An-Najah National University

Faculty of Engineering
Computer Engineering Department

Distributed Operating Systems Project Part 1 Report

Alia Tuqan & Ahmad Assi

This program was built using Django web framework, we chose it to earn some skills and get more familiar with it as it's required in the marketplace.
The program mainly consists of models to deal with databases as objects in Django.

The URL is used to control views, which are the first thing affected after hitting URLs, the serializer's main goal is to format the JSON response.

We also used Postman which is a collaboration platform for API development.

The design mainly consists of 1 database shared by two separate servers, the first server is the catalog server for managing queries like:info and search.
The second one is the order server which receives triggered purchase queries .

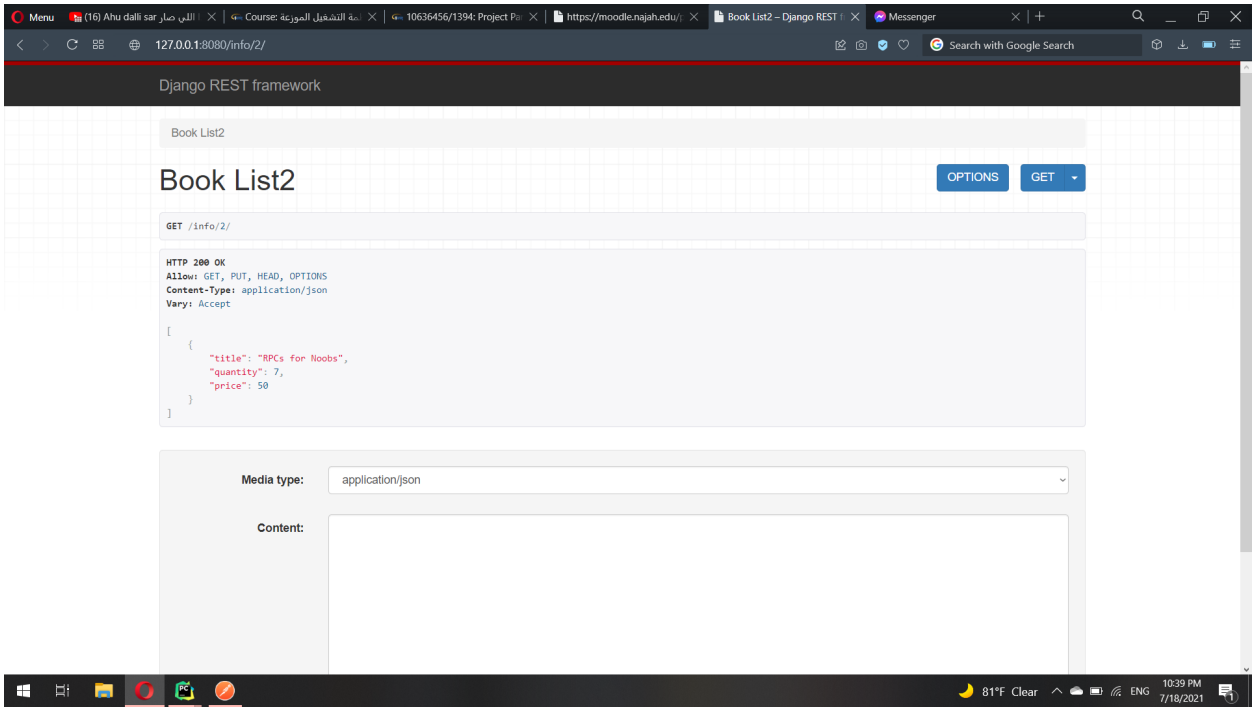
The main idea is that the second server hits APIs to the first server to get the desired purchase results. After hitting, data is fetched from the first server and changes are performed on the database stored items.

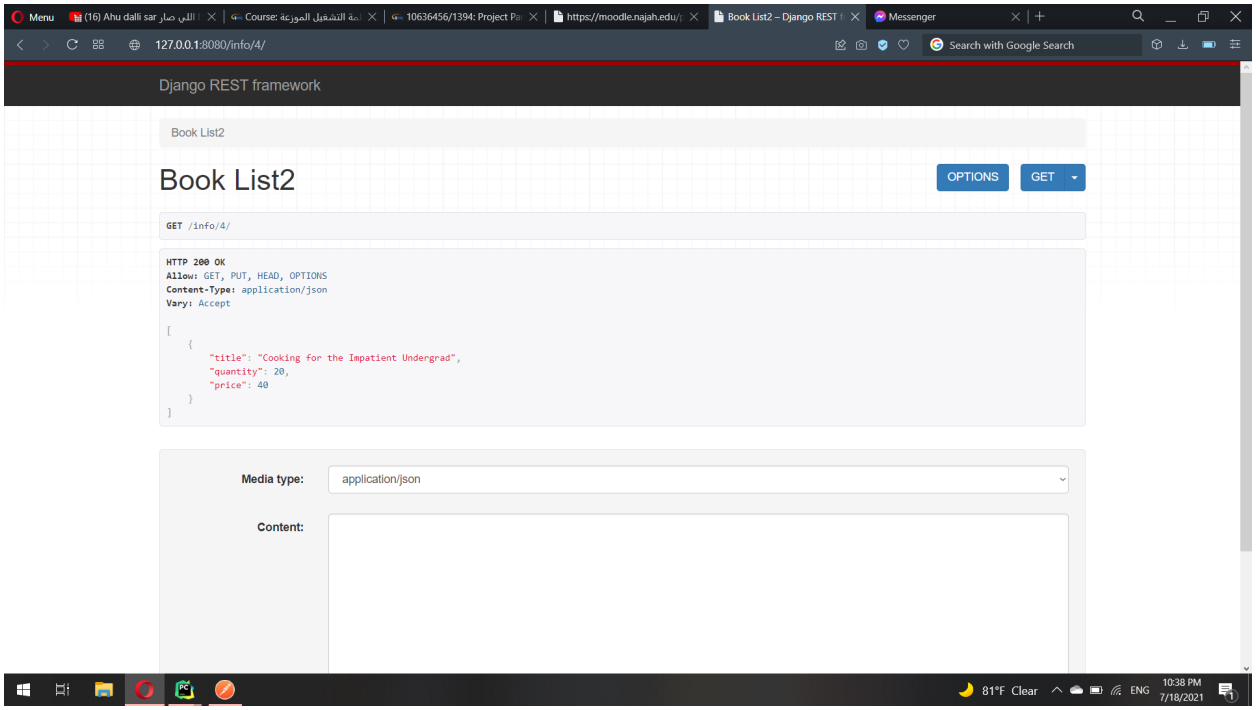
The program won't run unless server 1 is run on port 8080 and server 2 is run on port 8000

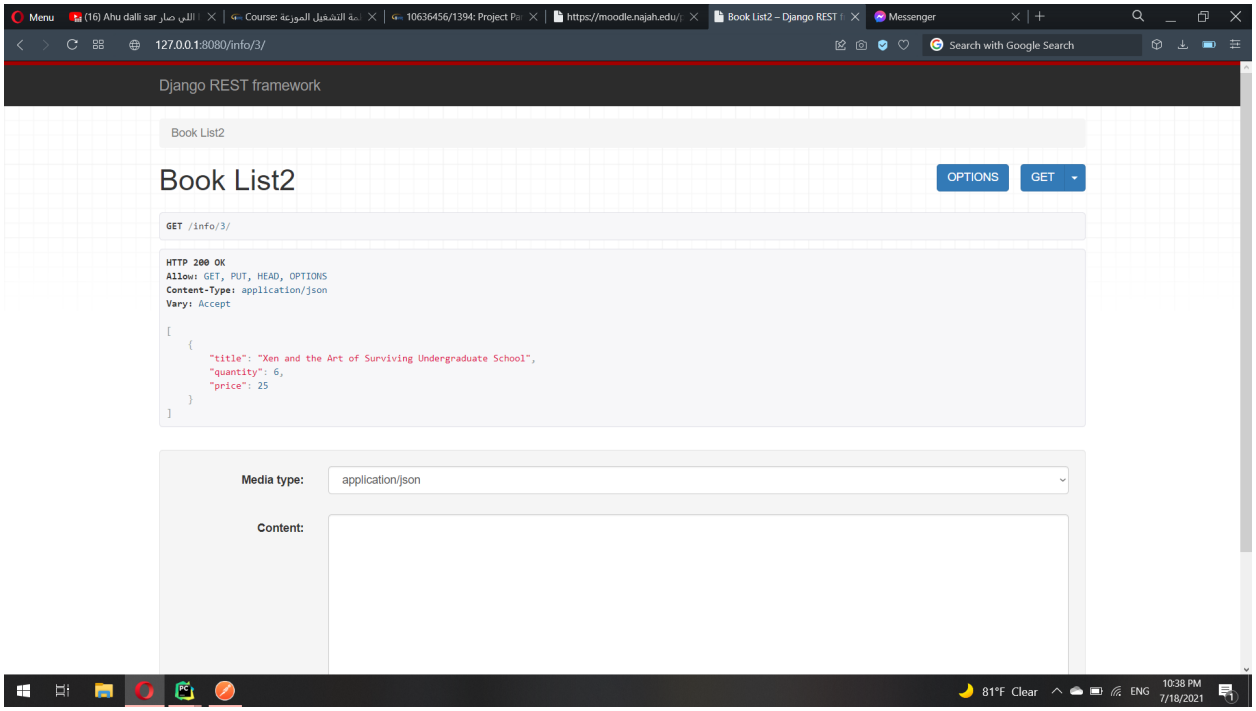
So this port specification makes the program functionality limited and creates the possibility of not running in all scenarios.

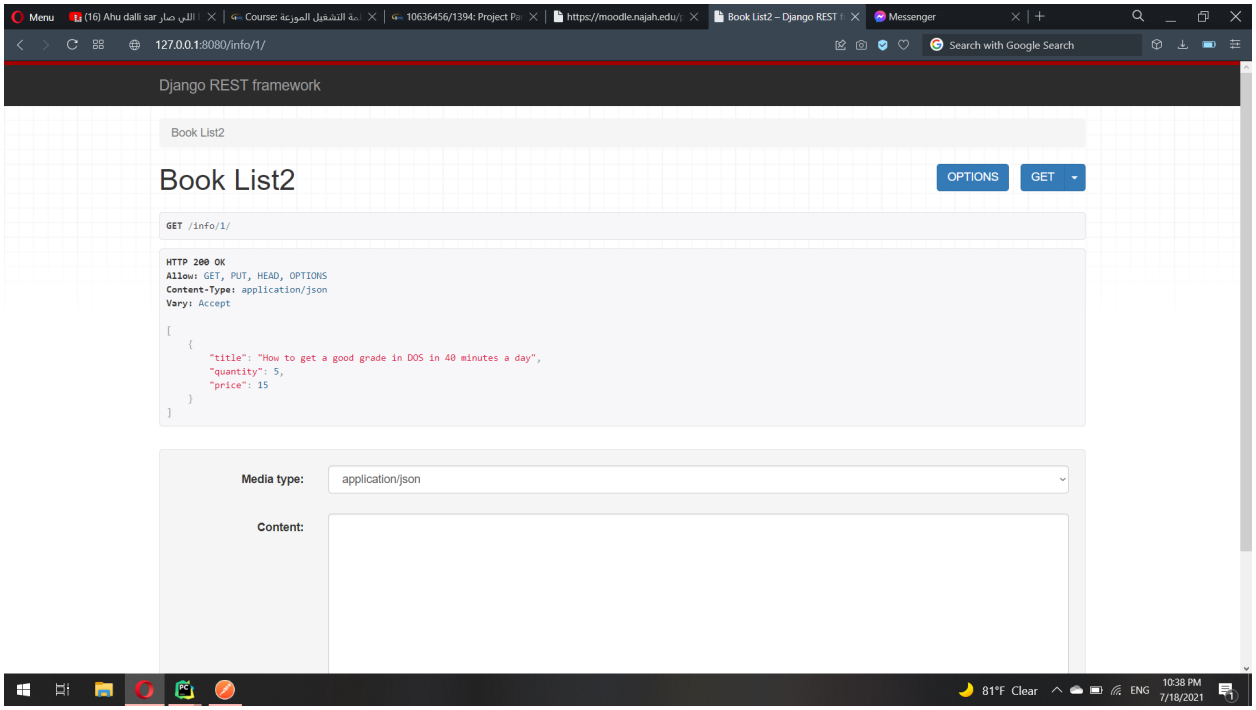
We also used Django restframework library to make it easier to build restful APIs, we also used the SQLite DB as it's lightweight.

Below are some screenshots of the program:
First we'll start by showing the results of requestig the info of each book.

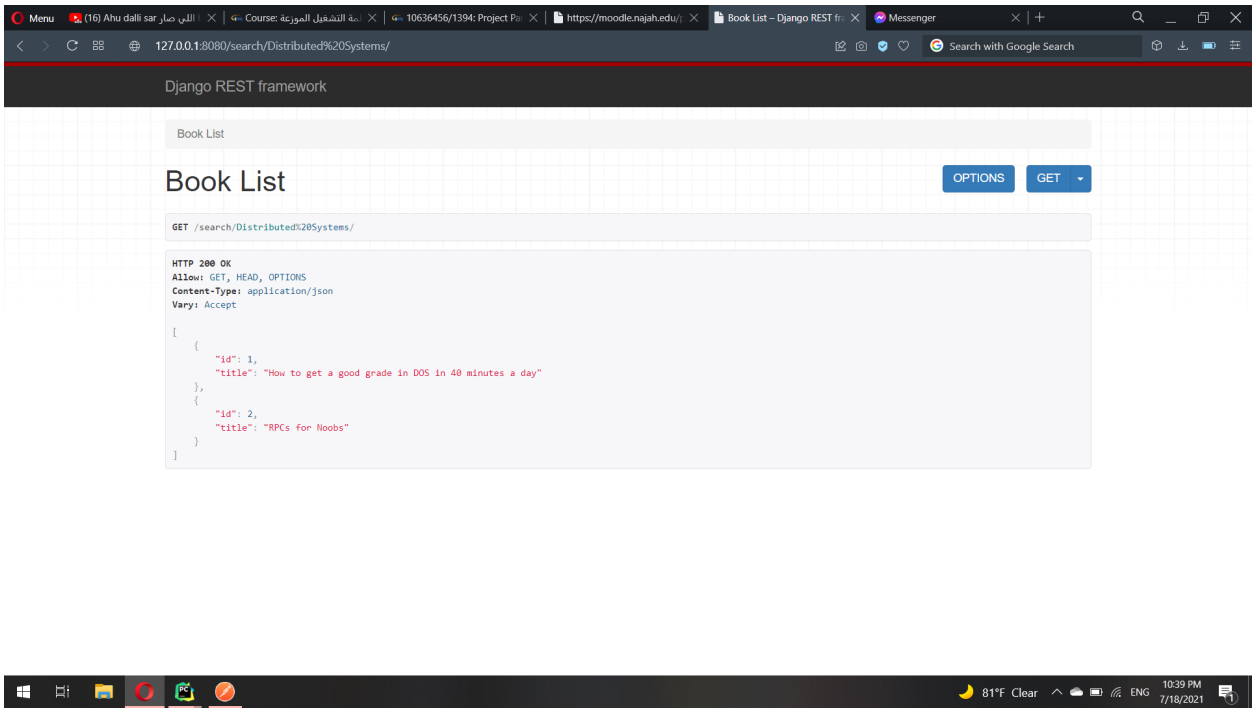


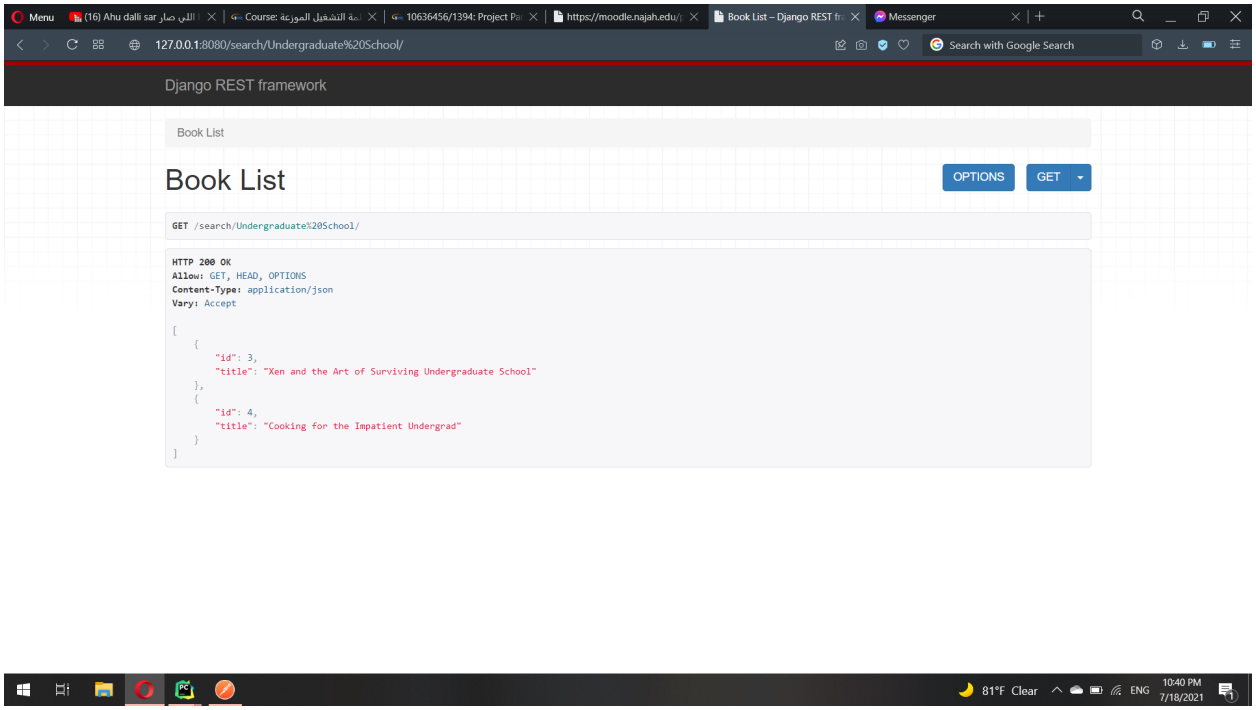

Menu
Book List2 – Django REST
Messenger
127.0.0.1:8080/info/4/
Search with Google Search
Django REST framework
Book List2
Book List2
OPTIONS
GET
GET /info/4/
HTTP 200 OK
Allow: GET, PUT, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[
{
"title": "Cooking for the Impatient Undergrad",
"quantity": 20,
"price": 40
}
]
Media type:
application/json
Content:
81°F Clear
ENG
10:38 PM
7/18/2021


Menu
Book List2 – Django REST
Messenger
127.0.0.1:8080/info/3/
Search with Google Search
Django REST framework
Book List2
Book List2
OPTIONS
GET
GET /info/3/
HTTP 200 OK
Allow: GET, PUT, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[
{
"title": "Xen and the Art of Surviving Undergraduate School",
"quantity": 6,
"price": 25
}
]
Media type:
application/json
Content:
81°F Clear
ENG
10:38 PM
7/18/2021

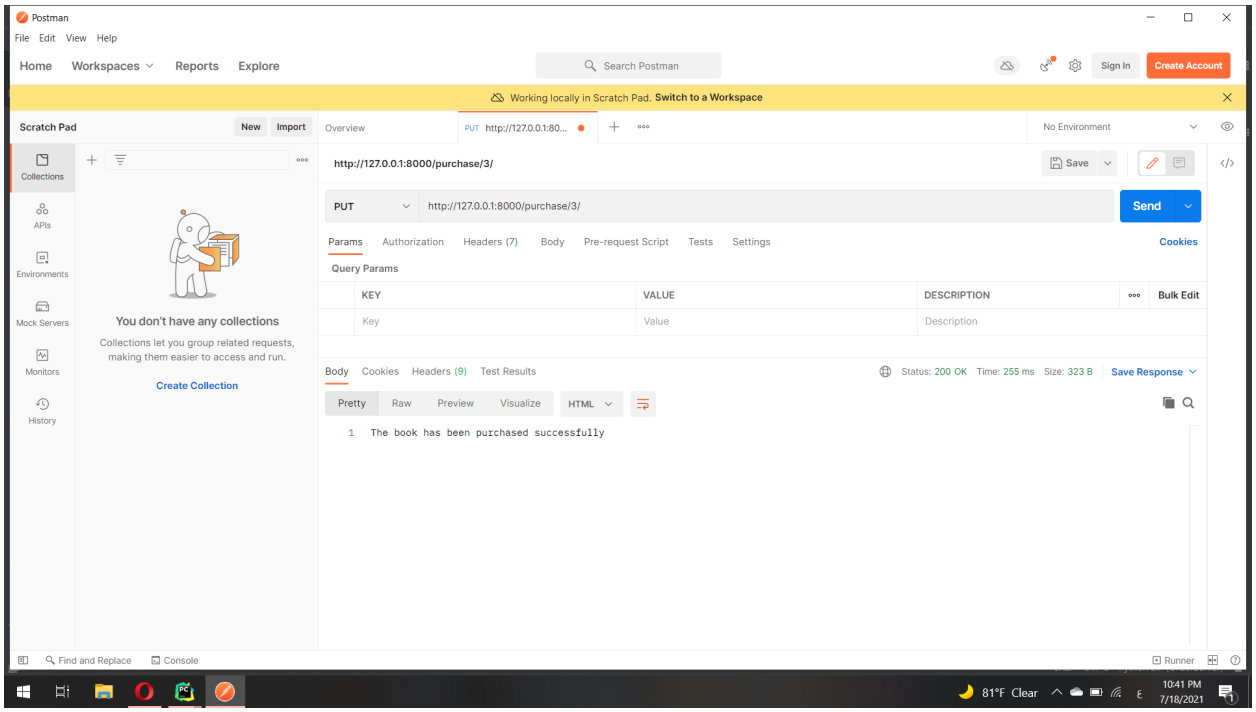


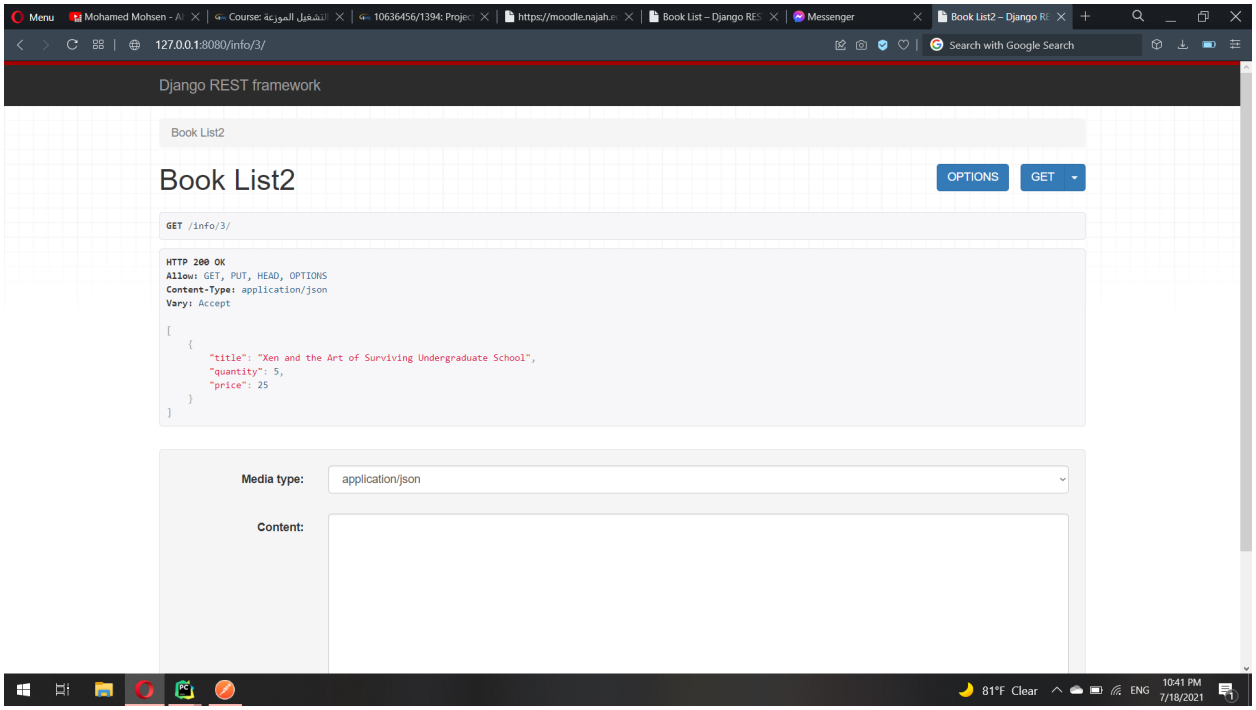
Then we will search for books using the topic of the book





Then we will perform a purchase operation on book id=3





We can see that the quantity has been decreased by 1.