

بسم الله الرحمن الرحيم

LAB 1: Bazar.com: A Multi-tier Online Book Store

الطالبان:

مطفى كمال سكر

11819630

مطفى عادل حب رمان

11820449

دكتور المساق: د. سامر العرندي

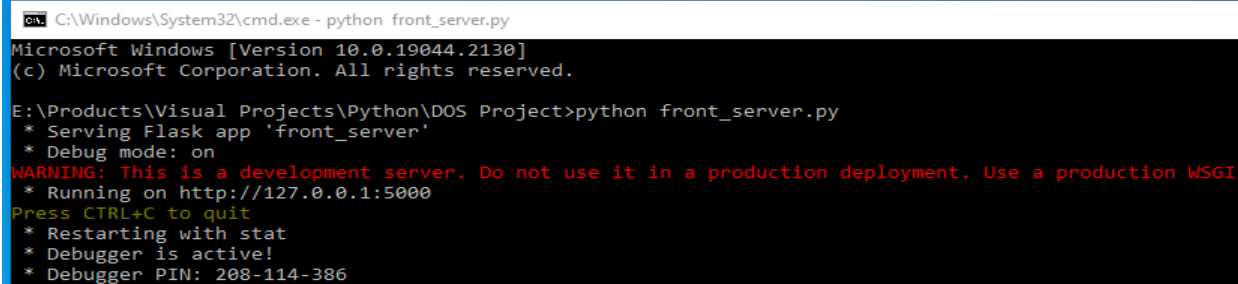
In this problem we create three tiers (servers):

1. Front end server.
2. Order server.
3. Catalog server.

Below is a simplified explanation of each server:

1. Front end server: accept user requests and perform initial processing.
2. Order server: server that i can buy from the bookstore,server maintains a list of all orders received for the books.
3. Catalog server: small database of books .

Front end server: running on host OS (windows)

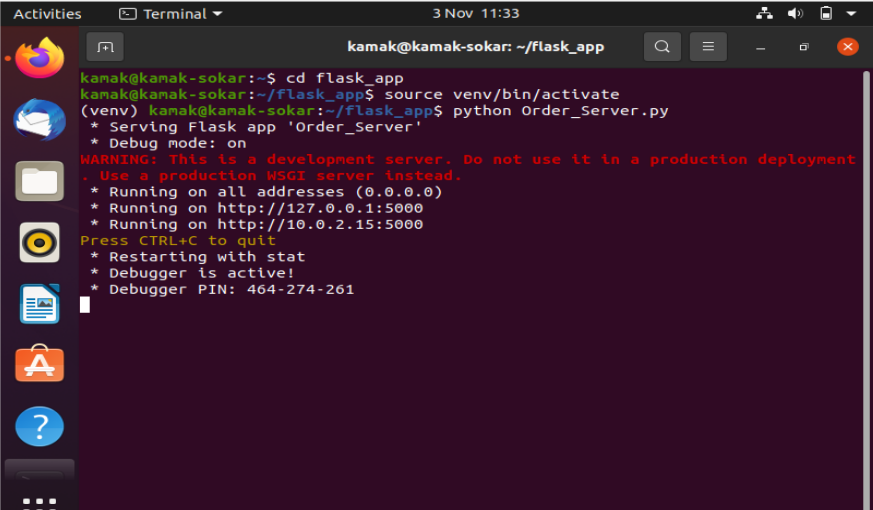
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\System32\cmd.exe - python front_server.py'. The window content shows the following text:

```
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

E:\Products\Visual Projects\Python\DOS Project>python front_server.py
* Serving Flask app 'front_server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 208-114-386
```

Order server: running on gest OS (Ubuntu)

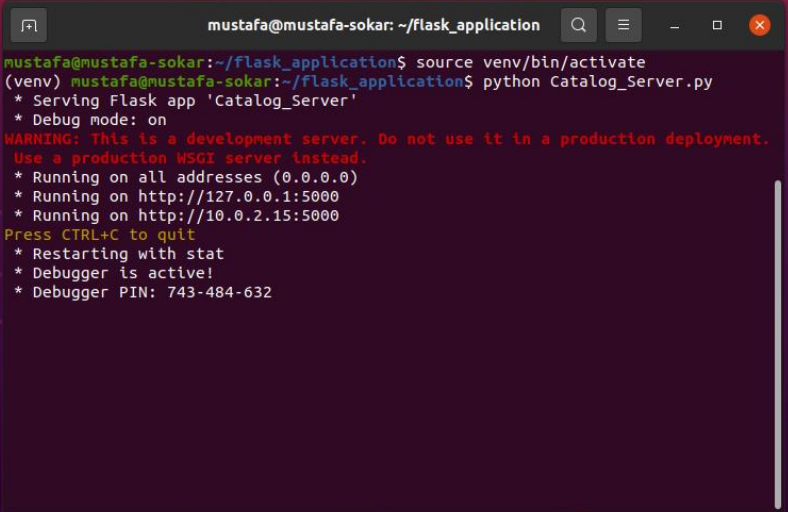
Ubuntu 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help



```
kamak@kamak-sokar:~$ cd flask_app
kamak@kamak-sokar:~/flask_app$ source venv/bin/activate
(venv) kamak@kamak-sokar:~/flask_app$ python Order_Server.py
* Serving Flask app 'Order_Server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment
. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.15:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 464-274-261
```

Catalog server: running on another gest OS (Ubuntu)

ubuntu 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help



```
mustafa@mustafa-sokar:~/flask_application$ source venv/bin/activate
(venv) mustafa@mustafa-sokar:~/flask_application$ python Catalog_Server.py
* Serving Flask app 'Catalog_Server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.15:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 743-484-632
```

In this project we use flask framework (python language), when user need request from book store the front server take this request and if the request:

1. Search about book by topic: then the request go to the Catalog server directly.
2. information about specific book(id): then the request go to the Catalog server directly.
3. Purchase: then the request go to Order server to verify that the item is in stock by querying the catalog server and then decrement the number of items in stock by one. The purchase request can fail if the item is out of stock.

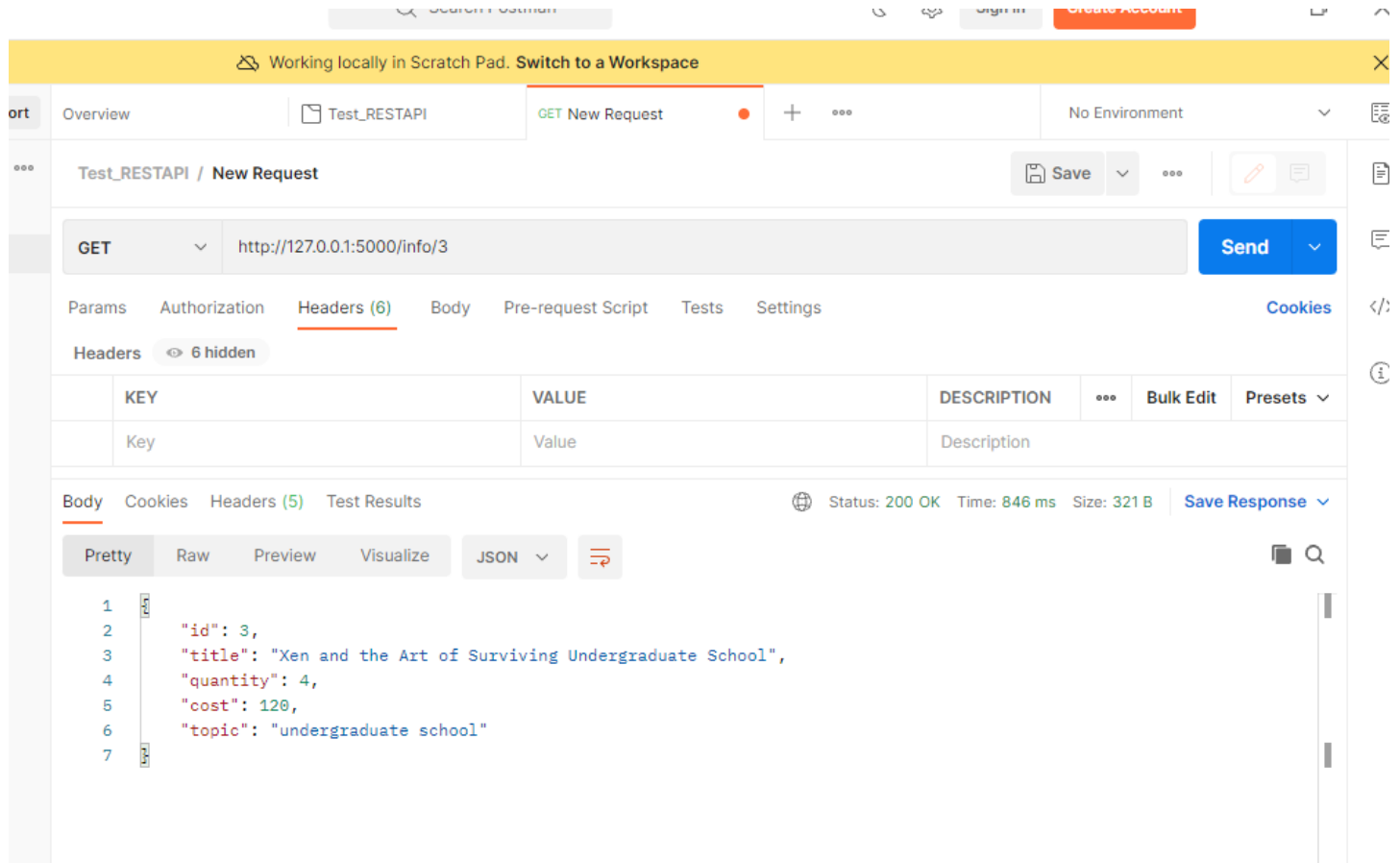
When we send a purchase request to the front server by using postman:

The screenshot shows the Postman interface for a PUT request. The request is sent to `http://127.0.0.1:5000/purchase/3` with a status of 200 OK, a time of 257 ms, and a size of 228 B. The response is displayed in the Body tab, showing a JSON message: `"message": "OK ,Book with id:3 Purchased Successfully"`.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 1
2  "message": "OK ,Book with id:3 Purchased Successfully"
3  1
```

When we send a info request about book with id=3 , to the front server by using postman:



When we send a search request to the front server by using postman:

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/search/distributed systems`. The request is successful, returning a 200 OK status with a response time of 838 ms and a size of 501 B. The response body is displayed in JSON format, showing a list of two items.

Request Details:

- Method: GET
- URL: `http://127.0.0.1:5000/search/distributed systems`
- Headers (6): Key, Value, Description, Bulk Edit, Presets

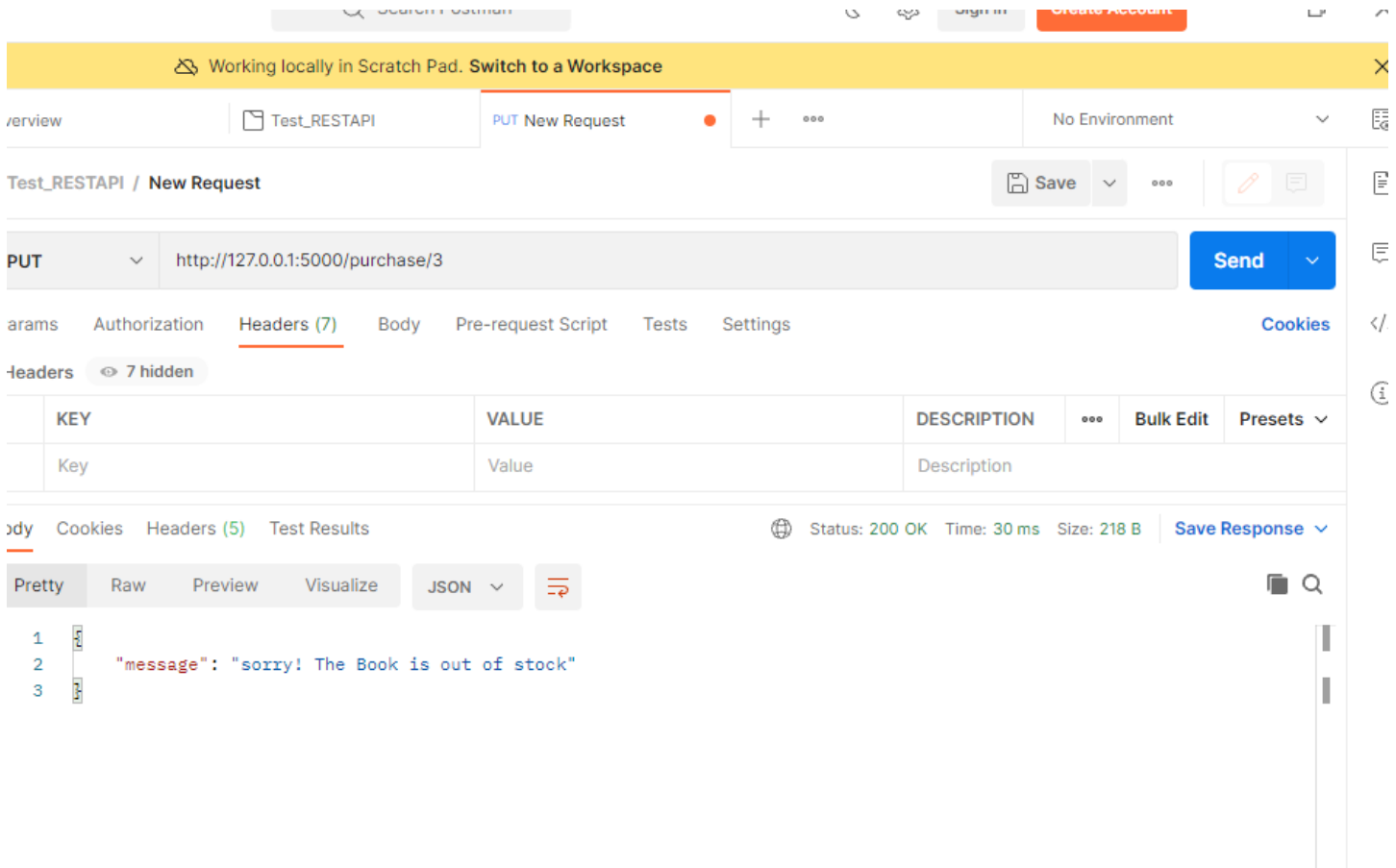
Response Details:

- Status: 200 OK
- Time: 838 ms
- Size: 501 B
- Save Response

Response Body (JSON):

```
1 {
2   {
3     "id": 1,
4     "title": "How to get a good grade in DOS in 40 minutes a day",
5     "quantity": 20,
6     "cost": 100,
7     "topic": "distributed systems"
8   },
9   {
10    "id": 2,
11    "title": "RPCs for Noobs",
12    "quantity": 8,
13    "cost": 70,
14    "topic": "distributed systems"
```

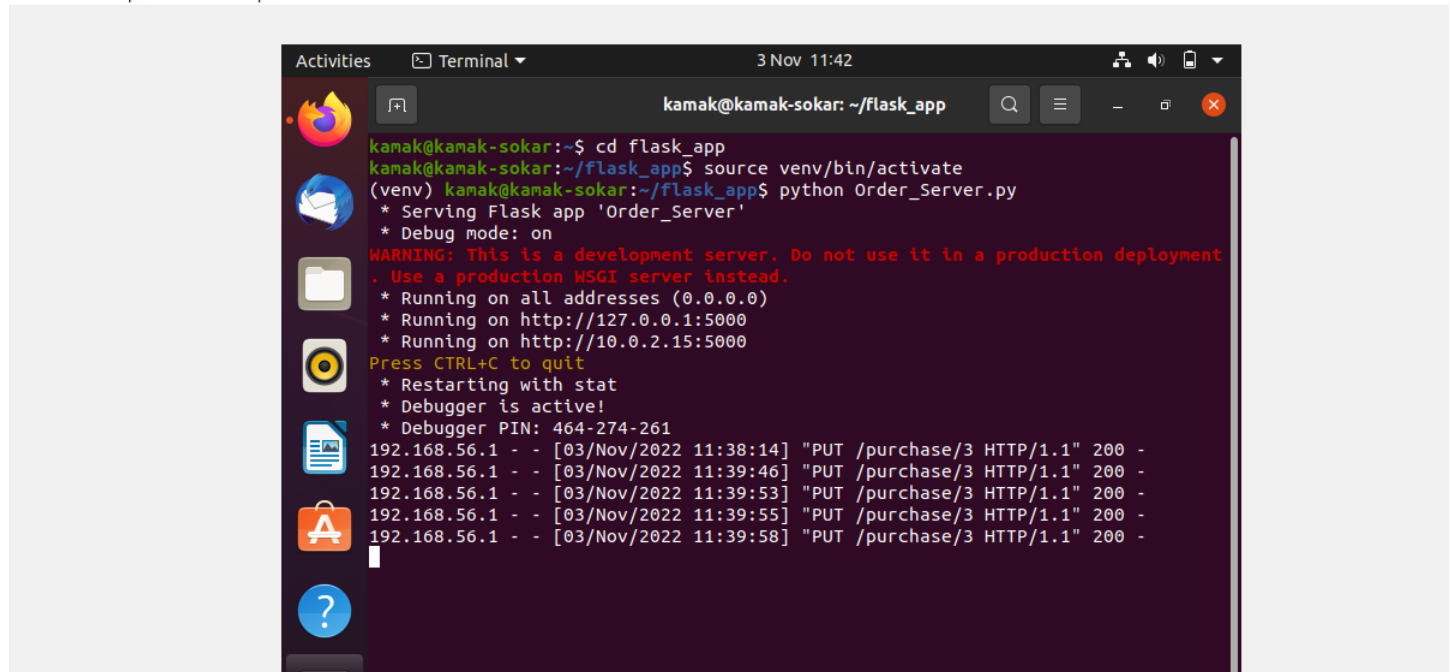
1. When we send a purchase request to the front server by using postman , but the request fail because item is out of stock:



And this receiving requests in servers:

o 1 [Running] - Oracle VM VirtualBox

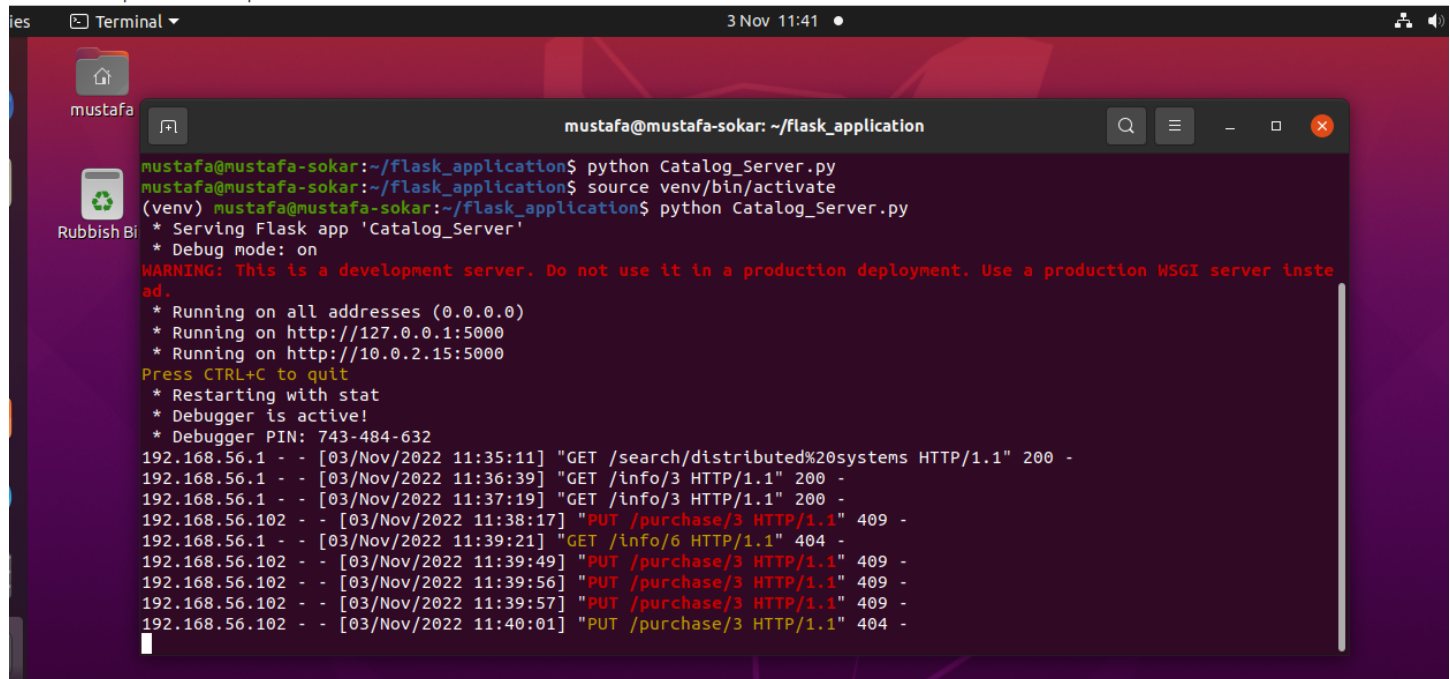
schine View Input Devices Help



```
Activities Terminal 3 Nov 11:42
kamak@kamak-sokar: ~/flask_app
kamak@kamak-sokar:~$ cd flask_app
kamak@kamak-sokar:~/flask_app$ source venv/bin/activate
(venv) kamak@kamak-sokar:~/flask_app$ python Order_Server.py
* Serving Flask app 'Order_Server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.15:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 464-274-261
192.168.56.1 - - [03/Nov/2022 11:38:14] "PUT /purchase/3 HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:39:46] "PUT /purchase/3 HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:39:53] "PUT /purchase/3 HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:39:55] "PUT /purchase/3 HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:39:58] "PUT /purchase/3 HTTP/1.1" 200 -
```

u 2 [Running] - Oracle VM VirtualBox

schine View Input Devices Help



```
ies Terminal 3 Nov 11:41
mustafa@mustafa-sokar: ~/flask_application
mustafa@mustafa-sokar:~/flask_application$ python Catalog_Server.py
mustafa@mustafa-sokar:~/flask_application$ source venv/bin/activate
(venv) mustafa@mustafa-sokar:~/flask_application$ python Catalog_Server.py
* Serving Flask app 'Catalog_Server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.15:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 743-484-632
192.168.56.1 - - [03/Nov/2022 11:35:11] "GET /search/distributed%20systems HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:36:39] "GET /info/3 HTTP/1.1" 200 -
192.168.56.1 - - [03/Nov/2022 11:37:19] "GET /info/3 HTTP/1.1" 200 -
192.168.56.102 - - [03/Nov/2022 11:38:17] "PUT /purchase/3 HTTP/1.1" 409 -
192.168.56.1 - - [03/Nov/2022 11:39:21] "GET /info/6 HTTP/1.1" 404 -
192.168.56.102 - - [03/Nov/2022 11:39:49] "PUT /purchase/3 HTTP/1.1" 409 -
192.168.56.102 - - [03/Nov/2022 11:39:56] "PUT /purchase/3 HTTP/1.1" 409 -
192.168.56.102 - - [03/Nov/2022 11:39:57] "PUT /purchase/3 HTTP/1.1" 409 -
192.168.56.102 - - [03/Nov/2022 11:40:01] "PUT /purchase/3 HTTP/1.1" 404 -
```