

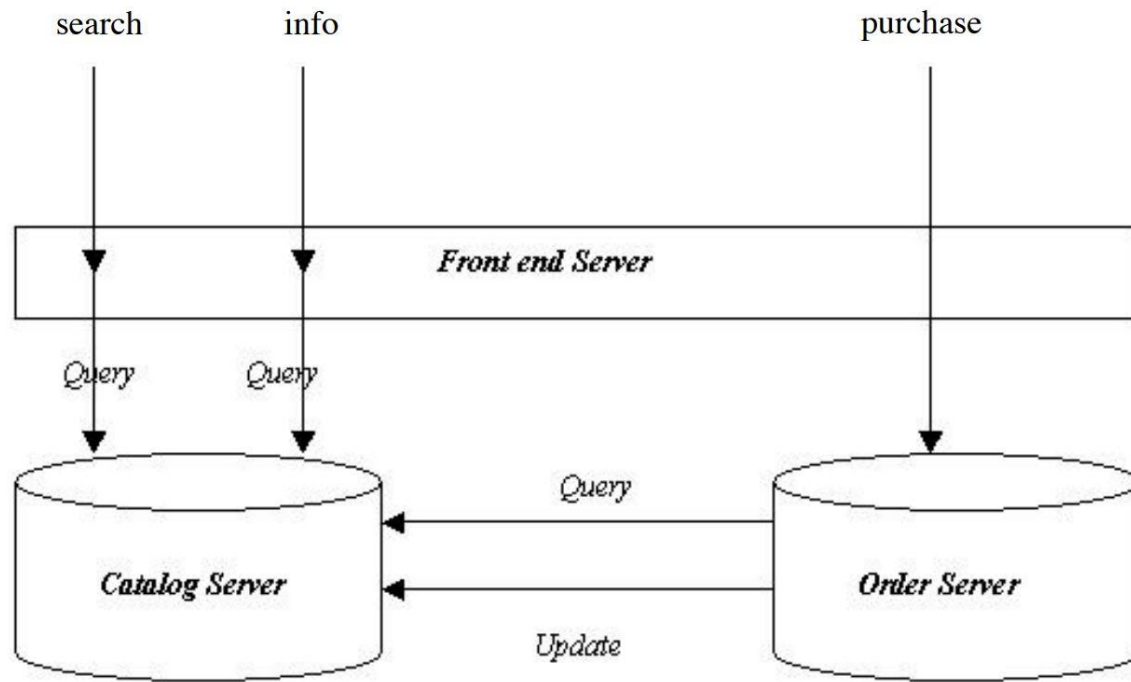
# Bazar.com

A Multi-tier Online Book Store

DOS PROJECT - LAB 1

WALA JANAJREH

## System Representation



## Servers implemented :

- Catalog server
- Order server
- Front end server

The catalog server keeps the books available in a json file

Each book is stored as an entity (id,title,topic,quantity ,price)

The catalog server supports two operations: query and update.

Two types of queries are supported: query-by-topic and query-by-id. In the first case, a topic is specified and the server returns all matching entries. In the second case, an item id is specified and all relevant details are returned.

# Catalog server

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace New Import

Collections + Postman Echo

APIs

Environments

Mock Servers

Monitors

History

GET http://127.0.0.1:5000/books?topic=distributed%20systems

No Environment

Save

Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	topic	distributed%20systems			
	Key	Value	Description		

Body 200 OK 48 ms 254 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "title": "How to get a good grade in DOS in 40 minutes a day."
5   },
6   {
7     "id": 2,
8     "title": "RPCs for Noobs."
9   }
10 }
```

Find and Replace Console

Bootcamp Runner Trash

Testing the  
get by topic  
search using  
postman  
case1 :  
ok ,200

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

My Workspace New Import

GET http://127.0.0.1:50...

No Environment

Save

Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	topic	algorithms			
	Key	Value	Description		

Body

500 INTERNAL SERVER ERROR 26 ms 463 B Save Response

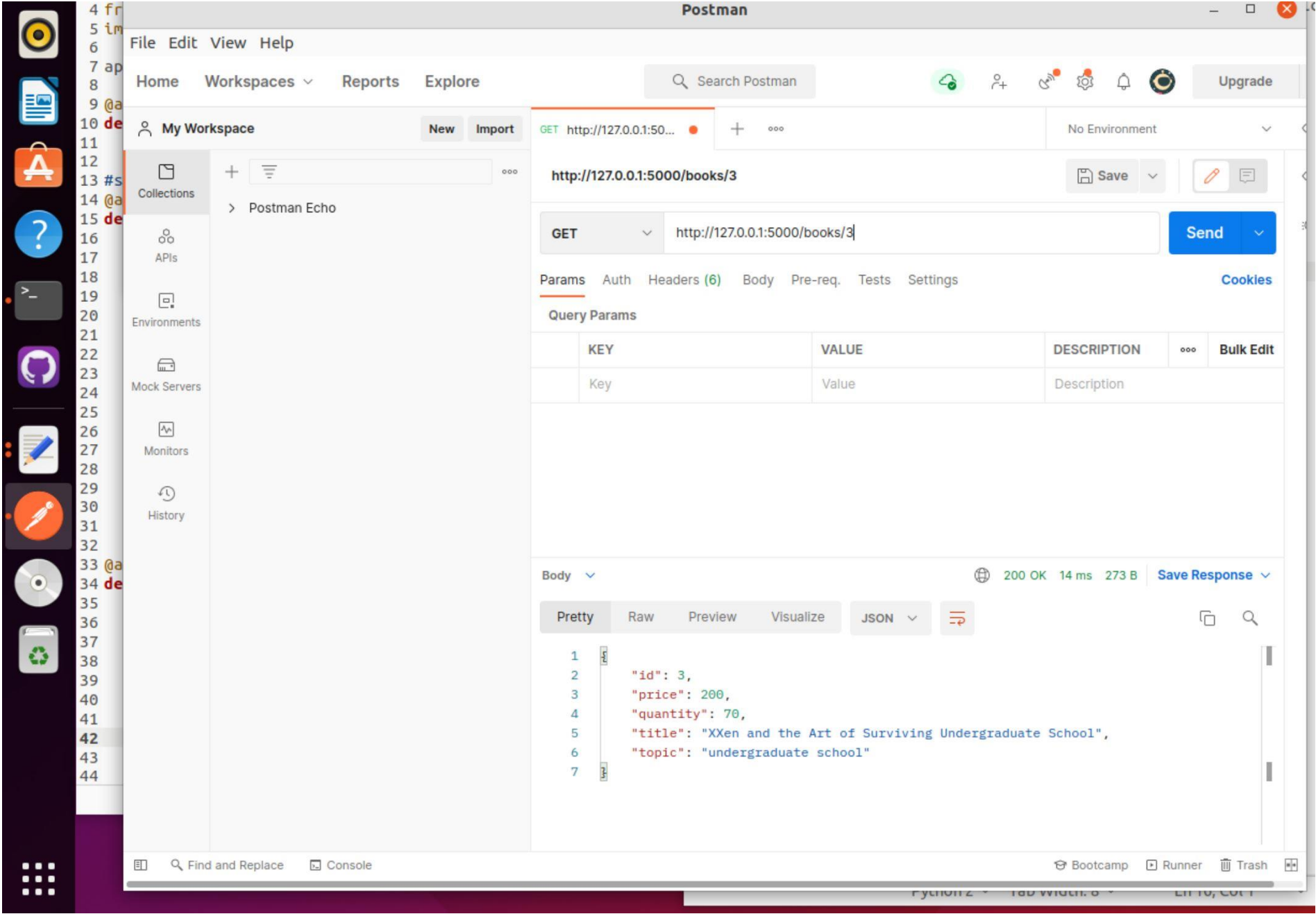
Pretty Raw Preview Visualize HTML

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <title>500 Internal Server Error</title>
3 <h1>Internal Server Error</h1>
4 <p>The server encountered an internal error and was unable to complete your
  request. Either the server is overloaded or
  there is an error in the application.</p>
```

CASE2 :  
searching for  
a non-existing  
topic such as  
algorithms

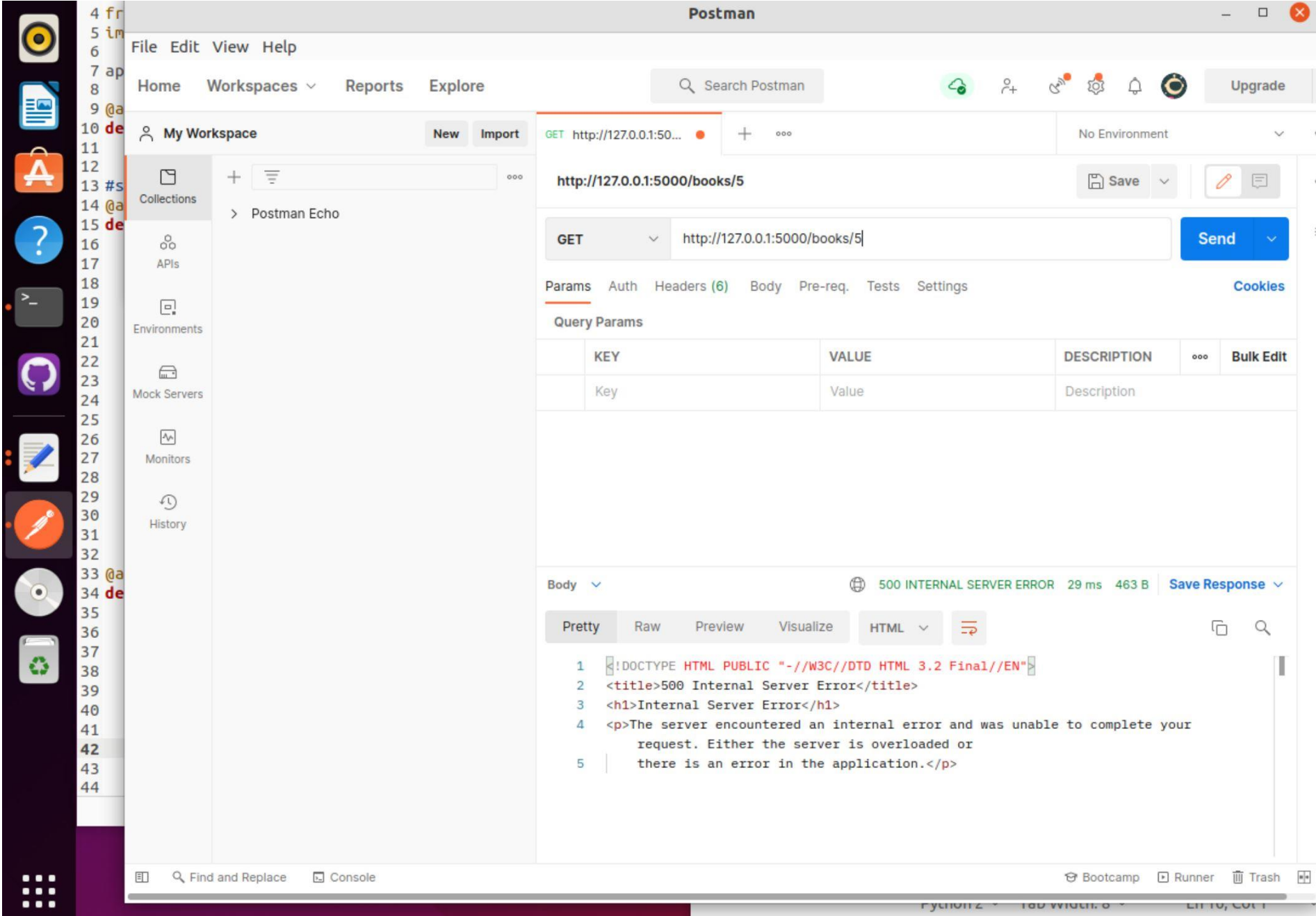
# SEARCHING BY ID

CASE 1 : EXISTING ID



# SEARCHING BY ID

CASE 2 : NON-EXISTING ID



It supports a single operation:  
`purchase(item_number)`.

Upon receiving a purchase request, the order server must first 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.

Order Server



The front-end server handles the client requests then returns the response with a friendly format instead of the received JSON response

It supports three operations:

- search(topic
- info(item\_number)
- purchase(item\_number)

Front End Server

Virtual machines

I created 3 virtual machine instances  
each machine runs one of the servers

Then used the ip addresses in the GET, POST, PUT URLs