

Visoka škola strukovnih studija za informacione i komunikacione tehnologije

DOKUMENTACIJA

Web programiranje PHP 2

github:

<https://github.com/MarijanaMitrovic/Laravel-readit-blog>

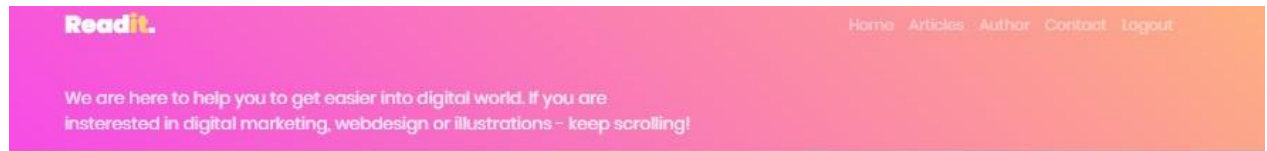
Student:

Marijana Mitrović 10/15

Sadržaj:

1. Slike stranica i opis funkcionalnosti
2. Struktura baze podataka
3. Slike organizacije foldera i fajlova
4. Kontroleri
5. Modeli
6. Custom request-ovi
7. Middleware-i
8. Rute
9. JS kod – ajax

1.



Digital sales - 2020 digital marketing trends

2020-03-24 13:18:00



Soft shadows in webdesign

2020-03-24 10:00:00

-Početna strana. Dinamički ispisan navigacioni meni i prikazano 5 najnovijih postova.



Digital sales - 2020 digital marketing trends

Digital sales trends 1. Assignment selling
Does your sales team use any of the content being produced by the



Soft shadows in webdesign

Soft shadows and floating elements create a pseudo-3D effect and make the design more layered and more interesting. - Alex Ivano ...



Search and seo digital marketing trends

The continued importance of local SEO
An increasing number of searchers are looking for businesses close to them, which has been

-Stranica Articles sa svim postovima, sortiranim od najnovijeg ka najstarijem. Klikom na određeni, može se videti njegov sadržaj.



Broken proportions Illustrations

As for digital illustration, the year 2019 let the trend of broken proportions get even more solid positions. It started getting established last year, yet this year it achieved the amazing diversity of styles, plots, exaggeration or tinification, and visual metaphors. Such an approach is definitely effective in setting visual originality, wished so much by brands. Still, to be done harmonically and beautifully, it requires real artistic talent from the illustrator: to break the rules, you need to know them perfectly!

VISITS : 3

1 Comments

Perica Peric

MARCH 26, 2020

Great post

Leave a comment

Type your comment here

Komentar

Post Comment

-Single post. Neautorizovan korisnik može videti fotografiju, sadržaj, komentare i broj poseta. Da bi ostavio komentar, mora biti prijavljen.

Register here

-Forma za registraciju. Provera kredencijala pomoću requesta i upis u bazu.

Login and access your account

You don't have account? [Register here](#)

-Login forma. Ukoliko korisnik postoji u bazi podataka, biće prijavljen.

Laza Lazic
MARCH 27, 2020

Komentar

DELETE

EDIT

Edit your comment

Type your reply here

Komentar

Update comment

-Autorizovan korisnik može menjati komentare koje je ranije ostavio i brisati ih. Takođe, administrator ima dozvolu da briše komentare. Korišćen Ajax.

Laza Lazic

MARCH 24, 2020

tekst komentara tekst komentara tekst komentaratekst komentara tekst komentara

REPLY

DELETE

Reply to user

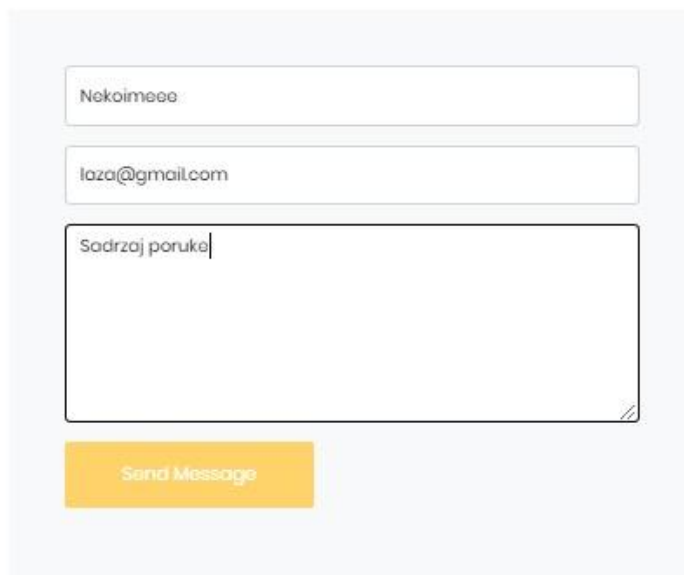
Type your reply here

Post Reply

-Administrator ispod svakog komentara ima dugme reply i mogućnost da odgovori korisniku na komentar.

Dodavanje odgovora od strane administratora, brisanje komentara i izmena od strane korisnika su rešeni uz upotrebu Ajax-a.

Feel free to send us message!



A contact form with three input fields and a submit button. The first field contains 'Nekoimeee', the second contains 'laza@gmail.com', and the third is a large text area with the placeholder 'Sadržaj poruke'. Below the fields is a yellow 'Send Message' button.

Nekoimeee

laza@gmail.com

Sadržaj poruke

Send Message

-Autorizovan korisnik može pristupiti stranici za kontakt. PHP mailerom se poruka šalje na mail administratoru. Email korisnika je automatski unet.

Napomena: Uklonjeni kredencijali administratora iz koda zbog bezbednosti, pa možda neće raditi. Sav ostali kod je vidljiv.

Marijana Mitrovic

My name is Marijana Mitrovic. I am 23 years old and I am currently attending my final year of studies at ICT college. This website was made using Laravel framework, for purposes of PHP2 course.



-Stranica O autoru sa kratkom biografijom i fotografijom.

ADMIN PANEL

The screenshot shows the Admin Panel interface. On the left is a dark sidebar with a menu containing: Posts, Users, Comments, Activity, and Pages. The main content area has a top navigation bar with links: Home, Articles, Author, Admin panel, and Logout. Below this is a section titled 'All posts' containing a table of posts.

Id	Title	Content	Updated at	Photo	Delete	Update
6	Digital sales - 2020 digital marketing trends	Digital sales trends 1. Assignment selling Doe ...	2020-03-24 13:18:00		Delete	Update
11	Soft shadows in webdesign	Soft shadows and floating elements create a pseudo ...	2020-03-24 10:00:00		Delete	Update
8	Search and seo digital marketing trends	The continued importance of local SEO An increas ...	2020-03-24 09:12:05		Delete	Update
13	Broken proportions illustrations	As for digital illustration, the year 2019 let the ...	2020-03-24 09:07:00		Delete	Update
7	Video marketing trends	Product-focused videos Fifty-two percent of consu ...	2020-03-24 07:23:14		Delete	Update

At the bottom of the table is a pagination control showing page 1 of 3, with links for previous, 1, 2, 3, and next.












-Inicijalno su prikazani svi postovi, sa opcijama za izmenu i brisanje. Klikom na Update, otvara se forma za izmenu popunjena trenutnim sadržajem posta.

Update post form

Title

Novi

Post content

Source Sans Pro
A


Izmenjeeeennnn

Category id (1- web, 2-marketing, 3-illustration)

2

File input

Choose file



Edit

-Forma za izmenu izabranog posta. Menja se i fotografija.

Maya Mitrovic

Posts

Users

Comments

Activity

Pages

[Home](#)
[Articles](#)
[Author](#)
[Admin panel](#)
[Logout](#)

All users

Id	Name	Surname	Email	Updated at	Role	Delete	Update
1	Marjana	Mitrovic	adminphp2saj1@gmail.com	0000-00-00 00:00:00	admin	Delete	Update
2	Pera	Peric	pera@gmail.com	0000-00-00 00:00:00	user	Delete	Update
3	Marjana	Mitrovic	mayaphp2@gmail.com	2020-03-17 12:16:00	admin	Delete	Update
4	Laza	Lazic	laza@gmail.com	2020-03-17 11:41:45	user	Delete	Update
5	Nekoizmenjen	Imelizmenjen	nekoizm@gmail.com	2020-03-26 10:27:29	user	Delete	Update

1
2
3

-Prikaz svih korisnika, kao i za postove, opcije za izmenu i brisanje.

Edit user form

Name
Nekoizmenjen

Last Name
Imeizmenjen

Email
nekoizm@gmail.com

Password
.....

Role id (1- admin, 2- user)
2

Edit

-Forma za izmenu izabranog korisnika.

Add user form

Name

Enter name

Last Name

Enter last name

Email

Enter email

Password

Password

Role id (1- admin, 2- user)

Enter role

Submit

-Forma za dodavanje novog korisnika

[Home](#)
[About Us](#)
[Contact Us](#)
[Privacy Policy](#)
[Terms & Conditions](#)

All comments

Id	Content	User's email	Post title	Updated at	Delete
16	Komentar	laza@gmail.com	Broken proportions Illustrations	2020-03-27 16:58:53	Delete
15	IZMENJEN	laza@gmail.com	Soft shadows in webdesign	2020-03-27 16:23:58	Delete
14	Opet alalalal	laza@gmail.com	Build a website in minutes with Templates	2020-03-27 12:44:09	Delete

[1](#)
[2](#)
[3](#)
[4](#)
[5](#)

-Prikaz svih komentara uz podatke korisnika.

Samo opcija za brisanje.

Napomena: urađena samo opcija za brisanje, jer ne bi bilo etički menjati sadržaj tuđih komentara ili ih dodavati u tuđe ime.

Select date:

03/26/2020

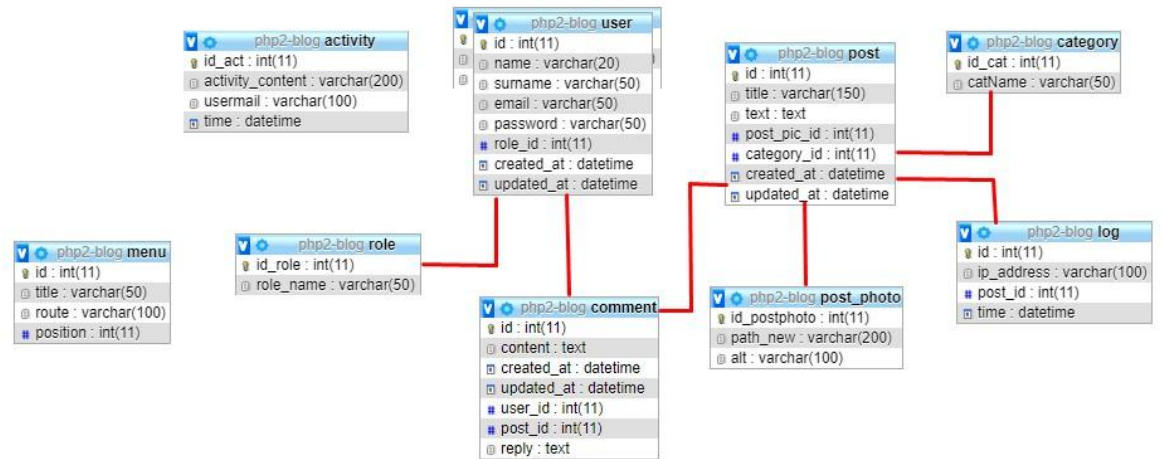
All activities

Activity	User	Time
User added comment	mika@gmail.com	2020-03-26 13:45:30
User added comment	pericinmail@mail.com	2020-03-26 13:44:06

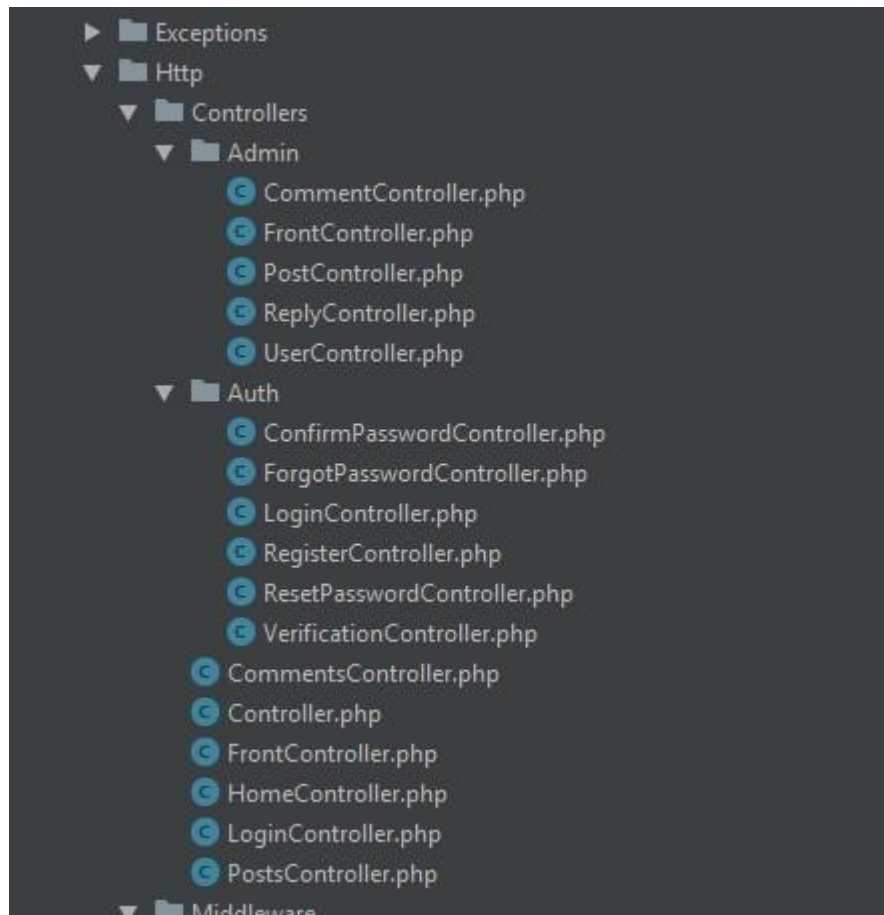
[1](#)
[2](#)
[3](#)

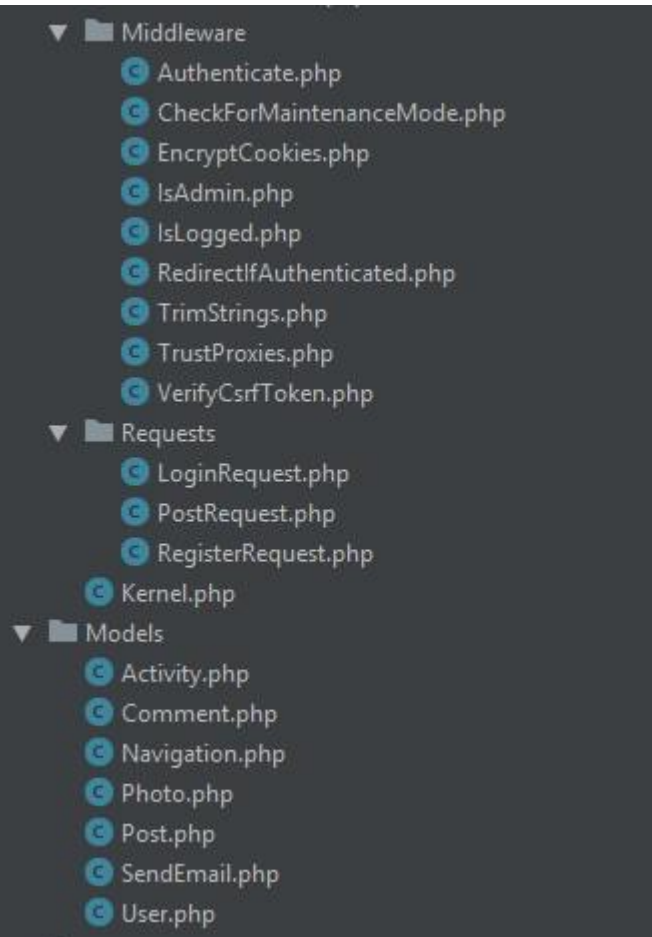
-Inicijalno prikazane sve ključne aktivnosti iz baze. Izborom datuma, uz upotrebu Ajax-a, biće prikazane samo aktivnosti za izabrani datum.

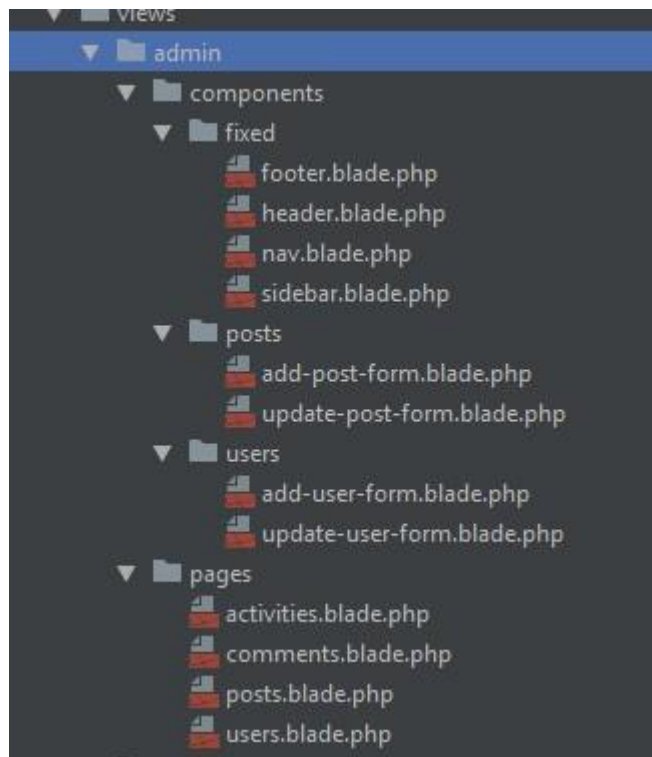
2. Struktura baze podataka

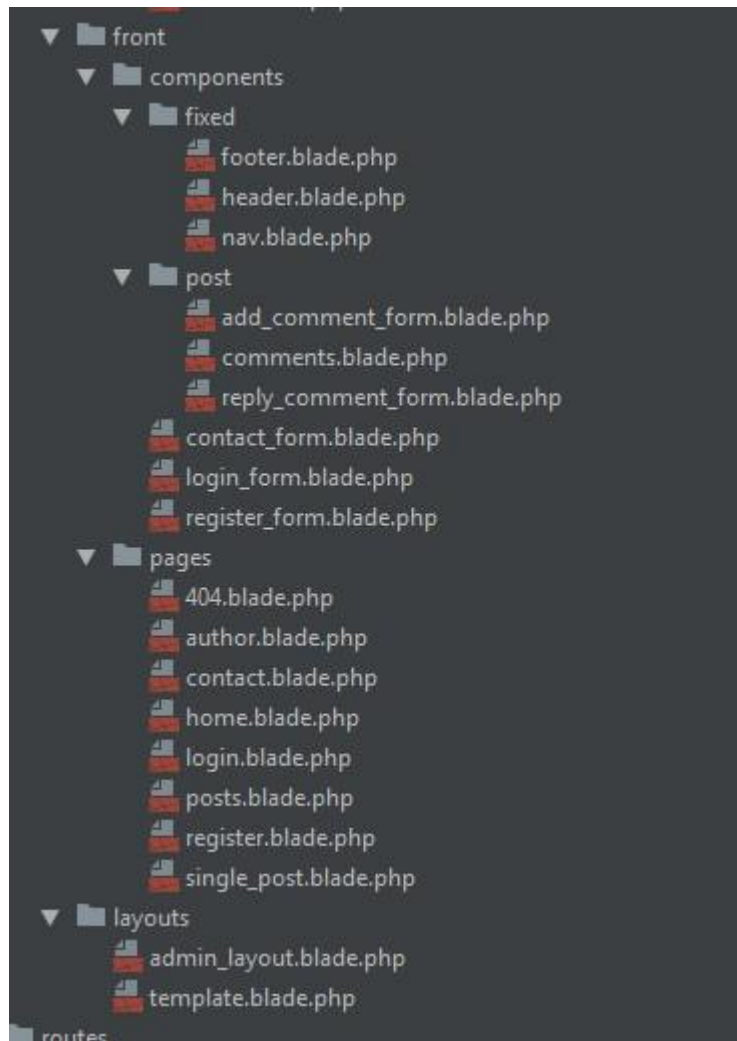


3. Slike organizacije foldera i fajlova









4. Kontroleri

Admin\CommentController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Comment;
use App\Models\Activity;
```

```

class CommentController extends Controller
{
    public function deleteComment($id)
    {
        $commModel = new Comment();
        try {
            $commModel->deleteAdminComment($id);
            $actModel=new Activity();
            $actModel->activity_content="Admin deleted comment";
            $actModel->usermail=session('user')->email;
            $actModel->addActivity();
            return redirect()->back()->with("success", "Comment successfully
deleted.");
        }
        catch (QueryException $ex) {
            \Log::error("Error while deleting " . $ex->getMessage());
            return redirect()->back()->with("warning", "An error has occurred, please
try again later.");
        }
    }
}

```

Admin\FrontController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Navigation;
use App\Models\Post;
use App\Models\User;
use App\Models\Comment;
use App\Models\Activity;
use Psy\Util\Json;

class FrontController extends Controller
{
    private $data=[];

    public function __construct(){
        $navModel= new Navigation();

        $navigation=$navModel->getMenuItems();
        $this->data= ["navigation"=>$navigation];
    }

    public function index(){

        $postsModel=new Post();
        $posts=$postsModel->getAdminPosts();
        $this->data+=["posts"=>$posts];

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

```

```

public function showUsers() {
    $usermodel=new User();
    $users=$usermodel->getUsers();
    $this->data+=["users"=>$users];

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

public function showComments() {
    $commmodel=new Comment();
    $comments=$commmodel->getAllComm();
    $this->data+=["comments"=>$comments];

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

public function addPostForm() {
    return view('admin.components.posts.add-post-form', $this->data);
}

public function addUserForm() {
    return view('admin.components.users.add-user-form', $this->data);
}

public function show($id)
{
    $model=new Post();
    $post=$model->find($id);
    $this->data+=["post"=>$post];

    return view('admin.components.posts.update-post-form', $this->data );
}

public function showUser($id)
{
    $model=new User();
    $user=$model->getOne($id);
    $this->data+=["user"=>$user];

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

public function activities() {
    $model=new Activity();
    $activities=$model->getAll();
    $this->data+=["activities"=>$activities];
    return view('admin.pages.activities', $this->data);
}

public function getFiltered($time) {
    $model=new Activity();
    $activities=$model->getForDate($time);
    // $this->data+=
    return Json :: encode ( $activities );
    // return view('admin.pages.activities', $this->data);
}

```

```
}
```

Admin\PostController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Database\QueryException;
use App\Http\Requests\PostRequest;
use App\Models\Post;
use App\Models\Photo;
use App\Models\Activity;

class PostController extends Controller
{

    public function index()
    {
        return view('layouts.admin_layout');
    }

    public function deletePost($id)
    {
        $postModel = new Post();
        try {
            $postModel->delete($id);
            $sactModel=new Activity();
            $sactModel->activity_content="Admin deleted post";
            $sactModel->usermail=session('user')->email;
            $sactModel->addActivity();
            return redirect()->back()->with("success", "Post successfully
deleted.");
        }
        catch (QueryException $ex) {
            \Log::error("Error while deleting " . $ex->getMessage());
            return redirect()->back()->with("warning", "An error has occurred,
please try again later.");
        }
    }

    public function store(PostRequest $request)
    {
        try {

            $postModel = new Post();
```



```

        $path_new = $request->file('photo');
        $directory = public_path("images/");
        $fileName = time() . "_" . $path_new->getClientOriginalName();
        $path_new->move($directory, $fileName);

        $photoModel = new Photo();
        $photoModel->path_new = $fileName;
        $photoModel->alt = "Blog photo";

        $postModel->post_pic_id = $photoModel->insert();
        $postModel->title = $request->get('title');
        $postModel->text = $request->get('text');
        $postModel->category_id = $request->get('category_id');
        $postModel->insert();
        $sactModel=new Activity();
        $sactModel->activity_content="Admin added post";
        $sactModel->usermail=session('user')->email;
        $sactModel->addActivity();
        return redirect()->back()->with("success", "Post successfully added!");
    } catch (QueryException $e) {
        \Log::error("Database error " . $e->getMessage());
    } catch (FileNotFoundException $e) {
        \Log::error("Picture upload error " . $e->getMessage());
    }
}

return redirect()->back()->with("error", "An error occured, please try again later");
}

```

```

public function update(PostRequest $request, $id)
{
    $model=new Post();

    $oldPhotoId = null;
    if ($request->hasFile('photo')) {
        $oldPictureId = $model->getPhotoId($id);
        try {
            $path_new = $request->file('photo');
            $directory = public_path("images/");
            $fileName = time() . "_" . $path_new->getClientOriginalName();
            $path_new->move($directory, $fileName);

            $photoModel = new Photo();
            $photoModel->path_new = $fileName;
            $photoModel->alt = "Blog post";
            $model->post_pic_id = $photoModel->insert();
            $sactModel=new Activity();
            $sactModel->activity_content="Admin updated post";
            $sactModel->usermail=session('user')->email;
            $sactModel->addActivity();

        } catch (QueryException $ex) {
            \Log::error("Error while updating " . $ex->getMessage());
        } catch (FileNotFoundException $ex) {
            \Log::error("Error in uploading photo " . $ex->getMessage());
        }
    }
}

```

```

$model->title = $request->get('title');
$model->text = $request->get('text');
$model->category_id = $request->get("category_id");
try {
    $model->update($id);

    try {
        if($oldPhotoId) {
            $photoModel = new Photo();
            $photo = $photoModel->find($oldPhotoId);
            unlink(public_path($photo->path_new));
            $photoModel->delete($oldPhotoId);
        }
    } catch(\Exception $ex) {
        \Log::error("Error while deleting old photo" . $ex->getMessage());
    }

    return redirect(route("admin-posts"))->with("success", "Post successfully
edited!");
} catch (QueryException $ex) {
    \Log::error("Error in update " . $ex->getMessage());
    return redirect()->back()->with("error", "An error occurred, please try
again later");
}

}

}

```

Admin\ReplyController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Comment;

class ReplyController extends Controller
{
    public function reply(Request $request, $commentId){

        $model=new Comment();

        try{
            $model->reply=$request->get('reply');
            $model->addReply($commentId);
            return response (null,200);
        } catch(\Exception $ex) {
            \Log::error("Error while adding reply".$ex->getMessage());
            return response(null,422);
        }
    }
}

```

```

    }
}

public function show($commentId)
{
    $model = new Comment();
    return response()->json($model->find($commentId));
}
}

```

Admin\UserController.php

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Http\Requests\RegisterRequest;
use App\Models\User;
use App\Models\Activity;

class UserController extends Controller
{
    public function deleteUser($id)
    {
        $userModel = new User();
        try {
            $userModel->delete($id);
            $actModel=new Activity();
            $actModel->activity_content="Admin deleted user";
            $actModel->usermail=session('user')->email;
            $actModel->addActivity();
            return redirect()->back()->with("success", "User successfully deleted.");
        }
        catch (QueryException $ex) {
            \Log::error("Error while deleting " . $ex->getMessage());
            return redirect()->back()->with("warning", "An error has occurred, please try again later.");
        }
    }

    public function store(RegisterRequest $request)
    {
        $userModel = new User();
        $userModel->name = $request->get('name');
        $userModel->surname = $request->get("surname");
        $userModel->email = $request->get('email');
        $userModel->password = $request->get("password");
        $userModel->role_id = $request->get("role_id");
    }
}

```

```

        try {
            $userModel->insertUser();
            $sactModel=new Activity();
            $sactModel->activity_content="Admin added user";
            $sactModel->usermail=session('user')->email;
            $sactModel->addActivity();
            return redirect(route("admin-users"))->with("success", "User successfully
added!");
        } catch(\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with("error", "An server error has occurred,
please try again later.");
        }
    }

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

        $userModel = new User();
        $userModel->name = $request->get('name');
        $userModel->surname = $request->get("surname");
        $userModel->email = $request->get('email');
        $userModel->password = $request->get("password");
        $userModel->role_id = $request->get("role_id");

        try {
            $userModel->update($id);
            $sactModel=new Activity();
            $sactModel->activity_content="Admin updated user";
            $sactModel->usermail=session('user')->email;
            $sactModel->addActivity();
            return redirect(route("admin-users"))->with("success", "User successfully
edited!");
        } catch(\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with("error", "An server error has occurred,
please try again later.");
        }
    }
}

```

CommentController.php

```

<?php

namespace App\Http\Controllers;
use App\Models\Comment;

use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Models\Activity;

```

```

class CommentsController extends Controller
{
    public function addComment(Request $request, $id){
        if($request->get('content')){
            $comModel=new Comment();
            $comModel->content=$request->get('content');
            $comModel->post_id=$id;
            try{
                $comModel->insertCom();
                $actModel=new Activity();
                $actModel->activity_content="User added comment";
                $actModel->usermail=session('user')->email;
                $actModel->addActivity();

                return redirect()->back()->with('success', 'Comment added
successfully');
            }

            catch(QueryException $ex){
                \Log::error("Error adding comment". $ex->getMessage());
                return redirect()->back()->with('error','Server error occured');
            }
        }

        return redirect()->back()->with('warning','Comment field can not be empty');
    }

    public function deleteComment($id)
    {
        $commentModel = new Comment();
        if ($commentModel->find($id)) {
            try {
                if ($commentModel->delete($id)) {
                    $actModel=new Activity();
                    $actModel->activity_content="User deleted comment";
                    $actModel->usermail=session('user')->email;
                    $actModel->addActivity();
                    return redirect()->back()->with("success", "Comment successfully
deleted.");
                } else {
                    \Log::critical("User without permission tried to delete comment.");
                    $actModel=new Activity();
                    $actModel->activity_content="User without permission tried to
delete comment";
                    $actModel->usermail=session('user')->email;
                    $actModel->addActivity();
                    return redirect()->back()->with("warning", "An error has occurred,
please try again later.");
                }
            } catch (QueryException $e) {
                \Log::error("Error while deleting " . $e->getMessage());
                return redirect()->back()->with("warning", "An error has occurred,
please try again later.");
            }
        }
        return redirect()->back();
    }
}

```

```

public function reply(Request $request, $commentId){

    $model=new Comment();

    try{
        $model->reply=$request->get('reply');
        $model->addReply($commentId);
        return response (null,200);
    } catch (\Exception $ex){
        \Log::error("Error while adding reply".$ex->getMessage());
        return response (null,422);
    }

}

public function showForm($commentId)
{
    $model = new Comment();
    return response()->json($model->findCom($commentId));
}

public function editCom(Request $request, $commentId)
{
    $commentModel = new Comment();
    try {
        $commentModel->content = $request->get('content');
        $commentModel->update($commentId);
        $actModel=new Activity();
        $actModel->activity_content="User edited comment";
        $actModel->usermail=session('user')->email;
        $actModel->addActivity();
        return redirect()->back()->with("success", "Comment successfully edited.");
    } catch (\Exception $e) {
        \Log::error("Error while editing " . $e->getMessage());
        return redirect()->back()->with("warning", "An error has occurred, please
try again later.");
    }

}

}

```

FrontController.php

```

<?php

namespace App\Http\Controllers;

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

use Illuminate\Http\Request;
use App\Models\Navigation;
use App\Models\Post;
use App\Models\Comment;
use App\Models\SendEmail;

```

```

use App\Models\Activity;

class FrontController extends Controller
{
    private $data=[];

    public function __construct(){
        $navModel= new Navigation();

        $navigation=$navModel->getMenuItems();
        $this->data= ["navigation"=>$navigation];
    }

    public function index(){
        $model=new Post();
        $posts=$model->getHomePosts();
        $this->data+=["posts"=>$posts];
        return view("front.pages.home",$this->data);
    }

    public function getAll(){
        $postsModel=new Post();
        $posts=$postsModel->getAllPosts();
        $this->data+=["posts"=>$posts];
        return view('front.pages.posts',$this->data);
    }

    public function getSinglePost(Request $request, $id)
    {
        $postModel = new Post();
        $post = $postModel->getSingle($id);
        if (!$post) {
            return view('front.pages.404');
        }

        $commentsModel = new Comment();
        $post->comments = $commentsModel->getPostComments($id);
        // $this->data['post'] = $post;
        $postModel->addVisit($id, $request->ip());
        $this->data+=["post"=>$post];
        return view("front.pages.single_post", $this->data);
    }

    public function showLoginForm(){
        return view('front.pages.login', $this->data);
    }

    public function showRegisterForm(){
        return view ('front.pages.register',$this->data);
    }

    public function contactForm(){
        return view ('front.pages.contact', $this->data);
    }

    public function getAuthor(){
        return view ('front.pages.author', $this->data);
    }
}

```

```

public function contactUs (Request $request)
{
    $name=$request->name;
    $email=$request->email;
    $message=$request->message;
    $ip=$request->ip();
    $emailSender = new SendEmail();
    try {
        $emailSender -> ContactAdmin($email,$name,$message);
        $actModel=new Activity();
        $actModel->activity_content="You have new message";
        $actModel->usermail=session('user')->email;
        $actModel->addActivity();

        return redirect()->back()->with('success', 'Message sent successfully');
    }
    catch (QueryException $ex){
        \Log::error("Error sending message". $ex->getMessage());
        return redirect()->back()->with('error','Server error occurred');
    }
}
}

```

LoginController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Requests\LoginRequest;
use App\Models\User;
use Dotenv\Exception\ValidationException;
use Illuminate\Database\QueryException;
use App\Http\Requests\RegisterRequest;
use App\Models\Activity;

class LoginController extends Controller
{

    public function register(RegisterRequest $request){
        $model=new User();

        $model->name=$request->get('name');
        $model->surname=$request->get('surname');
        $model->email=$request->get('email');
        $model->password=$request->get('password');
        $model->role_id=2;

        try {
            $model->insertUser();
            return redirect('/login')->with("success", "Successfully created account,
now log in");
            $actModel=new Activity();
            $actModel->activity_content="User with this ip registred";

```



```

        $sactModel->usermail=$request->ip();
        $sactModel->addActivity();
    } catch(QueryException $e) {
        \Log::error($e->getMessage());
        return redirect('/register')->with("error", "Something went wrong! Try
again");
    }
}

public function doLogin(LoginRequest $request){

    $email = $request->input("email");
    $password = $request->input("password");

    $model = new User();
    $user= $model->getUserAuth($email, $password);

    if($user){

        $request->session()->put("user", $user);
        $sactModel=new Activity();
        $sactModel->activity_content="User logged in";
        $sactModel->usermail=session('user')->email;
        $sactModel->addActivity();

        return \redirect("/")->with("message", "Successfully login!");
    } else {
        return redirect("/login")->with("error", "You need to register first!");
    }

}

public function logout(Request $request){
    $request->session()->forget("user");
    $request->session()->flush();
    $sactModel=new Activity();
    $sactModel->activity_content="User logged out";
    $sactModel->usermail=$request->ip();
    $sactModel->addActivity();
    return redirect("/login")->with("message", "You are logged out now");
}
}

```

5.Models

Models\Activity.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 26.3.2020.
 * Time: 23.59
 */

```

```

namespace App\Models;

class Activity
{
    public $activity_content;
    public $usermail;

    public function getAll(){
        return \DB::table('activity')
            ->orderBy('time','desc')
            ->paginate(10);
    }

    public function addActivity(){
        return \DB::table('activity')
            ->insertGetId([
                "activity_content"=>$this->activity_content,
                "usermail"=>$this->usermail,
                "time"=>date("Y:m:d H:i:s")
            ]);
    }

    public function getForDate($time){
        return \DB::table('activity')
            ->where("time","like","%".$time."%")
            ->get();
    }
}

```

Models\Comment.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 18.3.2020.
 * Time: 16.58
 */

namespace App\Models;

class Comment
{
    public $content;
    public $post_id;
    public $reply;

    public function getPostComments($id)
    {
        return \DB::table('comment AS c')
            ->join("user AS u", "c.user id", "=", "u.id")
    }
}

```

```

        ->where('post_id', $id)
        ->select('c.*', 'u.name', 'u.surname')
        ->orderBy('c.created_at', 'desc')
        ->get();
    }

    public function insertCom() {
        $this->reply=null;
        return \DB::table('comment')
            ->insert([
                'content'=>$this->content,
                'created_at'=>date("Y-m-d H:i:s"),
                'updated_at'=>date("Y-m-d H:i:s"),
                'user_id'=>session()->get('user')->id,
                'post_id'=>$this->post_id,
                'reply'=>$this->reply
            ]);
    }

    public function addReply($id)
    {
        return \DB::table('comment')
            ->where('id', $id)
            ->update([
                'reply' => $this->reply
            ]);
    }

    public function find($id)
    {
        return \DB::table('comment')
            ->where('id', $id)->get()->first();
    }

    public function delete($id)
    {
        if($this->canDelete($id)) {
            return \DB::table('comment')->delete($id);
        }
        return 0;
    }

    private function canDelete($id)
    {
        $comment = $this->find($id);

        return $comment ? (session()->get('user')->role_name == 'admin') || ($comment->user_id == session()->get('user')->id) : false;
    }

    public function getAllComm() {
        return \DB::table('comment AS c')
            ->join('user AS u', 'c.user_id', '=', 'u.id')
            ->join('post AS p', 'c.post_id', '=', 'p.id')
            ->select('c.*', 'u.email', 'p.title')
    }

```

```

        ->orderBy('c.updated_at', 'desc')
        ->paginate(3);
    }

    public function deleteAdminComment($id)
    {
        return \DB::table('comment')->delete($id);
    }

    public function findCom($id)
    {
        return \DB::table('comment')
            ->where('id', $id)->get()->first();
    }

    public function update($id)
    {
        return \DB::table('comment')
            ->where('id', $id)
            ->update([
                'content' => $this->content,
                'updated_at' => date("Y-m-d H:i:s")
            ]);
    }
}

```

Models\Navigation.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 16.3.2020.
 * Time: 19.45
 */

namespace App\Models;

class Navigation
{
    public function getMenuItems(){
        return \DB::table("menu")
            ->orderBy('position', 'asc')
            ->get();
    }
}

```

Models\Photo.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 26.3.2020.
 * Time: 16.28
 */

namespace App\Models;

class Photo
{
    public $path_new;
    public $alt;

    public function insert()
    {
        return \DB::table('post_photo')
            ->insertGetId([
                'path_new' => $this->path_new,
                'alt' => $this->alt
            ]);
    }
}

```

Models\Post.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 5.3.2020.
 * Time: 18.49
 */

namespace App\Models;

class Post
{
    public $title;
    public $text;
    public $category_id;
    public $post_pic_id;

    public function getHomePosts(){
        return \DB::table("post AS p")
            ->join("post_photo AS ph", "p.post_pic_id", "=", "ph.id_postphoto" )
            ->orderBy('updated_at','desc')
            ->limit(5)
            ->get();
    }
}

```

```

    }

    public function getAllPosts(){
        return \DB::table('post AS p')
            ->join("post_photo AS ph", "p.post_pic_id", "=", "ph.id_postphoto" )
            ->orderBy('updated_at','desc')
            ->paginate(3);
    }

    public function getSingle($id){
        return \DB::table('post as p')
            ->join("post_photo AS ph", "p.post_pic_id", "=", "ph.id_postphoto" )

            ->where('p.id', $id)
            ->select('p.*', 'ph.path_new', \DB::raw("(SELECT count(id) FROM log WHERE
post_id = $id) as visited"), \DB::raw("(SELECT count(id) FROM comment WHERE
post_id=$id) as num_of_com"))
            ->get()->first();
    }

    public function addVisit($id, $ip)
    {
        return \DB::table("log")
            ->insert([
                'ip_address' => $ip,
                'post_id' => $id,
                'time'=>date("Y-m-d H:i:s")
            ]);
    }

    public function getAdminPosts(){
        return \DB::table('post AS p')
            ->join("post_photo AS ph", "p.post_pic_id", "=", "ph.id_postphoto" )
            ->orderBy('updated_at','desc')
            ->paginate(5);
    }

    public function insert()
    {
        return \DB::table('post')
            ->insertGetId([
                'title' => $this->title,
                'text' => $this->text,
                'post_pic_id' => $this->post_pic_id,
                'category_id' => $this->category_id,
                'created_at'=>date("Y:m:d H:i:s"),
                'updated_at'=>date("Y:m:d H:i:s")
            ]);
    }

    public function find($id)
    {
        return \DB::table('post AS p')
            ->join("post_photo AS ph", "p.post_pic_id", "=", "ph.id_postphoto" )
            ->where('p.id', $id)
            ->select('p.*', 'ph.path_new', 'ph.alt')
            ->get()->first();
    }
}

```

```

    public function delete($id)
    {
        return \DB::table('post')->delete($id);
    }

    public function update($id)
    {
        $updateData = [
            'title' => $this->title,
            'text' => $this->text,
            'category_id' => $this->category_id,
            'updated_at' => date("Y-m-d H:i:s")
        ];
        if ($this->post_pic_id != null) {
            $updateData['post_pic_id'] = $this->post_pic_id;
        }
        return \DB::table('post')
            ->where('id', $id)
            ->update($updateData);
    }

    public function getPhotoId($postId)
    {
        return \DB::table('post')
            ->where('id', $postId)
            ->select("post.post_pic_id as id")
            ->get()->first()->id;
    }
}

```

Models\SendEmail.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 27.3.2020.
 * Time: 16.26
 */

namespace App\Models;
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

//require 'phpmailer\phpmailer\src\Exception';
//require 'phpmailer\phpmailer\src\PHPMailer';
//require 'phpmailer\phpmailer\src\SMTP';

class SendEmail
{

```

```

public function ContactAdmin ($email,$name,$message){
    $mail = new PHPMailer ( true );
    try {

$mail -> isSMTP ();
$mail -> Host = 'smtp.gmail.com' ;

$mail -> SMTPAuth = true ;
$mail -> Username = 'UKLONJENO'; // SMTP username
$mail -> Password = 'UKLONJENO' ; // SMTP password
$mail -> SMTPSecure = 'tls' ;
$mail -> Port = 587 ;
$mail -> setFrom ( 'UKLONJENO' , 'ReadIt blog' );
$mail -> addAddress ( ' ' , "Contact page" ); // Add a recipient
//Content
$mail -> isHTML ( true );
$mail -> Subject = 'New message - from ReadIt blog';
$mail -> Body = $name . " (e-mail: " . $email . ") sent you a
message:<br/>" . $message ;
$mail -> send ();
} catch ( Exception $e ) {
    echo 'Message could not be sent. Mailer Error: ' , $mail -> ErrorInfo ;
}
}
}
}

```

Models\User.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Marijana
 * Date: 17.3.2020.
 * Time: 21.34
 */

namespace App\Models;

class User
{
    public $name;
    public $surname;
    public $email;
    public $password;
    public $role_id;

    public function insertUser(){
        return \DB::table('user')
            ->insertGetId([
                'name'=>$this->name,
                'surname'=>$this->surname,
                'email'=>$this->email,
                'password'=>md5($this->password),
                'role_id'=>$this->role_id,
            ])
    }
}

```



```

        'created_at'=>date("Y:m:d H:i:s"),
        'updated_at'=>date("Y:m:d H:i:s")
    ];
}

public function getUserAuth($email, $password){
    return \DB::table('user AS u')
        ->join('role AS r', 'u.role_id','=', 'r.id_role')
        ->select('u.*', 'r.role_name')
        ->where([
            ["email", "=", $email],
            ["password", "=", md5($password)]
        ])
        ->first();
}

public function getUsers(){
    return \DB::table('user AS u')
        ->join('role AS r', 'u.role_id','=', 'r.id_role')
        ->select('u.*', 'r.role_name')
        ->paginate(5);
}

public function delete($id)
{
    return \DB::table('user')->delete($id);
}

public function getOne($id)
{
    return \DB::table('user')
        ->where("id", $id)->first();
}

public function update($id)
{
    return \DB::table('user')
        ->where("id", $id)
        ->update([
            'name' => $this->name,
            'surname' => $this->surname,
            'email' => $this->email,
            'password' => md5($this->password),
            'role_id' => $this->role_id,
            'updated_at'=>date("Y:m:d H:i:s")
        ]);
}
}

```

6. Request-ovi

LoginRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "email" => "required|email",
            "password" => ["required", "regex:/^[A-z\d]+$/" ]

        ];
    }

    public function message() {
        return ["required" => "This field is required"];
    }
}

```

PostRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class PostRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }
}

```

```

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'title' => 'required|min:3|max:200',
            'text' => 'required',
            'category_id' => 'required',
            'photo' => 'required|image|mimes:jpeg,png,jpg|max:3000'

        ];
    }
}

```

RegisterRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'name' => 'required|alpha|min:3|max:20',
            'surname' => 'required|alpha|min:3|max:30',
            'email' => 'required|email|unique:user',
            'password' => [
                'required',
                'min:6',
                'regex:/^[A-z\d]+$/'
            ]
        ];
    }
}

```

7. Middleware-i

IsAdmin.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (session()->has('user')) {
            if (session()->get('user')->role_name == 'admin') {
                return $next($request);
            }
        }
        return redirect(route("/"));
    }
}
```

IsLogged.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsLogged
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has("user"))
            return $next($request);
        else
            return redirect("/login");
    }
}
```

```
}  
}
```

8. Rute

routes/web.php

```
<?php  
  
/*  
|-----  
| Web Routes  
|-----  
|  
| Here is where you can register web routes for your application. These  
| routes are loaded by the RouteServiceProvider within a group which  
| contains the "web" middleware group. Now create something great!  
|  
*/  
  
Route::get("/", "FrontController@index");  
Route::get("/posts", "FrontController@getAll");  
  
Route::get("/author", "FrontController@author");  
Route::get("/posts/{id}", "FrontController@getSinglePost")->name("singlePost");  
Route::get("/contact", "FrontController@contactForm")->name("contact")->  
>middleware(["isLoggedIn"]);  
Route::post('/contact', 'FrontController@contactUs')-> name('contact-us')->  
>middleware(["isLoggedIn"]);  
  
Route::get("/login", "FrontController@showLoginForm")->name("loginForm");  
Route::get("/register", "FrontController@showRegisterForm")->name('registerForm');  
  
Route::group(['middleware'=>'isLoggedIn'], function() {  
    Route::post("/comments/{id}", "CommentsController@addComment")->name("addComment");  
    Route::get("/comments/{id}/delete", "CommentsController@deleteComment")->  
>name("deleteCom");  
    Route::get("/comments/{id}/form", "CommentsController@showForm")->name("show-  
edit");  
    Route::post("/comments/{id}/update-com", "CommentsController@editCom")->name("edit-  
com");  
});  
  
Route::post("/do-register", "LoginController@register")->name("register");  
Route::post("/do-login", "LoginController@doLogin")->name("login");  
Route::get("/logout", "LoginController@logout")->name("logout")->  
>middleware(["isLoggedIn"]);  
  
Route::group(['middleware' => 'isAdmin'], function(){  
    Route::get("/admin-panel/posts", "Admin\FrontController@index" )->name("admin-  
posts");
```

```

Route::get("/admin-panel/posts/{id}/delete", "Admin\PostController@deletePost" )-
>name("delete-post");
Route::get("/admin-panel/posts/add", "Admin\FrontController@addPostForm" )-
>name("add-post");
Route::post("/admin-panel/posts", "Admin\PostController@store" )->name("post-
store");
Route::get("/admin-panel/posts/{id}", "Admin\FrontController@show")->name("post-
show");
Route::post("/admin-panel/posts/{id}/update", "Admin\PostController@update")-
>name("post-update");

Route::get("admin-panel/users", "Admin\FrontController@showUsers")->name("admin-
users");
Route::get("/admin-panel/users/{id}/delete", "Admin\UserController@deleteUser" )-
>name("delete-user");
Route::get("/admin-panel/users/add", "Admin\FrontController@addUserForm" )-
>name("add-user");
Route::post("/admin-panel/users", "Admin\UserController@store" )->name("user-
store");
Route::get("/admin-panel/users/{id}", "Admin\FrontController@showUser")-
>name("user-show");
Route::post("/admin-panel/users/{id}/update", "Admin\UserController@update")-
>name("user-update");

Route::get("admin-panel/comments", "Admin\FrontController@showComments")-
>name("admin-comments");
Route::get("/admin-
panel/comments/{id}/delete", "Admin\CommentController@deleteComment" )->name("delete-
comment");

Route::get("/comments/{id}/reply", "Admin\ReplyController@show")->name("show");
Route::post("/comments/{id}/add-reply", "Admin\ReplyController@reply")-
>name("replyForm");

Route::get("/admin-panel/activities", "Admin\FrontController@activities" )-
>name("admin-activities");
Route::get("/admin-panel/activities/{time}", "Admin\FrontController@getFiltered");

});

```

9.JS kod / Ajax

adminpanel-ajax.js

```

document.querySelector("#datepicker").addEventListener("change", function() {

```

```

if(document.querySelector("#datepicker").value !== "" )
{
    $.ajax({
        url : "/admin-panel/activities/" +
        document.querySelector("#datepicker").value,
        dataType: "json",
        success:function(data){
            if (data.length === 0 )
            {
                $ ("#table").html(` <div class="alert alert-warning">
                There is no activity for choosen date.
                </div>`);
            }
            else
            {
                let print =` <table class="table table-bordered">
                <thead>
                <tr>

                <th>Activity</th>
                <th>User</th>
                <th>Time</th>

                </tr>
                </thead>
                <tbody>

                `;
                for(let d in data){
                    console.log(JSON.stringify(data));
                    print+="<tr>";

                    print+=`<td>${data[d].activity_content}</td>
                    <td>${data[d].usermail}</td>
                    <td>${data[d].time}</td>
                    </tr>`;
                }
                print+=`</tbody>
                </table>`;
                $("#tablecontent").html(print);
            }
        },
        error: function(error){
            console.log("Error filtering by date"+error);
        }
    });
}
});

```

comments-ajax.php

```

function show(id){
    $.ajax({

```

```

        "url" : BASE_URL + "/comments/"+id+"/reply",
        method: "get",
        type: 'json',
        success: function(data) {
            showForm(data);
        },
        error: function(error, xhr) {
            console.log("Error showing form");
            console.log(xhr.status);
        }
    });
}

function showForm(data) {
    var form="";
    form+=`
        <h3 class="mb-5">Reply to user</h3>

        <div class="form-group">
            <label for="message">Type your reply here</label>
            <textarea id="reply" name="reply" cols="30" rows="10" class="form-
control"></textarea>
        </div>
        <div class="form-group">
            <input type="submit" value="Post Reply" onclick="addReply(${data.id})"
class="btn py-3 px-4 btn-primary">
        </div>
    </div>
`;
    $("#komentar").html(form);
}

function addReply(id) {
    var replyData=$("#reply").val();
    $.ajax({
        "url": BASE_URL+"/comments/" + id + "/add-reply",
        method: "post",
        data: {
            'reply' : replyData,
            '_method' : "post",
            '_token' : TOKEN
        },
        type: 'json',
        success: function(data) {
            console.log("Reply added");
            hideForm(id, replyData);
        },
        error: function(error) {
            console.log("Error while replying");
        }
    });
}

function hideForm(id, replyData) {

```



```

        $("#komentar"+id).html(replyData);
        $("#komentar").html("<div class='alert alert-info'>Reply added
successfully!</div>");
        $("#rep").hide();
    }

<!-- IZMENA KOMENTARA -->

function editForm(id) {
    $.ajax({
        "url" : BASE_URL + "/comments/"+id+"/form",
        method: "get",
        type: 'json',
        success: function(data) {
            form(data);
        },
        error: function(error,xhr) {
            console.log("Error showing form");
            console.log(xhr.status);
        }
    });
}

function form(data) {
    var html = "";
    html += `
        <h3 class="mb-5">Edit your comment</h3>

        <div class="form-group">
            <label for="message">Type your reply here</label>
            <textarea id="content" name="content" cols="30" rows="10" class="form-
control">${data.content}</textarea>
        </div>
        <div class="form-group">
            <input type="submit" value="Update comment"
onclick="updateComment(${data.id})" class="btn py-3 px-4 btn-primary">
        </div>
    </div>
`;
    $("#komentar").html(html);
}

function updateComment(id) {
    var editedData = $("#content").val();
    $.ajax({
        "url" : BASE_URL + "/comments/" + id + "/update-com",
        method : "post",
        data : {
            'content' : editedData,
            '_method' : "post",
            '_token' : TOKEN
        },
        type : 'json',

```

```
        success : function(data) {
            hide(id, editedData);
        },
        error : function(error)
        {
            console.log("Something went wrong"+error);
        }
    });
}

function hide(id, editedData)
{
    $("#comment"+id).html(editedData);
    $("#komentar").html("<div class='alert alert-info'>Comment
editedsuccessfully!</div>");
}
```