

# DOS

Ola Awwad  
Takreem Abdullah

November 2020

## 1 project link on Github

<https://github.com/oulaawwad/DOS-NEW.git>

## 2 Output

In my system i used lumen framework using php langouge on ubuntu operating system using three virtual machines. We implement three services as wanted:

- Front-end service
- Order service
- Cataloge service

There are 3 operations and i will explain each one and how to works:

search

after sending the query from front-end controller service to the the cataloge service then the output will shown in the browser page with the book information : name and id of all books.

```

47 //2- search
48 public function getAllBooks($item){
49
50     // make a query
51     $startQuery2 = microtime(true);
52     if (cache::has($item))
53     {
54         $valueBook = cache::get($item);
55         $endCache2 = microtime(true);
56         echo "time of cache:" . ($endCache2 - $startQuery2);
57         return response()->json(json_decode($valueBook));
58     }
59
60     $url = "";
61     $randomum = rand(1,4);
62     if ($randomum < 2){
63         $url = "http://192.168.12.141:8888/query/booktopic/" . $item;
64     }
65     else {
66         $url = "http://192.168.12.141:8881/query/booktopic/" . $item;
67     }
68
69     ///////////////////////////////////////////////////
70     // get the topics from catalogue
71     $page = file_get_contents($url);
72     cache::put($topic . $page ,800);
73     $end = microtime(true);
74     echo "coget from cache:" . ($end - $startQuery2);
75     return response()->json(json_decode($page));
76
77 }

```

Figure 1: search/ front end controller

```

17 // show the specific book
18 public function showBook($topic){
19
20     // get the books information from text file with name books, its in catalogue folder
21     $file = new Illuminate\Filesystem\Filesystem();
22     $content = $file->get(_DIR_."/../books.txt");
23     $books = explode("\n",$content);
24     if (sizeof($books) < 2){
25         return response()->json([' ' => 'unfortunatly there is no books.... try later']);
26     }
27
28     $book_desc;
29     $json_inf;
30     $flag = false;
31     $count = -1;
32
33     // to display the books information
34     for ($i=0 ; $i<sizeof($books)-1 ; $i++){
35         $book_desc[$i] = explode(",",$books[$i]);
36         if ($book_desc[$i][3] == $topic){
37             $count++;
38             $flag = true;
39             $json_inf[$count]['Title'] = $book_desc[$i][0];
40             $json_inf[$count]['ID'] = $book_desc[$i][4];
41         }
42     }
43     if (!$flag){
44         return response()->json(['Message' => 'type another book .']);
45     }
46     return response()->json($json_inf);
47 }

```

Figure 2: search/ catalogue controller

lookup

after sending the query from front-end controller service with the id of book to the the catalogue service then the output will shown in the browser page with the book information : name and id of the specific books.

```

13 public function _construct()
14 {
15     ///////////////lookup
16     public function getBook($id){
17         $url = '';
18         $startQuery1 = microtime(true);
19         if (cache::has($id)){
20             $bookValue = cache::get($id);
21             $endCache = microtime(true);
22             echo "cache: " . ($endCache - $startQuery1);
23             return response()->json(json_decode($bookValue));
24         }
25
26         $num = rand(1,4);
27         if ($num < 3)
28         {
29             $url = 'http://192.168.12.141:8080/query/bookId/' . $id;
30         }
31         else
32         {
33             $url = 'http://192.168.12.141:8081/query/bookId/' . $id;
34         }
35
36         // get book from catalogue
37
38         $page = file_get_contents($url);
39         cache::put($id, $page, 3600);
40         $end = microtime(true);
41         echo "done put the book in the cache:" . ($end - $startQuery1);
42         return response()->json(json_decode($page));
43     }
44 }

```

Figure 3: lookup/ front end controller

```

79 //
80
81
82 public function showBook($id){
83     $file = new \Illuminate\Filesystem\Filesystem();
84     $content = $file->get('_DIR' . '/../..../books.txt');
85     $books = explode("\n", $content);
86     if (sizeof($books) < 2){
87         return response()->json(['Message' => 'There is no books in store.']);
88     }
89     $book_details;
90     $json_data;
91     $flag = false;
92     for ($i=0; $i<sizeof($books)-1; $i++){
93         $book_details[$i] = explode(" ", $books[$i]);
94         if ($book_details[$i][4] == $id){
95             $flag = true;
96             $json_data['Title'] = $book_details[$i][0];
97             $json_data['Count'] = $book_details[$i][1];
98             $json_data['Price'] = $book_details[$i][2];
99             $json_data['Topic'] = $book_details[$i][3];
100             $json_data['ID'] = $book_details[$i][4];
101         }
102     }
103     if (!$flag){
104         return response()->json(['Message' => 'Wrong ID , check it please.']);
105     }
106     return response()->json($json_data);
107 }
108
109
110 }

```

Figure 4: lookup/ catalogue controller

buy

after sending the query from front-end controller service with the id of book to the the order service then check if the book is available in catalogue service via sending 2 queries. the first is check if there is copies from this book, if not the second query will paused. otherwise the second query will send to stressing the success of process.

```

InEndService > app > http > Controllers > FrontEndController.php
44 public function buyBook($id){
45     $startQuery = microtime(true);
46     if (Cache::has('Buy' . $id))
47     {
48         // check if the item in cache
49         $bookidvalue = Cache::get('Buy' . $id);
50         $url = json_decode($bookidvalue)->Message;
51         if ($url != "Your buy operation done successfully.")
52         {
53             $endcache = microtime(true);
54             echo "cache:" . ($endcache - $startQuery);
55             return response()->json(json_decode($bookidvalue));
56         }
57         Cache::forget('Buy' . $id);
58     }
59
60     $url = "";
61     $num = rand(1,4);
62     if ($num < 3)
63     {
64         $url = "http://192.168.12.142:8080/buy/" . $id;
65     }
66     else
67     {
68         $url = "http://192.168.12.142:8081/buy/" . $id;
69     }
70
71     $page = file_get_contents($url);
72     Cache::put('Buy' . $id . $page, 60); // put it in cache
73     $end = microtime(true); // end time of query without cache
74     echo "put the book in the cache:" . ($end - $startQuery);
75     return response()->json(json_decode($page));
76 }

```

Figure 5: buy/ front end controller

```

InEndService > app > http > Controllers > CatalogController.php
47 public function find($id){
48     $file = new \Illuminate\Filesystem\Filesystem();
49     $content = $file->get(_DIR_ . '/../books.txt');
50     $books = explode("\n", $content);
51     if (sizeof($books) < 2){
52         return response()->json(['Message' => 'No']);
53     }
54     $book_details;
55     $flag = false;
56     for ($i=0; $i<sizeof($books)-1; $i++){
57         $book_details[$i] = explode(":", $books[$i]);
58         if ($book_details[$i][4] == $id){
59             if ($book_details[$i][1] < 1){
60                 return response()->json(['Message' => 'Zero']);
61             }
62             $book_details[$i][1]--;
63             $book_details[$i][1] = $book_details[$i][1]."";
64             $flag=true;
65         }
66     }
67     if (!$flag){
68         return response()->json(['Message' => 'Wrong']);
69     }
70     for ($i=0; $i<sizeof($books)-1; $i++){
71         $books[$i] = implode(":", $book_details[$i]);
72     }
73     $content = implode("\n", $books);
74     $file->put(_DIR_ . '/../books.txt', $content, false);
75     return response()->json(['Message' => 'Done']);
76 }
77 }
78 }

```

Figure 6: buy/ cataloge controller

```

InEndService > app > http > Controllers > OrderController.php
9
10 * Create a new controller instance.
11 *
12 * @return void
13 *
14 public function __construct(){}
15
16 public function buy($id){
17     $url = "http://192.168.12.141:8080/query/check/" . $id;
18     $page = file_get_contents($url);
19     $resp = response()->json(json_decode($page));
20     $flag = json_decode($page)->Message;
21     if ($flag == "Zero"){
22         return response()->json(['Message' => 'there is no copies of this book.']);
23     }
24     if ($flag == "No"){
25         return response()->json(['Message' => 'There is no books in the store.']);
26     }
27     if ($flag == "Wrong"){
28         return response()->json(['Message' => 'Wrong ID , please check it.']);
29     }
30     if ($flag == "Done"){
31         $url3 = "http://192.168.12.141:8081/query/check/" . $id;
32         $page3 = file_get_contents($url3);
33         $url2 = "http://192.168.12.141:8080/update/buy/" . $id;
34         $page2 = file_get_contents($url2);
35         return response()->json(json_decode($page2));
36     }
37 }
38 }
39 }

```

Figure 7: buy/ order controller

but unfortunately i didn't take the responses because our virtual boxes were crashed and we reinstall them but many issues happened and this make us very upset for this. Also we tried in 2 another computers but the same problems happened, we don't know why unfortunately

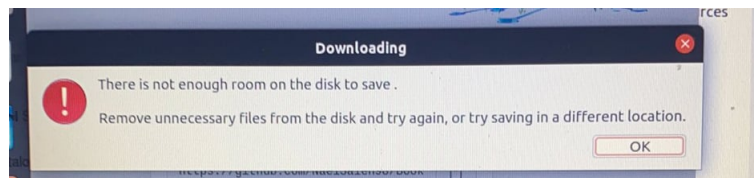


Figure 8: problem1

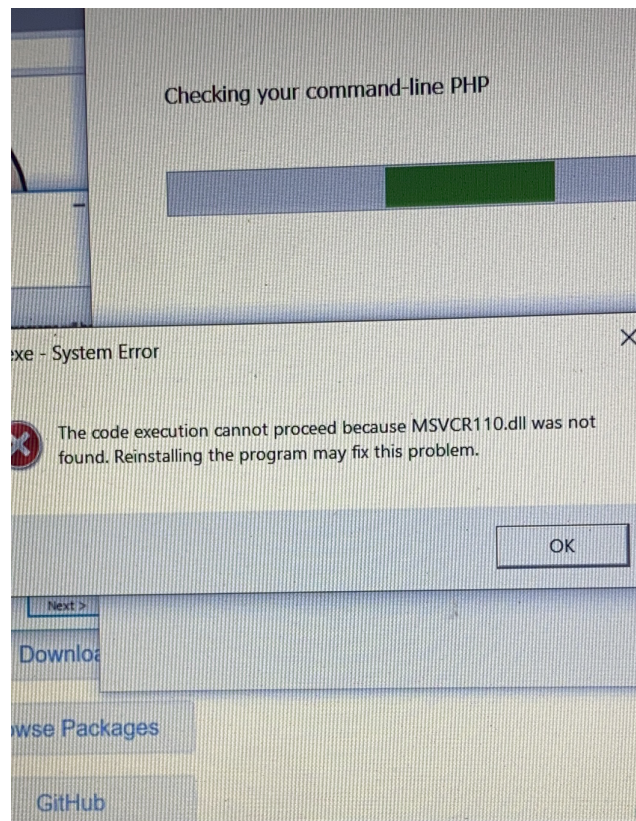


Figure 9: problem 2

ubuntu3 [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help

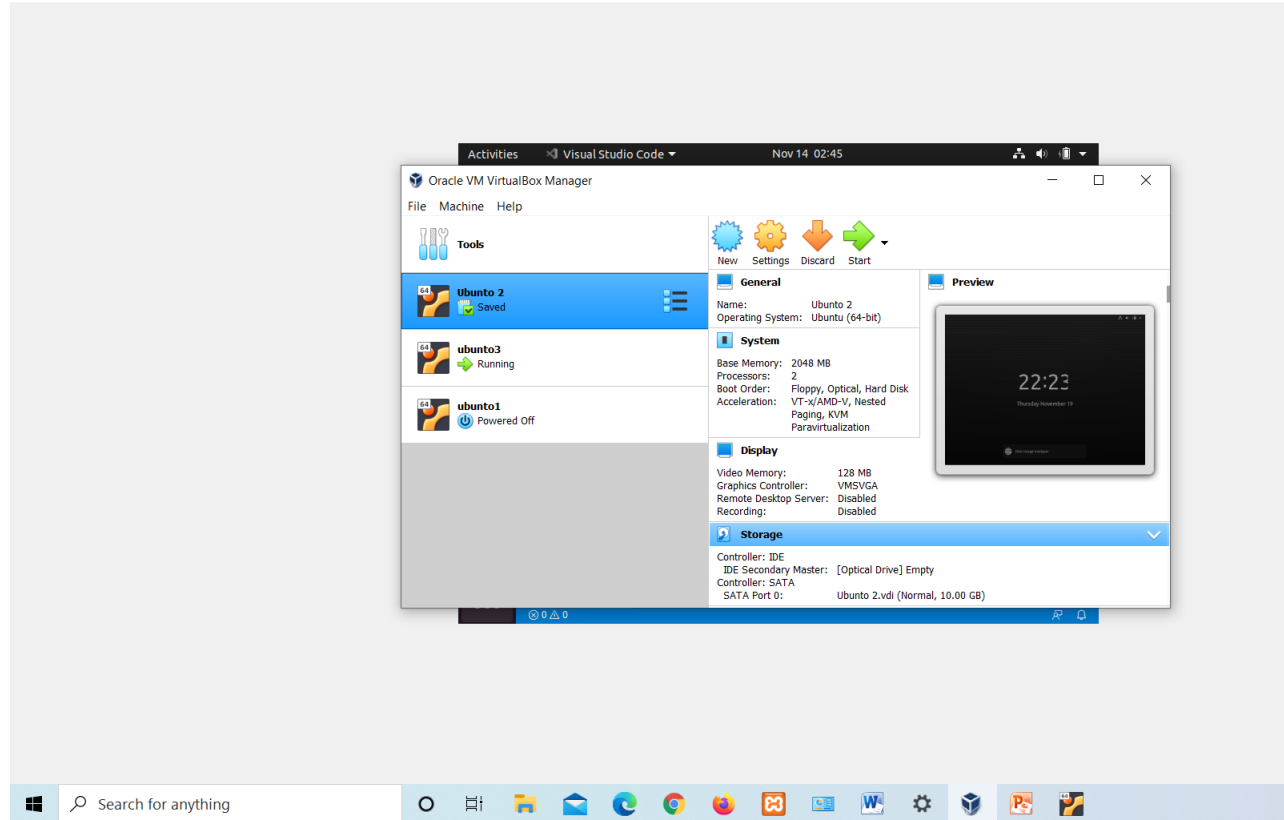


Figure 10: virtual 1

### 3 How to make it works

first: make the machines works with each other: create a network:

go to preferences then network, add anew network with networkcidr:192.168.1/24  
with name: lab1.

connect three virtuals to this network:

go to network in each virtual

change the port for each:

go to settings in network then to ip v4 then change the port for each, for exam-

ple : 192.168.12.141:8001

192.168.19.141:8000

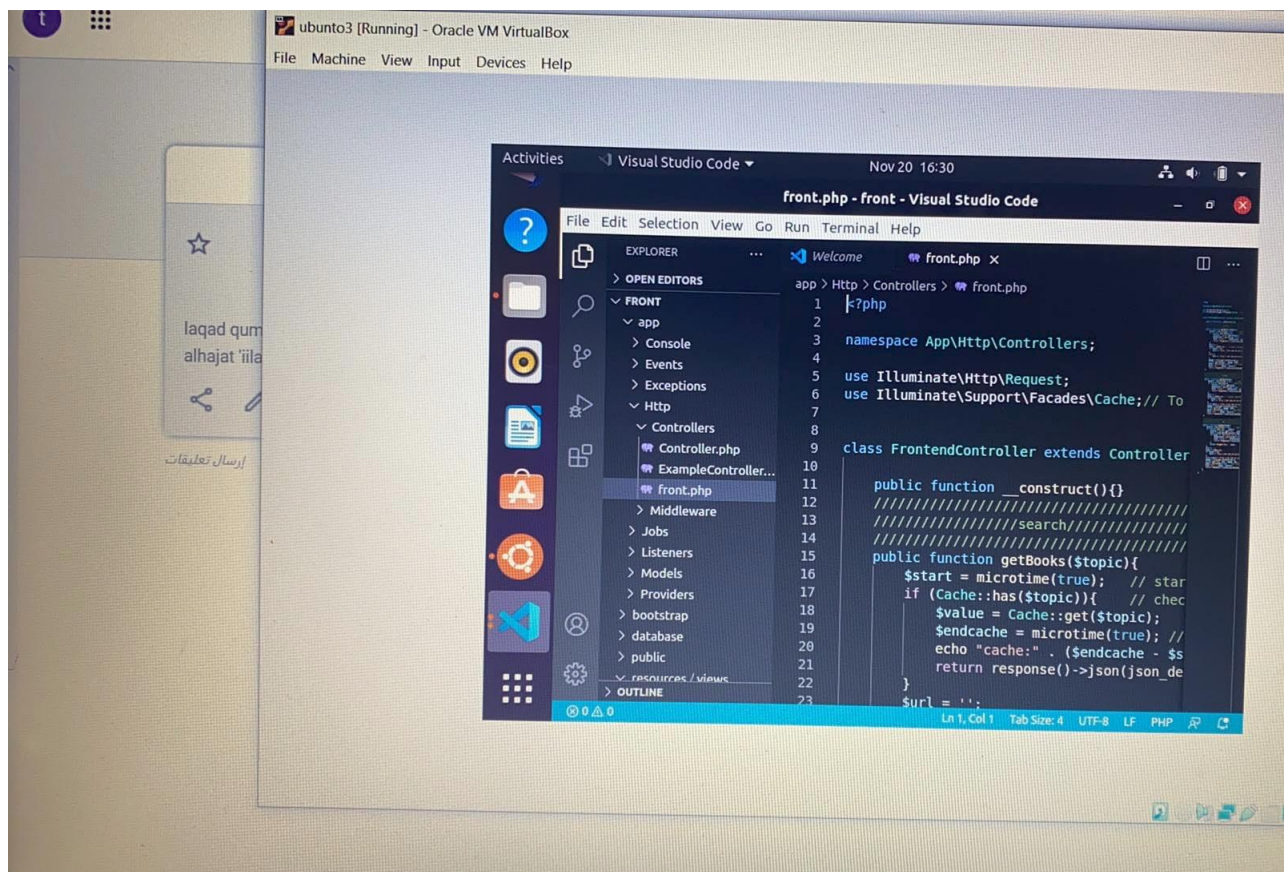


Figure 11: virtual 1/front-end



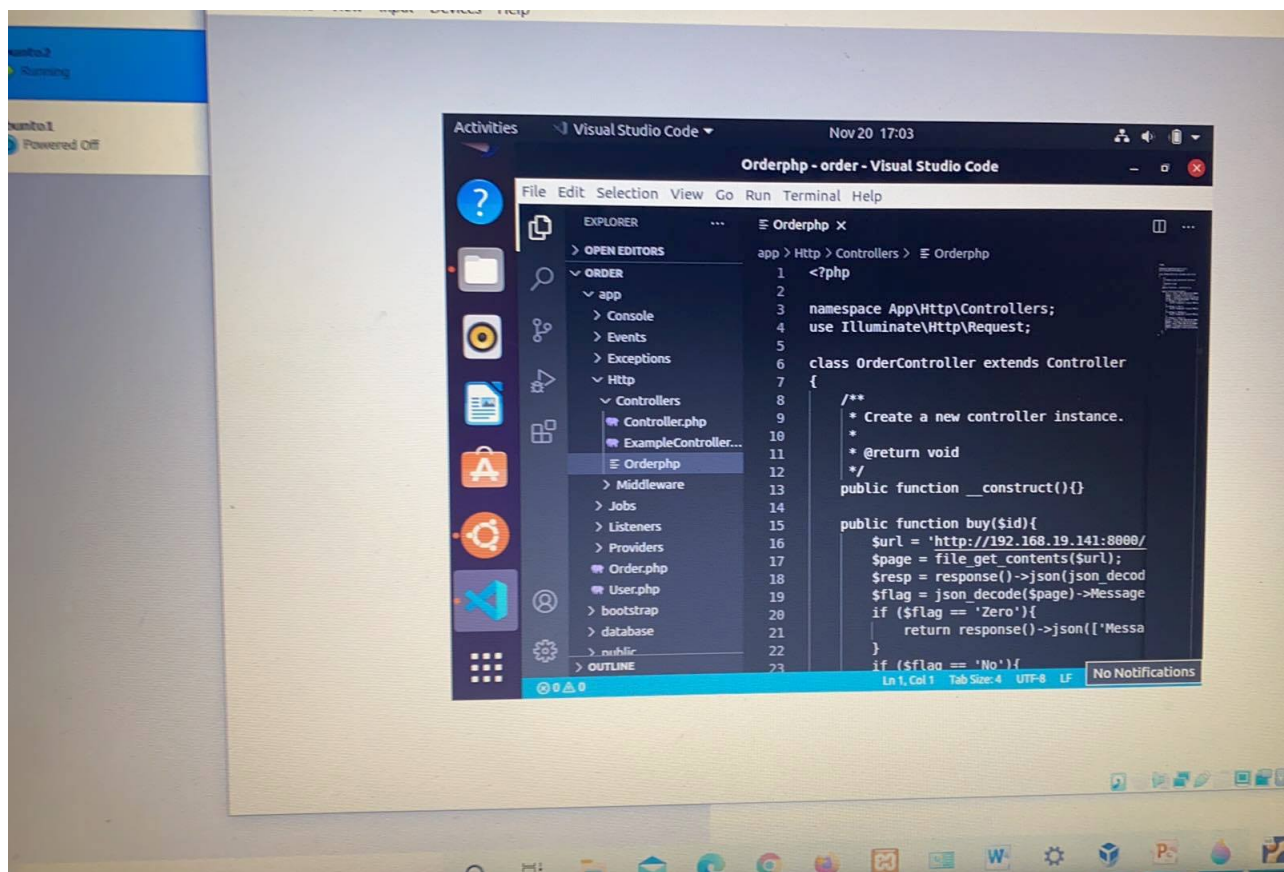


Figure 12: virtual 2/ order



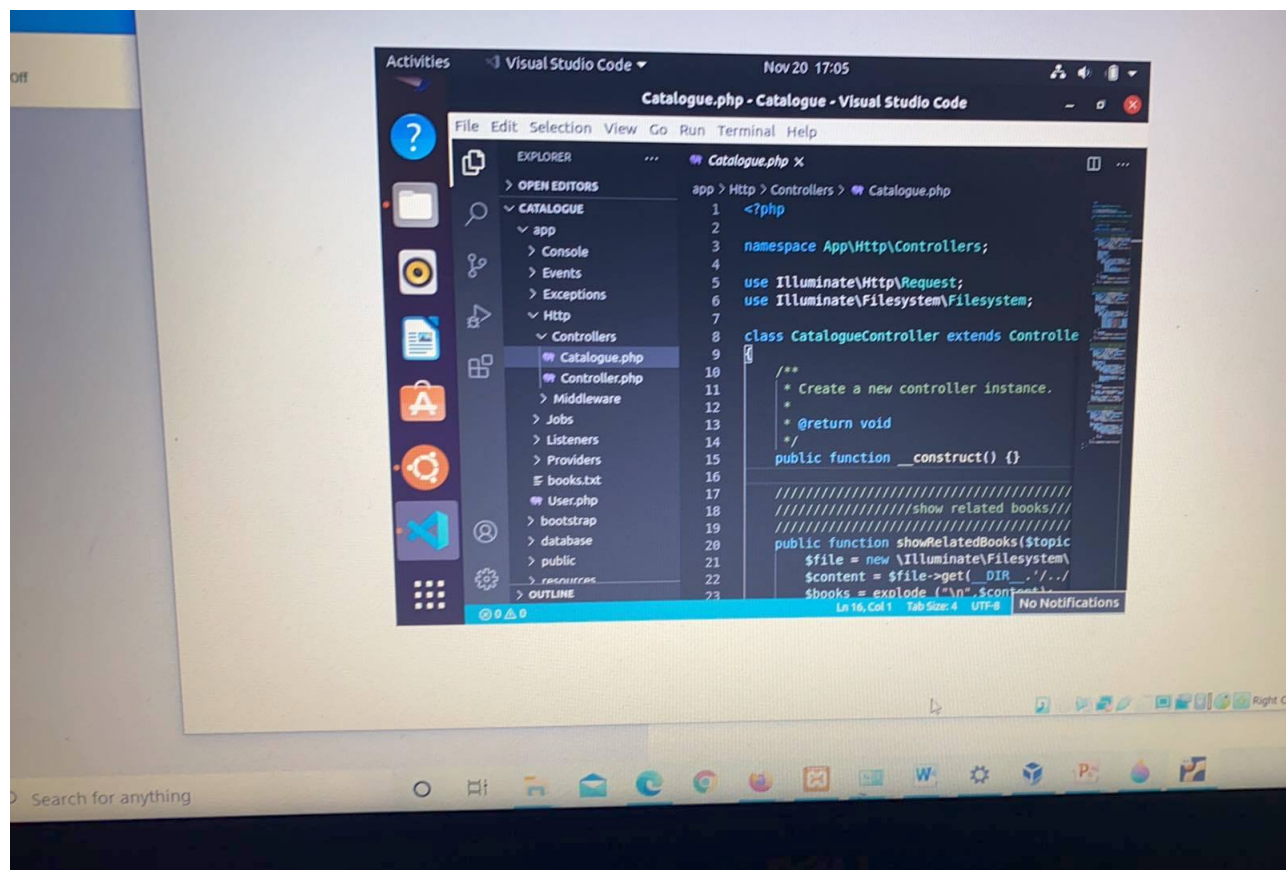


Figure 13: virtual 3/cataloge