Distributed Operating

System

LAB #1

Mohammad B. Shreem
Nidal S. Shaar
Omar M. Hannon

November 20th, 2020

Table of contents

Introduction

We used a Laravel framework to build the system with two tables of mysql database, our system has 4 servers (1 is extra server upon doctor Samir request), and all are laravel framework servers.

Why Laravel Framework?

Laravel provides an expressive, minimal API around the Guzzle HTTP client, allowing us to quickly make outgoing HTTP requests to communicate with other web applications. Laravel's wrapper around Guzzle is focused on its most common use cases and a wonderful developer experience. Note that DigitalOcean Online Database was used.

System contents

- **Front-end server**

  with a suitable GUI, we build or book store system, We have activated controls within the system, linking it with the rest of our servers, based on the user's input to the system.

- **Catalog Server**

  It communicates with its own database that contains book information, this server is considered the main in the process, it is the one that returns all the book information and performs various services of adjusting prices and quantities stored in Database as well, and it communicates with all system servers

- **Order Server**

  This server is dedicated to purchases. When receiving a purchase order from the user, it makes sure that the required book is in the system and the quantity of the required book is available through contact with the catalog server, after that it reduces the amount of books stored in the system by one and sends the update request to Catalog server to amend quantities,

- **Admin Server**

  This server is dedicated to a wider process of searching and buying for books, the most important of which is allowing modification to the databases, and it communicates only with the catalog server, it contains a database that contains the most important users who are allowed to modify the databases, when the fronted end server communicates with the catalog, which in turn will communicate with this server, The connection is confirmed in the event that the user is authorized to modify the Databases, so a special GUI has been allocated for the Admin users in the Front-end Server to be transferred to in the event of confirmation of the connection

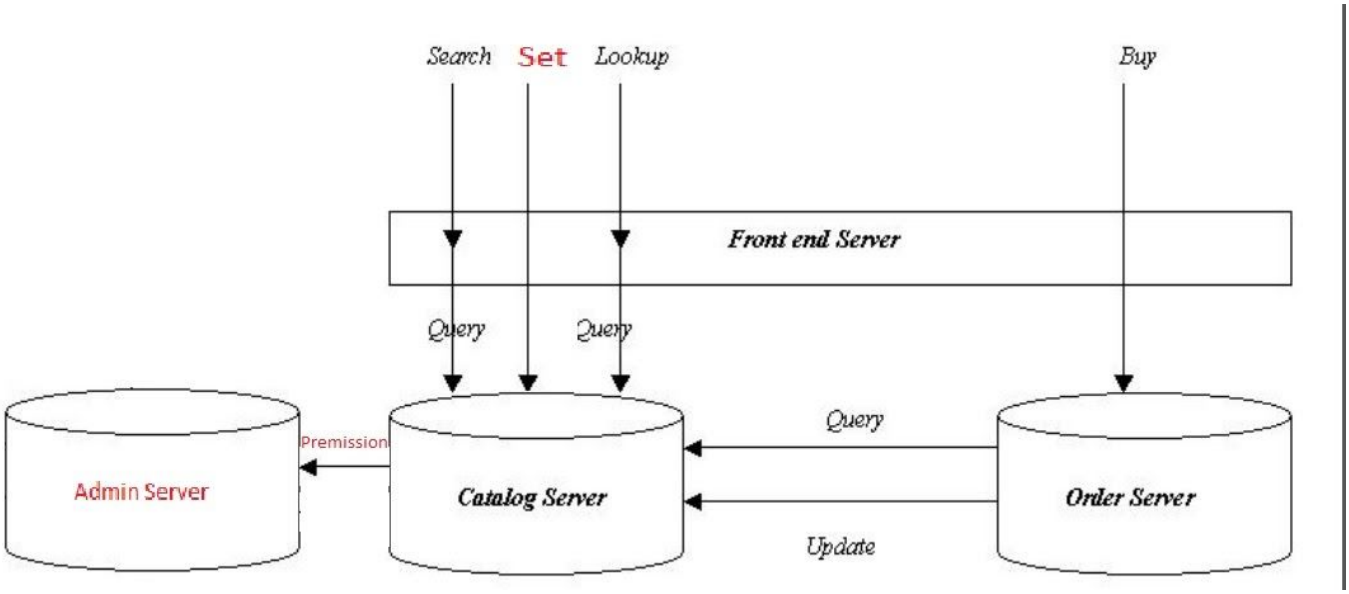


fig1 : Book Store System

How to run the System

we put our system on 4 different machine , the on every machine we need to install Laravel & PHP, the we deploy the system.

Installing Laravel

Laravel utilizes Composer to manage its dependencies. So, before using Laravel, make sure you have Composer installed on your machine.

Via Laravel Installer

First, download the Laravel installer using Composer:

```
composer global require laravel/installer
```

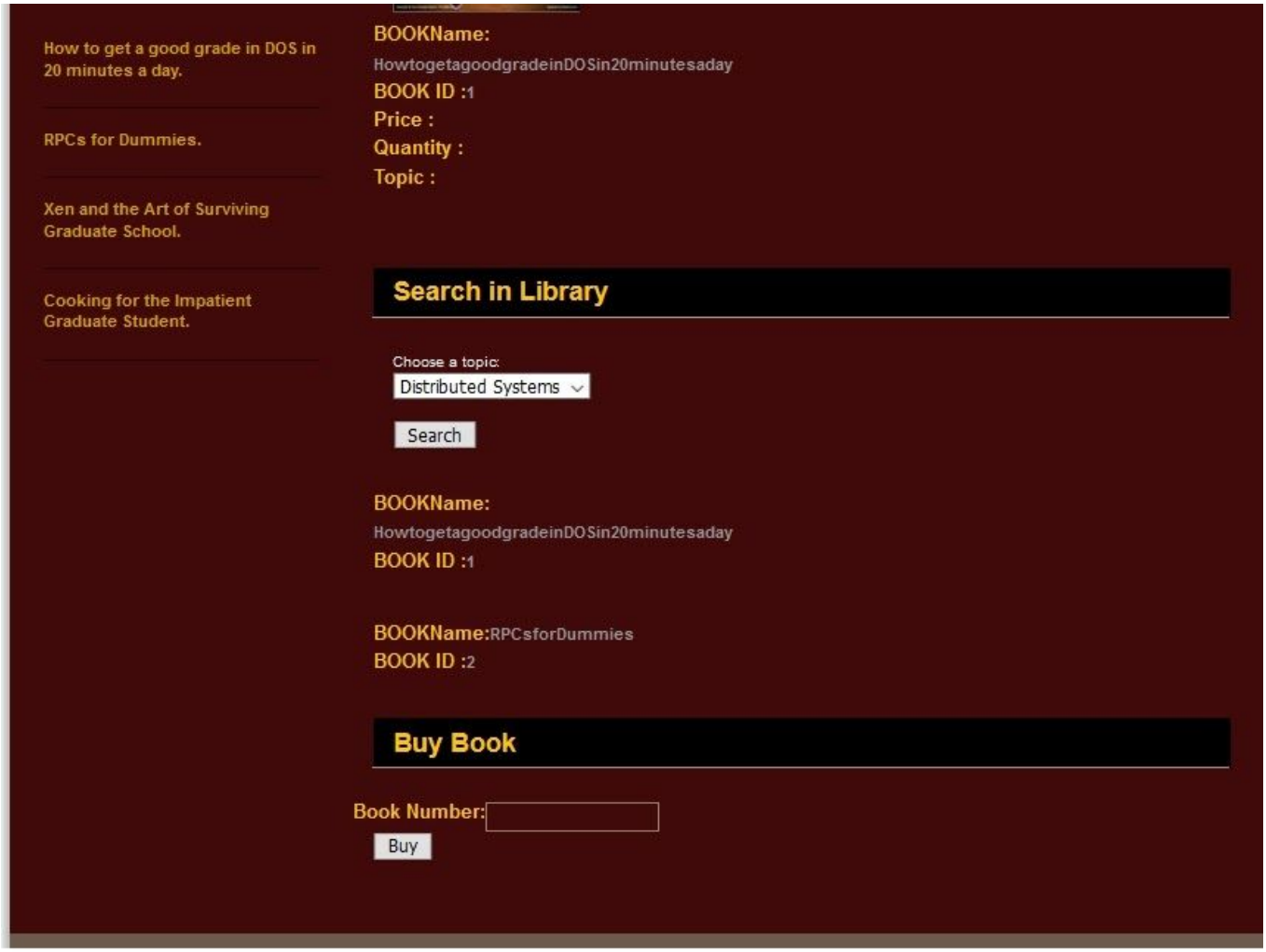
Development Server Deployment

```
php artisan serve --host **IP  --port **PORT
```

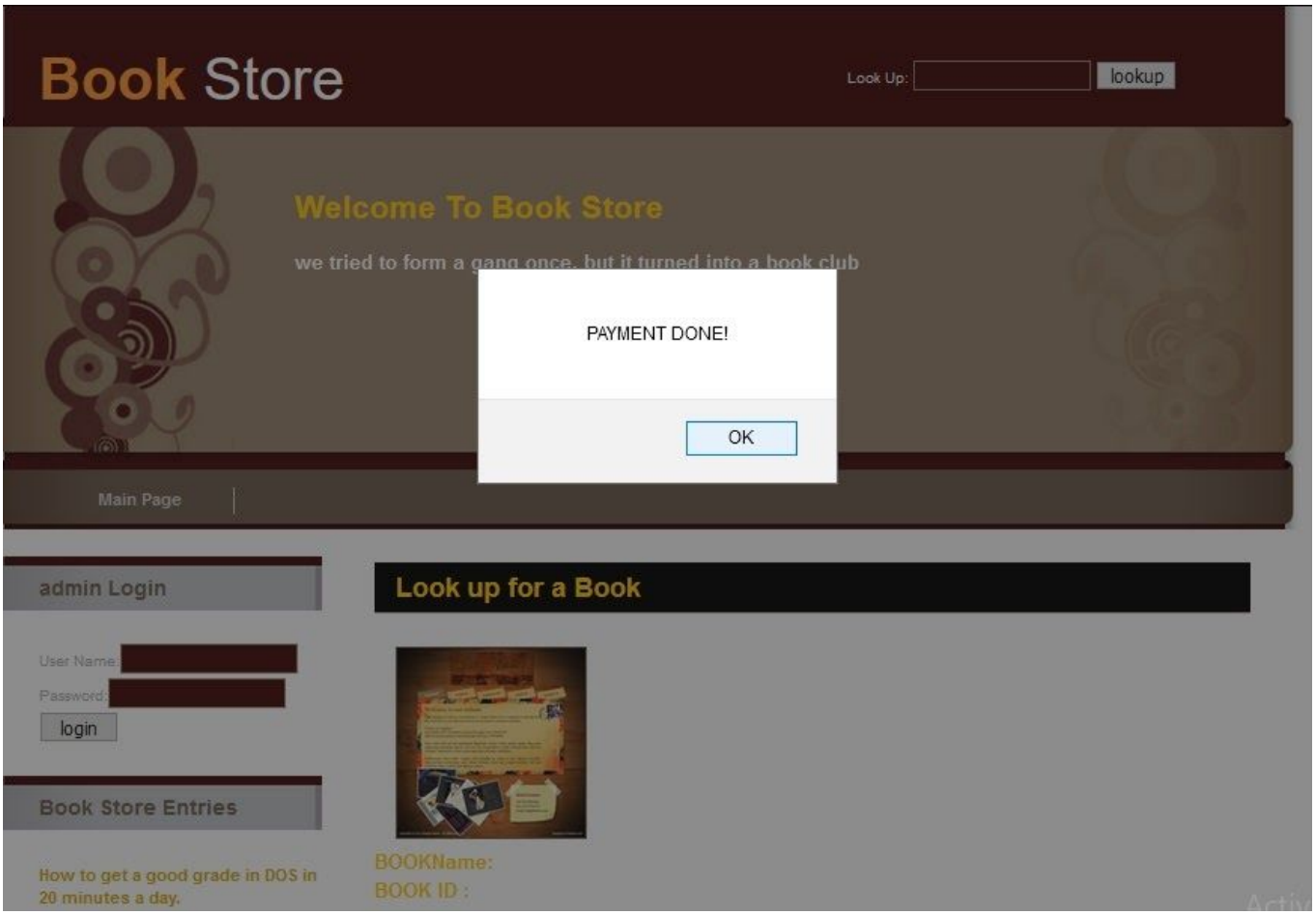
Clips during the work of the system

- **Look up '#BookNo' 2**

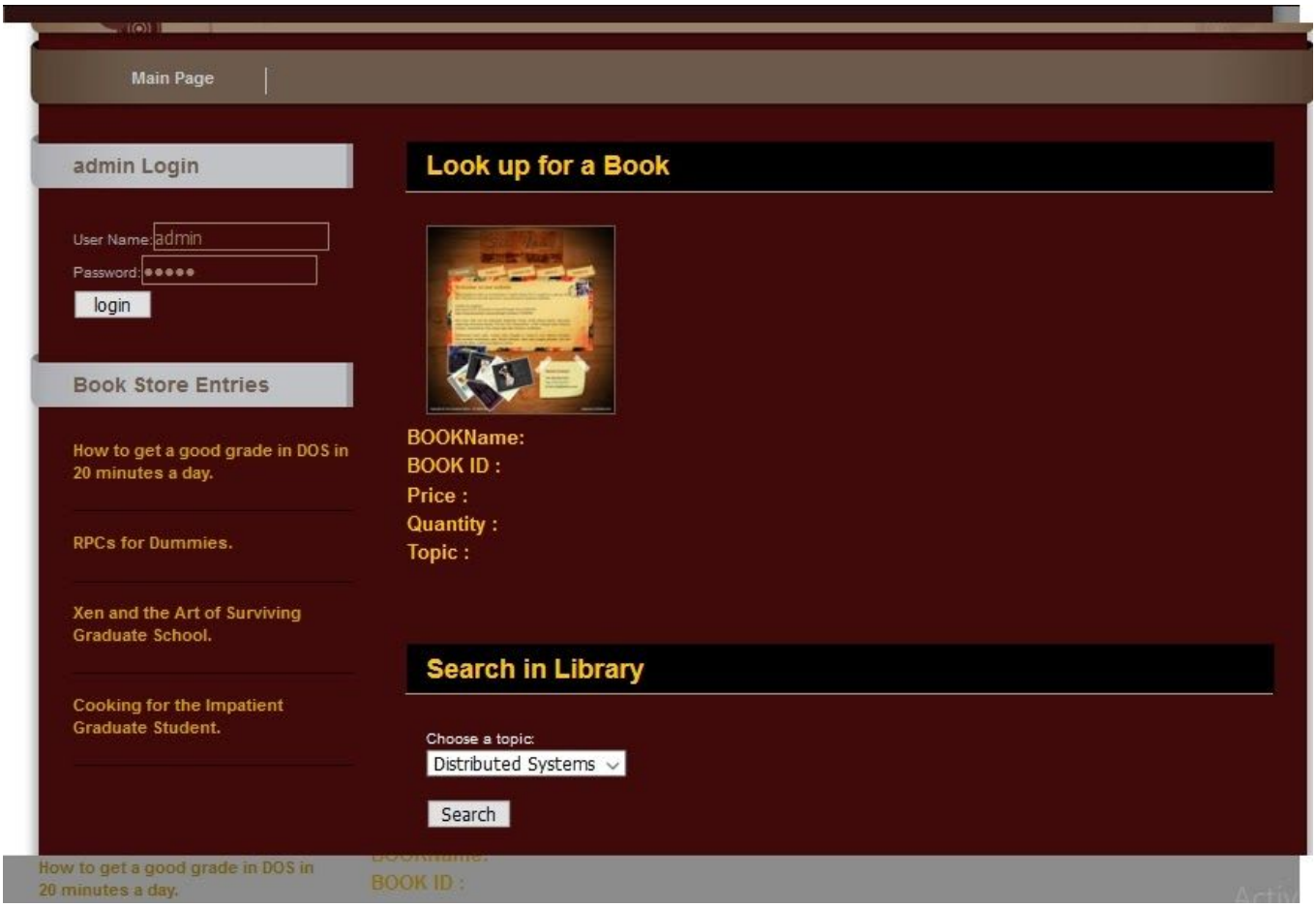
- **Search ‘#Topic’ Distributed System**



- **<u>Buy '#BookNo' 1</u>**



- **Admin Login ‘#username #password’ admin admin**




Book Store
logout
Welcome To Book Store
we tried to form a gang once, but it turned into a book club
Main Page
admin Login successfully
Results
Xen and the Art of Surviving Graduate School
RPCs for dummies
how to get good grades in Dos
BookName:
Cooking for the Impatient Graduate Student.
BOOK ID :4
Price :
30$
Stock:
500
update
BookName:
Xen and the Art of Surviving Graduate School.
BOOK ID :3
Price :
30$
Stock:
500
update
BookName:
RPCs for Dummies.
BOOK ID :2
Price :
30$
Stock:
500
update
BookName:
How to get a good grade in DOS in 20 minutes a day
BOOK ID :1
Price :
30$
Stock:
500
update

- **Set ‘#BookNo #PriceOrQuantity #Value'**