**DOS Project Part 1**

**First student name: Basel Batanjeh.**

**Second student name: Mohmmed Derbas.**

**Link of GitHub:** https://github.com/BASELBATANJEH/DOS

**Discussion:**

Microservices is one of most useful techniques in programming. In our project we use spring boot framework to work on it. And this framework is built on java language. We use two microservices to enhance the performance so the two server was implemented in two virtual machines.

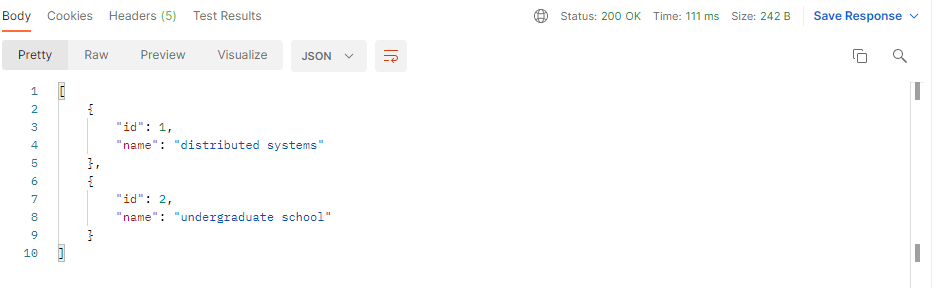
The first server is Catalog server and the second is order server. And absolute we use front end server. Addition to that, we build a friendly user interface that simplify on user treating with our interface.

The purpose of that as we say enhancement on performance and distribute the traffic to more than one machine and this will achieve the scalability as much as possible.

**CATALOG SERVER:**

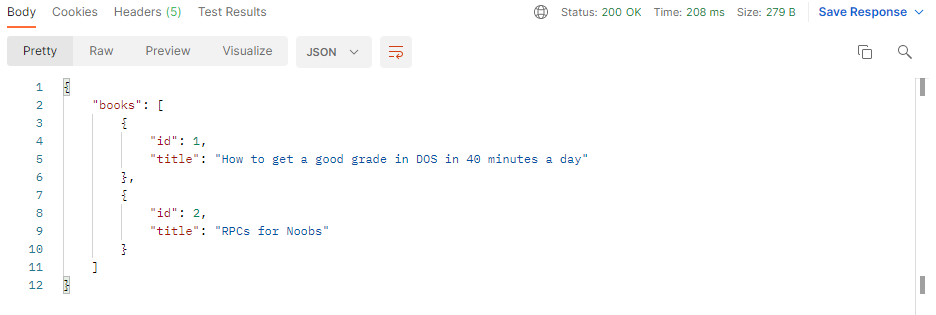
1. Search on categories:





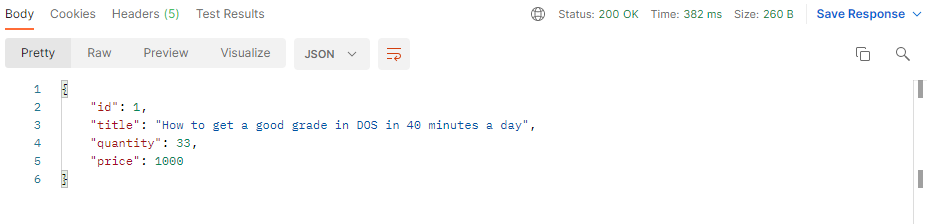
1. Search on topic





1. Information about book:



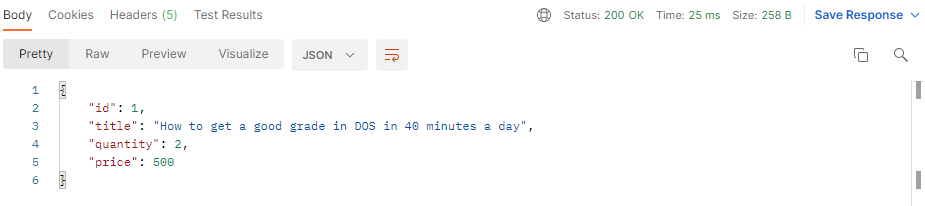


1. Update quantity / price



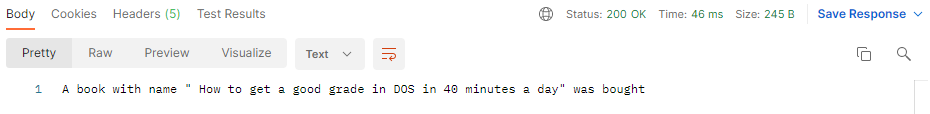


When we re-request info about book the result is:

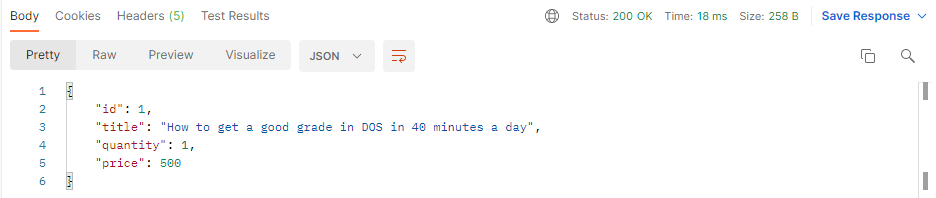


1. Buy a book(from second server)

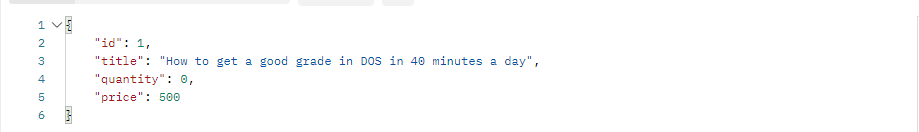




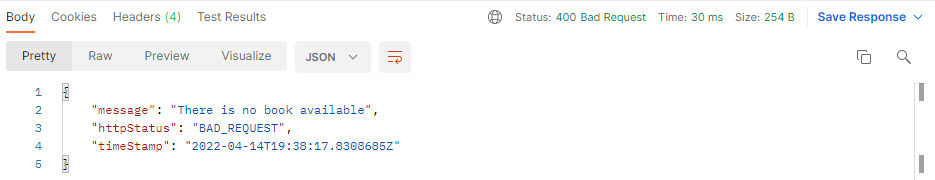
The result of info will be:



When we buy another book the info of book will be:



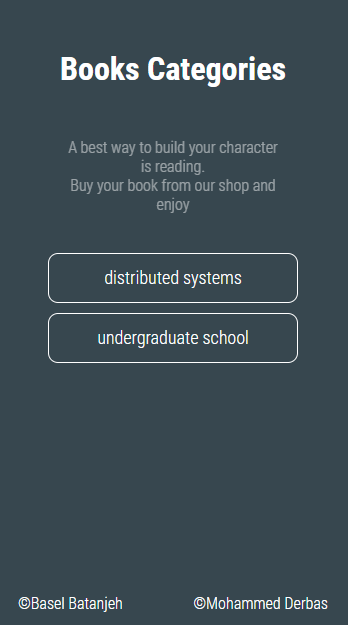
And now if we attempt to buy book the server will say:



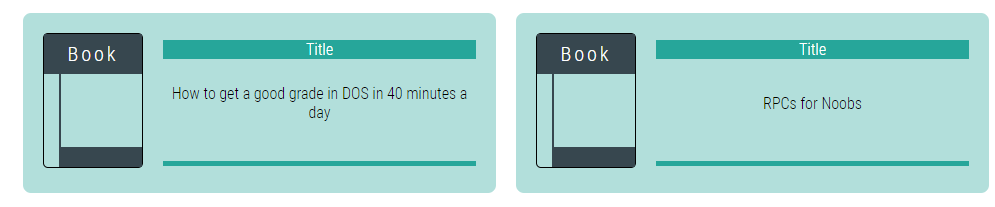
* **Frontend Server**

We made a good user interface that handle everything (html + css+ js )

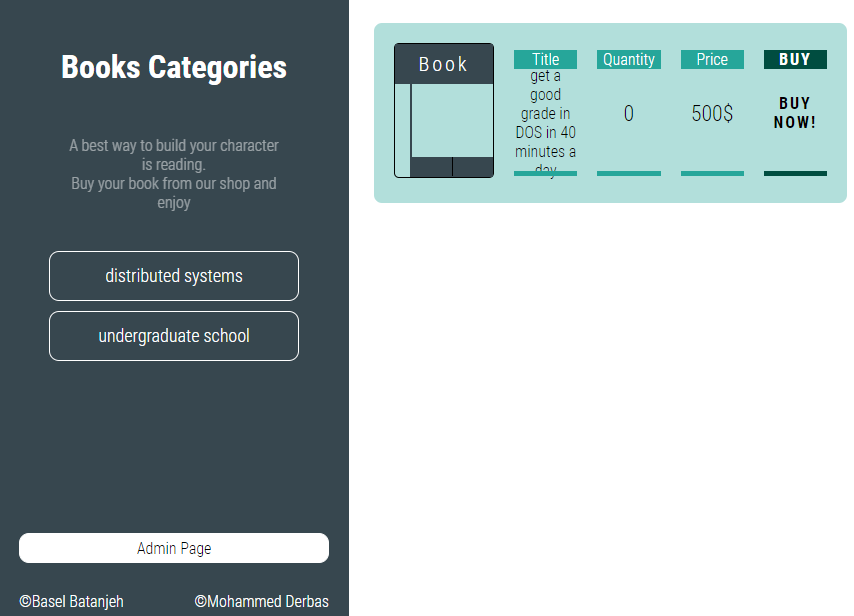




When we press in “distributed systems” (for example) we will make a sequence of div’s that its number equal to number of books in pressed category.

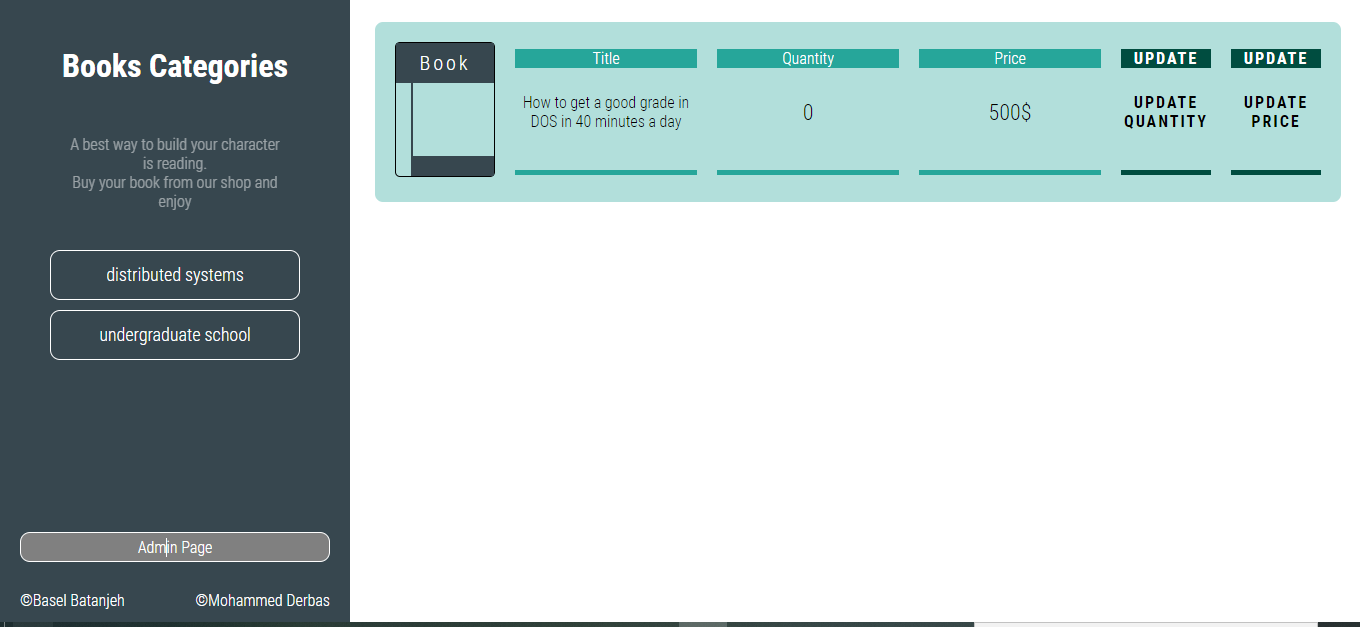


And when we press in any book we will show its details:

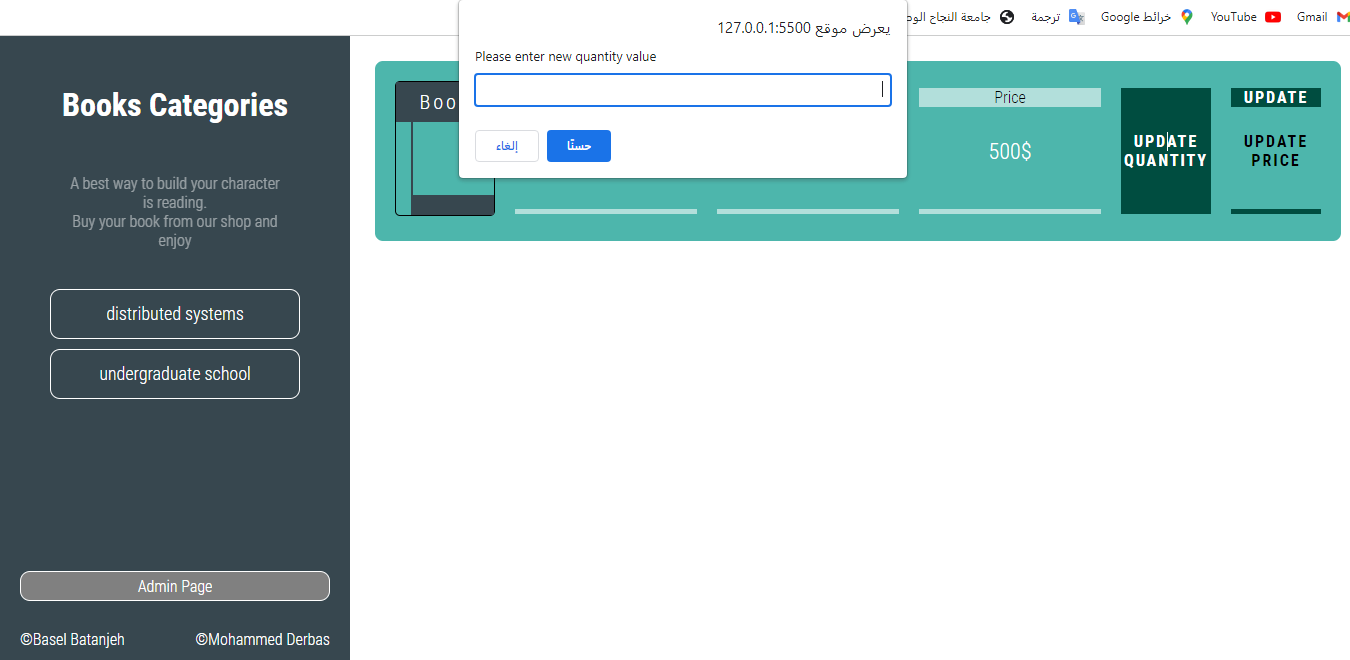


And we can easily buy book from the button as we see in screenshot.

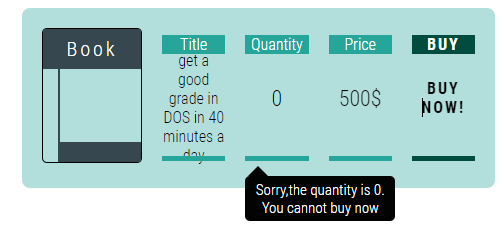
And if we notice we have “Admin Page” and if we click on it the update buttons will be shown.



If we click on update quantity for example we will show a prompt to user to enter the new quantity value.



If we return to user interface and make a lot of buy command until quantity reach 0. The user will receive message that is quantity reach zero without return to server (just using java script).



**IF WE WANT TO ENHANCE THIS PROJECT** we can for example replicate server so the traffic will be distributed to more than one server.