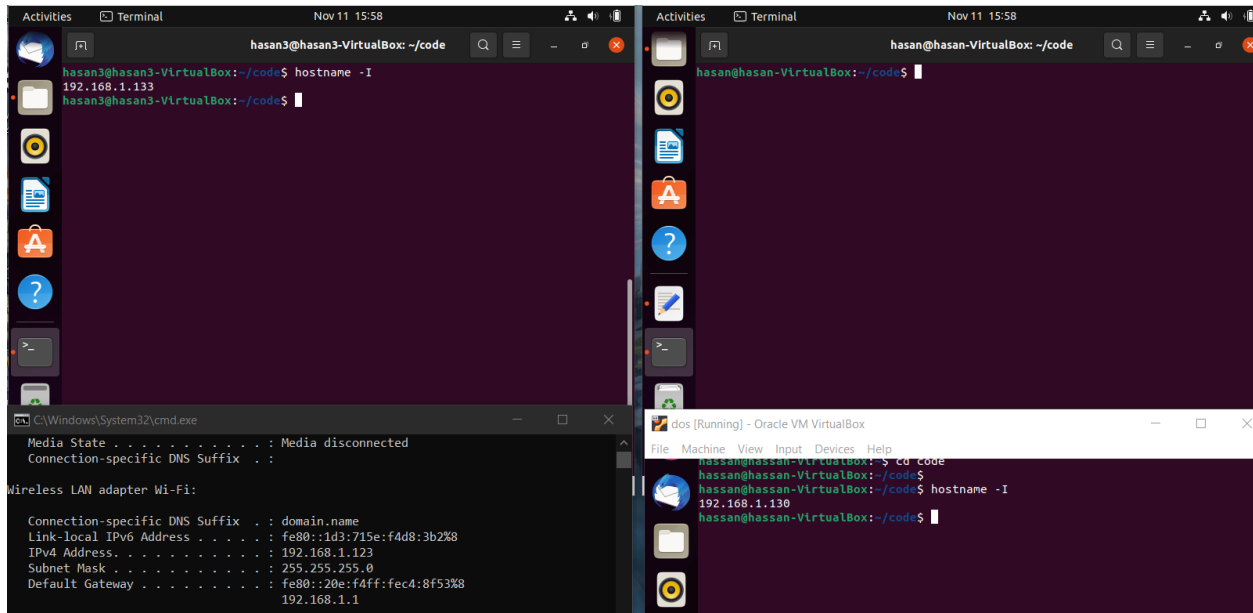


Hostname-I and IPCONFIG Outputs:



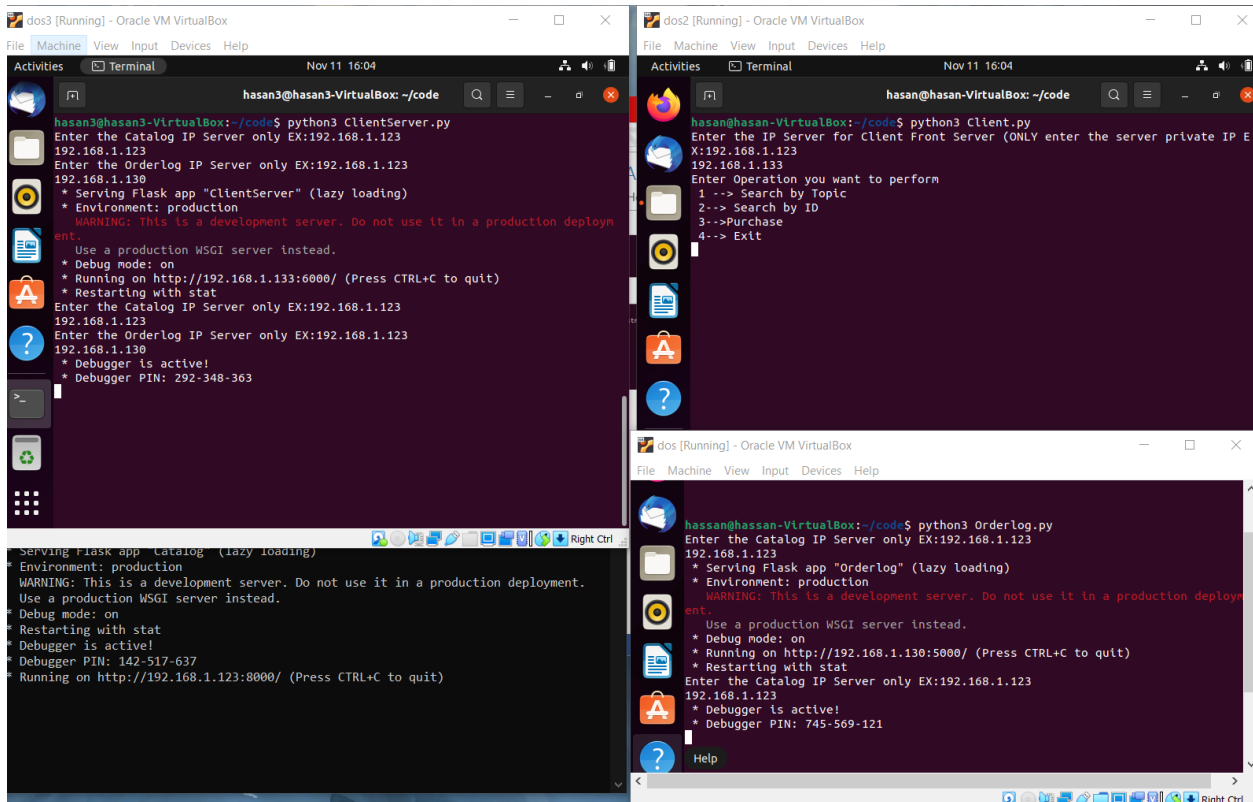
The image shows two terminal windows side-by-side. The left window is titled 'hasan3@hasan3-VirtualBox: ~/code' and shows the output of the 'hostname -I' command, which returns '192.168.1.133'. Below this, a Windows command prompt window titled 'cmd.exe' shows the output of the 'ipconfig' command, displaying network configuration for a wireless LAN adapter, including the IP address '192.168.1.123' and the default gateway '192.168.1.1'. The right window is also titled 'hasan@hasan-VirtualBox: ~/code' and shows the output of the 'hostname -I' command, which returns '192.168.1.130'.

```
hasan3@hasan3-VirtualBox: ~/code
hasan3@hasan3-VirtualBox:~/code$ hostname -I
192.168.1.133
hasan3@hasan3-VirtualBox:~/code$

C:\Windows\System32\cmd.exe
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix  . :
Wireless LAN adapter Wi-Fi:
Connection-specific DNS Suffix  . : domain.name
Link-local IPv6 Address . . . . . : fe80::1d3:715e:f4d8:3b2%8
IPv4 Address. . . . . : 192.168.1.123
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::20e:f4ff:fec4:8f53%8
192.168.1.1

hasan@hasan-VirtualBox: ~/code
hasan@hasan-VirtualBox:~/code$ cd code
hasan@hasan-VirtualBox:~/code$ hostname -I
192.168.1.130
hasan@hasan-VirtualBox:~/code$
```

Running Servers:



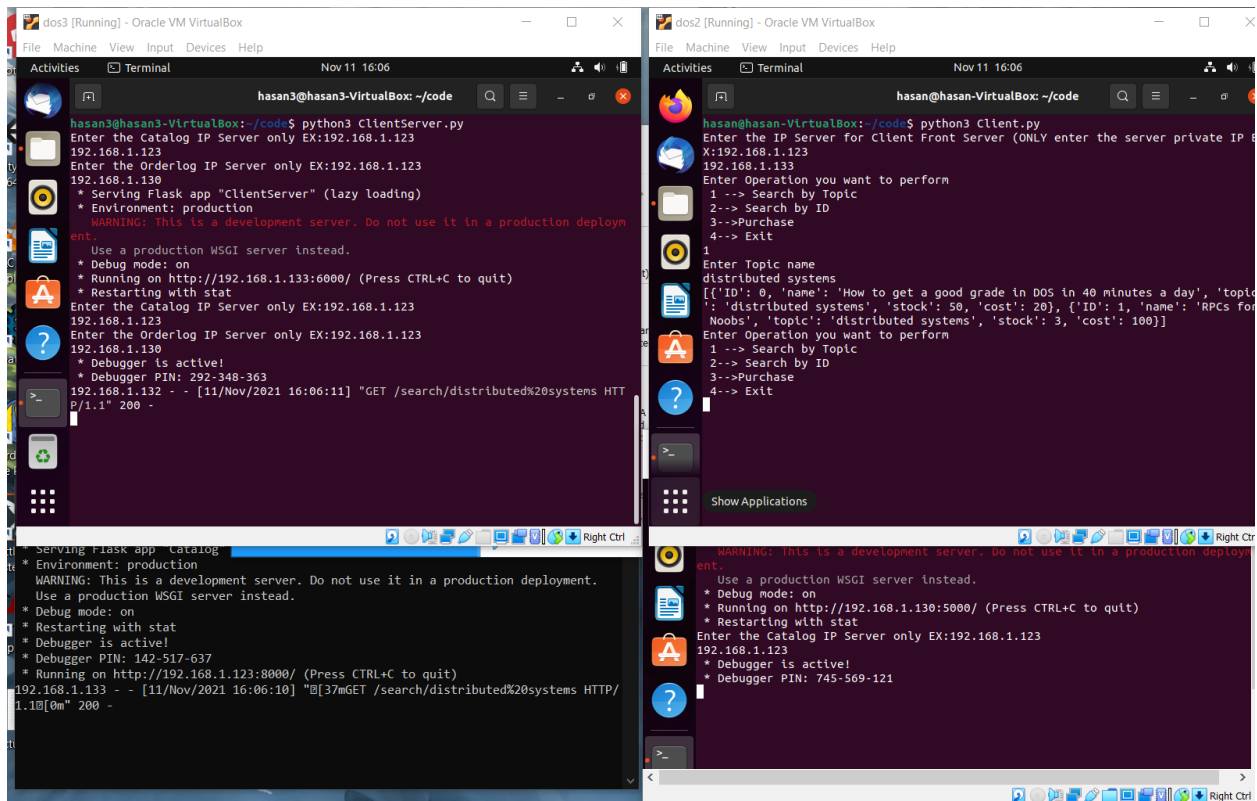
The image shows two terminal windows side-by-side, both titled 'dos3 [Running] - Oracle VM VirtualBox' and 'dos2 [Running] - Oracle VM VirtualBox'. The left window shows the output of running 'python3 ClientServer.py', which prompts for a catalog IP (192.168.1.123) and an orderlog IP (192.168.1.123). It then displays server status: 'Serving Flask app "ClientServer" (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', 'Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)', and 'Restarting with stat'. The right window shows the output of running 'python3 Client.py', which prompts for an IP server for the client front server (192.168.1.123) and an IP server for the client front server (192.168.1.133). It then displays a menu: 'Enter Operation you want to perform', '1 --> Search by Topic', '2 --> Search by ID', '3 --> Purchase', '4 --> Exit'. Below this, it shows the output of running 'python3 Orderlog.py', which prompts for a catalog IP (192.168.1.123) and an orderlog IP (192.168.1.123). It then displays server status: 'Serving Flask app "Orderlog" (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', 'Running on http://192.168.1.130:5000/ (Press CTRL+C to quit)', and 'Restarting with stat'.

```
hasan3@hasan3-VirtualBox: ~/code
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Serving Flask app "ClientServer" (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
* Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Debugger is active!
* Debugger PIN: 292-348-363

hasan@hasan-VirtualBox: ~/code
hasan@hasan-VirtualBox:~/code$ python3 Client.py
Enter the IP Server for Client Front Server (ONLY enter the server private IP E
X:192.168.1.123
192.168.1.133
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
1

hasan@hasan-VirtualBox: ~/code$ python3 Orderlog.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
* Serving Flask app "Orderlog" (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
* Running on http://192.168.1.130:5000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
* Debugger is active!
* Debugger PIN: 745-569-121
```

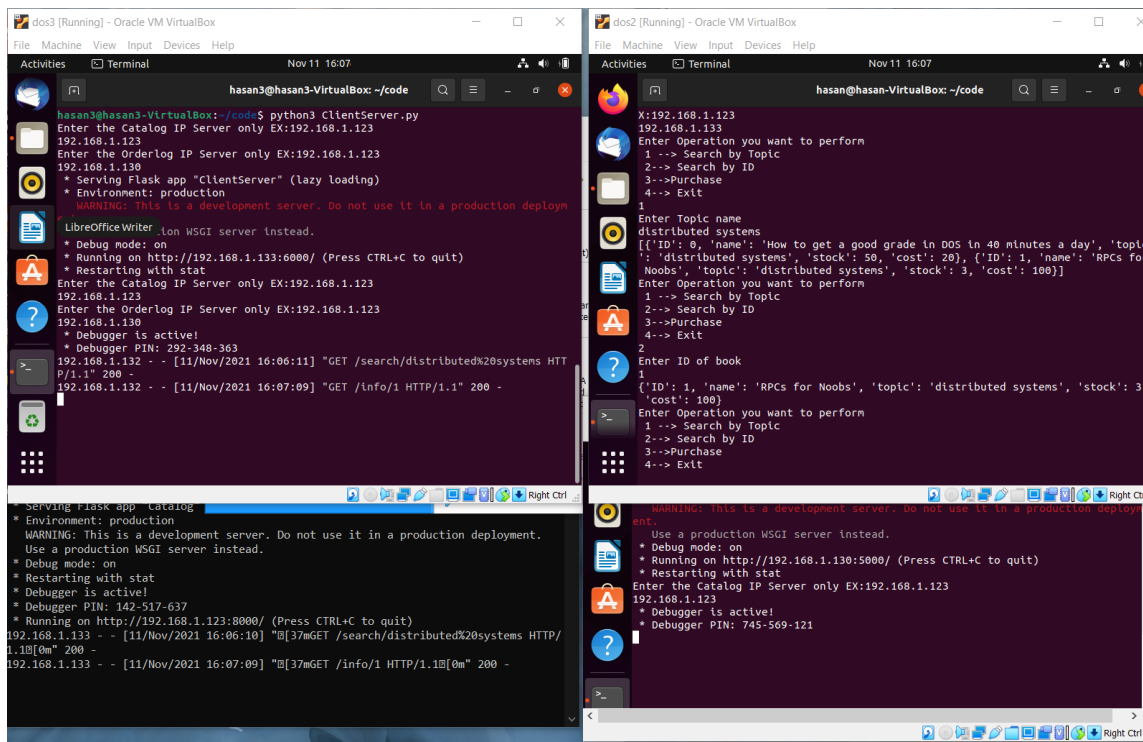
Search:



```
hasan3@hasan3-VirtualBox: ~/code
python3 ClientServer.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Serving Flask app "ClientServer" (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
* Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Debugger is active!
* Debugger PIN: 292-348-363
192.168.1.132 - - [11/Nov/2021 16:06:11] "GET /search/distributed%20systems HTTP/1.1" 200 -
```

```
hasan@hasan-VirtualBox: ~/code
python3 Client.py
Enter the IP Server for Client Front Server (ONLY enter the server private IP)
X:192.168.1.123
192.168.1.133
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
1
Enter Topic name
distributed systems
[{'ID': 0, 'name': 'How to get a good grade in DOS in 40 minutes a day', 'topic': 'distributed systems', 'stock': 50, 'cost': 20}, {'ID': 1, 'name': 'RPCs for Noobs', 'topic': 'distributed systems', 'stock': 3, 'cost': 100}]
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
```

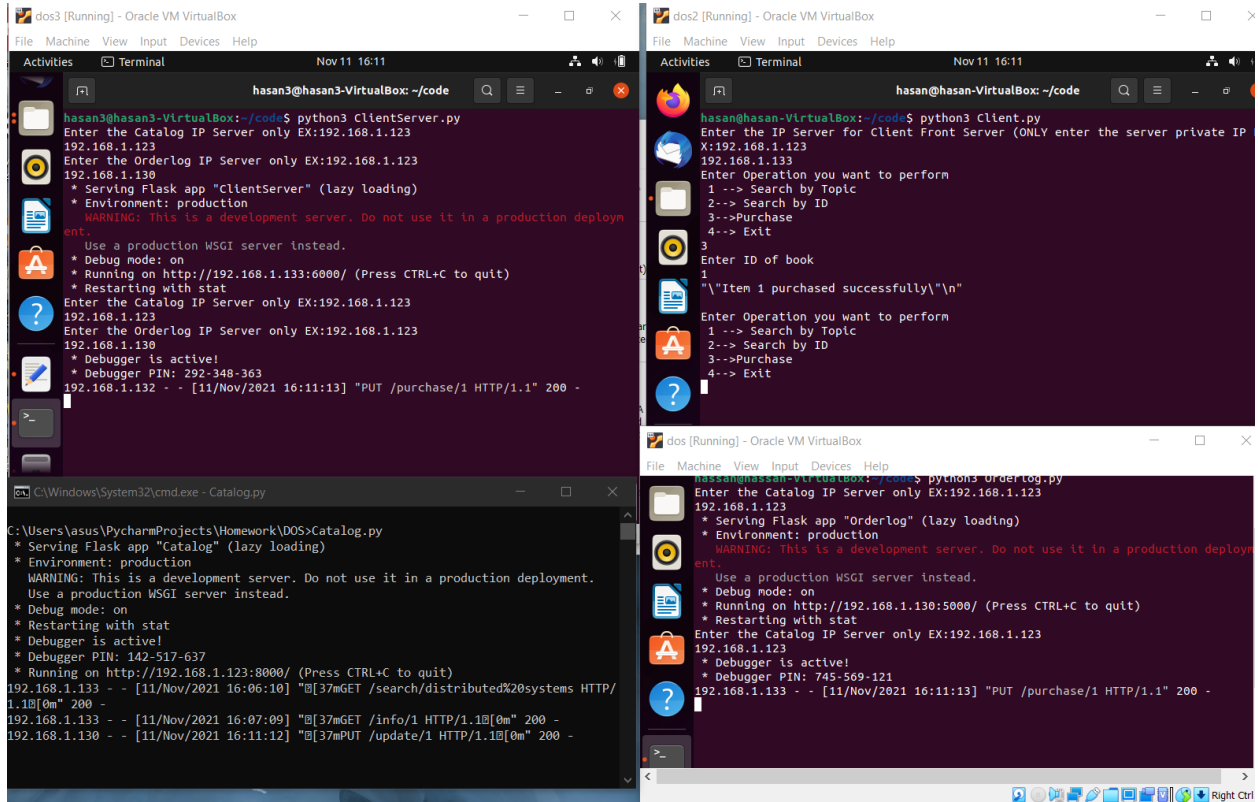
Info:



```
hasan3@hasan3-VirtualBox: ~/code
python3 ClientServer.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Serving Flask app "ClientServer" (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
* Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Debugger is active!
* Debugger PIN: 292-348-363
192.168.1.132 - - [11/Nov/2021 16:06:11] "GET /search/distributed%20systems HTTP/1.1" 200 -
192.168.1.132 - - [11/Nov/2021 16:07:09] "GET /info/1 HTTP/1.1" 200 -
```

```
hasan@hasan-VirtualBox: ~/code
X:192.168.1.123
192.168.1.133
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
1
Enter Topic name
distributed systems
[{'ID': 0, 'name': 'How to get a good grade in DOS in 40 minutes a day', 'topic': 'distributed systems', 'stock': 50, 'cost': 20}, {'ID': 1, 'name': 'RPCs for Noobs', 'topic': 'distributed systems', 'stock': 3, 'cost': 100}]
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
2
Enter ID of book
1
[{'ID': 1, 'name': 'RPCs for Noobs', 'topic': 'distributed systems', 'stock': 3, 'cost': 100}]
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
```

Purchase:



The screenshot displays three terminal windows from Oracle VM VirtualBox, showing the execution of a Flask application. The top-left window (dos3) shows the initial setup and the first purchase request. The top-right window (dos2) shows the application's response to the purchase request. The bottom window (dos) shows the application's response to the purchase request.

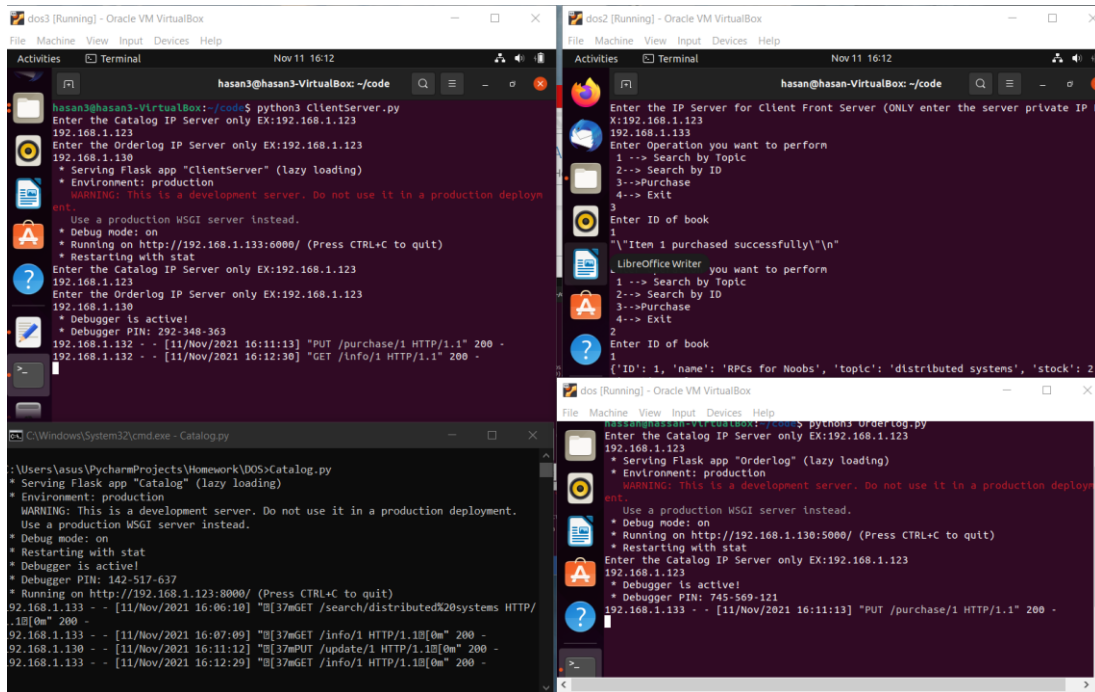
```
hasan3@hasan3-VirtualBox: ~/code
hasan3@hasan3-VirtualBox:~/code$ python3 ClientServer.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Serving Flask app "ClientServer" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Debugger is active!
* Debugger PIN: 292-348-363
192.168.1.132 - - [11/Nov/2021 16:11:13] "PUT /purchase/1 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe - Catalog.py
C:\Users\asus\PycharmProjects\Homework\DOS>Catalog.py
* Serving Flask app "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: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 142-517-637
* Running on http://192.168.1.123:8000/ (Press CTRL+C to quit)
192.168.1.133 - - [11/Nov/2021 16:06:10] "B[37mGET /search/distributed%20systems HTTP/1.1B[0m" 200 -
192.168.1.133 - - [11/Nov/2021 16:07:09] "B[37mGET /info/1 HTTP/1.1B[0m" 200 -
192.168.1.130 - - [11/Nov/2021 16:11:12] "B[37mPUT /update/1 HTTP/1.1B[0m" 200 -

hasan@hasan-VirtualBox: ~/code
hasan@hasan-VirtualBox:~/code$ python3 Client.py
Enter the IP Server for Client Front Server (ONLY enter the server private IP
X:192.168.1.123
192.168.1.133
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
3
Enter ID of book
1
"\Item 1 purchased successfully\`\n"
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit

hasan@hasan-VirtualBox: ~/code$ python3 Orderlog.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
* Serving Flask app "Orderlog" (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
* Running on http://192.168.1.130:5000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
* Debugger is active!
* Debugger PIN: 745-569-121
192.168.1.133 - - [11/Nov/2021 16:11:13] "PUT /purchase/1 HTTP/1.1" 200 -
```

DB proof (Stock becomes 2):



The screenshot displays three terminal windows from Oracle VM VirtualBox, showing the execution of a Flask application. The top-left window (dos3) shows the initial setup and the first purchase request. The top-right window (dos2) shows the application's response to the purchase request. The bottom window (dos) shows the application's response to the purchase request.

```
hasan3@hasan3-VirtualBox: ~/code
hasan3@hasan3-VirtualBox:~/code$ python3 ClientServer.py
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Serving Flask app "ClientServer" (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
* Running on http://192.168.1.133:6000/ (Press CTRL+C to quit)
* Restarting with stat
Enter the Catalog IP Server only EX:192.168.1.123
192.168.1.123
Enter the Orderlog IP Server only EX:192.168.1.123
192.168.1.130
* Debugger is active!
* Debugger PIN: 292-348-363
192.168.1.132 - - [11/Nov/2021 16:11:13] "PUT /purchase/1 HTTP/1.1" 200 -
192.168.1.132 - - [11/Nov/2021 16:12:30] "GET /info/1 HTTP/1.1" 200 -

C:\Windows\System32\cmd.exe - Catalog.py
C:\Users\asus\PycharmProjects\Homework\DOS>Catalog.py
* Serving Flask app "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: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 142-517-637
* Running on http://192.168.1.123:8000/ (Press CTRL+C to quit)
192.168.1.133 - - [11/Nov/2021 16:06:10] "B[37mGET /search/distributed%20systems HTTP/1.1B[0m" 200 -
192.168.1.133 - - [11/Nov/2021 16:07:09] "B[37mGET /info/1 HTTP/1.1B[0m" 200 -
192.168.1.130 - - [11/Nov/2021 16:11:12] "B[37mPUT /update/1 HTTP/1.1B[0m" 200 -
192.168.1.133 - - [11/Nov/2021 16:12:29] "B[37mGET /info/1 HTTP/1.1B[0m" 200 -

hasan@hasan-VirtualBox: ~/code
hasan@hasan-VirtualBox:~/code$ python3 Client.py
Enter the IP Server for Client Front Server (ONLY enter the server private IP E
X:192.168.1.123
192.168.1.133
Enter Operation you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
3
Enter ID of book
1
"\Item 1 purchased successfully\`\n"
LibreOfficeWriter you want to perform
1 --> Search by Topic
2 --> Search by ID
3 --> Purchase
4 --> Exit
2
Enter ID of book
1
("ID": 1, 'name': 'RPCs for Noobs', 'topic': 'distributed systems', 'stock': 2,
```

Orderlog proof:

```
136
137 {
138   "ID": 1,
139   "name": "RPCs for Noobs",
140   "topic": "distributed systems",
141   "stock": 2,
142   "cost": 100
143 }
144
```