GEEKYFREAK – comics site

“Comics are a gateway drug to literacy.” @ Art Spiegelman

Team members:

Farrukh Tulkunov	U1610069
Jasur Turaev	U1610095
Ibrokhim Orifjonov	U1610076
Nodirjon Utkirov	U1610170
Saidazimkhon Jasurbekov	U1610195

Link to GITHUB repo: https://github.com/iuthub/design-project-geekyfreak

Link to WEBSITE: https://geekyfreak.000webhostapp.com/

Brief information about site:

Simple, convenient, innovative, creative… These are the main basics on which our site was created.

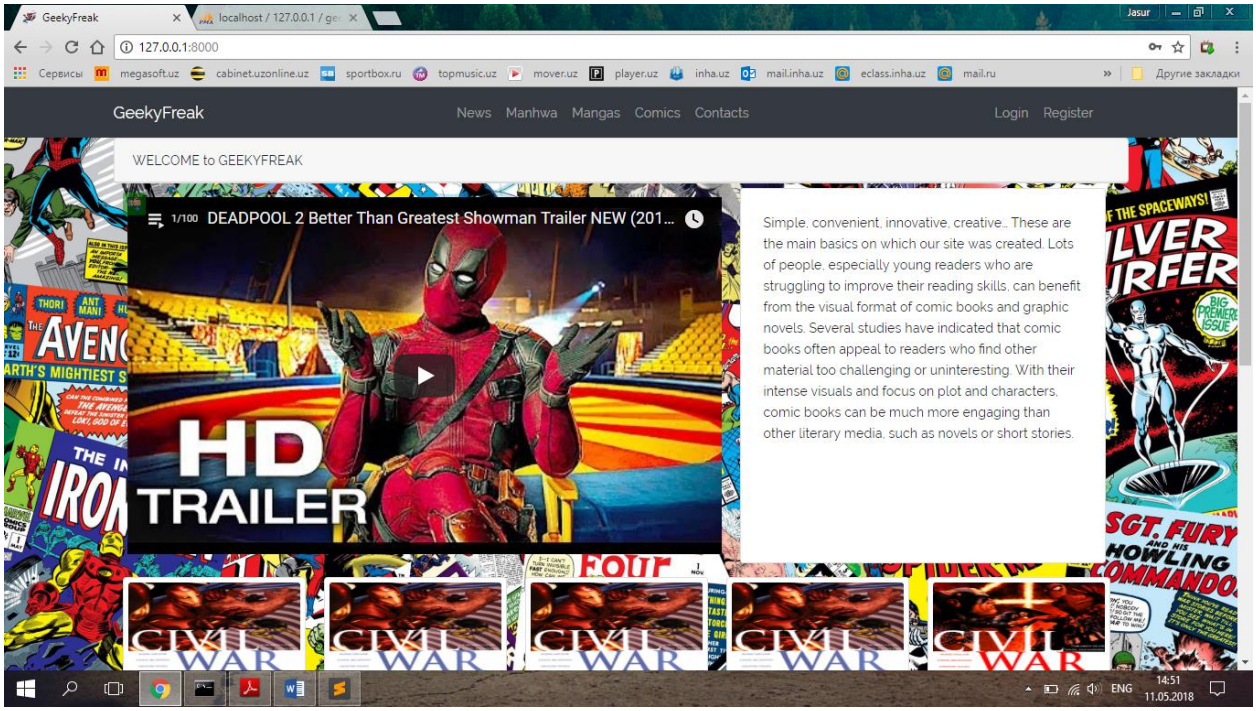
Lots of people, especially young readers who are struggling to improve their reading skills, can benefit from the visual format of comic books and graphic novels. Several studies have indicated that comic books often appeal to readers who find other material too challenging or uninteresting. With their intense visuals and focus on plot and characters, comic books can be much more engaging than other literary media, such as novels or short stories.

Our site provides with the main and the most needed information about *comics, manga, manhwa*.

You can also find news about latest releases of comics by simple keeping the watching the video on our home page.

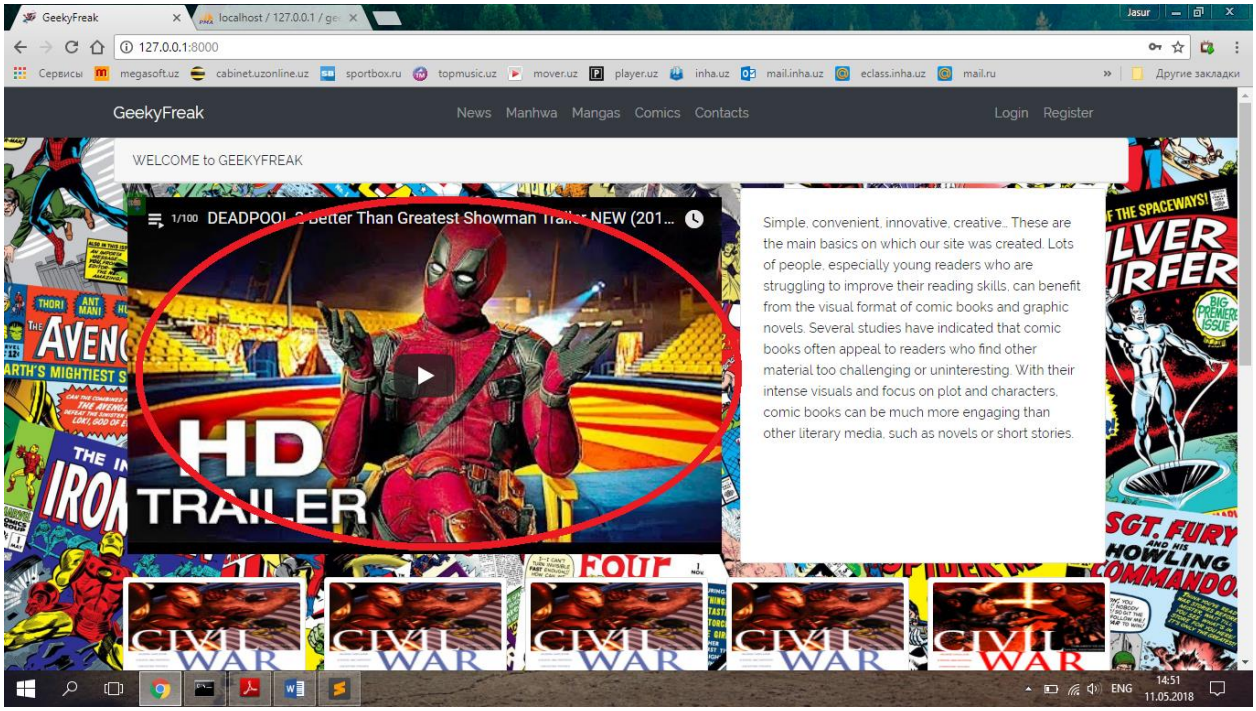
The site has register-login operations, that means that you can have your favorite books in your device immediately without searching. In order to improve our site by giving advices you are able to connect with us via our contacts and also the comments for discussion with people worldwide are open for you just after authorizing.

Developer side:

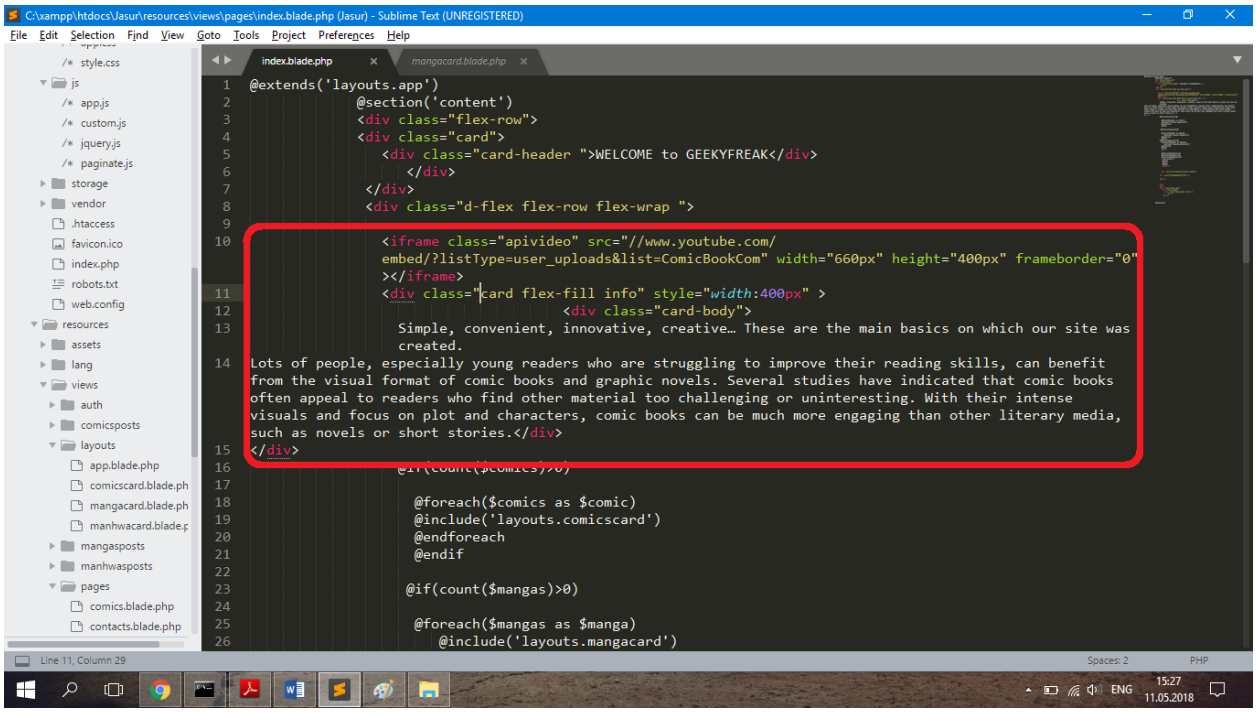


Home page consists of all comic, manga and manhwa books and each time updating video. On the top you can see the navbar on each of your pages.

The video was added using API feature. The <iframe> tag was used for completion.
In this row, you can also see some brief inspirational information about site.

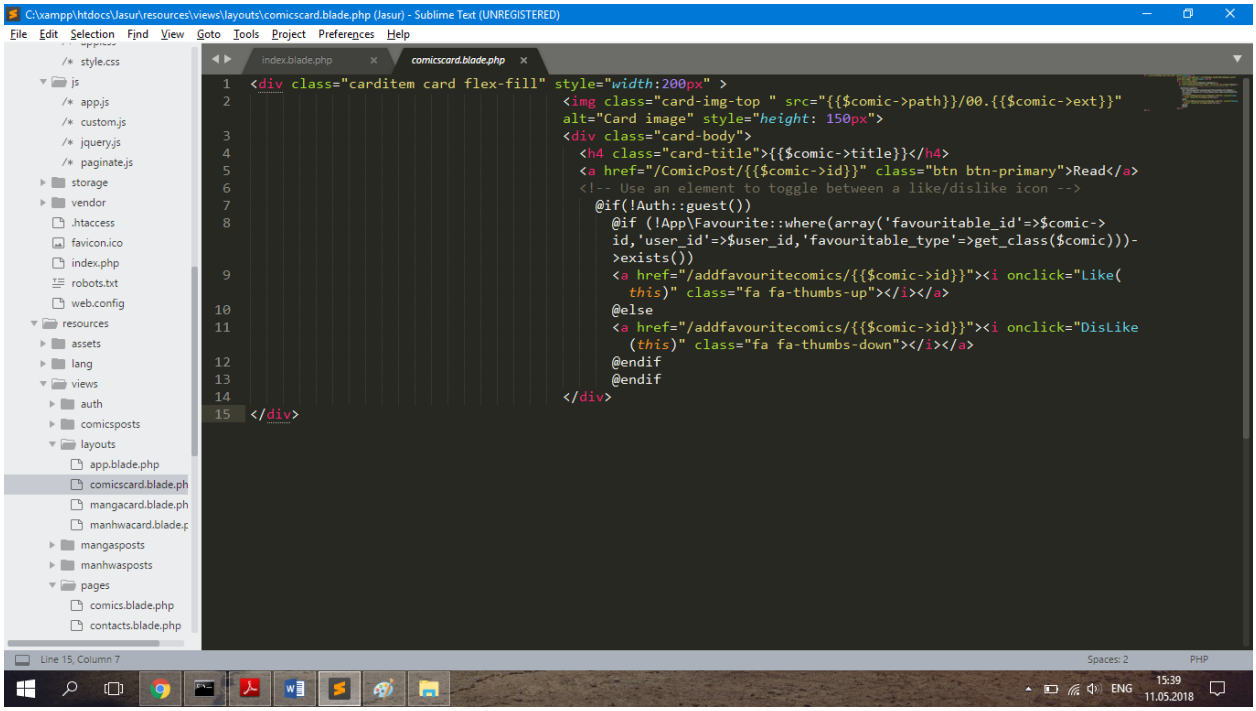


This is how we implemented with code:



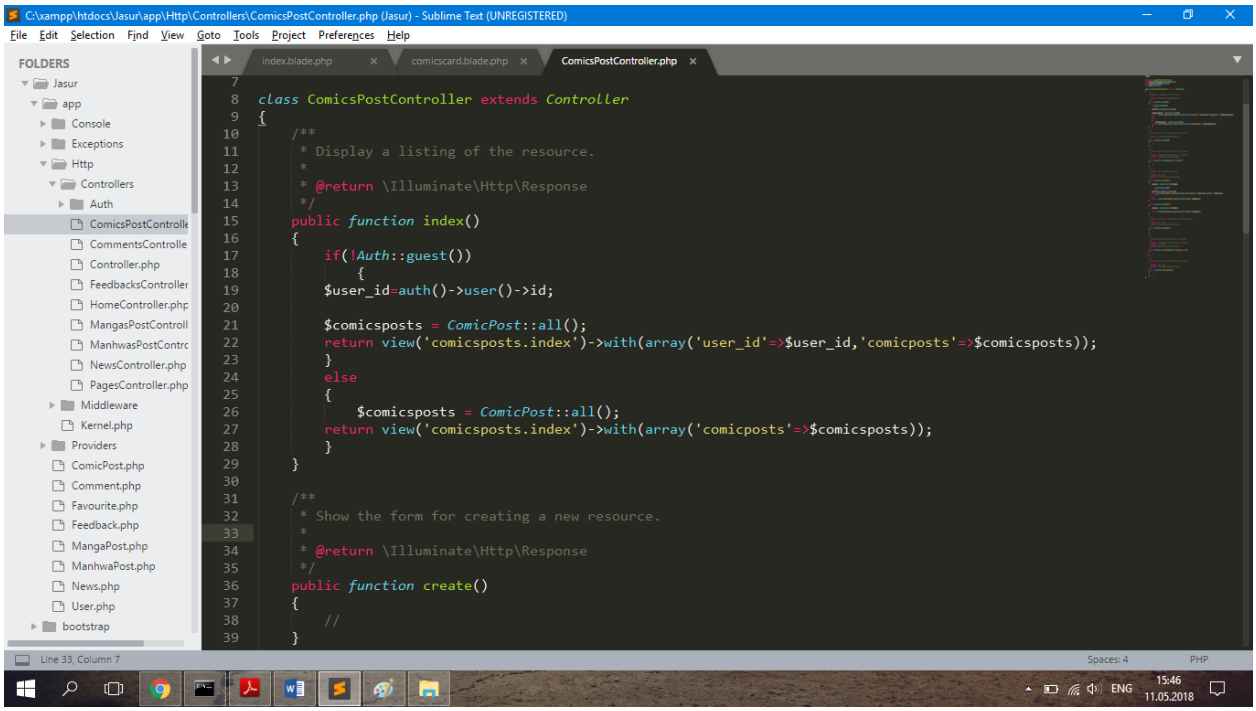
The comics-cards gives you possibility to read them if you are logged in and also the icon “thumbs up” allows you to add the comics book to your favorite catalogue.

Here is the code

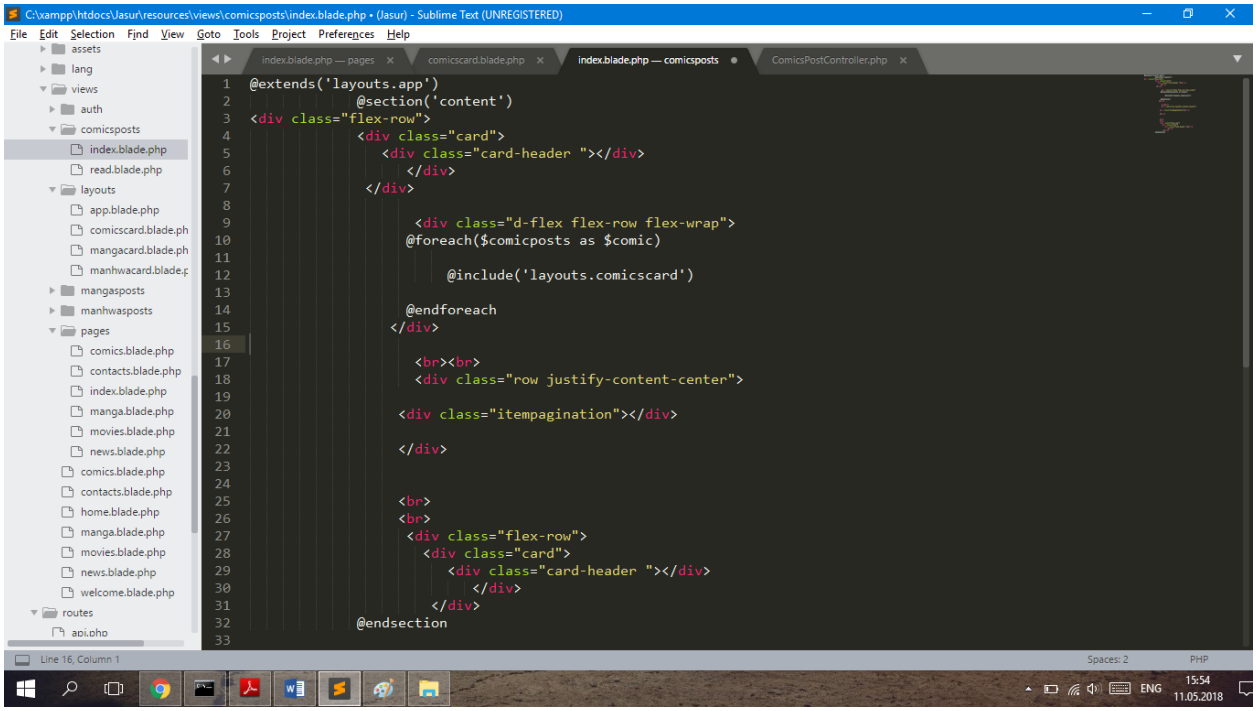


To achieve this, we needed model named “comicspost”, controller named “comicspostcontroller”, and a route to given page and most importantly a data inside database;

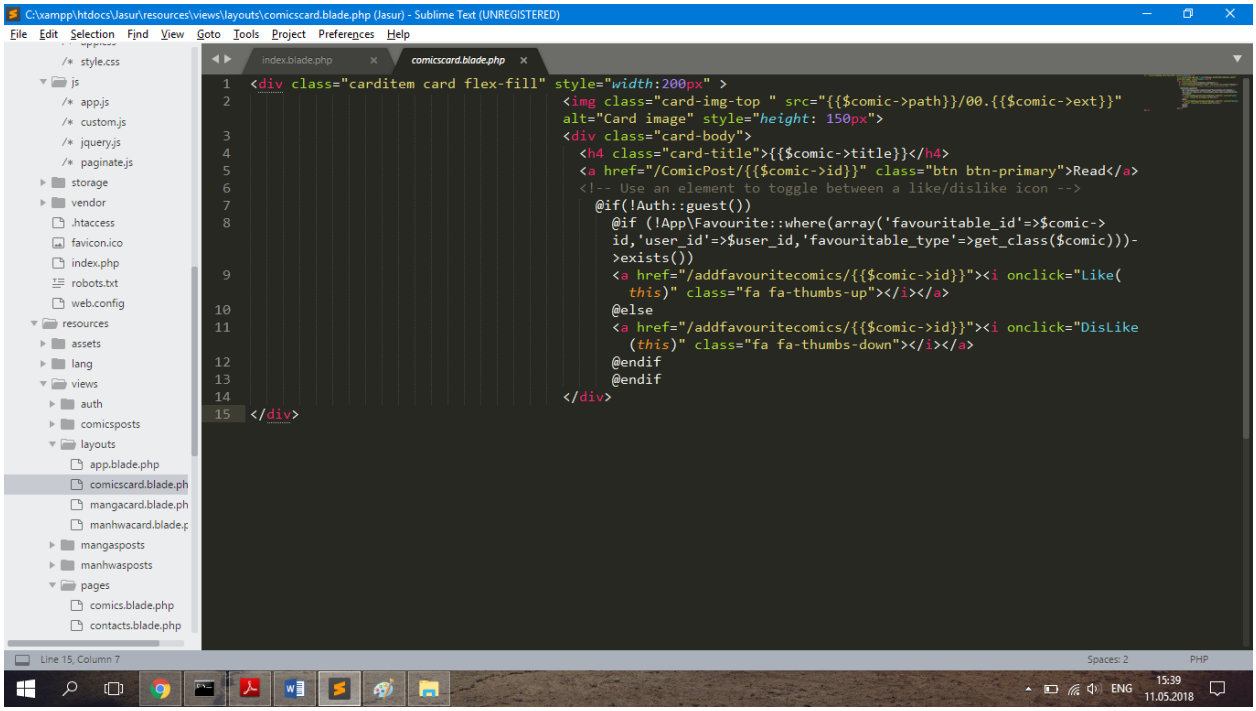
Route::resource('ComicPost','ComicsPostController'); this route will call the index function when /ComicPost is being opened



Here in the controller index function checks if you are logged in or not. It will get all the comics from database and pass it in array to the view comicpost. In case you are logged in it will also include your user id.

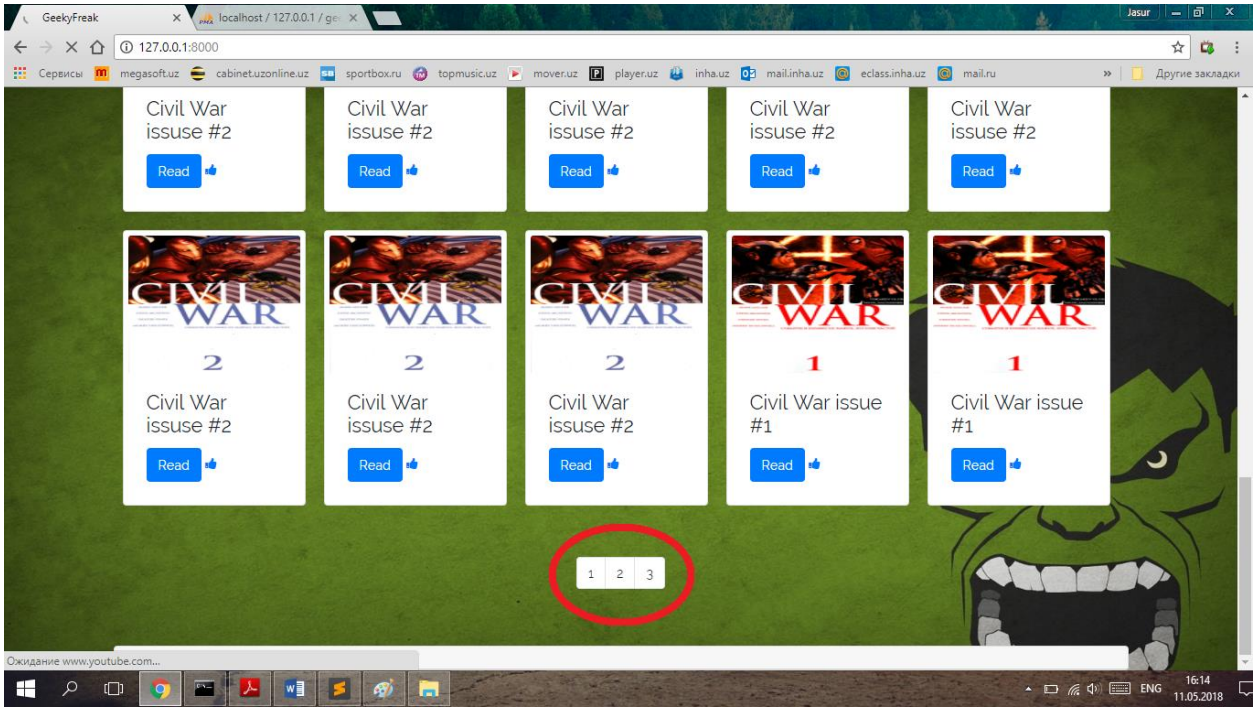


Here as you can see in index.blade we include all the necessary tags by extending layout and also we include comicscard layout which can be seen below.



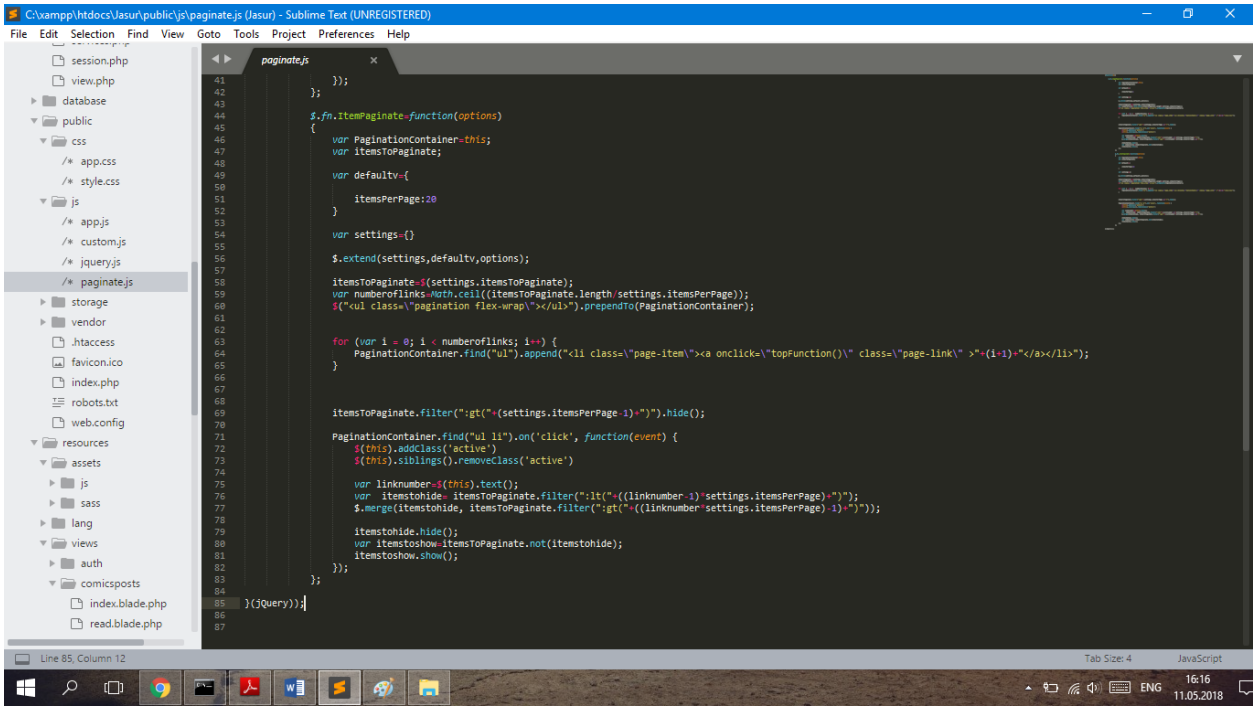
In this view you can see variable $comics, this is a variable given by controller and it is model with its own attributes which are specified in database. It will load all the requested information from database and show in our page.

To provide feature to add favorite item firstly we implemented many-to-many relation between comics and users. One user can have many favorite comics and one comics can belong to many users. So before showing button to add favorite we check if it isn't already a favorite item. if it is we provide a button to remove it from favorites.



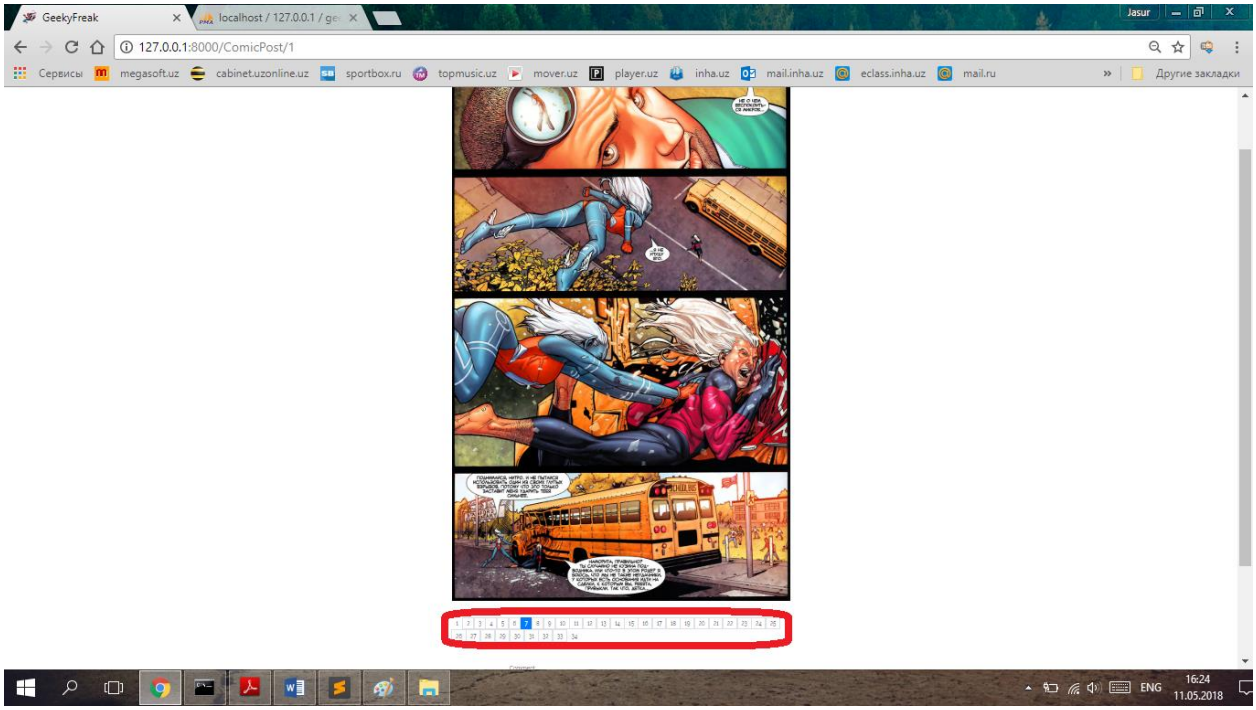
When the number of comic-cards reaches 20, at the bottom you can see the page number and using them it's easy to reach any page you want.

JQUERY and JS were used to implement the pagination.

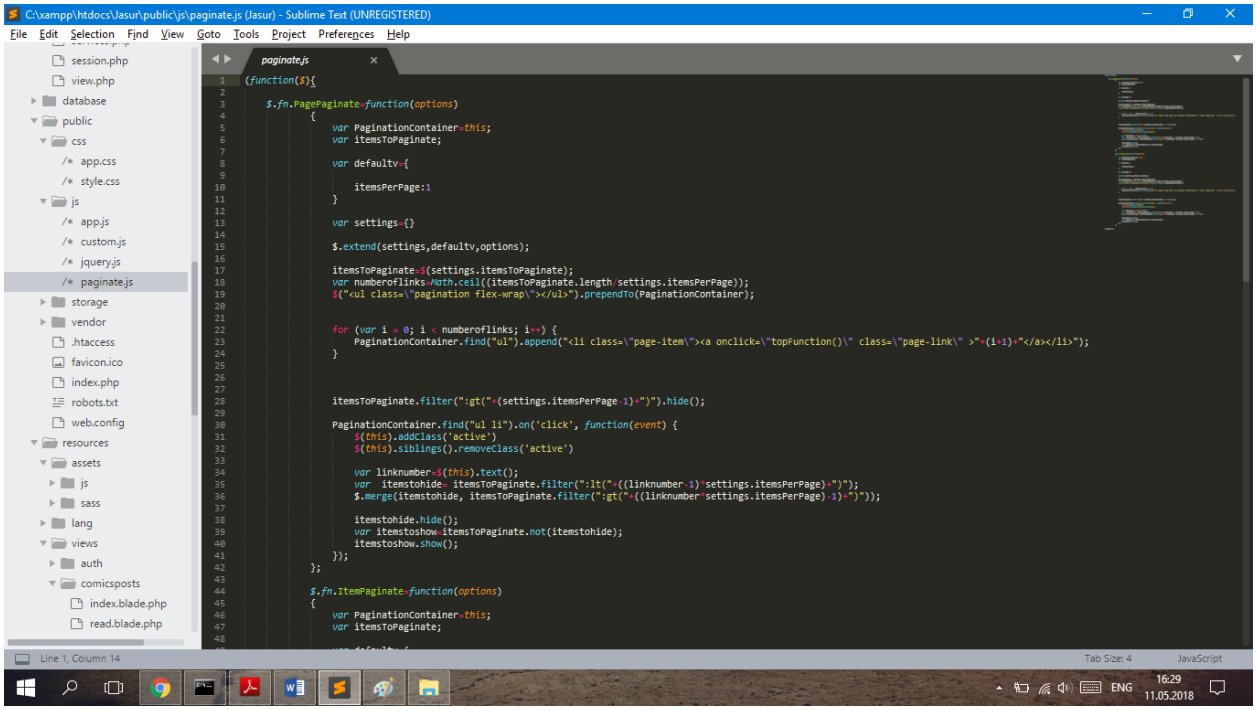


Comics, manhwa and manga pages are all implemented in the same way. The only difference is that comics page has comics, manhwa has manhwa and so on. But home and favorites pages contain all of them.

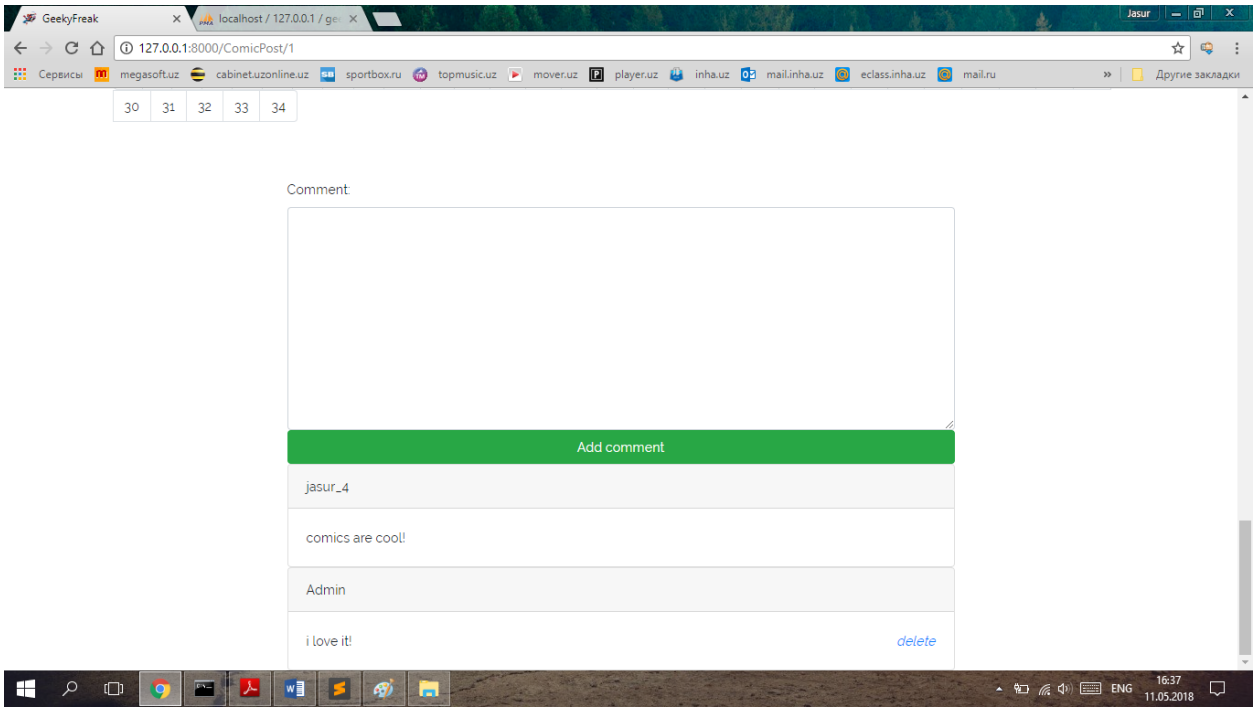
The process of reading the books:



As you know, comics are the books that are created with many images. So, in order to make that books readable, we made the same principle as in pagination in the home page. The difference is that here only the images are changing instead of the whole page refreshing.

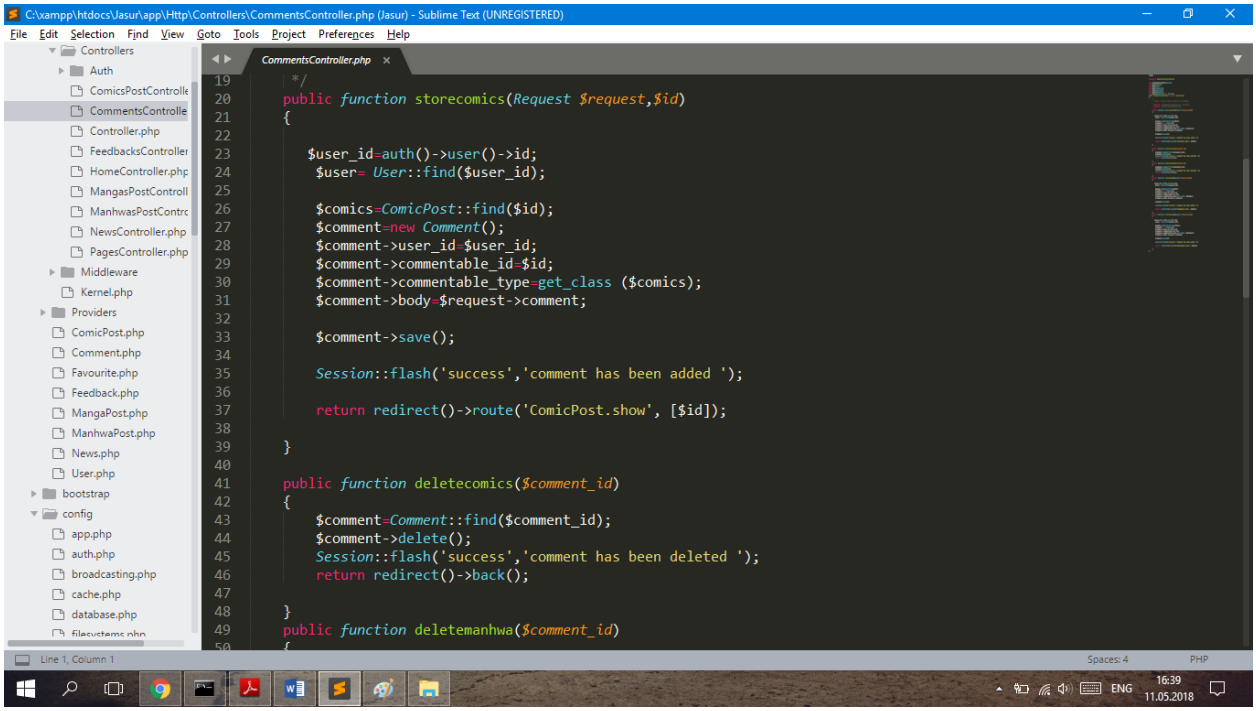


At the bottom of each book you can find comments box. And also, there is feature of deleting them.



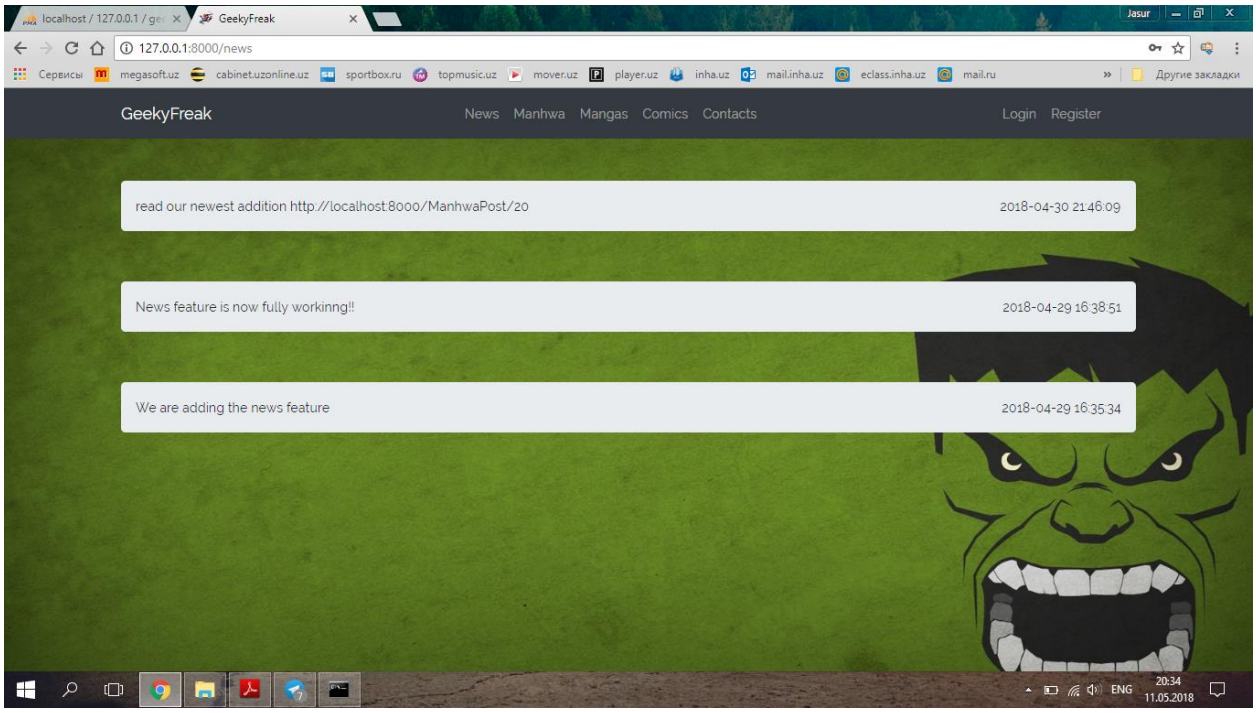
The implementation was learnt from documentation of laravel.com.

Polymorphic relations were used to link several model with one association. The owning of model to return when accessing the commentable relation is determined by ORM.

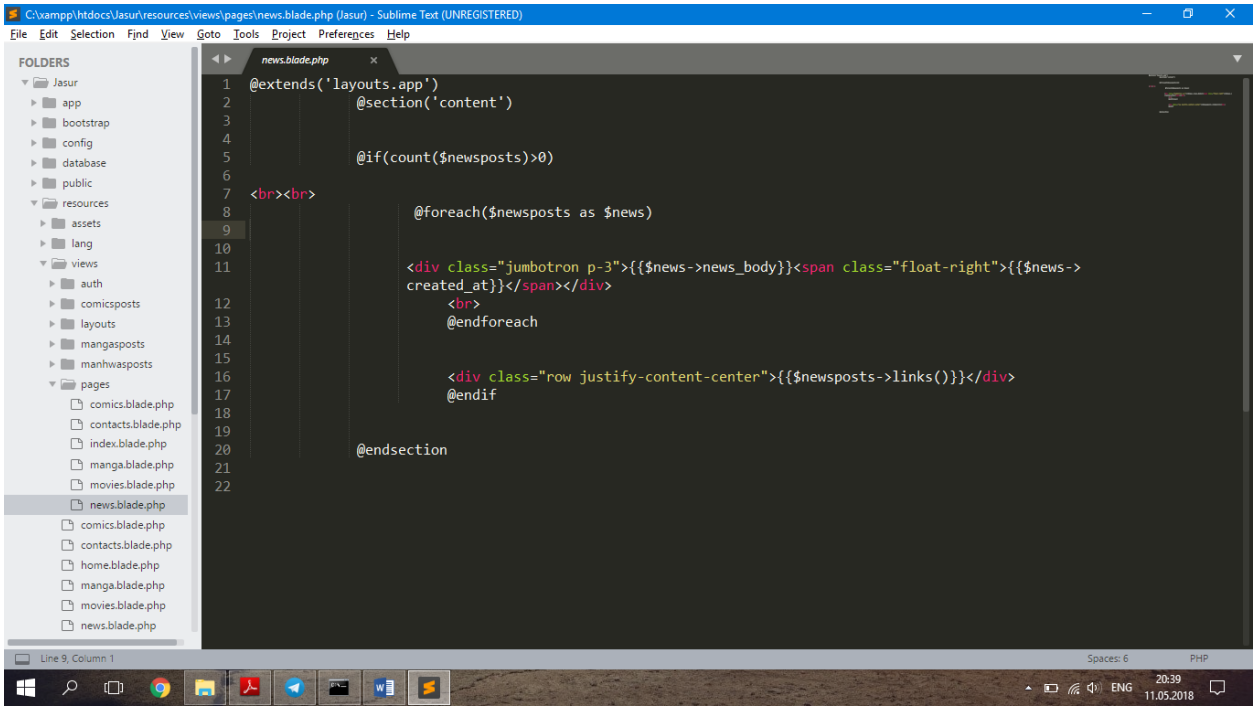


News.

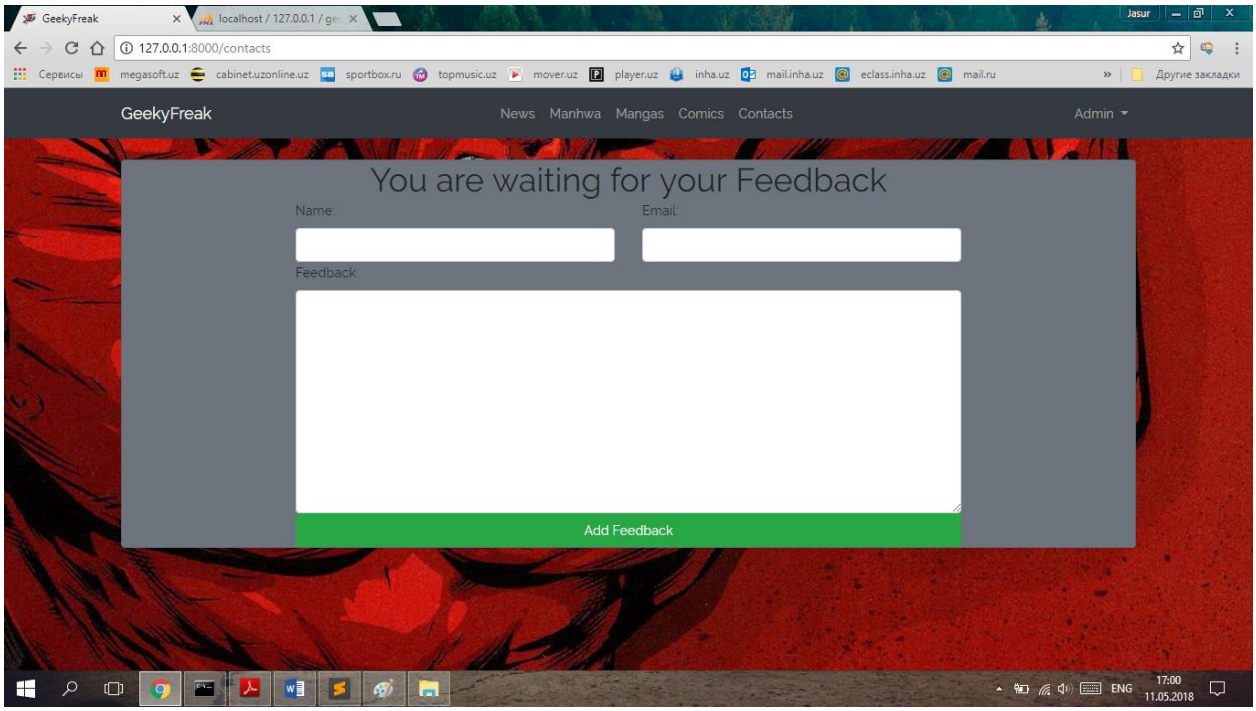
In news page all the news is loaded from database and they are ordered by latest modification.



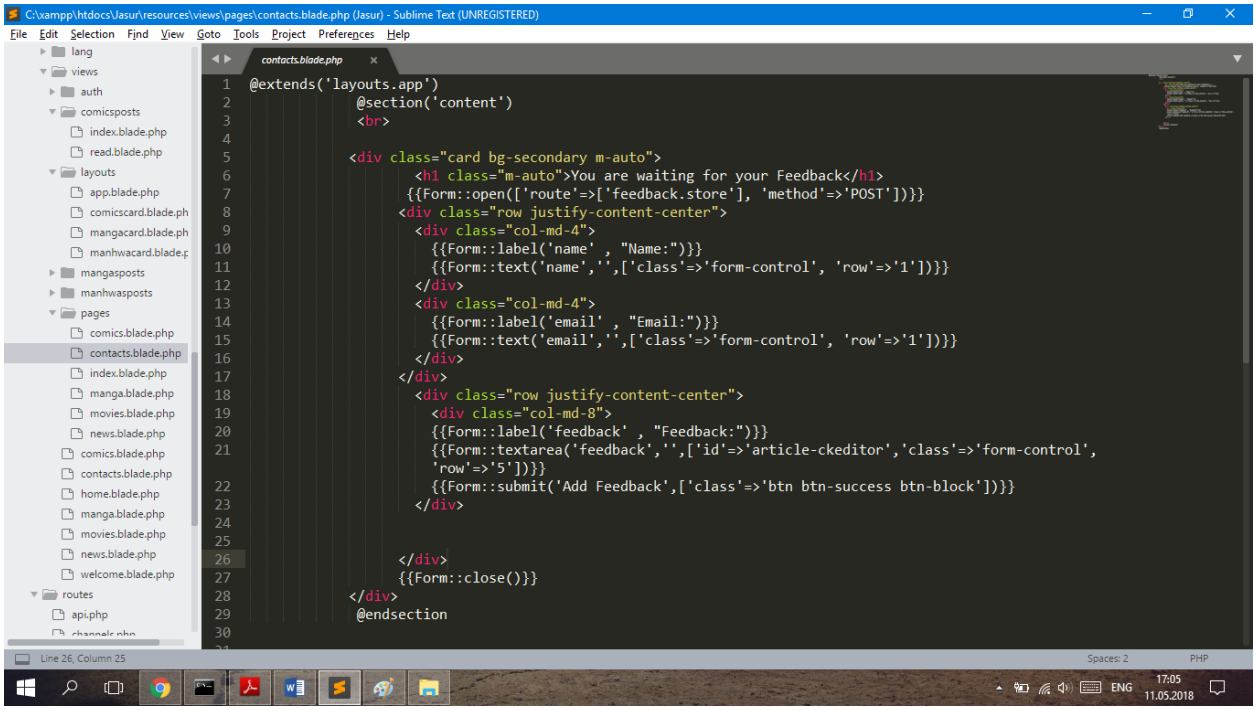
Implementation simple:



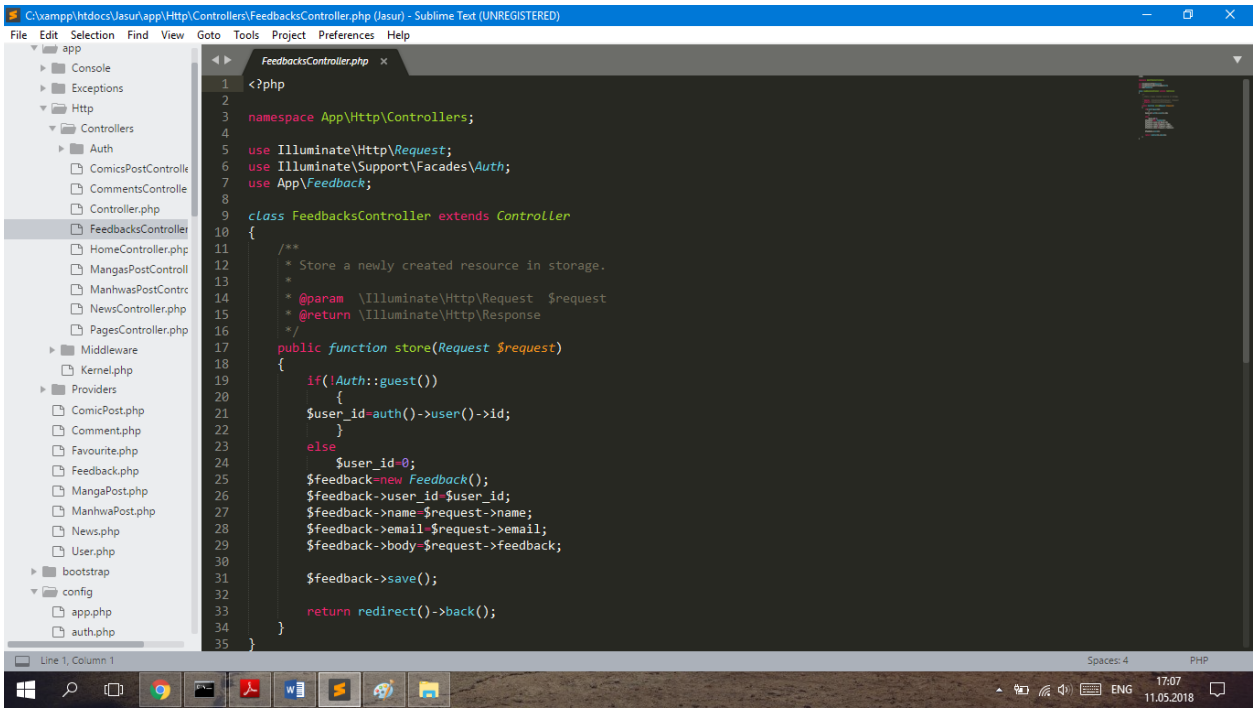
The “Contacts” were also added as in other sites in order to provide the user with feedback if he/she has questions, complains, suggestions or gratitude.



The implementation was made with simple forms.



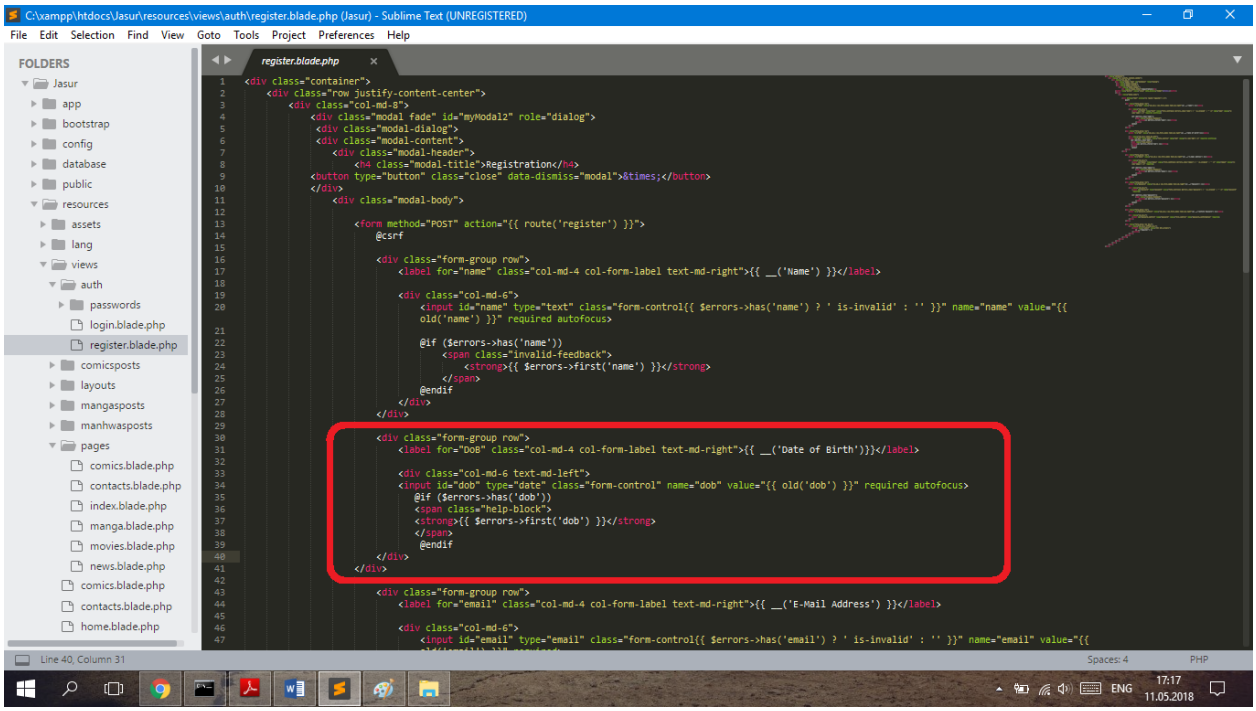
Form sends the message to database, feedback.store function. Here is it:



```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Feedback;

class FeedbacksController extends Controller
{
    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        if(!Auth::guest())
            {
        $user_id=auth()->user()->id;
            }
        else
            $user_id=0;
        $feedback=new Feedback();
        $feedback->user_id=$user_id;
        $feedback->name=$request->name;
        $feedback->email=$request->email;
        $feedback->body=$request->feedback;

        $feedback->save();

        return redirect()->back();
    }
}
```

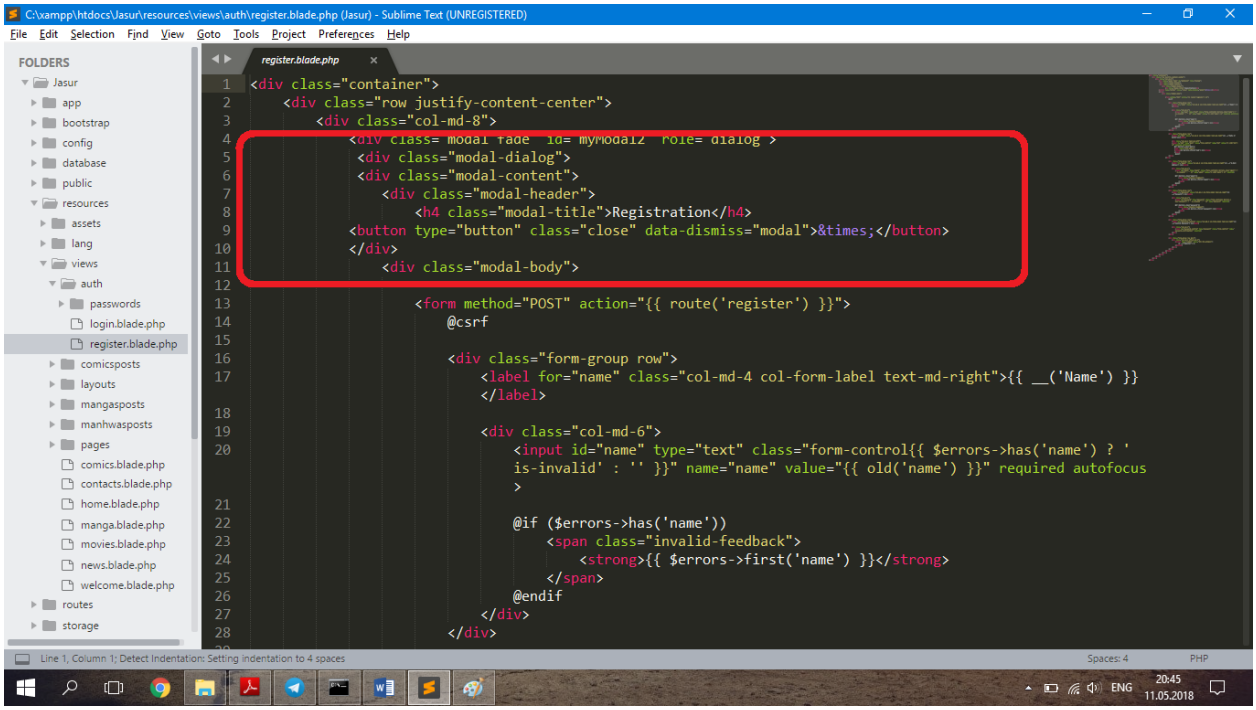
Finally, register-login operations are made easily as they are built-in laravel project. We just slightly modified it. “Date of birth” option was added. Also, you can notice the popup window, that was created using Modal class in bootstrapping.

Date of birth:



```
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="modal fade" id="myModal2" role="dialog">
             <div class="modal-dialog">
             <div class="modal-content">
                <div class="modal-header">
                    <h4 class="modal-title">Registration</h4>
            <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>
                <div class="modal-body">

                    <form method="POST" action="{{ route('register') }}">
                        @csrf

                        <div class="form-group row">
                            <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>

                            <div class="col-md-6">
                                <input id="name" type="text" class="form-control{{ $errors->has('name') ? ' is-invalid' : '' }}" name="name" value="{{
                                old('name') }}" required autofocus>

                                @if ($errors->has('name'))
                                    <span class="invalid-feedback">
                                        <strong>{{ $errors->first('name') }}</strong>
                                    </span>
                                @endif
                            </div>
                        </div>

                        <div class="form-group row">
                            <label for="DoB" class="col-md-4 col-form-label text-md-right">{{ __('Date of Birth')}}</label>

                            <div class="col-md-6 text-md-left">
                            <input id="dob" type="date" class="form-control" name="dob" value="{{ old('dob') }}" required autofocus>
                                @if ($errors->has('dob'))
                                <span class="help-block">
                                <strong>{{ $errors->first('dob') }}</strong>
                                </span>
                                @endif
                            </div>
                        </div>

                        <div class="form-group row">
                            <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>

                            <div class="col-md-6">
                                <input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{
```

Modal addition:



Summary of each member's contribution to the project:

Overall, the project was fairly done by whole group. There were many minor additions and changes by each member, but here are the main contributions of every person:

Nodirjon Utkirov: started the project, created the layouts, worked on css, backgrounds

Jasur Turaev: register-login models, API from YouTube

Saidazimkhon Jasurbekov: created routes models for comics' migrations

Ibrohim Orifjonov: extended comics to manga and manhwa

Farrukh Tulkunov: comments, pagination, favorite, and some changes to layouts

You can check site in authorized view (profiles):

1) Login: jasur.turayev@mail.ru

   Password: definition

2) Login: ftulkunov@gmail.com

   Password: qwerty