



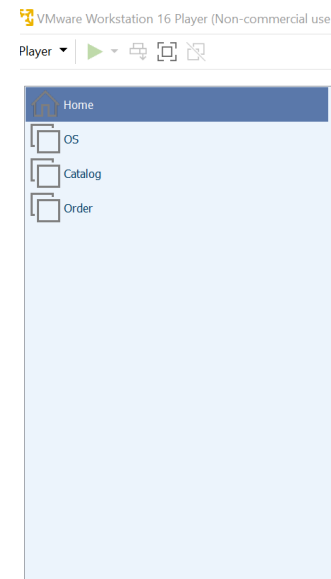
**Computer Engineering Department**  
**Distributed and Operating Systems**  
**Executing Document**

Shima Osamah	11716299
Shatha Abu Hanesh	11717771

## How to Run our Design?

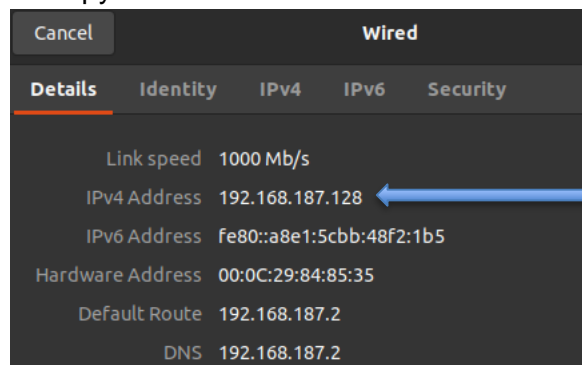
### 1. Create three virtual machines:

- OS machine will contain frontend services.
- Catalog will handle query services and DB.
- Order will handle purchase service.

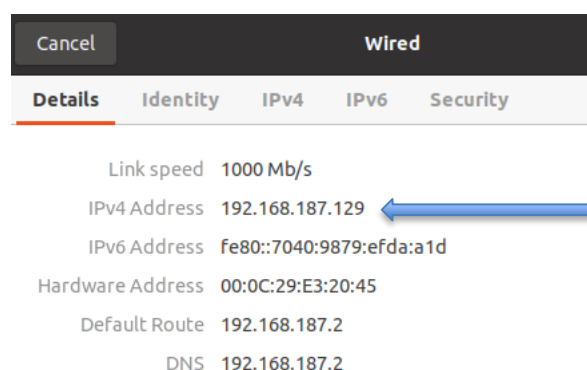


2. On each machine there will be its python implementation (each VM will have one folder from our repository).
3. Now we have to change the host address of each python file:  
\*host address is the IPv4 for each machine.

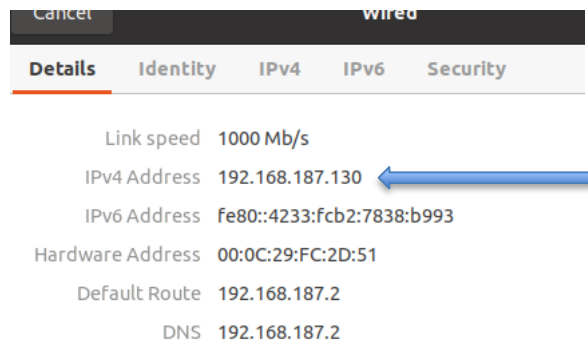
- o main.py in front folder.



- o order.py in order folder.



- query.py in catalog folder.



- \* On each file we will replace the host value with its IPv4 address.  
For example, in main.py the host address will be like this:

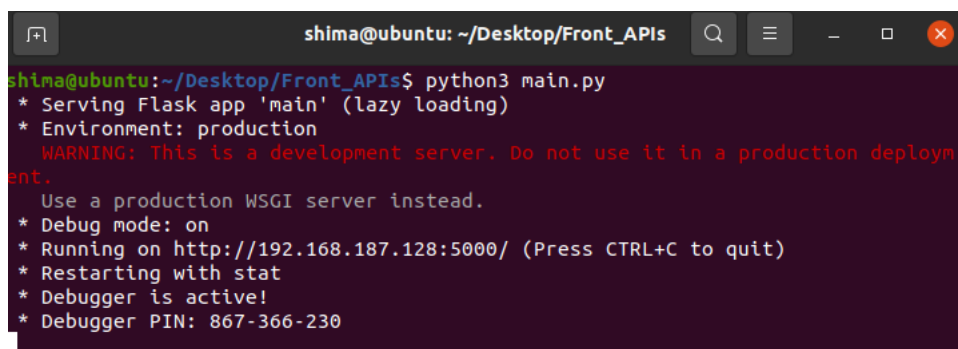
```
51 |
52 | if __name__ == "__main__":
53 |     app.run(debug=True, host="192.168.187.128")
```

4. more modification in main.py and order.py:

- in main.py:
  - ⇒ lines (23,28,33,38) the request address must be as the host address in catalog VM.
  - ⇒ line 34 the request address must be as the host address in order VM.
- In order.py:
  - ⇒ Lines (23,31) the request address must be as the host address in catalog VM.

5. Finally, we will run the three python files each on its VM.

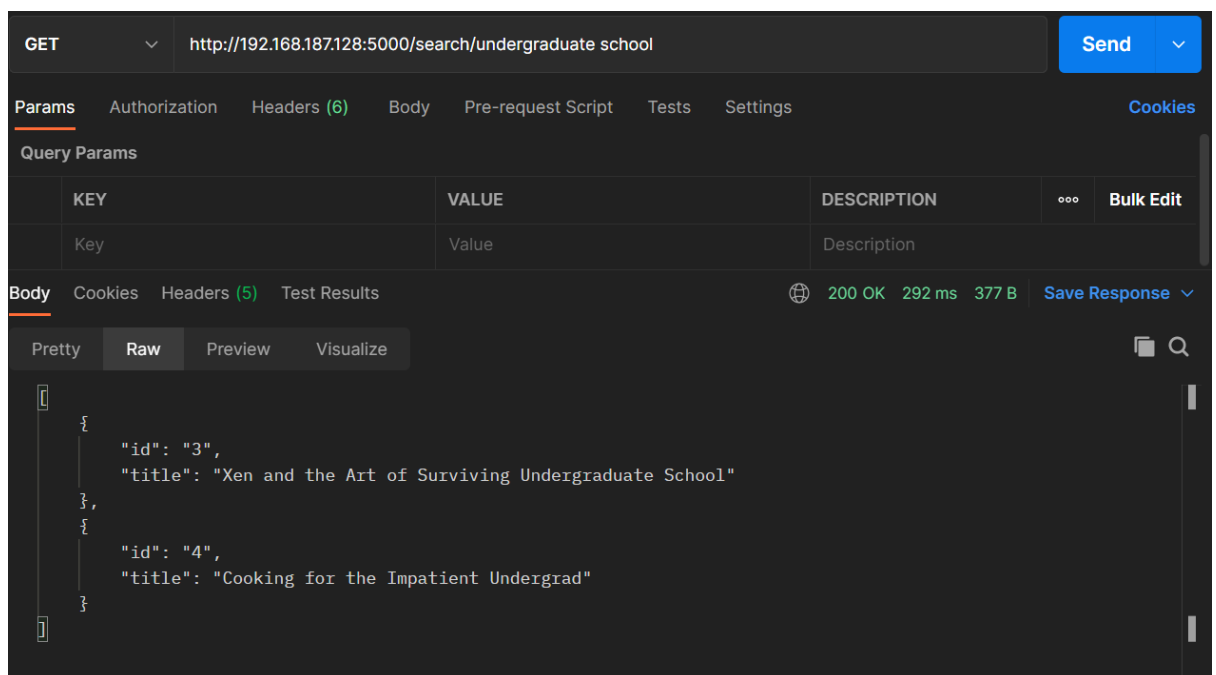
\*make sure that (flask, flask\_cors, and pandas) are installed. If not, then simply run (pip install <lib\_name>) on the terminal.



```
shima@ubuntu: ~/Desktop/Order
shima@ubuntu:~/Desktop/Order$ python3 order.py
* Serving Flask app 'order' (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://192.168.187.129:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 650-192-507
```

```
shima@ubuntu: ~/Desktop/Catalog
shima@ubuntu:~/Desktop/Catalog$ python3 query.py
* Serving Flask app 'query' (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://192.168.187.130:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 957-460-075
```

6. On Postman we will try to buy a book:



GET

http://192.168.187.128:5000/info/4

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK 117 ms 305 B

Save Response

Pretty

Raw

Preview

Visualize

```
[
  {
    "title": "Cooking for the Impatient Undergrad",
    "quantity": "2",
    "price": "35"
  }
]
```

PUT

http://192.168.187.128:5000/purchase/4

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK 148 ms 233 B

Save Response

Pretty

Raw

Preview

Visualize

```
bought book Cooking for the Impatient Undergrad
```