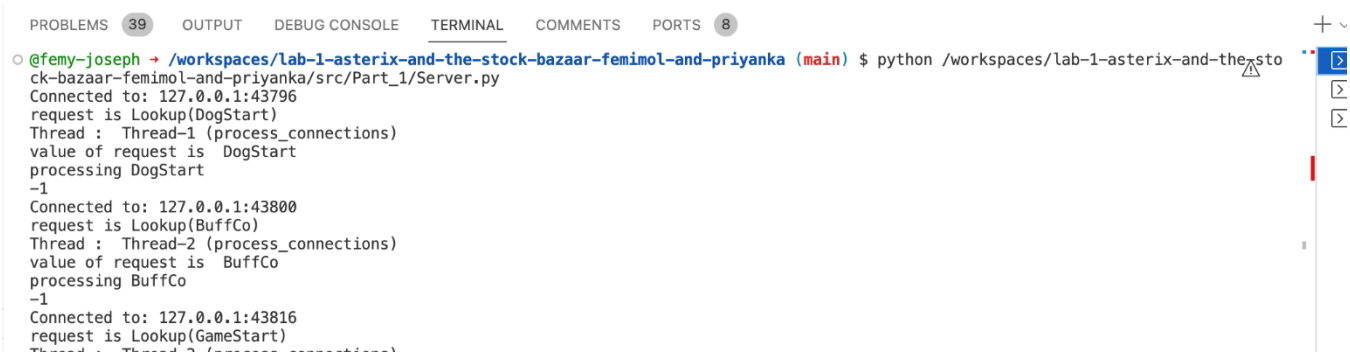
**PART 1**

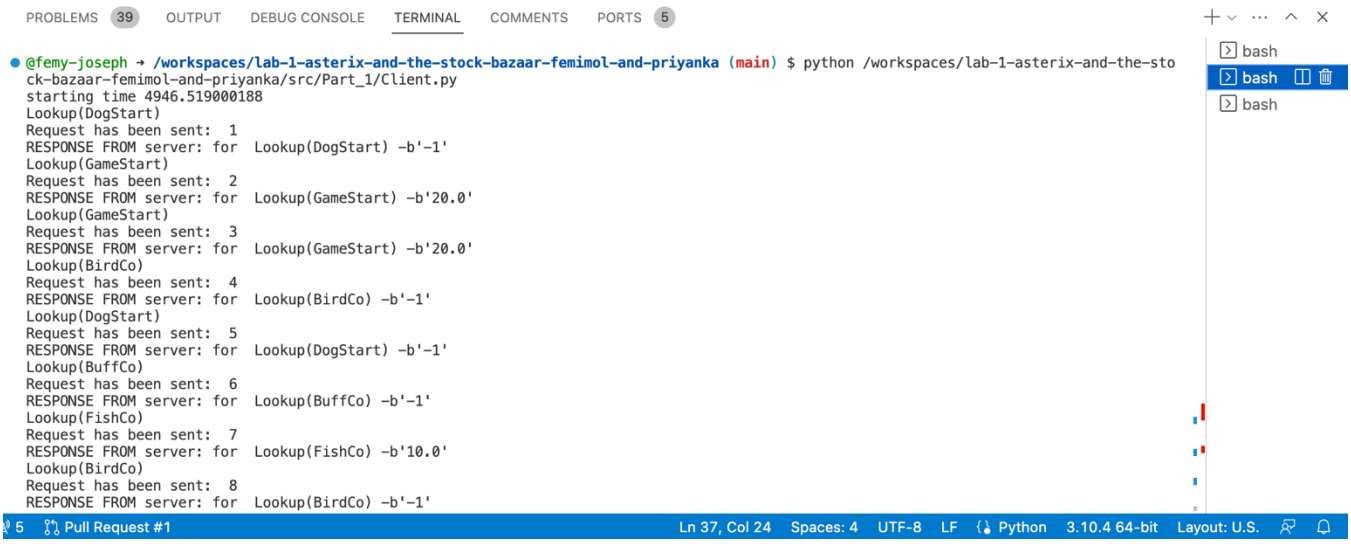
1. Server.py




Server starts given worker threads on startup and keeps polling for requests. Once the server accepts client connection the request is put to the queue which is then taken by the idle thread to process.

The above screenshot shows the idle threads processing the requests – DogStart, BuffCo and GameStart from the queue.

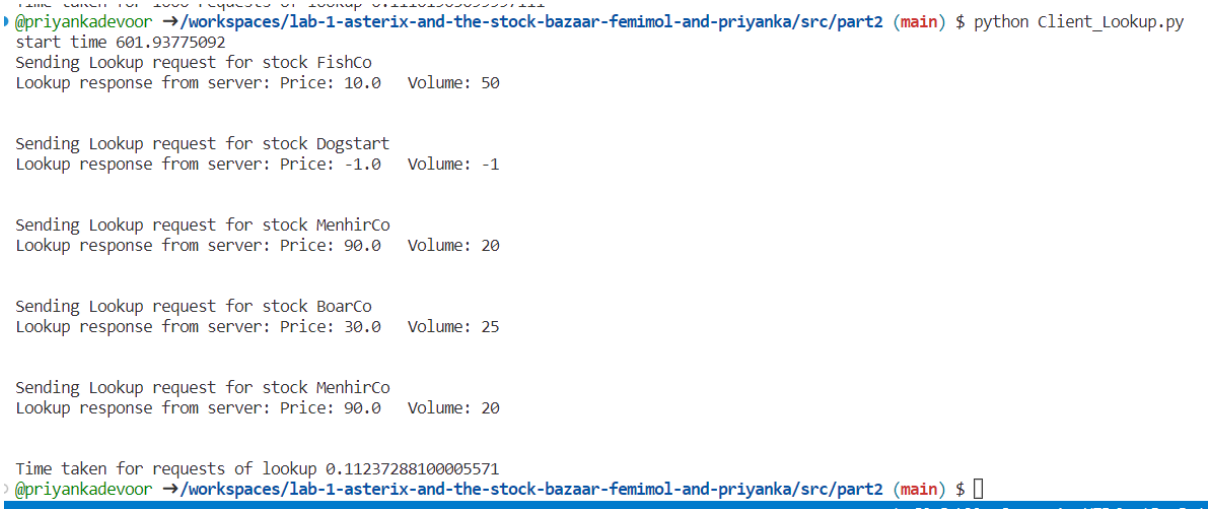
2. **Client.py**




Client sending multiple request from a for loop by randomly picking the stock names from list. The response from the server is also printed at the client side.

**PART 2**

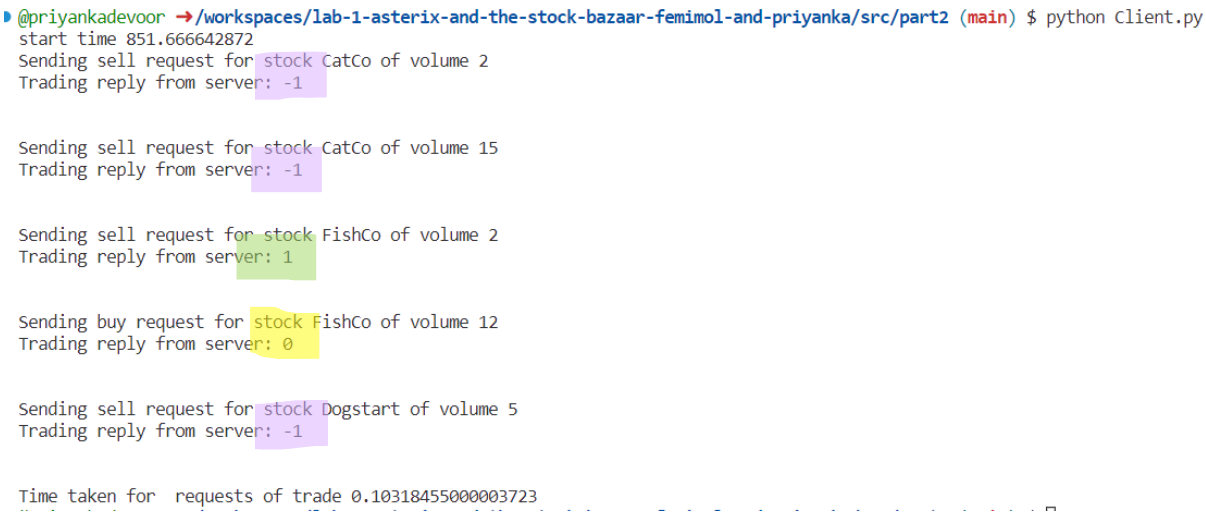
### 1. Single client making lookup requests




Client_Lookup python file sends requests to the server by randomly picking the stocks given in the stock list. If the stock is present in stock bazar the server returns its respective stock price and volume. If it is an invalid stock name then it returns -1.

Here, the output shows 4 positive use cases and 1 negative use case which is DogStart stock which is not present in the stock bazaar.

### 2. Single client making trade requests

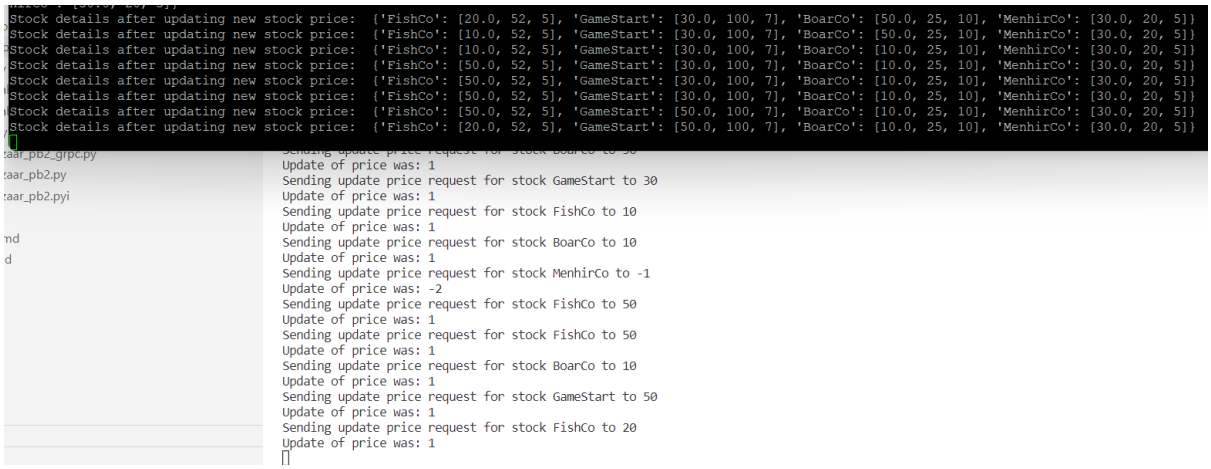



Client.py sends only trading requests to the server. The attached screenshot shows 3 use cases where if the buy/sell of stock request is present in the stock bazaar then the server updates the bazaar and responds to the client as 1 meaning the trade was successful.

If trade request is made for a stock which is not present in bazaar then -1 is returned.

If buy/sell of stock is made more than the max fixed volume then 0 is sent back to the client meaning the trade was suspended.

### 3. Client updating stock prices every 3sec




Client_update.py sends stock price update request periodically by picking random stocks from the stock list and a random price. If the stock price was updated successfully in the server then it returns 1. If a negative/zero stock price update was sent by the client then the sever responds with -2.

White terminal -> Client sending price update requests.

Black terminal -> Server printing updated prices.

### 4. Multi client requests to server

```
@priyankadevoor →/workspaces/lab-1-asterix-and-the-stock-bazaar-femimol-and-priyanka/src/part2 (main) $ sh run_parallel_lookup.sh
start time 1708.113916796
Sending Lookup request for stock Dogstart
start time 1708.11794197
Sending Lookup request for stock MenhirCo
start time 1708.131980731
Sending Lookup request for stock BoarCo
Lookup response from server: Price: -1.0   Volume: -1


Sending Lookup request for stock FishCo
Lookup response from server: Price: 30.0   Volume: 22


Sending Lookup request for stock GameStart
Lookup response from server: Price: 20.0   Volume: 52


Sending Lookup request for stock Dogstart
Lookup response from server: Price: 10.0   Volume: 28


Sending Lookup request for stock Dogstart
Lookup response from server: Price: 10.0   Volume: 100


Sending Lookup request for stock GameStart
Lookup response from server: Price: -1.0   Volume: -1
```

Run_parallel_lookup.sh script will start the multiple clients parallelly thereby avoiding manual delay of multiple client startup.

Above screenshot shows response for 3 client connections to the server.