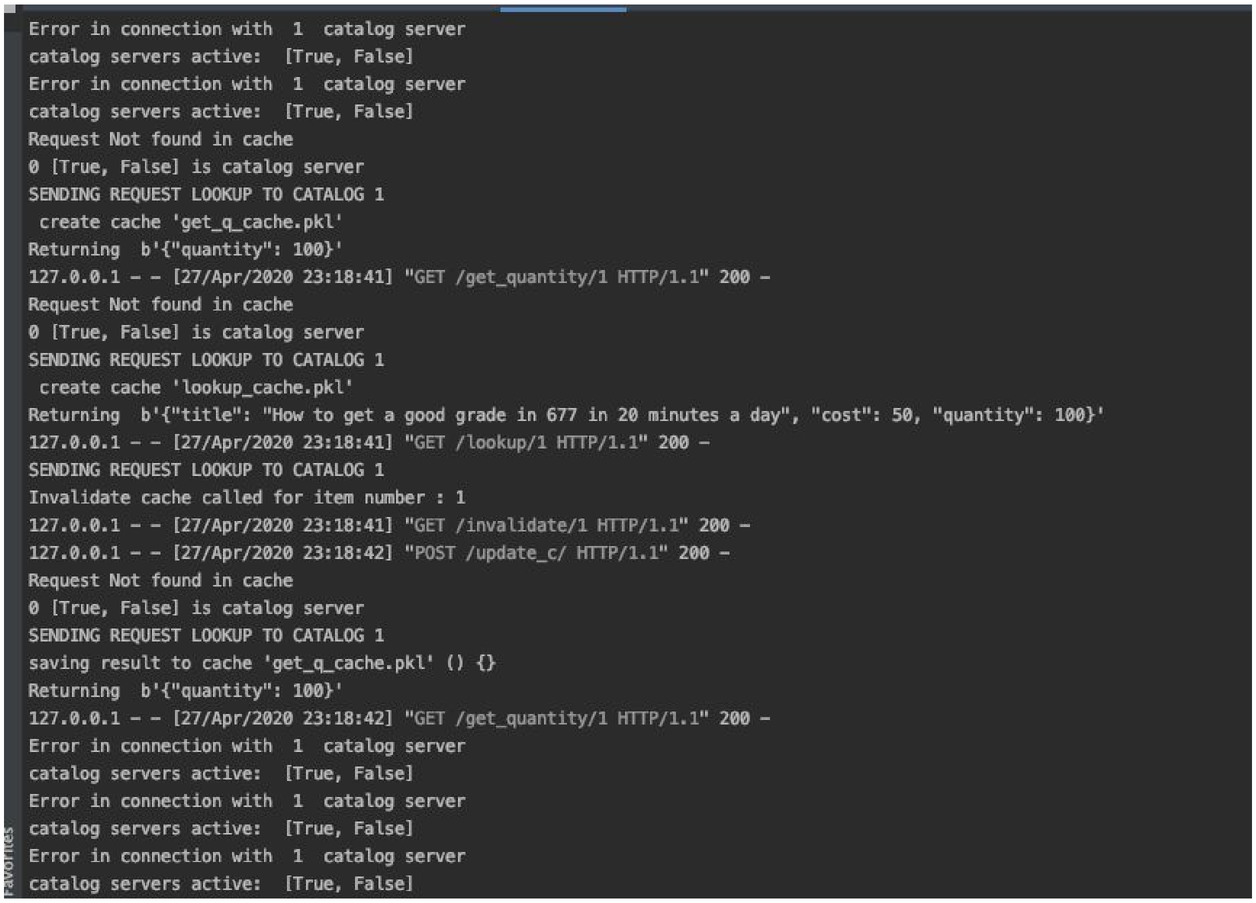
Catalog 2 is killed or is down-

In such a case all the requests are directed to the first catalog server by the front end



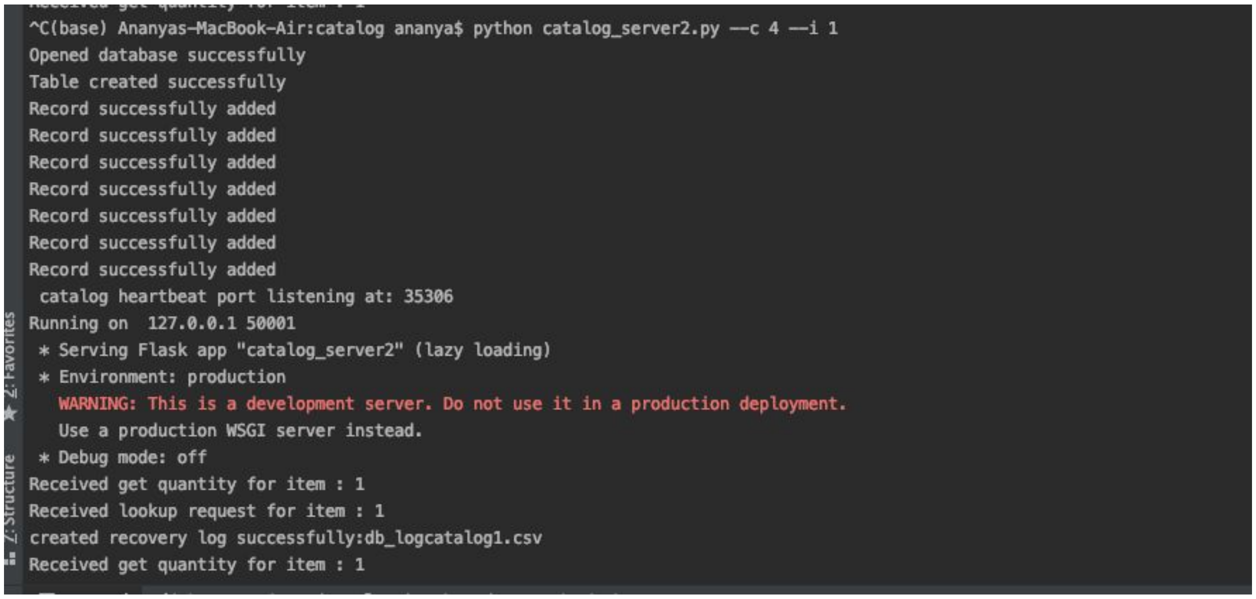
```
(base) Ananyas-MacBook-Air:lab-3-lab-3-gupta-kapadia ananya$ python src/tests/outputs.py --c 4
Client started running tests for json post requests
==================================================
Running some tests
get quantity test for item  1
{'quantity': 100}
==================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 50, 'quantity': 100}
==================================================
update cost test for item  1  to  10
Cost of item :1 successfully updated to:10
==================================================
get quantity test for item  1
{'quantity': 100}
==================================================
```

Front end server knows from heartbeat status that the catalog server 2 is down so forwards all the requests to the catalog server instance number 1



```
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Request Not found in cache
0 [True, False] is catalog server
SENDING REQUEST LOOKUP TO CATALOG 1
 create cache 'get_q_cache.pkl'
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:18:41] "GET /get_quantity/1 HTTP/1.1" 200 -
Request Not found in cache
0 [True, False] is catalog server
SENDING REQUEST LOOKUP TO CATALOG 1
 create cache 'lookup_cache.pkl'
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:18:41] "GET /lookup/1 HTTP/1.1" 200 -
SENDING REQUEST LOOKUP TO CATALOG 1
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 23:18:41] "GET /invalidate/1 HTTP/1.1" 200 -
127.0.0.1 - - [27/Apr/2020 23:18:42] "POST /update_c/ HTTP/1.1" 200 -
Request Not found in cache
0 [True, False] is catalog server
SENDING REQUEST LOOKUP TO CATALOG 1
saving result to cache 'get_q_cache.pkl' () {}
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:18:42] "GET /get_quantity/1 HTTP/1.1" 200 -
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Error in connection with  1  catalog server
catalog servers active:  [True, False]
```

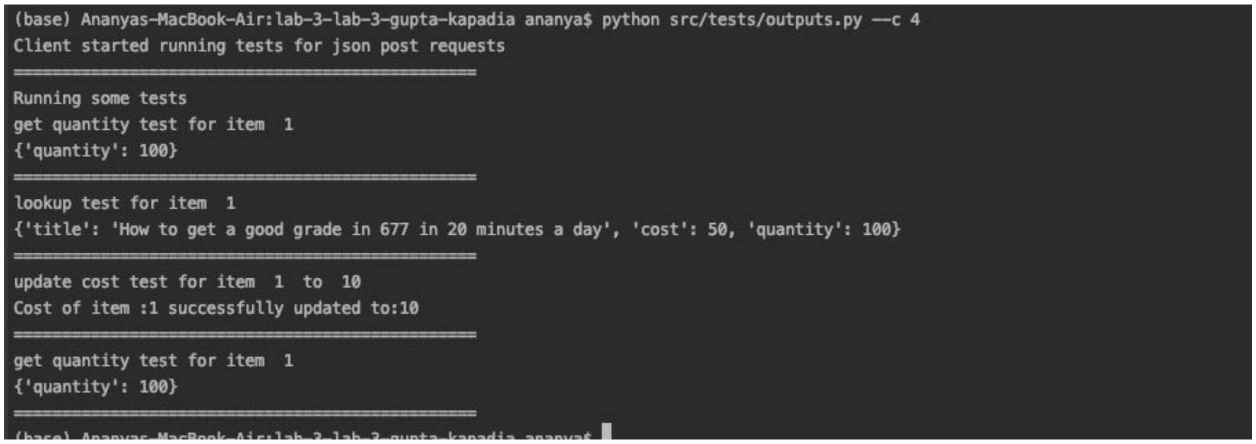
Catalog server 1 getting all the 3 requests



```
^C(base) Ananyas-MacBook-Air:catalog ananya$ python catalog_server2.py --c 4 --i 1
Opened database successfully
Table created successfully
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
Record successfully added
 catalog heartbeat port listening at: 35306
Running on  127.0.0.1 50001
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received get quantity for item : 1
Received lookup request for item : 1
created recovery log successfully:db_logcatalog1.csv
Received get quantity for item : 1
```

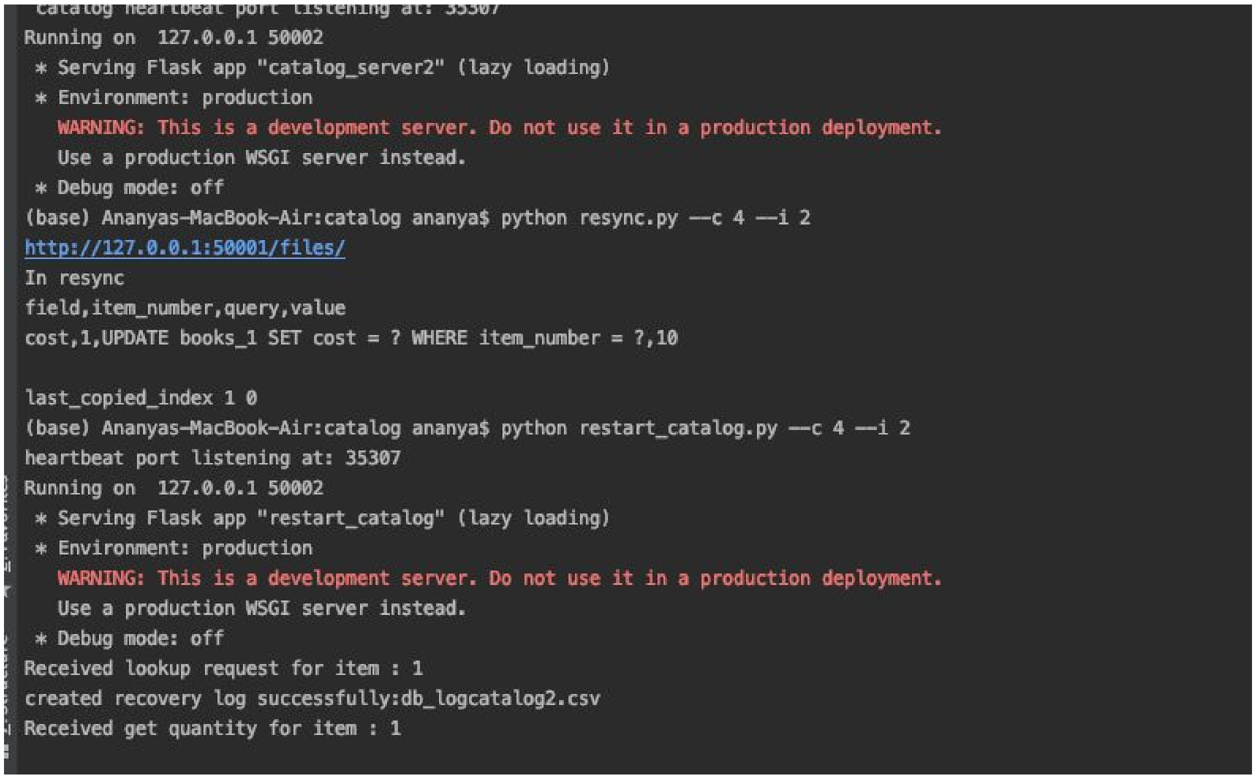
Failure of catalog server and resync and restart of catalog server and resync of logs

Sent requests after restarting and resyncing the catalog server 2, request served by the catalog server2 after its database resyncs with the other catalog instance



```
(base) Ananyas-MacBook-Air:lab-3-lab-3-gupta-kapadia ananya$ python src/tests/outputs.py --c 4
Client started running tests for json post requests
================================================
Running some tests
get quantity test for item  1
{'quantity': 100}
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 50, 'quantity': 100}
================================================
update cost test for item  1  to  10
Cost of item :1 successfully updated to:10
================================================
get quantity test for item  1
{'quantity': 100}
================================================
```

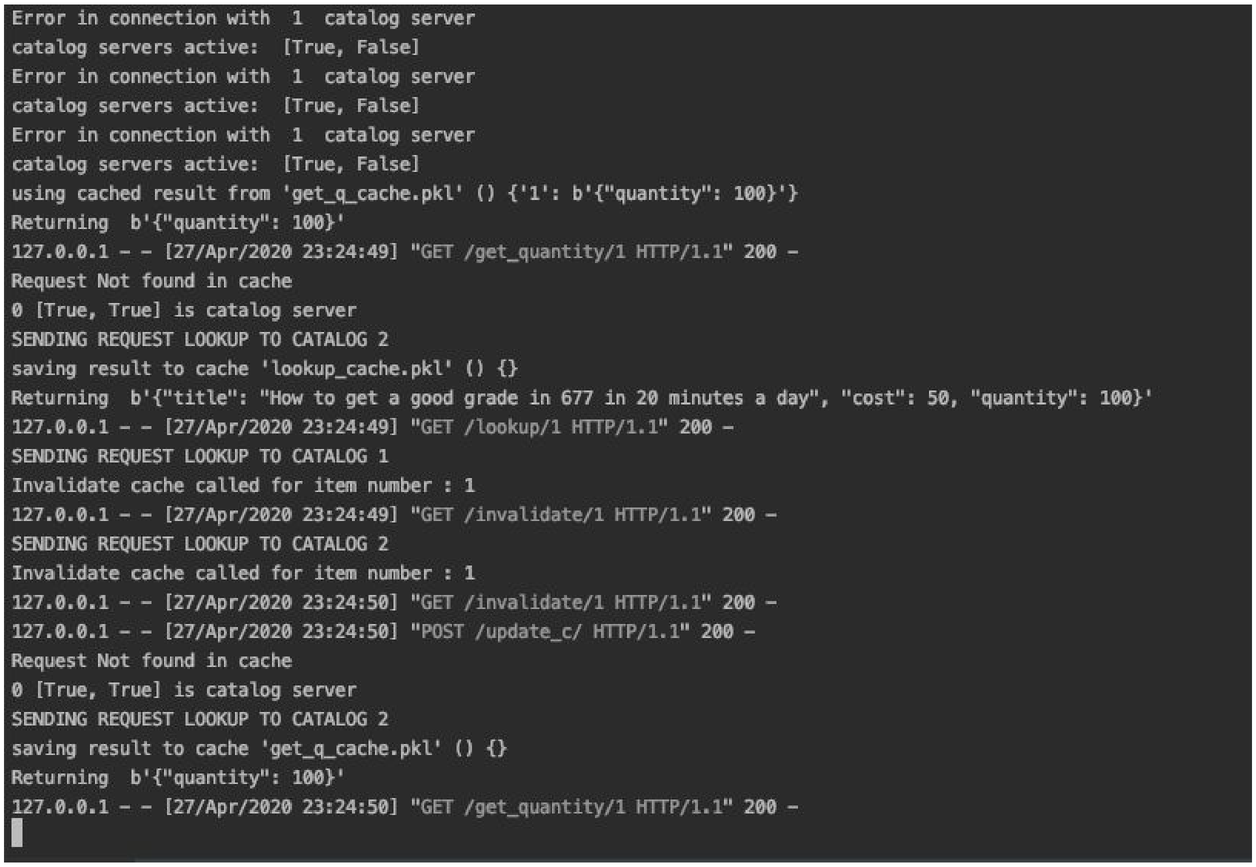
Catalog server 2 serving the request with the updated value (while it was down got updated in catalog server 1 but it resyncs and restarts so is consistent)



```
 catalog heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
(base) Ananyas-MacBook-Air:catalog ananya$ python resync.py --c 4 --i 2
http://127.0.0.1:50001/files/
In resync
field,item_number,query,value
cost,1,UPDATE books_1 SET cost = ? WHERE item_number = ?,10

last_copied_index 1 0
(base) Ananyas-MacBook-Air:catalog ananya$ python restart_catalog.py --c 4 --i 2
heartbeat port listening at: 35307
Running on  127.0.0.1 50002
 * Serving Flask app "restart_catalog" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Received lookup request for item : 1
created recovery log successfully:db_logcatalog2.csv
Received get quantity for item : 1
```

Front end server now forwarding the request to catalog server 2



```
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Error in connection with  1  catalog server
catalog servers active:  [True, False]
Error in connection with  1  catalog server
catalog servers active:  [True, False]
using cached result from 'get_q_cache.pkl' () {'1': b'{"quantity": 100}'}
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:24:49] "GET /get_quantity/1 HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
SENDING REQUEST LOOKUP TO CATALOG 2
saving result to cache 'lookup_cache.pkl' () {}
Returning  b'{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 50, "quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:24:49] "GET /lookup/1 HTTP/1.1" 200 -
SENDING REQUEST LOOKUP TO CATALOG 1
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 23:24:49] "GET /invalidate/1 HTTP/1.1" 200 -
SENDING REQUEST LOOKUP TO CATALOG 2
Invalidate cache called for item number : 1
127.0.0.1 - - [27/Apr/2020 23:24:50] "GET /invalidate/1 HTTP/1.1" 200 -
127.0.0.1 - - [27/Apr/2020 23:24:50] "POST /update_c/ HTTP/1.1" 200 -
Request Not found in cache
0 [True, True] is catalog server
SENDING REQUEST LOOKUP TO CATALOG 2
saving result to cache 'get_q_cache.pkl' () {}
Returning  b'{"quantity": 100}'
127.0.0.1 - - [27/Apr/2020 23:24:50] "GET /get_quantity/1 HTTP/1.1" 200 -
```

Other miscellaneous Test Cases:

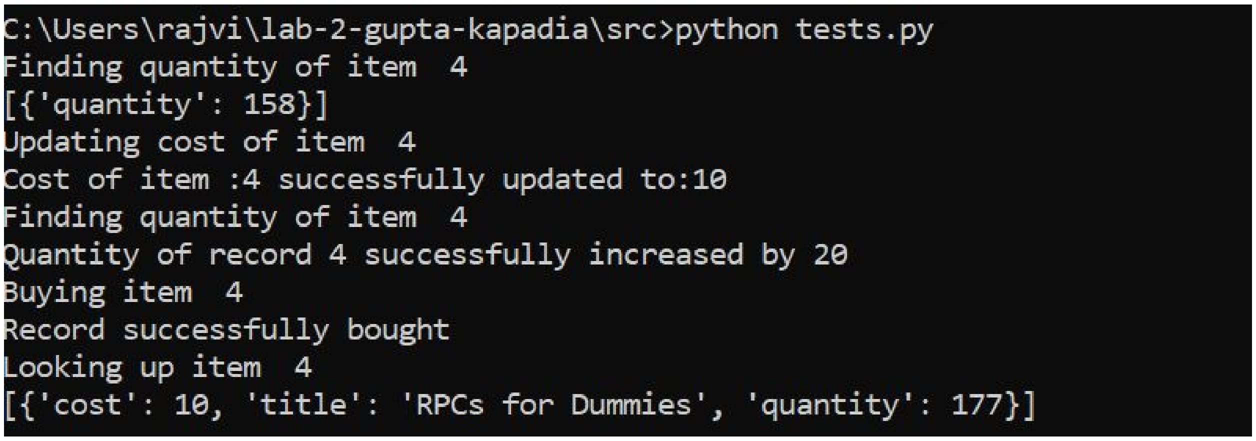
We have tested with the following test cases to test the correctness of our system. This includes running the whole system on one single machine, plus testing the system while running different processes on different machines, which is mentioned in the configuration files 1 and 2.

So depending on the configuration file argument passed the servers are started on those respective ip and ports, as obtained from the config files.
We also tested the **GET and POST HTTP** requests for each of the servers as well the front end. For this during testing we also used postman for sending requests and verifying outputs.

Failed test cases: We faced some issues in case of overloading of requests, when a server crashed and threw connection errors.

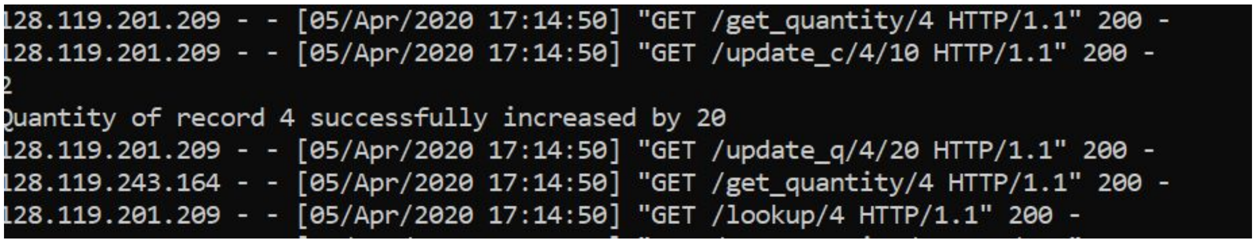
The proof and evidence of correctness can be found in subsequent sections.

Testcase 1:
We ran the frontend, order,catalog server and test client on different machines. We created a database, added records , increased quantity of item 4 to , updated its cost to 10, and bought a record of the same item. We display the results of the final values of the item we modified by a lookup.
Updating and buying using the GET HTTP requests



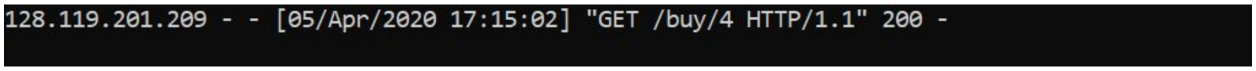
```
C:\Users\rajvi\lab-2-gupta-kapadia\src>python tests.py
Finding quantity of item  4
[{'quantity': 158}]
Updating cost of item  4
Cost of item :4 successfully updated to:10
Finding quantity of item  4
Quantity of record 4 successfully increased by 20
Buying item  4
Record successfully bought
Looking up item  4
[{'cost': 10, 'title': 'RPCs for Dummies', 'quantity': 177}]
```

Fig 1:Screenshot of output of client executed locally



```
128.119.201.209 - - [05/Apr/2020 17:14:50] "GET /get_quantity/4 HTTP/1.1" 200 -
128.119.201.209 - - [05/Apr/2020 17:14:50] "GET /update_c/4/10 HTTP/1.1" 200 -
2
Quantity of record 4 successfully increased by 20
128.119.201.209 - - [05/Apr/2020 17:14:50] "GET /update_q/4/20 HTTP/1.1" 200 -
128.119.243.164 - - [05/Apr/2020 17:14:50] "GET /get_quantity/4 HTTP/1.1" 200 -
128.119.201.209 - - [05/Apr/2020 17:14:50] "GET /lookup/4 HTTP/1.1" 200 -
```

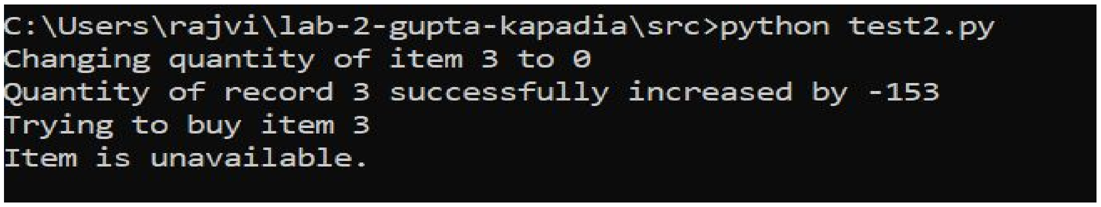
Fig 2: Screenshot of edlab 1: requests made to catalog server



```
128.119.201.209 - - [05/Apr/2020 17:15:02] "GET /buy/4 HTTP/1.1" 200 -
```

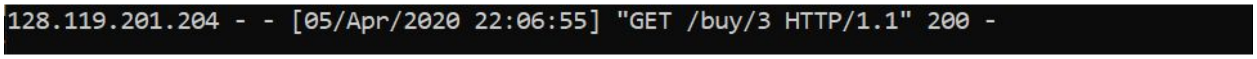
Fig 3: Screenshot of edlab 2: Requests made to order server

Testcase 2:
We decreased the quantity of item 3 to 0 and tried to buy it, giving the following error message.



```
C:\Users\rajvi\lab-2-gupta-kapadia\src>python test2.py
Changing quantity of item 3 to 0
Quantity of record 3 successfully increased by -153
Trying to buy item 3
Item is unavailable.
```

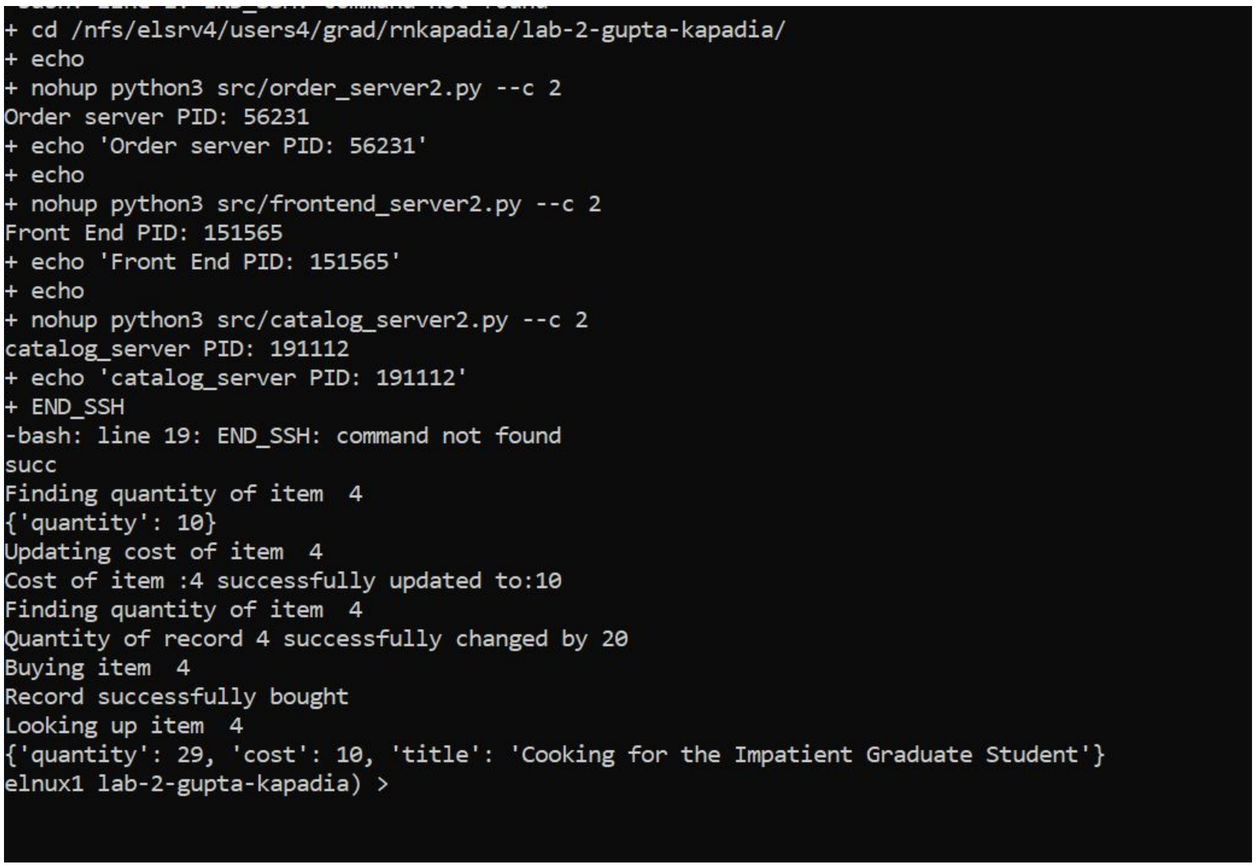
Fig 4:Screenshot of client



```
128.119.201.204 - - [05/Apr/2020 22:06:55] "GET /buy/3 HTTP/1.1" 200 -
```

Fig 5:Screenshot of order server on edlab 2

Testcase 3:
Here all the microservices run on the same edlab machine 2 and the testing client runs on edlab 1. It repeats the same requests as in test case 1.

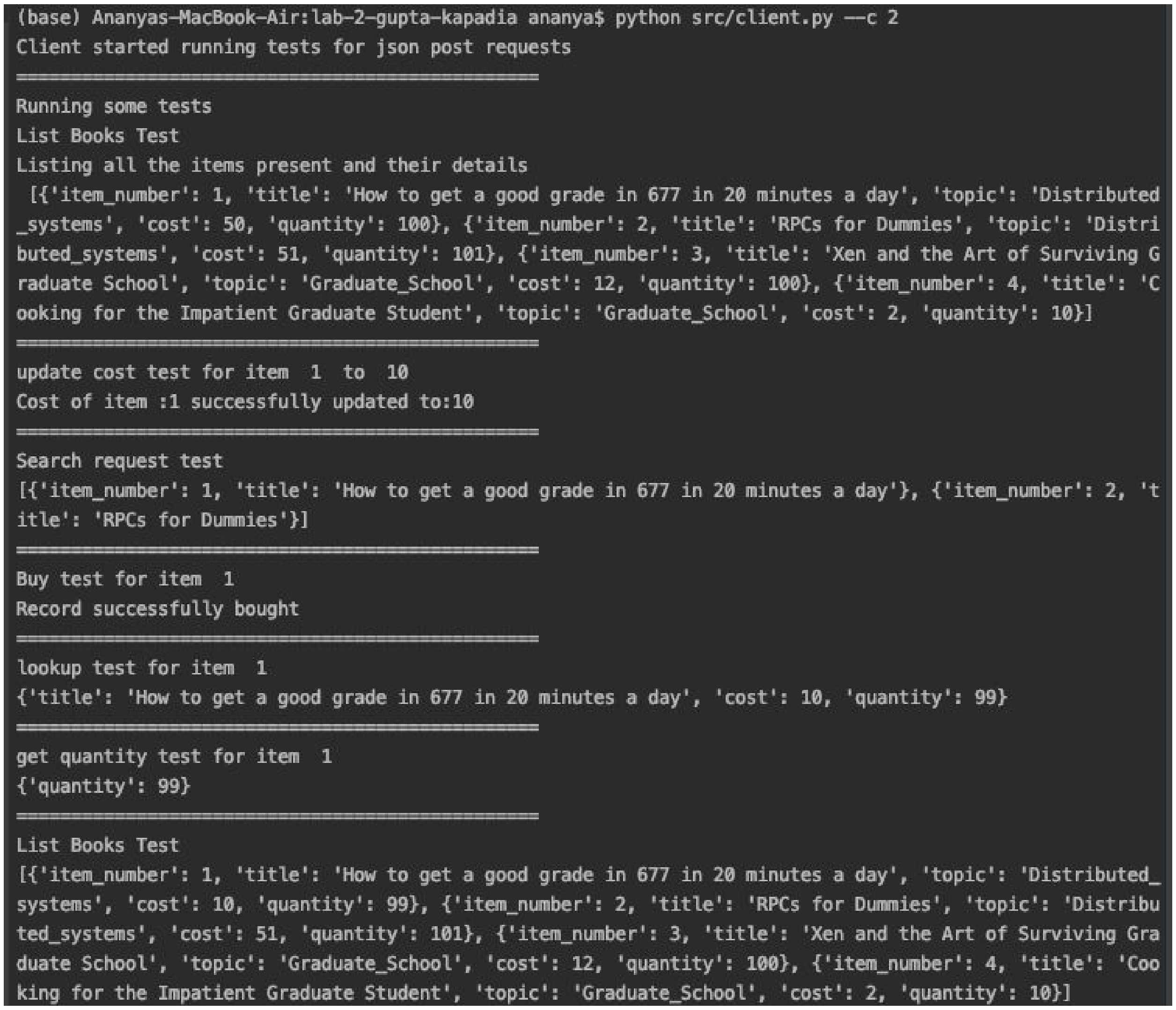


```
+ cd /nfs/elsrv4/users4/grad/rnkapadia/lab-2-gupta-kapadia/
+ echo
+ nohup python3 src/order_server2.py --c 2
Order server PID: 56231
+ echo 'Order server PID: 56231'
+ echo
+ nohup python3 src/frontend_server2.py --c 2
Front End PID: 151565
+ echo 'Front End PID: 151565'
+ echo
+ nohup python3 src/catalog_server2.py --c 2
catalog_server PID: 191112
+ echo 'catalog_server PID: 191112'
+ END_SSH
-bash: line 19: END_SSH: command not found
succ
Finding quantity of item  4
{'quantity': 10}
Updating cost of item  4
Cost of item :4 successfully updated to:10
Finding quantity of item  4
Quantity of record 4 successfully changed by 20
Buying item  4
Record successfully bought
Looking up item  4
{'quantity': 29, 'cost': 10, 'title': 'Cooking for the Impatient Graduate Student'}
elnux1 lab-2-gupta-kapadia) >
```

Fig 3: All services on the same machine for test case 3

Testcase 4:
Here src/client.py runs and it sends POST HTTP requests to the front end server and hence the entire system gets tested. The following screenshots are outputs of this test-

Running Client.py (Tests system by sending post requests)



```
(base) Ananyas-MacBook-Air:lab-2-gupta-kapadia ananya$ python src/client.py --c 2
Client started running tests for json post requests
================================================
Running some tests
List Books Test
Listing all the items present and their details
 [{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 minutes a day', 'topic': 'Distributed
_systems', 'cost': 50, 'quantity': 100}, {'item_number': 2, 'title': 'RPCs for Dummies', 'topic': 'Distri
buted_systems', 'cost': 51, 'quantity': 101}, {'item_number': 3, 'title': 'Xen and the Art of Surviving G
raduate School', 'topic': 'Graduate_School', 'cost': 12, 'quantity': 100}, {'item_number': 4, 'title': 'C
ooking for the Impatient Graduate Student', 'topic': 'Graduate_School', 'cost': 2, 'quantity': 10}]
================================================
update cost test for item  1  to  10
Cost of item :1 successfully updated to:10
================================================
Search request test
[{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 minutes a day'}, {'item_number': 2, 't
itle': 'RPCs for Dummies'}]
================================================
Buy test for item  1
Record successfully bought
================================================
lookup test for item  1
{'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 10, 'quantity': 99}
================================================
get quantity test for item  1
{'quantity': 99}
================================================
List Books Test
[{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 minutes a day', 'topic': 'Distributed_
systems', 'cost': 10, 'quantity': 99}, {'item_number': 2, 'title': 'RPCs for Dummies', 'topic': 'Distribu
ted_systems', 'cost': 51, 'quantity': 101}, {'item_number': 3, 'title': 'Xen and the Art of Surviving Gra
duate School', 'topic': 'Graduate_School', 'cost': 12, 'quantity': 100}, {'item_number': 4, 'title': 'Coo
king for the Impatient Graduate Student', 'topic': 'Graduate_School', 'cost': 2, 'quantity': 10}]
```

The client.py file tests the entire system by sending requests to the front_end_server with POST requests where the request body has content in json format. It does the following tests-

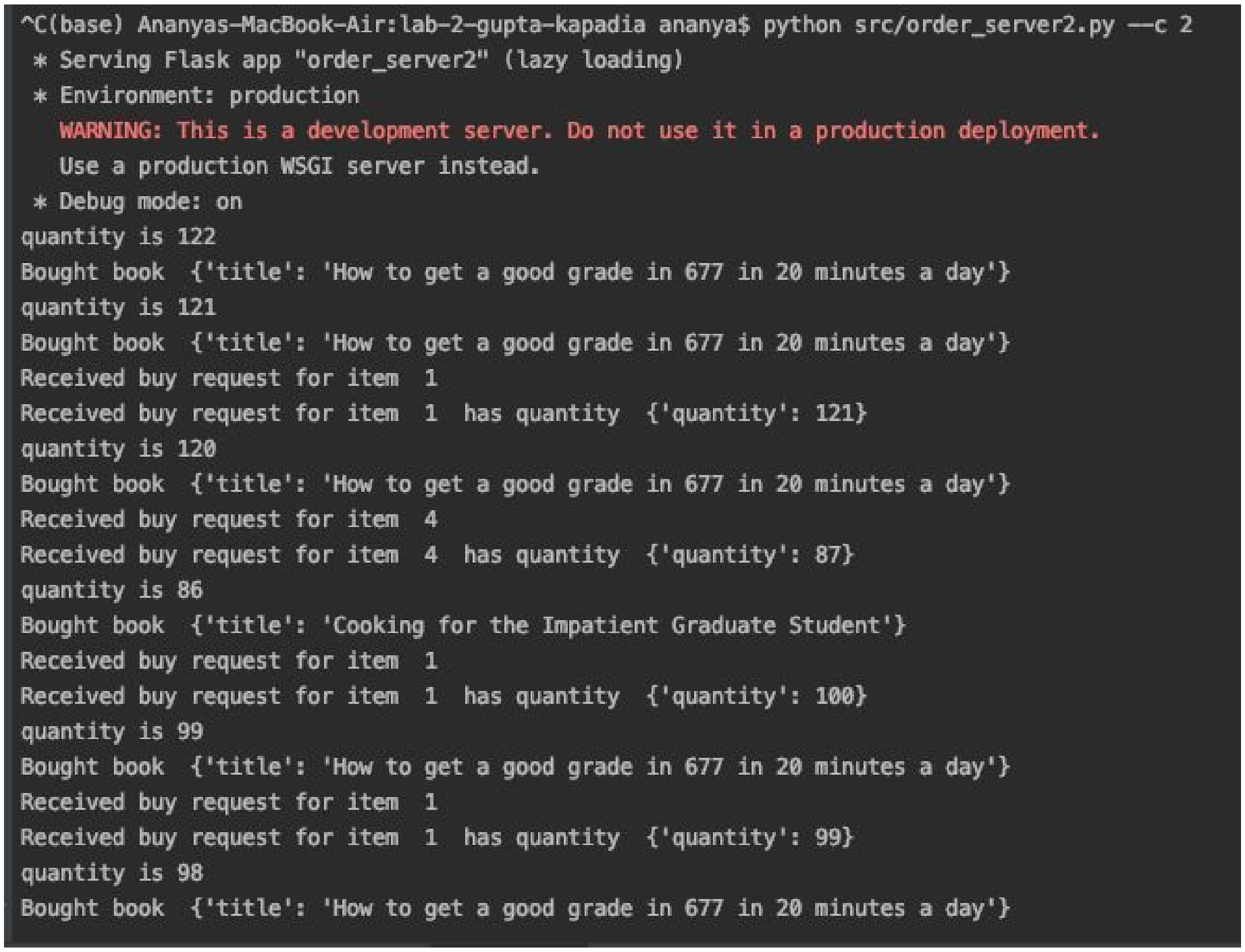
list() -> which lists all the items present in the table books_1 (Item 1 has cost=50, quantity=100)

update_cost() -> which updates the cost of the item 1 to 10

search() -> which returns all items with topic Distributed_systems

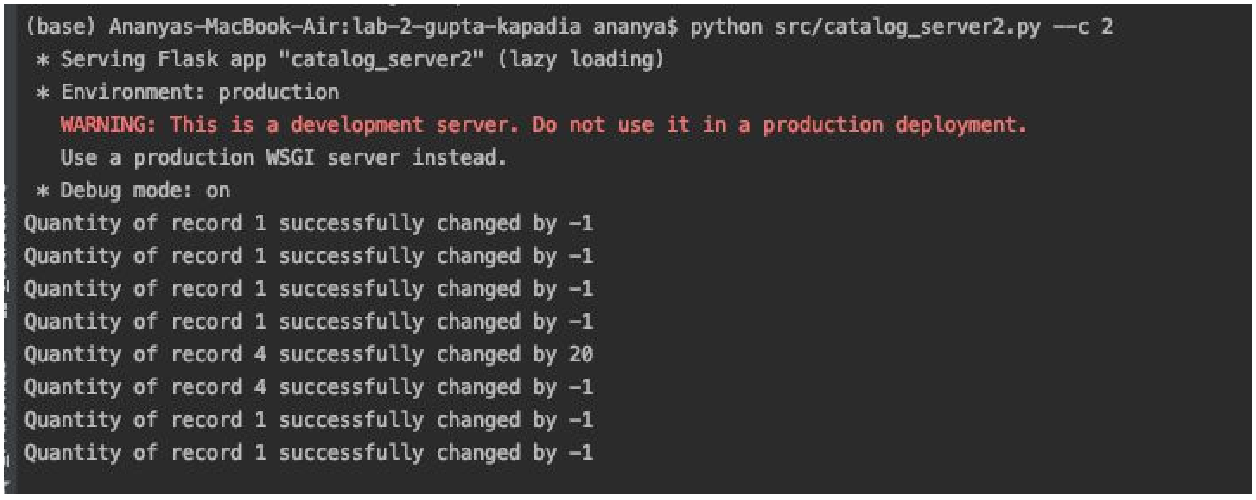
buy() -> which buys the item 1 and reduces the quantity of item by 1.(The corresponding logs get printed and logged in other server components as seen below)
lookup() ->which looks up item book with item_number 1
get_quantity() -> which returns the updated quantity of the item 1(99 from 100)
list() -> which returns the updated cost(10) and quantity(99) of the item 1 and hence verifies the accuracy of the system as a whole.

Running the order server which gets requests about the buying of items, it checks the quantity first and then if item present then proceeds with decreasing count else it just returns error and unavailable item message, shows bought in the following screenshot-



```
^C(base) Ananyas-MacBook-Air:lab-2-gupta-kapadia ananya$ python src/order_server2.py --c 2
 * Serving Flask app "order_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
quantity is 122
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
quantity is 121
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
Received buy request for item  1
Received buy request for item  1  has quantity  {'quantity': 121}
quantity is 120
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
Received buy request for item  4
Received buy request for item  4  has quantity  {'quantity': 87}
quantity is 86
Bought book  {'title': 'Cooking for the Impatient Graduate Student'}
Received buy request for item  1
Received buy request for item  1  has quantity  {'quantity': 100}
quantity is 99
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
Received buy request for item  1
Received buy request for item  1  has quantity  {'quantity': 99}
quantity is 98
Bought book  {'title': 'How to get a good grade in 677 in 20 minutes a day'}
```

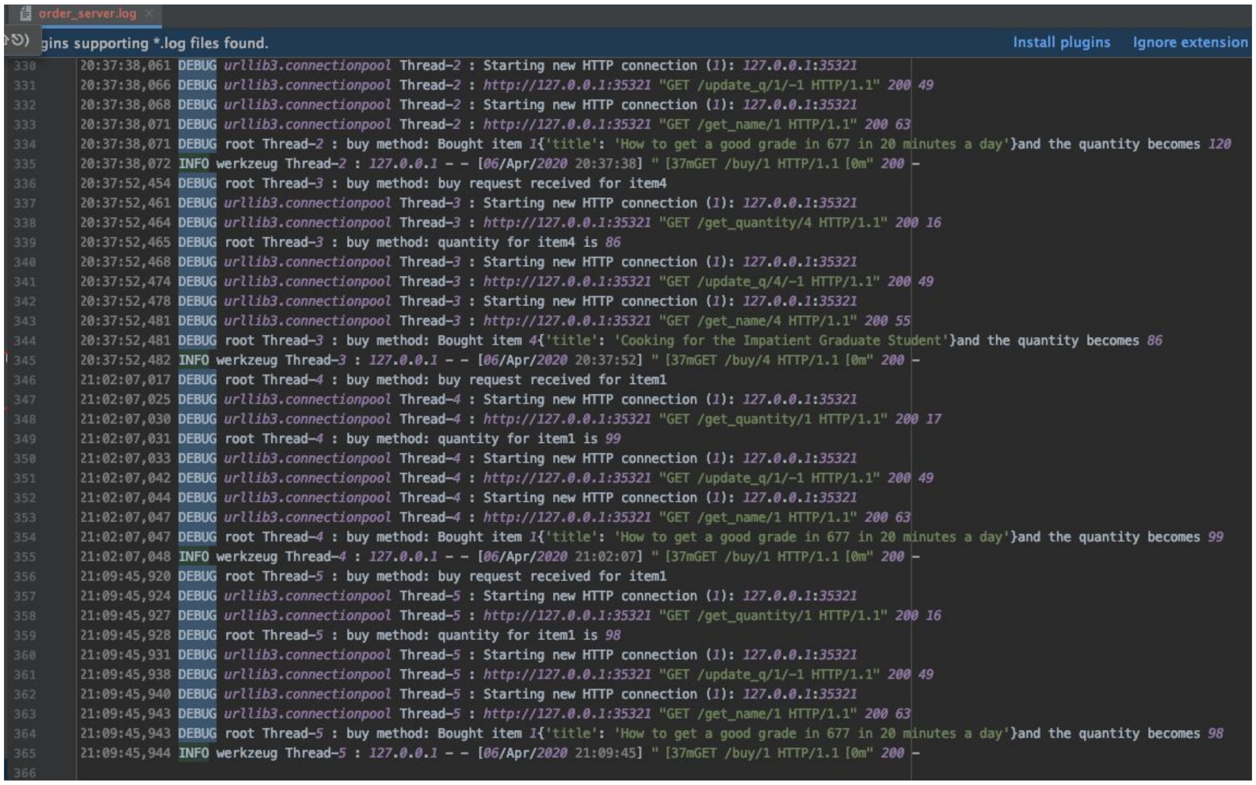
Running the catalog server which gets updates about the increment and decrement of quantity of items



```
(base) Ananyas-MacBook-Air:lab-2-gupta-kapadia ananya$ python src/catalog_server2.py --c 2
 * Serving Flask app "catalog_server2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
Quantity of record 1 successfully changed by -1
Quantity of record 1 successfully changed by -1
Quantity of record 1 successfully changed by -1
Quantity of record 1 successfully changed by -1
Quantity of record 4 successfully changed by 20
Quantity of record 4 successfully changed by -1
Quantity of record 1 successfully changed by -1
Quantity of record 1 successfully changed by -1
```

LOGS on the servers- They have timestamp, thread and other log related information

Logs for order server

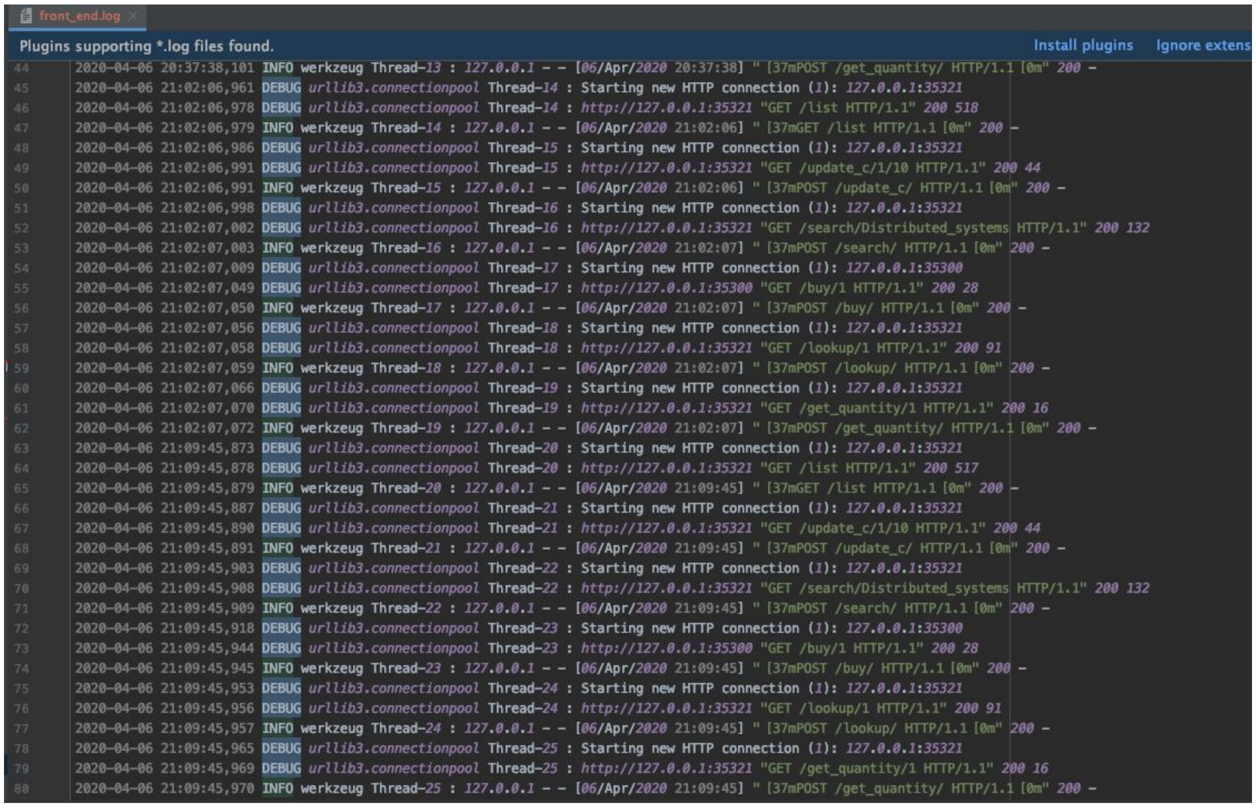


order_server.log

gins supporting *.log files found. Install plugins Ignore extension

```
20:37:38,061 DEBUG urllib3.connectionpool Thread-2 : Starting new HTTP connection (1): 127.0.0.1:35321
20:37:38,066 DEBUG urllib3.connectionpool Thread-2 : http://127.0.0.1:35321 "GET /update_q/1/-1 HTTP/1.1" 200 49
20:37:38,068 DEBUG urllib3.connectionpool Thread-2 : Starting new HTTP connection (1): 127.0.0.1:35321
20:37:38,071 DEBUG urllib3.connectionpool Thread-2 : http://127.0.0.1:35321 "GET /get_name/1 HTTP/1.1" 200 63
20:37:38,071 DEBUG root Thread-2 : buy method: Bought item 1{'title': 'How to get a good grade in 677 in 20 minutes a day'}and the quantity becomes 120
20:37:38,072 INFO werkzeug Thread-2 : 127.0.0.1 - - [06/Apr/2020 20:37:38] " [37mGET /buy/1 HTTP/1.1 [0m" 200 -
20:37:52,454 DEBUG root Thread-3 : buy method: buy request received for item4
20:37:52,461 DEBUG urllib3.connectionpool Thread-3 : Starting new HTTP connection (1): 127.0.0.1:35321
20:37:52,464 DEBUG urllib3.connectionpool Thread-3 : http://127.0.0.1:35321 "GET /get_quantity/4 HTTP/1.1" 200 16
20:37:52,465 DEBUG root Thread-3 : buy method: quantity for item4 is 86
20:37:52,468 DEBUG urllib3.connectionpool Thread-3 : Starting new HTTP connection (1): 127.0.0.1:35321
20:37:52,474 DEBUG urllib3.connectionpool Thread-3 : http://127.0.0.1:35321 "GET /update_q/4/-1 HTTP/1.1" 200 49
20:37:52,478 DEBUG urllib3.connectionpool Thread-3 : Starting new HTTP connection (1): 127.0.0.1:35321
20:37:52,481 DEBUG urllib3.connectionpool Thread-3 : http://127.0.0.1:35321 "GET /get_name/4 HTTP/1.1" 200 55
20:37:52,481 DEBUG root Thread-3 : buy method: Bought item 4{'title': 'Cooking for the Impatient Graduate Student'}and the quantity becomes 86
20:37:52,482 INFO werkzeug Thread-3 : 127.0.0.1 - - [06/Apr/2020 20:37:52] " [37mGET /buy/4 HTTP/1.1 [0m" 200 -
21:02:07,017 DEBUG root Thread-4 : buy method: buy request received for item1
21:02:07,025 DEBUG urllib3.connectionpool Thread-4 : Starting new HTTP connection (1): 127.0.0.1:35321
21:02:07,030 DEBUG urllib3.connectionpool Thread-4 : http://127.0.0.1:35321 "GET /get_quantity/1 HTTP/1.1" 200 17
21:02:07,031 DEBUG root Thread-4 : buy method: quantity for item1 is 99
21:02:07,033 DEBUG urllib3.connectionpool Thread-4 : Starting new HTTP connection (1): 127.0.0.1:35321
21:02:07,042 DEBUG urllib3.connectionpool Thread-4 : http://127.0.0.1:35321 "GET /update_q/1/-1 HTTP/1.1" 200 49
21:02:07,044 DEBUG urllib3.connectionpool Thread-4 : Starting new HTTP connection (1): 127.0.0.1:35321
21:02:07,047 DEBUG urllib3.connectionpool Thread-4 : http://127.0.0.1:35321 "GET /get_name/1 HTTP/1.1" 200 63
21:02:07,047 DEBUG root Thread-4 : buy method: Bought item 1{'title': 'How to get a good grade in 677 in 20 minutes a day'}and the quantity becomes 99
21:02:07,048 INFO werkzeug Thread-4 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mGET /buy/1 HTTP/1.1 [0m" 200 -
21:09:45,920 DEBUG root Thread-5 : buy method: buy request received for item1
21:09:45,924 DEBUG urllib3.connectionpool Thread-5 : Starting new HTTP connection (1): 127.0.0.1:35321
21:09:45,927 DEBUG urllib3.connectionpool Thread-5 : http://127.0.0.1:35321 "GET /get_quantity/1 HTTP/1.1" 200 16
21:09:45,928 DEBUG root Thread-5 : buy method: quantity for item1 is 98
21:09:45,931 DEBUG urllib3.connectionpool Thread-5 : Starting new HTTP connection (1): 127.0.0.1:35321
21:09:45,938 DEBUG urllib3.connectionpool Thread-5 : http://127.0.0.1:35321 "GET /update_q/1/-1 HTTP/1.1" 200 49
21:09:45,940 DEBUG urllib3.connectionpool Thread-5 : Starting new HTTP connection (1): 127.0.0.1:35321
21:09:45,943 DEBUG urllib3.connectionpool Thread-5 : http://127.0.0.1:35321 "GET /get_name/1 HTTP/1.1" 200 63
21:09:45,943 DEBUG root Thread-5 : buy method: Bought item 1{'title': 'How to get a good grade in 677 in 20 minutes a day'}and the quantity becomes 98
21:09:45,944 INFO werkzeug Thread-5 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /buy/1 HTTP/1.1 [0m" 200 -
```

Logs for front end server

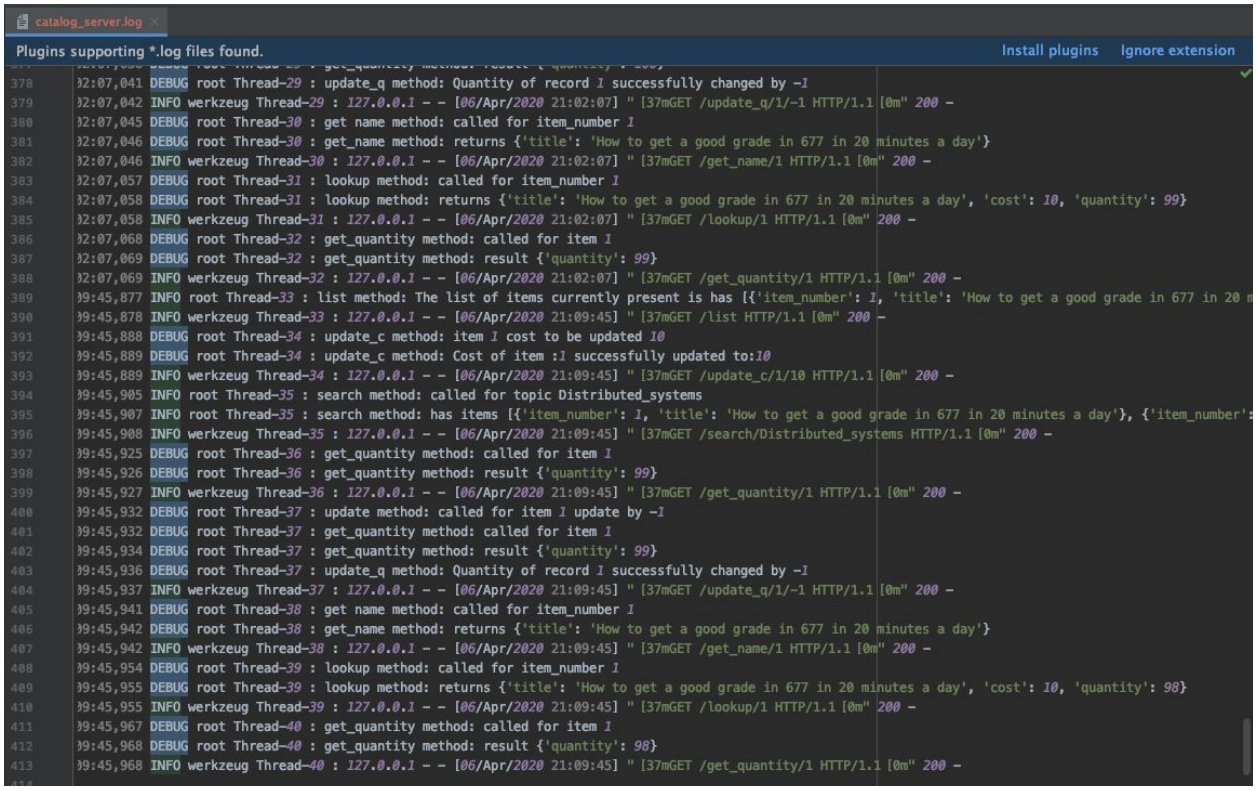


front_end.log

Plugins supporting *.log files found. Install plugins Ignore extens

```
2020-04-06 20:37:38,101 INFO werkzeug Thread-13 : 127.0.0.1 - - [06/Apr/2020 20:37:38] " [37mPOST /get_quantity/ HTTP/1.1 [0m" 200 -
2020-04-06 21:02:06,961 DEBUG urllib3.connectionpool Thread-14 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:02:06,978 DEBUG urllib3.connectionpool Thread-14 : http://127.0.0.1:35321 "GET /list HTTP/1.1" 200 518
2020-04-06 21:02:06,979 INFO werkzeug Thread-14 : 127.0.0.1 - - [06/Apr/2020 21:02:06] " [37mGET /list HTTP/1.1 [0m" 200 -
2020-04-06 21:02:06,986 DEBUG urllib3.connectionpool Thread-15 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:02:06,991 DEBUG urllib3.connectionpool Thread-15 : http://127.0.0.1:35321 "GET /update_c/1/10 HTTP/1.1" 200 44
2020-04-06 21:02:06,991 INFO werkzeug Thread-15 : 127.0.0.1 - - [06/Apr/2020 21:02:06] " [37mPOST /update_c/ HTTP/1.1 [0m" 200 -
2020-04-06 21:02:06,998 DEBUG urllib3.connectionpool Thread-16 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:02:07,002 DEBUG urllib3.connectionpool Thread-16 : http://127.0.0.1:35321 "GET /search/Distributed_systems HTTP/1.1" 200 132
2020-04-06 21:02:07,003 INFO werkzeug Thread-16 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mPOST /search/ HTTP/1.1 [0m" 200 -
2020-04-06 21:02:07,009 DEBUG urllib3.connectionpool Thread-17 : Starting new HTTP connection (1): 127.0.0.1:35300
2020-04-06 21:02:07,049 DEBUG urllib3.connectionpool Thread-17 : http://127.0.0.1:35300 "GET /buy/1 HTTP/1.1" 200 28
2020-04-06 21:02:07,050 INFO werkzeug Thread-17 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mPOST /buy/ HTTP/1.1 [0m" 200 -
2020-04-06 21:02:07,056 DEBUG urllib3.connectionpool Thread-18 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:02:07,058 DEBUG urllib3.connectionpool Thread-18 : http://127.0.0.1:35321 "GET /lookup/1 HTTP/1.1" 200 91
2020-04-06 21:02:07,059 INFO werkzeug Thread-18 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mPOST /lookup/ HTTP/1.1 [0m" 200 -
2020-04-06 21:02:07,066 DEBUG urllib3.connectionpool Thread-19 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:02:07,070 DEBUG urllib3.connectionpool Thread-19 : http://127.0.0.1:35321 "GET /get_quantity/1 HTTP/1.1" 200 16
2020-04-06 21:02:07,072 INFO werkzeug Thread-19 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mPOST /get_quantity/ HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,873 DEBUG urllib3.connectionpool Thread-20 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:09:45,878 DEBUG urllib3.connectionpool Thread-20 : http://127.0.0.1:35321 "GET /list HTTP/1.1" 200 517
2020-04-06 21:09:45,879 INFO werkzeug Thread-20 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /list HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,887 DEBUG urllib3.connectionpool Thread-21 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:09:45,890 DEBUG urllib3.connectionpool Thread-21 : http://127.0.0.1:35321 "GET /update_c/1/10 HTTP/1.1" 200 44
2020-04-06 21:09:45,891 INFO werkzeug Thread-21 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mPOST /update_c/ HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,903 DEBUG urllib3.connectionpool Thread-22 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:09:45,908 DEBUG urllib3.connectionpool Thread-22 : http://127.0.0.1:35321 "GET /search/Distributed_systems HTTP/1.1" 200 132
2020-04-06 21:09:45,909 INFO werkzeug Thread-22 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mPOST /search/ HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,918 DEBUG urllib3.connectionpool Thread-23 : Starting new HTTP connection (1): 127.0.0.1:35300
2020-04-06 21:09:45,944 DEBUG urllib3.connectionpool Thread-23 : http://127.0.0.1:35300 "GET /buy/1 HTTP/1.1" 200 28
2020-04-06 21:09:45,945 INFO werkzeug Thread-23 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mPOST /buy/ HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,953 DEBUG urllib3.connectionpool Thread-24 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:09:45,956 DEBUG urllib3.connectionpool Thread-24 : http://127.0.0.1:35321 "GET /lookup/1 HTTP/1.1" 200 91
2020-04-06 21:09:45,957 INFO werkzeug Thread-24 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mPOST /lookup/ HTTP/1.1 [0m" 200 -
2020-04-06 21:09:45,965 DEBUG urllib3.connectionpool Thread-25 : Starting new HTTP connection (1): 127.0.0.1:35321
2020-04-06 21:09:45,969 DEBUG urllib3.connectionpool Thread-25 : http://127.0.0.1:35321 "GET /get_quantity/1 HTTP/1.1" 200 16
2020-04-06 21:09:45,970 INFO werkzeug Thread-25 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mPOST /get_quantity/ HTTP/1.1 [0m" 200 -
```

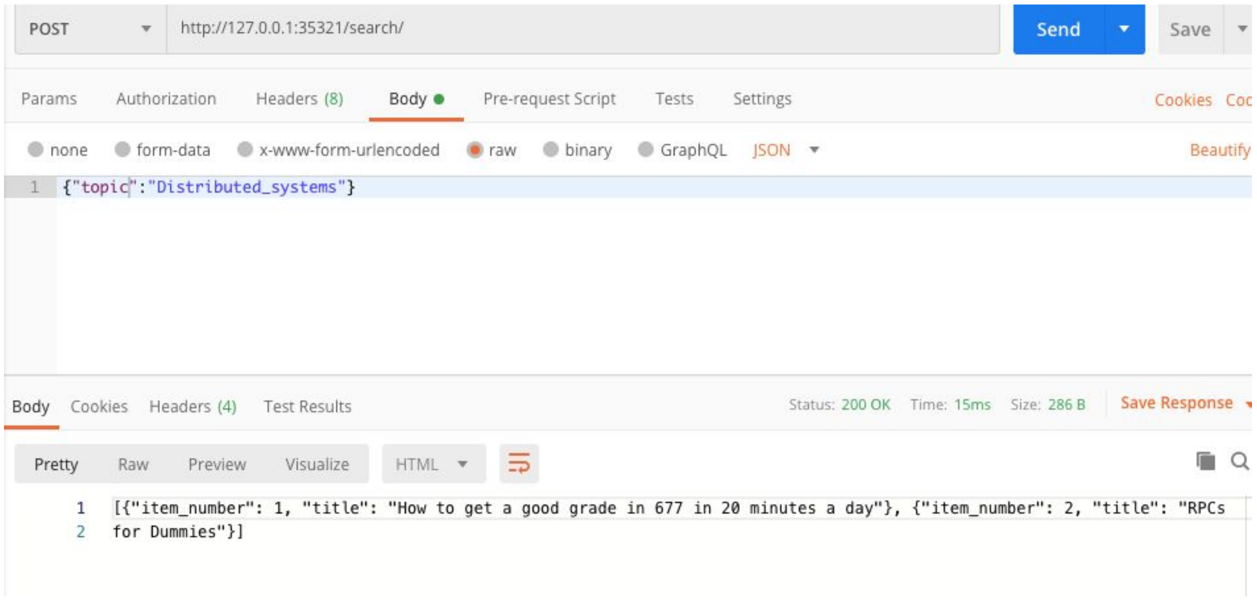
Logs for catalog server



```
2:07,041 DEBUG root Thread-29 : update_q method: Quantity of record 1 successfully changed by -1
2:07,042 INFO werkzeug Thread-29 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mGET /update_q/1/-1 HTTP/1.1 [0m" 200 -
2:07,045 DEBUG root Thread-30 : get name method: called for item_number 1
2:07,046 DEBUG root Thread-30 : get_name method: returns {'title': 'How to get a good grade in 677 in 20 minutes a day'}
2:07,046 INFO werkzeug Thread-30 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mGET /get_name/1 HTTP/1.1 [0m" 200 -
2:07,057 DEBUG root Thread-31 : lookup method: called for item_number 1
2:07,058 DEBUG root Thread-31 : lookup method: returns {'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 10, 'quantity': 99}
2:07,058 INFO werkzeug Thread-31 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mGET /lookup/1 HTTP/1.1 [0m" 200 -
2:07,068 DEBUG root Thread-32 : get_quantity method: called for item 1
2:07,069 DEBUG root Thread-32 : get_quantity method: result {'quantity': 99}
2:07,069 INFO werkzeug Thread-32 : 127.0.0.1 - - [06/Apr/2020 21:02:07] " [37mGET /get_quantity/1 HTTP/1.1 [0m" 200 -
9:45,877 INFO root Thread-33 : list method: The list of items currently present is has [{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 
9:45,878 INFO werkzeug Thread-33 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /list HTTP/1.1 [0m" 200 -
9:45,888 DEBUG root Thread-34 : update_c method: item 1 cost to be updated 10
9:45,889 DEBUG root Thread-34 : update_c method: Cost of item :1 successfully updated to:10
9:45,889 INFO werkzeug Thread-34 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /update_c/1/10 HTTP/1.1 [0m" 200 -
9:45,905 INFO root Thread-35 : search method: called for topic Distributed_systems
9:45,907 INFO root Thread-35 : search method: has items [{'item_number': 1, 'title': 'How to get a good grade in 677 in 20 minutes a day'}, {'item_number':
9:45,908 INFO werkzeug Thread-35 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /search/Distributed_systems HTTP/1.1 [0m" 200 -
9:45,925 DEBUG root Thread-36 : get_quantity method: called for item 1
9:45,926 DEBUG root Thread-36 : get_quantity method: result {'quantity': 99}
9:45,927 INFO werkzeug Thread-36 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /get_quantity/1 HTTP/1.1 [0m" 200 -
9:45,932 DEBUG root Thread-37 : update method: called for item 1 update by -1
9:45,932 DEBUG root Thread-37 : get_quantity method: called for item 1
9:45,934 DEBUG root Thread-37 : get_quantity method: result {'quantity': 99}
9:45,936 DEBUG root Thread-37 : update_q method: Quantity of record 1 successfully changed by -1
9:45,937 INFO werkzeug Thread-37 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /update_q/1/-1 HTTP/1.1 [0m" 200 -
9:45,941 DEBUG root Thread-38 : get name method: called for item_number 1
9:45,942 DEBUG root Thread-38 : get_name method: returns {'title': 'How to get a good grade in 677 in 20 minutes a day'}
9:45,942 INFO werkzeug Thread-38 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /get_name/1 HTTP/1.1 [0m" 200 -
9:45,954 DEBUG root Thread-39 : lookup method: called for item_number 1
9:45,955 DEBUG root Thread-39 : lookup method: returns {'title': 'How to get a good grade in 677 in 20 minutes a day', 'cost': 10, 'quantity': 98}
9:45,955 INFO werkzeug Thread-39 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /lookup/1 HTTP/1.1 [0m" 200 -
9:45,967 DEBUG root Thread-40 : get_quantity method: called for item 1
9:45,968 DEBUG root Thread-40 : get_quantity method: result {'quantity': 98}
9:45,968 INFO werkzeug Thread-40 : 127.0.0.1 - - [06/Apr/2020 21:09:45] " [37mGET /get_quantity/1 HTTP/1.1 [0m" 200 -
```

TESTS DONE WITH POSTMAN REQUESTS for front end server-

Search for topic with post request



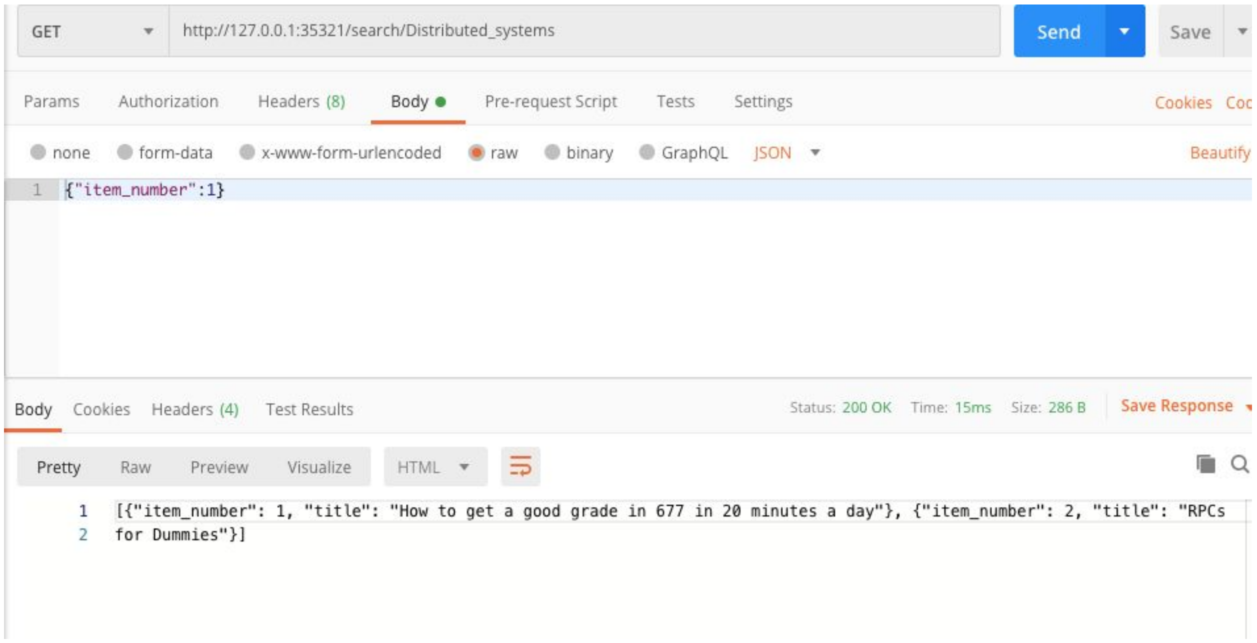
http://127.0.0.1:35321/search/

```
{"topic":"Distributed_systems"}
```

Status: 200 OK Time: 15ms Size: 286 B

```
[{"item_number": 1, "title": "How to get a good grade in 677 in 20 minutes a day"}, {"item_number": 2, "title": "RPCs
for Dummies"}]
```

Search for topic with get request



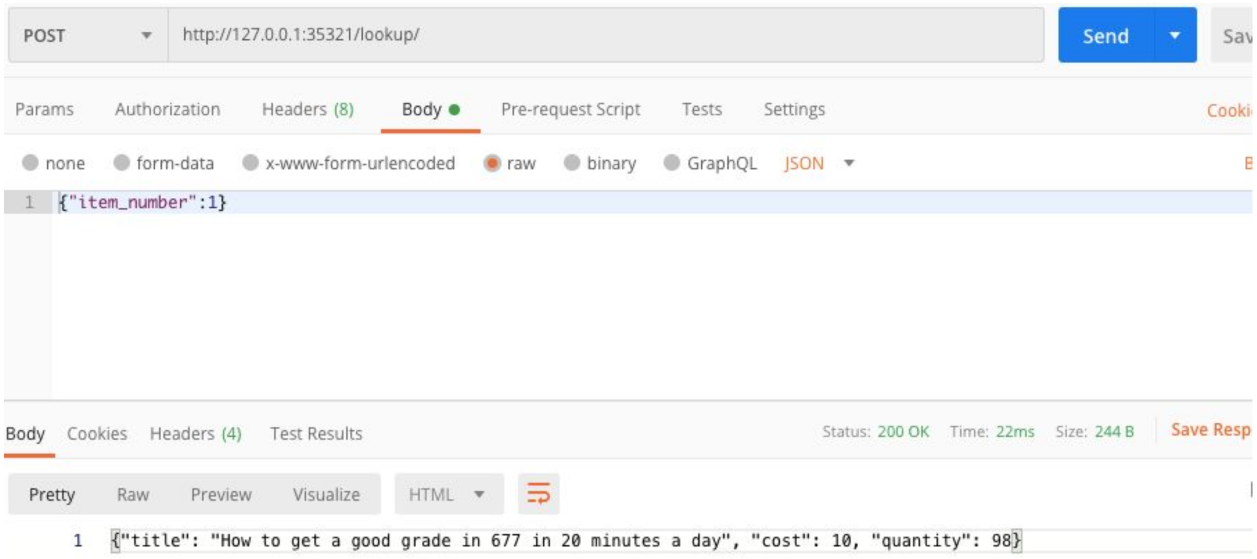
http://127.0.0.1:35321/search/Distributed_systems

```
{"item_number":1}
```

Status: 200 OK Time: 15ms Size: 286 B

```
[{"item_number": 1, "title": "How to get a good grade in 677 in 20 minutes a day"}, {"item_number": 2, "title": "RPCs
for Dummies"}]
```

Search for item with post request



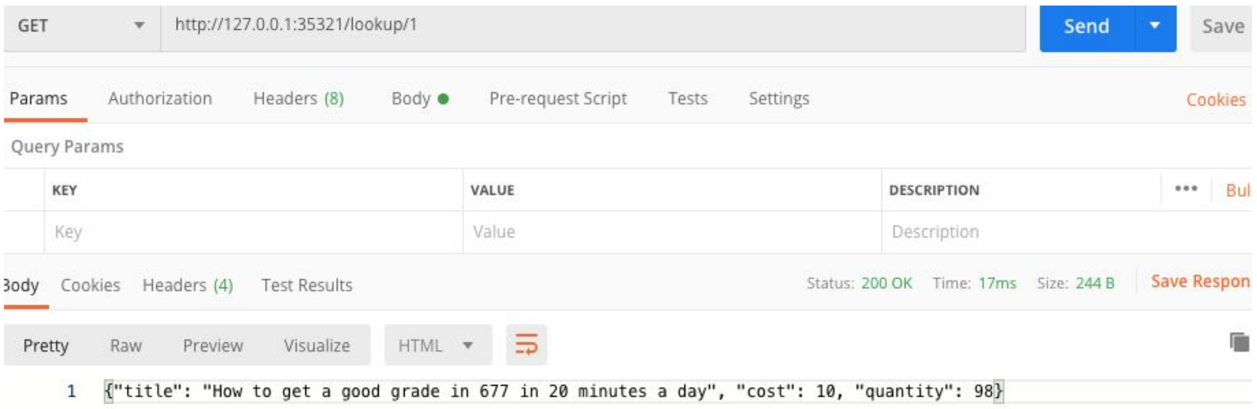
POST http://127.0.0.1:35321/lookup/

```
{"item_number":1}
```

Status: 200 OK Time: 22ms Size: 244 B

```
{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 10, "quantity": 98}
```

Search for item with get request

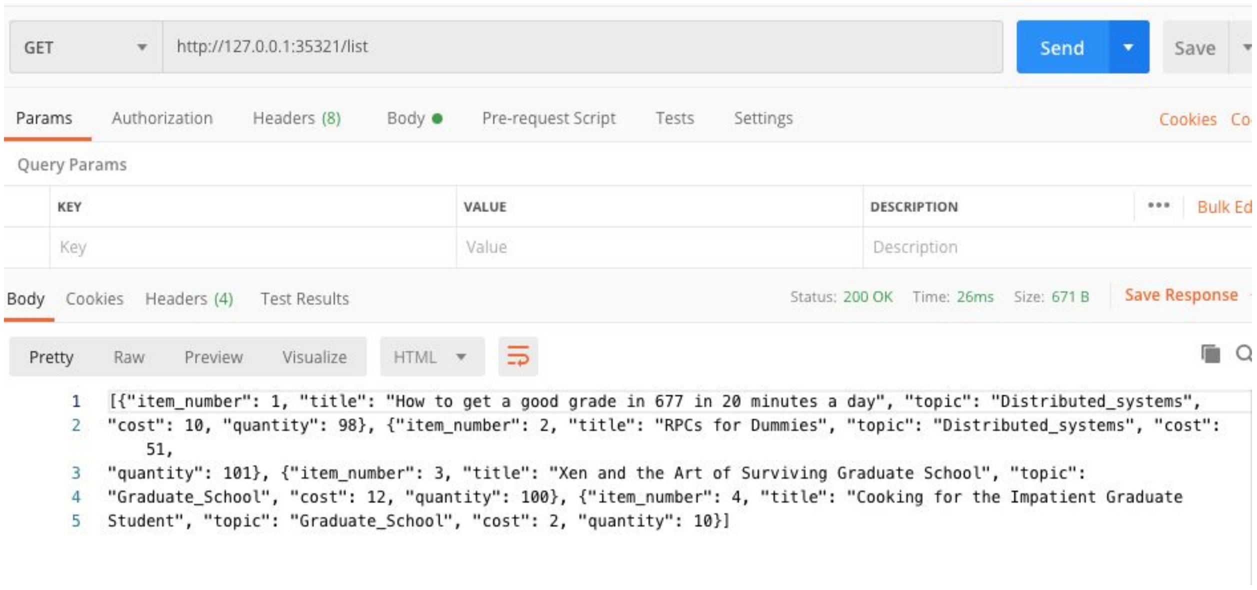


GET http://127.0.0.1:35321/lookup/1

Status: 200 OK Time: 17ms Size: 244 B

```
{"title": "How to get a good grade in 677 in 20 minutes a day", "cost": 10, "quantity": 98}
```

Search for all the items

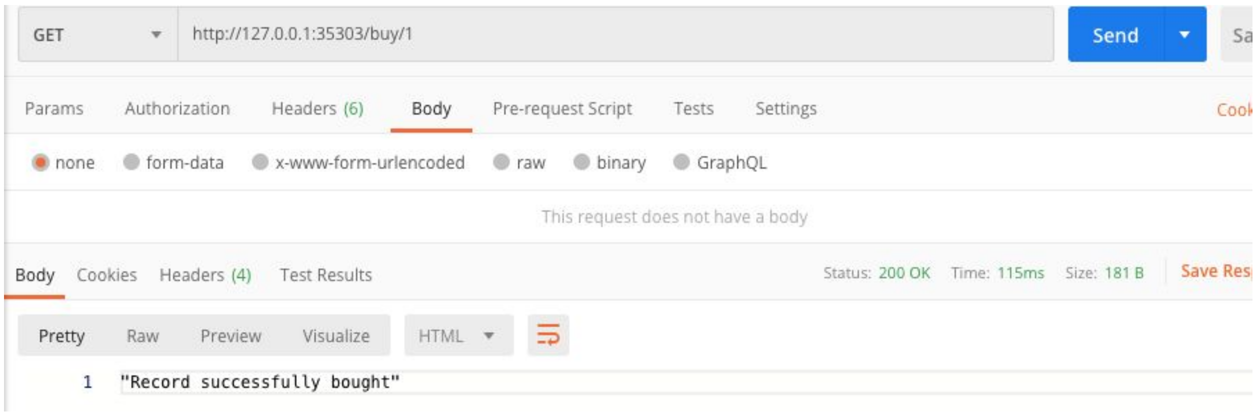


http://127.0.0.1:35321/list

Status: 200 OK Time: 26ms Size: 671 B

```
[{"item_number": 1, "title": "How to get a good grade in 677 in 20 minutes a day", "topic": "Distributed_systems",
"cost": 10, "quantity": 98}, {"item_number": 2, "title": "RPCs for Dummies", "topic": "Distributed_systems", "cost":
    51,
"quantity": 101}, {"item_number": 3, "title": "Xen and the Art of Surviving Graduate School", "topic":
"Graduate_School", "cost": 12, "quantity": 100}, {"item_number": 4, "title": "Cooking for the Impatient Graduate
Student", "topic": "Graduate_School", "cost": 2, "quantity": 10}]
```

Buy Request



http://127.0.0.1:35303/buy/1

This request does not have a body

Status: 200 OK Time: 115ms Size: 181 B

```
"Record successfully bought"
```