



WEB PROGRAMIRANJE PHP2

DOKUMENTACIJA PROJEKTA

Student:

Никола Митровић 147/14

Beograd 2018.

SADRŽAJ

1. Opis funkcionalnosti.....	1
2. Skica struktura stranica.....	1
3. Dijagram baze podataka.....	2
3. MVC Organizacija.....	3
Kontroleri.....	3
Modeli.....	3
View.....	3
4. javascript kod.....	4
5. PHP kod.....	5






1.Opis funkcionalnosti




При приступу сајта приказује се страница за регистрацију или логовање, неауторизовани корисник не може ићи даље од ње. Валидација регуларним изразима за регистрацију (email). Када се корисник региструје/улогује добија приступ сајту где може да оставља твитове, коментарише на твитове, гласа на твитове, мења своју слику. Пагинација је урађена на твитовима, макс пет твита по страни. Коментари нису учитани на страну док год се не кликне на view comments и онда се за одређени твит довлаче из базе преко ајакса и исписују кориснику. Анкета је урађена преко ајакса, корисник који гласа једном на твит не може то урадити поново. Ако се улогује администратор добија приказ додатних линкова (такође из базе, на свакој страници су подешени у AppServiceProvider) и може управљати ентитетима на сајту (Брисање, едитовање, додавање). Свака акција корисника се записује у лог фајл. Урађени изузеци.

Korišćeni jezici	<ul style="list-style-type: none">• HTML4 i HTML 5• CSS• JavaScript• jQuery• PHP• Laravel• MySQL• AJAX
Korišćeni HTML šabloni (framework-ci)	Bootstrap https://getbootstrap.com/
Korišćene biblioteke i njihovi url-ovi odakle ste ih preuzeli	<ul style="list-style-type: none">• JQuery• https://jquery.com/
Korišćene biblioteke iz framework-a	<ul style="list-style-type: none">• LARAVEL
Korišćene php biblioteke van fremework-a	
Koje funkcionalnosti su realizove putem AJAX-a	Ispisivanje komentara odredjenog tvita kada se klikne na view comments, glasanje na svakom tvitu
Navești lokacije gde je napisan AJAX kod (fajlovi ili delovi html stranice)	public/js/script.js
Navești lokacije gde je napisan sopstveni javaScript kod (fajlovi ili delovi html stranice)	public/js/scriptj.s
URL sajta/aplikacije	https://jellytot95.000webhostapp.com/public/index.php
Pristupni parametri	Poslati su na mail prof. Danijeli Nikitin

2. Skica struktura stranica

<p>Name</p> <input type="text" value="Name"/>	<p>Email</p> <input type="text" value="Email"/>
<p>Email</p> <input type="text" value="Email"/>	<p>Password</p> <input type="text" value="Password"/>
<p>Password</p> <input type="text" value="Password"/>	<p>Sign in</p>
<p>Confirm password</p> <input type="text" value="Repeat password"/>	
<p>Register</p>	

Home Profile Users Comments Tweets Links Ratings About Logout Documentation		
 <p>Edit your profile</p>	<p>Tweet:</p> <input type="text" value="What's happening?"/> <p>Tweet it!</p> <hr/> <div> Korisnik 45 minutes ago</div> <p>LET'S GO</p> <p>☆☆☆☆</p> <div> Korisnik 45 minutes ago</div> <p>View comments</p> <p>0 Comments</p> <p>PAGINATE</p>	<div> Korisnik</div> <div> Administrator</div>

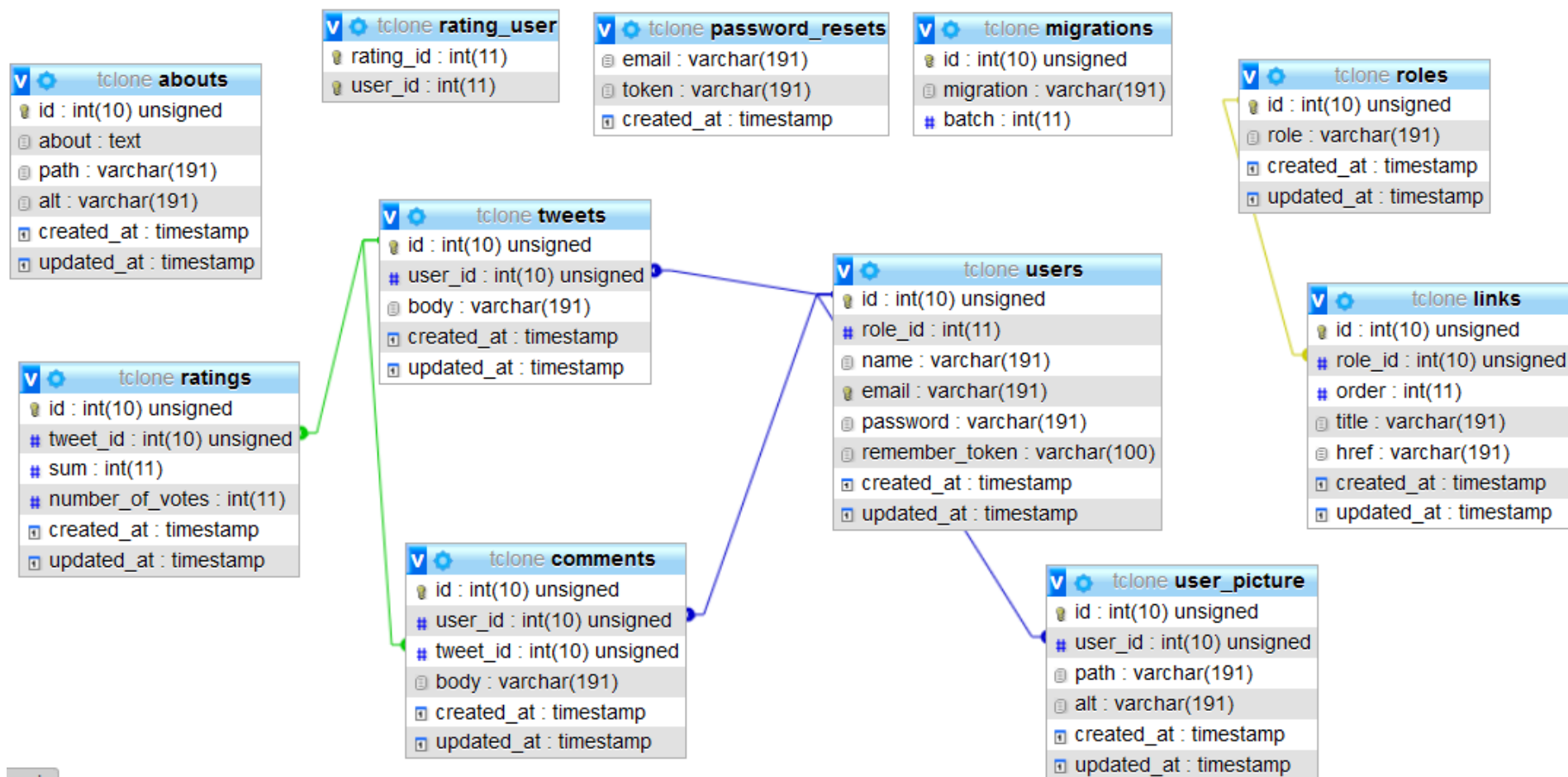
Home Profile Users Comments Tweets Links Ratings About Logout Documentation		
<div><p>Administrator</p><hr/><p><input type="text" value="Choose File"/> Upload</p><p>Change picture</p></div>	<p>Tweet:</p> <input type="text" value="What's happening?"/> <p>Tweet it!</p> <hr/> <div> Administrator 3 hours ago</div> <p>Welcome, everybody!</p> <p>Delete</p> <p>View comments</p> <p>2 comment</p> <p>Rating: 5.0</p> <div> Administrator 3 hours ago</div> <p>Delete</p> <p>Feel free to start tweeting, commenting</p>	

ID	Comment	Author	Tweet	Created	Updated	Edit	Delete
1	sadadada	Korisnik	Feel free to start tweeting, commenting, rating, changing your profile etc	2018-03-17 18:45:16	2018-03-17 18:45:16	Edit	Delete
2	sadada	Korisnik	Welcome, everybody!	2018-03-17 18:54:38	2018-03-17 18:54:38	Edit	Delete
3	sadadsadasdsada	Korisnik	Welcome, everybody!	2018-03-17 18:57:31	2018-03-17 18:57:31	Edit	Delete

Comment:

Tweet:

3. Dijagram baze podataka



3.MVC Organizacija

Kontroleri

Admin/ CommentsController
index(\$id)
store()
update(\$id)
destroy(\$id)

Admin/LinksController
index(\$id)
store()
update(\$id)
destroy(\$id)

Admin/RatingsController
index(\$id)
store()
update(\$id)
destroy(\$id)

Admin/TweetsController
index(\$id)
store()
update(\$id)
destroy(\$id)

Admin/UsersController
index(\$id)
store()

update(\$id)
destroy(\$id)

AboutController
index()

AuthController
construct()
index()
register()
login()
destroy()

CommentsController
construct()
store(\$tweet)

ProfileController
construct()
showUser(\$id)
editPicture()

TweetsController
construct()
index()
store()
showTweetComments()
delete(\$tweet)
vote(\$id, \$value)

About

Modeli

Comment
tweet()
user()
showComments()

Link
role()
links()

Rating
tweet()
users()

Role
users()
links()

Tweet
user()
comments()
rating()
latestTweets()
numberOfComments()
addComment(\$comment)

User
tweets()
comments()
picture()
role()
ratings()
postTweet(\$tweet)

UserPicture
user()

View

Spisak view fajlova
admin.blade.php
comments.blade.php
links.blade.php
users.blade.php
errors.blade.php
footer.blade.php
nav.blade.php
layout.blade.php
comment.blade.php
tweet.blade.php
about.blade.php
home.blade.php
user.blade.php
welcome.blade.php

4. JavaScript kod

public/js/script.js

```
$(document).ready(function(){

    $(".show-comments").on('click', function(){

        $(".response-comments").empty();

        var tweet_id = this.id;

        $.ajax({
            type: 'GET',
            // headers: {
            //     'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            // },
            url : BASE_URL + "/comments",
            _token : TOKEN,
            // url: "http://localhost/clone/public/comments",
            data: {
                id : tweet_id
            },
            success: function (response) {
                console.log(response);
                var text = "";
                if (response.length>0){
                    $.each(response, function(index, value){
                        console.log(value['body']);
                        text+="
```

```

$(".vote").on('click', function(){
    var tweet_id = this.id;
    var value = this.value;
    var parent = $(this).parent();
    $.ajax({
        url: BASE_URL + 'vote/' +tweet_id +'/value/' + value,
        type: "GET",
        data: {
            tweet_id : tweet_id,
            value : value,
            _token : TOKEN
        },
        success: function(response){
            var text = '<i>Rating: ' +response+ '</i>';
            parent.html(text);
        }
    });
});

document.getElementById("uploadBtn").addEventListener('change', function(){
    document.getElementById("uploadFile").value = this.value;
});

});

function mail(){
    var mail = document.getElementById('email');
    var regMail = /^\\w+@[a-zA-Z_]+?\\.[a-zA-Z]{2,3}$/;
    if(!regMail.test(mail.value)) {
        mail.style.borderColor='red';
    }
    else {
        mail.style.borderColor='green';
    }
}

```

5. PHP kod

app/Providers/AppServiceProvider.php

```
public function boot(){  
    view()->composer('*', function($view){  
        $view->with('links', \App\Link::links());  
    });  
}
```

Controllers/Admin/CommentsController.php

```
<?php
```

```
namespace App\Http\Controllers\Admin;
```

```
use Illuminate\Http\Request;  
use App\Http\Controllers\Controller;  
use App\Comment;  
use App\Tweet;
```

```
class CommentsController extends Controller  
{
```

```
    private $data = [];
```

```
    public function index($id = null)  
    {
```

```
        //
```

```
        $this->data['comments'] = Comment::with('tweet', 'user')->get();  
        $this->data['tweets'] = Tweet::all();
```

```
        if (isset($id)) {  
            $this->data['single'] = Comment::find($id);
```

```

    }

    return view('admin.comments', $this->data);
}

public function store(Request $request)
{
    //
    $validated = $request->validate([
        'comment' => 'required',
        'tweet' => 'required|not_in:0'
    ]);

    try {
        $comm = Comment::create([
            'body' => $validated['comment'],
            'tweet_id' => $validated['tweet'],
            'user_id' => auth()->user()->id
        ]);

        \Log::info(auth()->user()->name . " has commented through admin panel. Comment: "
        . $comm->body . ", TweetID: " . $comm->tweet_id);
        return redirect()->back();
    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

public function update(Request $request, $id)
{
    //
    $validated = $request->validate([
        'comment' => 'required',
        'tweet' => 'required|not_in:0'
    ]);

    try {
        $comment = Comment::find($id);

        $comment->body = $validated['comment'];
        $comment->tweet_id = $validated['tweet'];

        $comment->save();
    }
}

```

```

        \Log::info(auth()->user()->name . " has updated through admin panel. Comment: " .
$comment->body . ", TweetID: " . $comment->tweet_id);
        return redirect()->back();
    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

public function destroy($id)
{
    //
    try {
        $comm = Comment::destroy($id);
        \Log::info(auth()->user()->name . " has deleted comment ");
        return redirect()->back();
    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}
}

```

Controller/Admin/LinksController.php

```
<?php
```

```
namespace App\Http\Controllers\Admin;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Http\Request;
```

```
use App\Link;
```

```
use App\Role;
```

```
class LinksController extends Controller
```

```
{
```

```
    private $data = [];
```

```
    public function index($id = null)
```

```

{
    //
    $this->data['links'] = Link::all();
    $this->data['roles'] = Role::all();

    if(isset($id)){
        $this->data['single'] = Link::find($id);
    }

    return view('admin.links', $this->data);
}

public function store(Request $request)
{
    //

    $validated = $request->validate([
        'role' => 'required|not_in:0',
        'order' => 'required',
        'title' => 'required',
        'href' => 'required'
    ]);

    try {
        Link::create([
            'role_id' => $validated['role'],
            'order' => $validated['order'],
            'title' => $validated['title'],
            'href' => $validated['href']
        ]);

        \Log::info('Link inserted through admin panel by ' . auth()->user()->name);
        return redirect()->back();
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

public function update(Request $request, $id)
{

```



```

//
$validated = $request->validate([
    'role' => 'required|not_in:0',
    'order' => 'required',
    'href' => 'required',
    'title' => 'required'
]);

try {
    $link = Link::find($id);

    $link->role_id = $validated['role'];
    $link->order = $validated['order'];
    $link->href = $validated['href'];
    $link->title = $validated['title'];

    $link->save();

    \Log::info('Link updated through admin panel by '.auth()->user()->name);
    return redirect('/admin/links');
}
catch (\Illuminate\Database\QueryException $ex) {
    \Log::error($ex->getMessage());
    return redirect()->back();
}
}

public function destroy($id)
{
    try {
        Link::destroy($id);

        \Log::info('Link deleted via admin panel by '.auth()->user()->name);
        return redirect()->back();
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}
}

```

Controllers/Admin/RatingsController.php

```
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Rating;

class RatingsController extends Controller
{
    //
    private $data = [];

    public function index($id = null){

        $this->data['ratings'] = Rating::with('tweet', 'users')->get();

        if(isset($id)){

            $this->data['single'] = Rating::find($id);

        }

        return view('admin.ratings', $this->data);
    }

    public function store(){

        $validated = request()->validate([
            'tweet' => 'required|not_in:0',
            'sum' => 'required|integer',
            'number' => 'required|integer'
        ]);

        try {
            Rating::create([
                'tweet_id' => $validated['tweet'],
                'sum' => $validated['sum'],
                'number_of_votes' => $validated['number']
            ]);
        }
        \Log::info('New rating added by '. auth()->user()->name);
        return redirect('admin/ratings');
```

```

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

```

```

public function destroy($id){

```

```

    try {
        Rating::destroy($id);
        \Log::info('Rating destroyed!');
        return redirect()->back();
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

```

```

public function update($id){

```

```

    $validated = request()->validate([
        'tweet' => 'required|not_in:0',
        'sum' => 'required|integer',
        'number' => 'required|integer'
    ]);

    try {
        $rate = Rating::find($id);

        $rate->sum = $validated['sum'];
        $rate->number_of_votes = $validated['number'];
        $rate->tweet_id = $validated['tweet'];

        $rate->save();
        \Log::info('Rating updated');
        return redirect('admin/ratings');
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

```

```
}
```

Controllers/Admin/TweetsController.php

```
<?php
```

```
namespace App\Http\Controllers\Admin;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Tweet;
```

```
class TweetsController extends Controller
```

```
{
```

```
    private $data = [];
```

```
    public function index($id = null)
```

```
    {
```

```
        //
```

```
        $this->data['tweets'] = Tweet::with('user')->get();
```

```
        if(!empty($id)){
```

```
            $this->data['single'] = Tweet::with('user')->where('id',$id)->first();
```

```
        }
```

```
        return view('admin.tweets', $this->data);
```

```
    }
```

```
    public function store(Request $request)
```

```
    {
```

```
        //
```

```
        $validated = $request->validate([
```

```
            'tweet' => 'required'
```

```
        ]);
```

```
        try {
```

```
            auth()->user()->postTweet(new Tweet(['body' => $validated['tweet']]));
```

```
            \Log::info(auth()->user()->name. ' created new tweet');
```

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

```
        }
```

```

        catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back();
        }
    }

    public function update(Request $request, $id)
    {
        //
        $validated = $request->validate([
            'tweet' => 'required'
        ]);

        try {
            $tweet = Tweet::find($id);

            $tweet->body = $validated['tweet'];

            $tweet->save();
            \Log::info('Tweet updated by '.auth()->user()->name);
            return redirect()->back();
        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back();
        }
    }

    public function destroy($id)
    {
        try {
            Tweet::destroy($id);
            \Log::info('Tweet is deleted by '.auth()->user()->name);
            return redirect()->back();
        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back();
        }
    }
}

```

Controllers/Admin/UsersController.php

```
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\User;
use App\Role;
use App\UserPicture;

class UsersController extends Controller
{

    private $data = [];

    public function index($id = null)
    {
        $this->data['roles'] = Role::all();
        $this->data['users'] = User::with('picture')->get();

        if(!empty($id)){
            $this->data['single'] = User::with('picture')->where('id',$id)->first();
        }

        return view('admin.users', $this->data);
    }

    public function store(Request $request)
    {
        //
        $validated = $request->validate([
            'name' => 'required|min:5|alpha',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|min:1',
            'role' => 'required|integer|not_in:0',
            'picture' => 'image|mimes:png,jpg,jpeg'
        ]);

        try {
            $user = User::create([
                'name' => $validated['name'],
                'password' => bcrypt($validated['password']),
                'role_id' => $validated['role'],
            ]);
        } catch (\Exception $e) {
            //
        }
    }
}
```

```

        'email' => $validated['email']
    ]);

    UserPicture::create(['user_id' => $user->id]);
    \Log::info('New user was created: '.$user->name.", ".$user->id.", by user ".$user->name());
    return redirect()->back();
} catch (\Illuminate\Database\QueryException $ex) {
    \Log::error($ex->getMessage());
    return redirect()->back();
}
}

public function update(Request $request, $id)
{
    //
    $request->validate([
        'name' => 'required|min:5|alpha',
        'email' => 'required|email',
        'password' => 'required|min:1',
        'role' => 'required|integer|not_in:0',
        'picture' => 'image'
    ]);

    try {
        $user = User::find($id);

        $user->email = $request->email;
        $user->name = $request->name;
        $user->password = bcrypt($request->password);
        $user->role_id = $request->role;

        $user->save();

        //

        $image = request()->file('picture');

        $name = "user_id_" . $id . "." . $image->getClientOriginalExtension();

        $image->move(public_path("/images/profile"), $name);
    }
}

```

```

        $picture = UserPicture::where('user_id', $id)->first();

        $picture->path = $name;

        $picture->save();

        \Log::info('Picture updated!');

        return redirect('admin/users');
    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}

public function destroy($id)
{
    //
    try {
        $user = User::with('picture')->where('id',$id)->first();

        $path = public_path('images/profile/'.$user->picture->path);

        $user->delete();

        \File::delete($path);
        \Log::info('User and his picture deleted');
        return redirect()->back();
    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }
}
}

```

Controllers/AboutController.php

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App>About;

```



```

class AboutController extends Controller
{
    //
    public function index(){

        $about = About::all();

        return view('pages.about', compact('about'));

    }
}

```

Controllers/AuthController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\User;
use App\UserPicture;

class AuthController extends Controller
{

    public function __construct()
    {

        $this->middleware('guest', ['except' => 'destroy']);

    }

    public function index()
    {

        return view('pages.welcome');

    }

    public function register(Request $request)
    {

        $validation_rules = [

```

```

'name' => 'required|regex:/^[A-z]{4,15}+$/',
'email' => 'required|email|unique:users,email',
'password' => 'required|min:6',
'password' => 'required|confirmed'
];

$validation_messages = [
'regex' => 'Name must be atleast 4 characters and must contain only letters',
];

$data = $request->validate($validation_rules, $validation_messages);

try {
    $user = User::create(['role_id' => 2, 'email' => $data['email'], 'name' => $data['name'],
'password' => bcrypt($data['password'])]);

    UserPicture::create(['user_id' => $user->id]);

    auth()->login($user);
    \Log::info("New user registered! ".$user->email."; ".$user->name);
    return redirect()->home();
} catch (Illuminate\Database\QueryException $ex) {
    \Log::error($ex->getMessage());
    return redirect()->back();
}

}

public function login(Request $request)
{
    if (auth()->attempt(['email' => $request->email, 'password' => $request->password]))
    {
        \Log::info('User logged in! ' . auth()->user()->name);
        return redirect('/home');
    }
    return back()->with("Message", "Wrong email or password");
}

public function destroy()
{
    auth()->logout();
    return redirect('/');
}

```

```
}  
}
```

Controllers/CommentController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Tweet;
```

```
class CommentsController extends Controller  
{
```

```
    public function __construct()  
    {
```

```
        $this->middleware('auth');  
    }
```

```
    public function store(Tweet $tweet)  
    {
```

```
        try {  
            $comment = request()->validate([  
                'comment' => 'required|min:5|max:50'  
            ]);
```

```
            $tweet->addComment($comment['comment']);  
            \Log::info('Comment added by user ' . auth()->user->email);  
            return back();  
            return redirect()->back();  
        }
```

```
        catch (\Illuminate\Database\QueryException $ex) {  
            \Log::error($ex->getMessage());  
            return redirect()->back();  
        }
```

```
    }
```

```
}
```

Controllers/ProfileController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\User;
```

```
use App\UserPicture;
```

```
class ProfileController extends Controller
```

```
{
```

```
    private $model;
```

```
    public function __construct()
```

```
    {
```

```
        $this->middleware('auth');
```

```
        $this->model = new User;
```

```
    }
```

```
    public function showUser($id = null)
```

```
    {
```

```
        $user = $id ? User::find($id) : User::find(auth()->user()->id);
```

```
        return view('pages.user', compact('user'));
```

```
    }
```

```
    public function editPicture()
```

```
    {
```

```
        $rules = [
```

```
            'picture' => 'required|image'
```

```
        ];
```

```
        $validation_message = [
```

```
            'image' => 'Image is not in supported format',
```

```
        ];
```

```
        request()->validate($rules, $validation_message);
```

```

try {
    $image = request()->file('picture');

    $name = "user_id_" . auth()->user()->id . "." . $image->getClientOriginalExtension();

    $image->move(public_path("/images/profile"), $name);

    $picture = UserPicture::where('user_id', auth()->user()->id)->first();

    $picture->path = $name;

    $picture->save();
    \Log::info('User ' . auth()->user()->email . " has updated his profile picture");
    return redirect()->back();
}
catch (\Illuminate\Database\QueryException $ex) {
    \Log::error($ex->getMessage());
    return redirect()->back();
}
}
}

```

Controllers/TweetsController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Tweet;
use App\Rating;
use App\User;
use App\Comment;

class TweetsController extends Controller
{

    public function __construct()
    {

```

```

        $this->middleware('auth');

    }

    public function index()
    {

        $user = new User;

        $users = $user->orderBy('created_at', 'desc')->paginate(5);

        $tweet = new Tweet;

        $tweets = $tweet->latestTweets();

        return view('pages.home', compact('tweets', 'users'));

    }

    public function store(Request $request)
    {

        $validated = $request->validate([
            'tweet' => 'required|min:5|max:150'
        ]);

        auth()->user()->postTweet(new Tweet(['body' => $validated['tweet']]));

        return back();
    }

    public function showTweetComments()
    {

        if (request()->has('id')) {

            try {
                \Log::info("User ".auth()->user()->email. 'is viewing comments');
                return Comment::showComments();
            }
            catch (\Illuminate\Database\QueryException $ex) {
                \Log::error($ex->getMessage());
                return redirect()->back();
            }
        }
    }

```

```

    }
}

public function delete(Tweet $tweet)
{

    try {
        $tweet->delete();
        \Log::info("Tweet deleted by ' .auth()->user()->name);
        return redirect()->back();
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }

}

public function vote($id, $value)
{

    try {
        $rating = Rating::where('tweet_id', $id)->first();

//      $user = User::find(auth()->user()->id);

        $rating->users()->attach(auth()->user()->id);

        $rating->sum += $value;
        $rating->number_of_votes += 1;

        $rating->save();
        \Log::info('User ' . auth()->user()->email." successfully voted on tweet with the id of " .
$rating->id);
        return number_format((float)$rating->sum / $rating->number_of_votes, 1, '.', '');
    }
    catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back();
    }

}

```

```
}
```

App/About.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class About extends Model
```

```
{
```

```
    //
```

```
}
```

App/Comment.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Comment extends Model
```

```
{
```

```
    //
```

```
    protected $guarded = [];
```

```
    public function tweet()
```

```
{
```



```

        return $this->belongsTo(Tweet::class);

    }

    public function user()
    {

        return $this->belongsTo(User::class);

    }

    public static function showComments() {

        return static::where('tweet_id', request()->id)->with('tweet', 'user')-
>orderBy('created_at', 'desc')->get();

    }
}

```

App\Link.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Link extends Model
```

```
{
```

```
//
```

```
protected $guarded = [];
```

```

public function role()
{
    return $this->belongsTo(Role::class);
}

public static function links()
{
    return static::with('role')->orderBy('order','asc')->get();
}
}

```

App\Rating.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Rating extends Model
{
    //
    protected $guarded = [];
    public $timestamps = false;

    public function tweet()

```

```

{

    return $this->belongsTo(Tweet::class);

}

public function users()
{
    return $this->hasMany(User::class);
}
}

```

App\Role.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
    //

    public function users()
    {

        return $this->hasMany(User::class);
    }
}

```

```

    }

    public function links()
    {

        return $this->hasMany(Link::class);

    }
}

```

App\Tweet.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Tweet extends Model
{

    protected $guarded = [];

    public function user()

```

```
{

    return $this->belongsTo(User::class);

}

public function comments()
{

    return $this->hasMany(Comment::class);

}

public function rating()
{

    return $this->hasOne(Rating::class);

}

public function latestTweets()
{

    return $this->with('user')->orderBy('created_at', 'desc')->paginate(5);

}
```

```
public function numberOfComments()
{

    return $this->comments->where('tweet_id', $this->id)->count();

}
```

```
public function addComment($comment)
{

    $this->comments()->create(['body' => $comment, 'user_id' => auth()->id()]);

}
}
```

App\User.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Notifications\Notifiable;
```

```
use Illuminate\Foundation\Auth\User as Authenticatable;
```

```
use App\Rating;
```

```

class User extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'role_id',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public function tweets()
    {

        return $this->hasMany(Tweet::class);
    }
}

```

```
}
```

```
public function comments()
```

```
{
```

```
    return $this->hasMany(Comment::class);
```

```
}
```

```
public function picture()
```

```
{
```

```
    return $this->hasOne(UserPicture::class);
```

```
}
```

```
public function role()
```

```
{
```

```
    return $this->belongsTo(Role::class);
```

```
}
```

```
public function ratings()
```

```
{
```

```
    return $this->belongsToMany(Rating::class);
```

```
}
```



```

public function postTweet(Tweet $tweet)
{

    $this->tweets()->save($tweet);

    Rating::create([
        'tweet_id' => $tweet->id,
        'sum' => 0,
        'number_of_votes' => 0
    ]);

}
}

```

App\UserPicture.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class UserPicture extends Model
{
    protected $guarded = [];

    protected $table = 'user_picture';
}

```

```

public function user()
{

    return $this->belongsTo(User::class);

}
}

```

[app/Http/Middleware/ActivityLog.php](#)

```

<?php

namespace App\Http\Middleware;

use Closure;

class ActivityLog
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (auth()->user()) {
            \Log::info('UserID: ' . auth()->user()->id . ' with the ip of ' . $request->ip() . ", using " .
            $request->userAgent() . " is visiting " . $request->getRequestUri() . " with " . $request-
            >method() . " method");
        } else {
            \Log::info('Unregistered user with the ip of ' . $request->ip() . ", using " . $request-
            >userAgent() . " is visiting " . $request->getRequestUri() . " with " . $request->method() . "
            method");
        }
        return $next($request);
    }
}

```

```
}
```

[app/Http/Middleware/AdminMiddleware.php](#)

```
<?php
```

```
namespace App\Http\Middleware;
```

```
use Closure;
```

```
class AdminMiddleware
```

```
{
```

```
    /**
```

```
     * Handle an incoming request.
```

```
     *
```

```
     * @param \Illuminate\Http\Request $request
```

```
     * @param \Closure $next
```

```
     * @return mixed
```

```
     */
```

```
    public function handle($request, Closure $next)
```

```
    {
```

```
        if (auth()->check()){
```

```
            if (auth()->user()->role_id == '1'){
```

```
                return $next($request);
```

```
            }
```

```
        }
```

```
        return abort(404);
```

```
    }
```

```
}
```

[views/admin/admin.blade.php](#)

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
    <!-- Required meta tags -->
```

```

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="csrf-token" content="{{ csrf_token() }}">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/
font-awesome.min.css">

<!-- Bootstrap CSS -->

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

        integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

<link rel="stylesheet" href="{{ asset('css/admin.css') }}">

<title>@yield('title')</title>

<script>

    var BASE_URL = "{{ asset('/') }}";

    var TOKEN = "{{ csrf_token() }}";

</script>

</head>

<body>

@include ('layouts.partials.nav')


{{--<div class="main">--}}

    @yield('content')

{{--</div>--}}


@include ('layouts.partials.footer')

```

views/admin/comments.blade.php

views/admin/links.blade.php

views/admin/ratings.blade.php

views/admin/tweets.blade.php

views/admin/users.blade.php

Мењају се само атрибути

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<!-- Required meta tags -->
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

```
<meta name="csrf-token" content="{{ csrf_token() }}">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/
font-awesome.min.css">
```

```
<!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
```

```
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
<link rel="stylesheet" href="{{ asset('css/admin.css') }}">
```

```
<title>@yield('title')</title>
```

```
<script>
```

```
var BASE_URL = "{{ asset('/') }}";
```

```
var TOKEN = "{{ csrf_token() }}";
```

```
</script>
```

```
</head>
```

```
<body>
```

```
@include ('layouts.partials.nav')
```

```
{{--<div class="main">--}}
```

```
@yield('content')
```

```
{{--</div>--}}
```

```
@include ('layouts.partials.footer')
```

```
views/layouts/partials/errors.blade.php
```

```
@if(count($errors))
```

```
<div class="alert alert-danger">
```

```
<ul>
```

```
@foreach ($errors->all() as $error)
```

```
<li>{{ $error }}</li>
```

```
@endforeach
```

```
</ul>
```

```
</div>
```

```
@endif
```

```
views/layouts/partials/footer.blade.php
```

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
```

```
integrity="sha384-KJ3o2DKtIkVYIK3UENzmM7KCKr/rE9/Qpg6aAZGJwFDMVNA/
GpGFF93hXpG5KkN"
```

```
crossorigin="anonymous"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
```

```
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvX
usvfa0b4Q"
```

```
crossorigin="anonymous"></script>
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
```

```
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+7
6PVCmYI"
```

```
crossorigin="anonymous"></script>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

```
@yield('script')
```

```
</body>
```

```
</html>
```

[views/layouts/partials/nav.blade.php](#)

```
<nav class="navbar navbar-expand-md bg-info navbar-dark">
```

```
<div class="container">
```

```
@if(auth()->check())
```

```
<ul class="navbar-nav">
```

```
<div class="collapse navbar-collapse" id="collapsibleNavbar">
```

```
@if(auth()->user()->role->id == '2')
```

```
@foreach($links->where('role_id','2') as $link)
```

```
<li class="nav-item">
```

```
<a class="nav-link"
```

```
href="{{asset('/'.$link->href)}}">{{ $link->title }}</a>
```

```

        </li>

    @endforeach

    @else

        @foreach($links as $link)

            <li class="nav-item">

                <a class="nav-link"
href="{{asset('/'.$link->href)}}">{{ $link->title }}</a>

            </li>

        @endforeach

    @endif

</ul>

@endif

</div>

</div>

</nav>

```

views/pages/about.blade.php

```

@extends('layouts.layout')

@section('content')

<div class="container jumbotron">

    @foreach($about as $a)

        <p>{{ $a->about }}</p>

        alt }}" width="300px" height="400px">

    @endforeach

</div>

@endsection

```


views/pages/home.blade.php

```
@extends('layouts.layout')
```

```
@section('title', 'Home')
```

```
@section('content')
```

```
<div class="container">
```

```
<div class="jumbotron">
```

```
<div class="row content">
```

```
<div class="col-lg-3 sidenav">
```

```
<div class="jumbotron user">
```

```
<a href="{{ asset('/user/'.auth()->user()->id) }}">user()->picture->alt }}">
```

```
<p>Edit your profile</p></a>
```

```
</div>
```

```
</div>
```

```
<div class="col-lg-6 posts tweets">
```

```
@include('layouts.partials.errors')
```

```
<div class="form-group">
```

```

<form action="{{asset('/tweets')}}" method="POST">

    {{ csrf_field() }}

    <div class="tweets">

        <div class="form-group">

            <label for="tweet">Tweet:</label>

            <input type="text" class="form-control" id="tweet"
placeholder="What's happening?"
                name="tweet">

            </div>

            <button type="submit" class="btn btn-success">Tweet it!</button>

        </div>

    </form>

</div>

<hr>

<div class="blog-main">

    @foreach($tweets as $tweet)

        @include('pages.partials.tweet')

    @endforeach

    <div class="links">

        {{ $tweets->links() }}

    </div>

</div>

</div>

<div class="col-lg-3 sidenav">

    <div class="jumbotron popular">

```

```

@foreach($users as $user)

    <a href="{{asset('/user/'.$user->id)}}">picture->alt }}"> {{ $user->name }} </a><br/>

    @endforeach

    {{ $users->links() }}

</div>

</div>

</div>

</div>

</div>

@endsection

@section('script')

    <script src="{{asset('/js/script.js')}}"></script>

@endsection

views/pages/user.blade.php

@extends('layouts.layout')

@section('title', 'User profile')

```

```

@section('content')

<div class="container">

    <div class="jumbotron">

        <div class="row content">

            <div class="user-image col-md-5">

                {{ $user->name }}

                <hr>

                picture->alt }}" />

            </div>

            @if($user->id == auth()->user()->id)

                <form action="{{ asset('/user') }}" method="POST" enctype="multipart/form-
data">

                    {{ csrf_field() }}

                    <input id="uploadFile" placeholder="Choose File" disabled="disabled" />

                    <div class="fileUpload btn btn-primary">

                        <span>Upload</span>

                        <input id="uploadBtn" type="file" class="upload" name="picture" />

                    </div>

                    <button type="submit">Change picture</button>

                </form>

            @endif

        </div>

```

```

<div class="offset-1"></div>

<div class="col-md-6 user-tweet">

    @include('layouts.partials.errors')

    <form action="{{ asset('/tweets') }}" method="POST">

        {{ csrf_field() }}

        <div class="col-lg-12 posts tweets">

            <div class="form-group">

                <label for="tweet">Tweet:</label>

                <input type="text" class="form-control" id="tweet" placeholder="What's
happening?" name="tweet">

            </div>

            <button type="submit" class="btn btn-success">Tweet it!</button>

        </div>

    </form>

    <hr>

    @foreach($user->tweets as $tweet)

        @include('pages.partials.tweet')

    @endforeach

</div>

</div>

</div>

</div>

@endsection

```

```
@section('script')
```

```
<script src="{{asset('/js/script.js')}}"></script>
```

```
@endsection
```

`views/pages/welcome.blade.php`

```
@extends('layouts.layout')
```

```
@section('title', 'Welcome')
```

```
@section('content')
```

```
<div class="container-fluid">
```

```
<div class="row content">
```

```
<div class="col-md-6 register">
```

```
@include('layouts.partials.errors')
```

```
<form method='post' action='welcome/register' class="form-horizontal"  
role="form">
```

```
{{ csrf_field() }}
```

```
<div class="form-group">
```

```

        <label for="name" class="col-lg-3 control-label">Name</label>

        <div class="col-lg-10">

            <input type="text" class="form-control" id="name" placeholder="Name"
name="name" required>

        </div>

    </div>

    <div class="form-group">

        <label for="inputEmail1" class="col-lg-3 control-label">Email</label>

        <div class="col-lg-10">

            <input type="email" class="form-control" id="email" onkeyup="mail();"
placeholder="Email" name="email"

                required>

        </div>

    </div>

    <div class="form-group">

        <label for="password" class="col-lg-3 control-label">Password</label>

        <div class="col-lg-10">

            <input type="password" class="form-control" id="password"
placeholder="Password"

                name="password" required>

        </div>

    </div>

    <div class="form-group">

        <label for="password" class="col-lg-3 control-label">Confirm
password</label>

        <div class="col-lg-10">

            <input type="password" class="form-control" id="password_confirmation"

                placeholder="Repeat password" name="password_confirmation"
required>

```

```

        </div>

    </div>

    <div class="form-group">

        <div class="col-lg-offset-2 col-lg-10">

            <button type="submit" class="btn btn-default"
name="register">Register</button>

        </div>

    </div>

</form>

</div>

<div class="col-md-6 login">

    <form class="form-horizontal" role="form" method="POST"
action="welcome/login">

        {{ csrf_field() }}

        <div class="form-group">

            <label for="inputEmail1" class="col-lg-2 control-label">Email</label>

            <div class="col-lg-10">

                <input type="email" class="form-control" id="inputEmail1"
placeholder="Email" name="email"

                    required>

            </div>

        </div>

        <div class="form-group">

            <label for="inputPassword1" class="col-lg-2 control-label"
required>Password</label>

            <div class="col-lg-10">

```



```
        <input type="password" class="form-control" id="inputPassword1"
placeholder="Password"
```

```
        name="password" required>
```

```
    </div>
```

```
</div>
```

```
<div class="form-group">
```

```
    <div class="col-lg-offset-2 col-lg-10">
```

```
        <button type="submit" class="btn btn-default">Sign in</button>
```

```
    </div>
```

```
</div>
```

```
</form>
```

```
@if(session('Message'))
```

```
    {{session('Message')}}
```

```
@endif
```

```
</div>
```

```
</div>
```

```
</div>
```

```
@endsection
```

```
<script src="{{asset('/js/script.js')}}"></script>
```

[views/pages/partials/comment.blade.php](#)

```
<div id="mymodal-{{ $tweet->id }}" class="modal fade" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel"
```

```
aria-hidden="true">
```

```
<div class="modal-dialog">
```

```
  <div class="modal-content">
```

```
    <div class="modal-header">
```

```
      <button type="button" class="close" data-dismiss="modal" aria-hidden="true">×</button>
```

```
    </div>
```

```
    <div class="modal-body">
```

```
      {{--TWEET BEGIN--}}
```

```
      <a href="{{asset('user/'.$tweet->user->id)}}">
```

```
        <span class="profile-icon">
```

```
          user->picture->alt }}">
```

```
        </span>{{ $tweet->user->name }}
```

```
      </a>
```

```
      {{ $tweet->created_at->diffForHumans() }}
```

```
    <div class='tweet-body'>
```

```
      {{ $tweet->body }}
```

```
    </div>
```

```
{{--TWEET END--}}
```

```
<div class="form-group">
```

```
    <form class="form-inline" method="POST" action="{{asset('/comments/'.  
$tweet->id)}}">
```

```
        {{ csrf_field() }}
```

```
        <input type="text" name="comment" class="form-control input-sm"
```

```
            placeholder="Type your response here">
```

```
        <button type="submit" id="add-comment" class="btn  
btn-info">Submit</button>
```

```
    </form>
```

```
</div>
```

```
<div class="form-group">
```

```
    <div class="response-comments" id="tweetid_{{ $tweet->id }}">
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
<div class="modal-footer">
```

```
    <button class="btn" data-dismiss="modal" aria-hidden="true">Close</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

views/pages/partials/tweet.blade.php

```
<a href="{{asset('user/'.$tweet->user->id)}}">
```

```
    <span class="profile-icon">
```

```
        user->picture->alt }}">
```

```
    </span>{{ $tweet->user->name }}
```

```
</a>
```

```
    {{ $tweet->created_at->diffForHumans() }}
```

```
<div class="post row">
```

```
    <div class='col-lg-8 tweet-body'>
```

```
        {{ $tweet->body }}
```

```
    </div>
```

```
<div class="col-lg-4">
```

```
    @if (auth()->id() == $tweet->user_id)
```

```
        <a class="btn btn-danger" href="{{asset('/delete/'.$tweet->id)}}">Delete</a>
```

```
    @endif
```

```
    <a href="#myModal-{{ $tweet->id }}" id='{{ $tweet->id }}' role="button" value='test'
```

```
        class="btn btn-info show-comments" data-toggle="modal">View comments</a>
```

```

    @if ($tweet->numberOfComments()==1)

        {{ $tweet->numberOfComments() }} comment

    @elseif ($tweet->numberOfComments()>1)

        {{ $tweet->numberOfComments() }} comment

    @else

        0 Comments

    @endif

    <br/>

</div>

@include('pages.partials.comment')

</div>

<div class="rates">

    @if(auth()->user()->ratings()->where('rating_id', $tweet->id)->exists())

        <div class="rating">

            <i>Rating: {{ number_format((float)$tweet->rating->sum/$tweet->rating-
            >number_of_votes, 1, '.', '') }}</i>

        </div>

    @else

        <div class="rating" id="rating">

            <button class='vote' id='{{ $tweet->id }}' value='5'>☆</button>

            <button class='vote' id='{{ $tweet->id }}' value='4'>☆</button>

            <button class='vote' id='{{ $tweet->id }}' value='3'>☆</button>

            <button class='vote' id='{{ $tweet->id }}' value='2'>☆</button>

            <button class='vote' id='{{ $tweet->id }}' value='1'>☆</button>

        </div>

    @endif

```

</div>

<hr>