VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

# WEB PROGRAMIRANJE PHP2

# DOKUMENTACIJA PROJEKTA

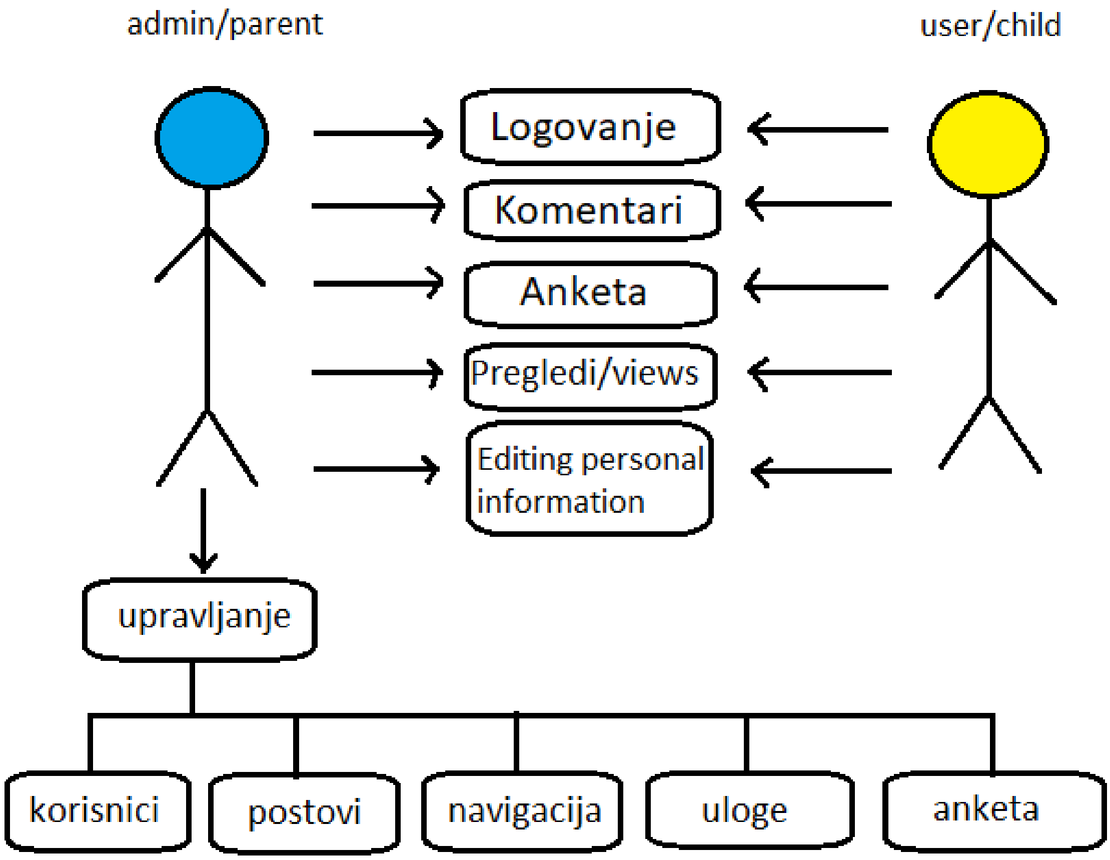
Student:

Milos Simic 113/13

Beograd 2018.

SADRŽAJ

## 1. Opis funkcionalnosti



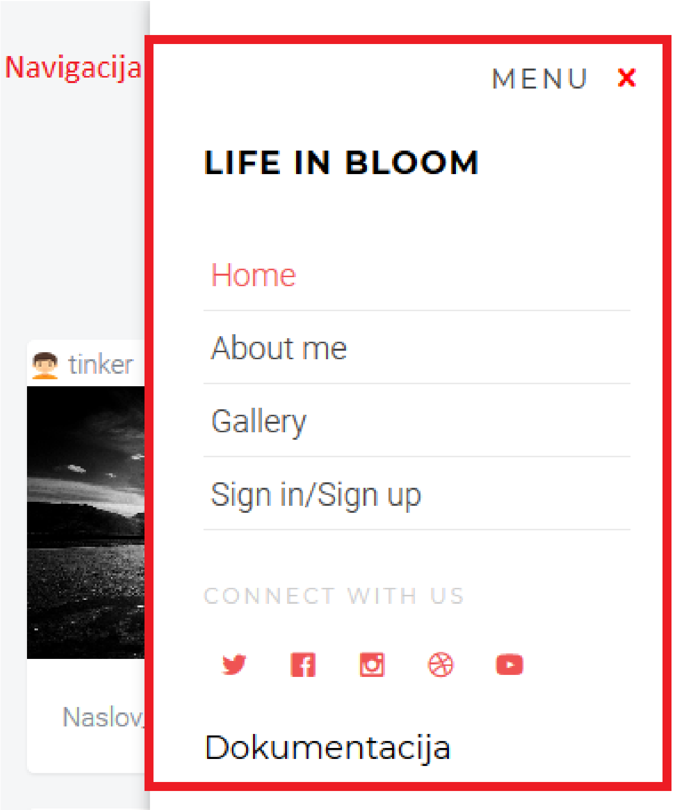
## Kratan opis funkcionalnosti:

Komentare mogu da unesu samo autorizovani korisnici, pored toga mogu da obrisu i promene sopstvene komentare, dok administrator moze svaki komentar da obrise i menja kao i da unese sopstveni komentar. Anketu mogu da vide samo autorizovani korisnici na koju mogu da glasaju i vide rezultate ankete iste, dok administrator moze da unese novu anketu, izbrise postojece ankete i da menja ankete. Svi korisnici koji posete bilo koji od postova belezi broj poseta tog posta. Autorizovani korisnici mogu da izmene licne informacije kao i upravljaju sopstvenim postovima. Administrator upravlja celim sadrzajem na sajtu i to: korisnicima, postovima, navigacionim linkovima, ulogama i anketama.

| | |
|---|---|
| **Koriščeni jezici** | • HTML 4 i HTML 5<br>• CSS<br>• javaScript<br>• PHP<br>• MySQL |
| **Korišćeni HTML šabloni (framework-ci)** | https://freehtml5.co/hydrogen-free-html5-bootstrap-template/ |
| **Koriščene biblioteke i njihovi url-ovi odakle ste ih preuzeli** | • jQuery<br>• Bootstrap<br>• sve biblioteke koje su koriscene su vec bile u okviru template-a |
| **Korišćene biblioteke iz framework-a** | • Laravel |
| **Korišćene php biblioteke van fremework-a** | |
| **Koje funkcionalnosti su realizove putem AJAX-a** | • Stranicenje<br>• Anketa |
| **Navesti lokacije gde je napisan AJAX kod (fajlovi ili delovi html stranice)** | • public/js/ajax.js<br>• public/js/mine.js |
| **Navesti lokacije gde je napisan sopstveni javaScript kod (fajlovi ili delovi html stranice)** | • public/js/provera.js |
| **URL sajta/aplikacije** | • lifeinabloom.000webhostapp.com |
| **Pristupni parametri** | • poslati su na mail |

## 2. Skica struktura stranica

### Pocetna strana:



Navigacija

MENU ×

LIFE IN BLOOM

Home

About me

Gallery

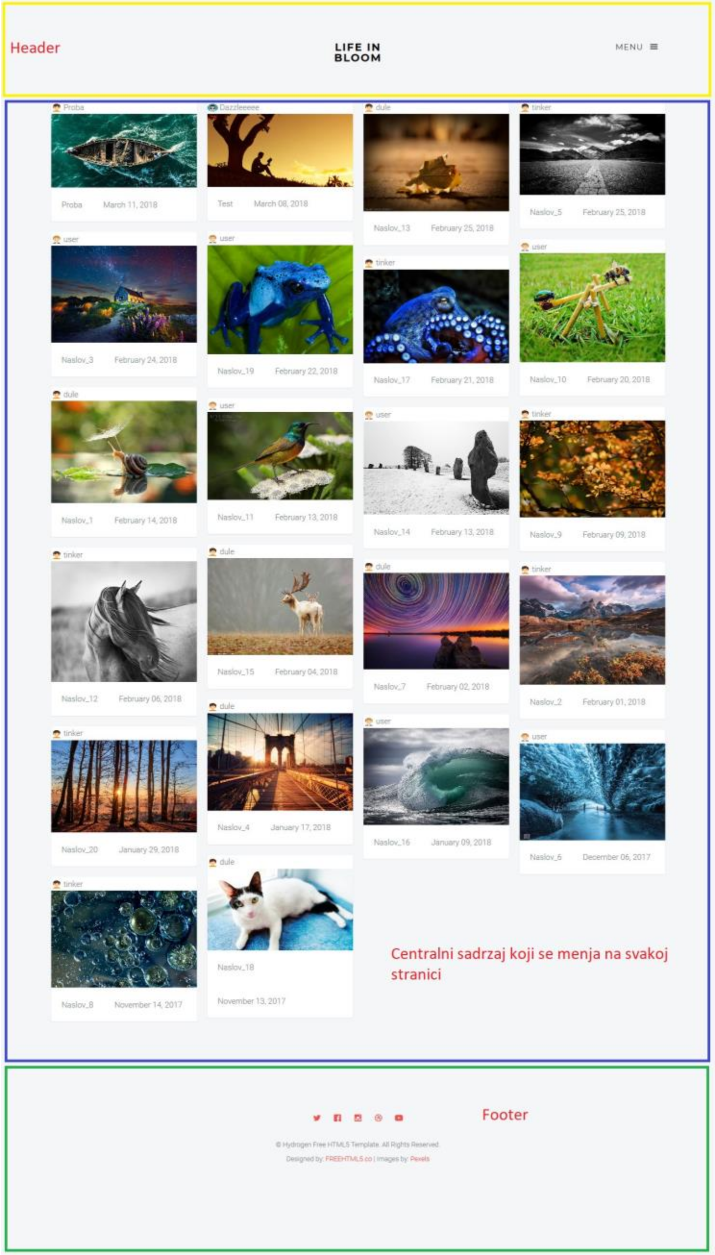
Sign in/Sign up

CONNECT WITH US

Dokumentacija

tinker

Naslov



Header

LIFE IN
BLOOM
MENU ≡
Proba
Proba March 11, 2018
Dazzleeeee
Test March 08, 2018
dule
Naslov_13 February 25, 2018
tinker
Naslov_5 February 25, 2018
user
Naslov_3 February 24, 2018
user
Naslov_19 February 22, 2018
tinker
Naslov_17 February 21, 2018
user
Naslov_10 February 20, 2018
dule
Naslov_1 February 14, 2018
user
Naslov_11 February 13, 2018
user
Naslov_14 February 13, 2018
tinker
Naslov_9 February 09, 2018
tinker
Naslov_12 February 06, 2018
dule
Naslov_15 February 04, 2018
dule
Naslov_7 February 02, 2018
tinker
Naslov_2 February 01, 2018
tinker
Naslov_20 January 29, 2018
dule
Naslov_4 January 17, 2018
user
Naslov_16 January 09, 2018
user
Naslov_6 December 06, 2017
tinker
Naslov_8 November 14, 2017
dule
Naslov_18
November 13, 2017

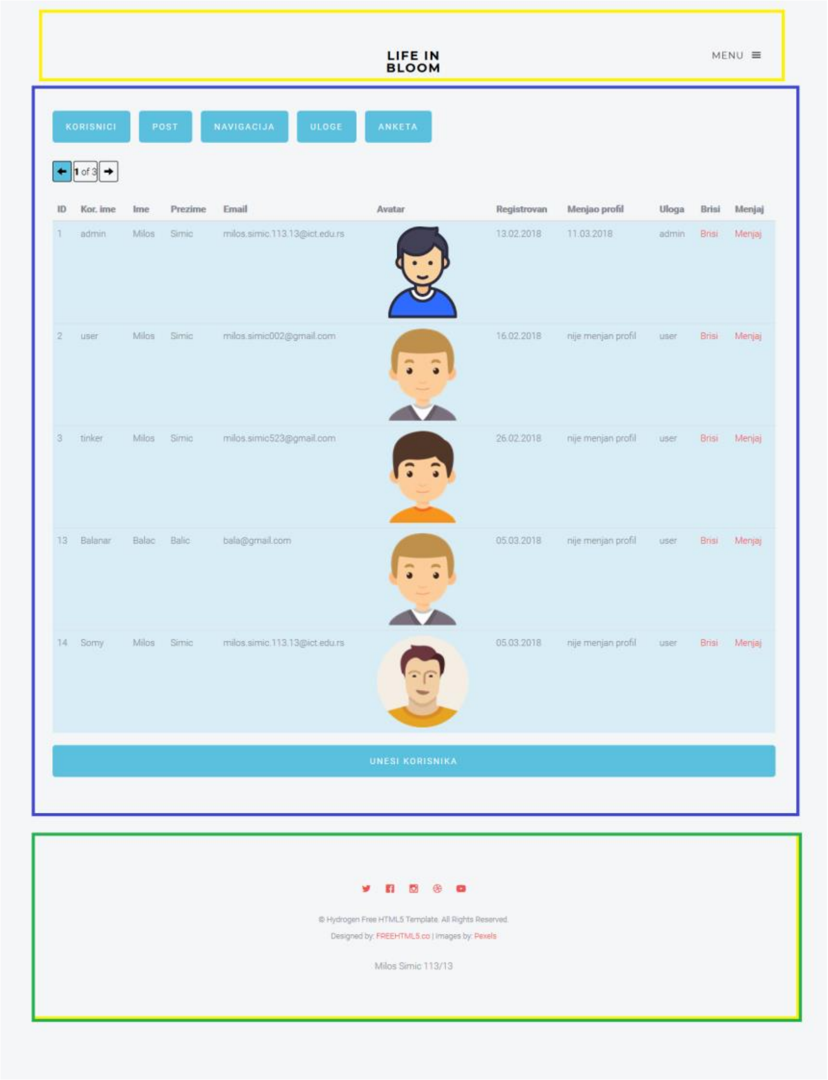
Centralni sadrzaj koji se menja na svakoj stranici

Footer

© Hydrogen Free HTML5 Template. All Rights Reserved.

Designed by: FREEHTML5.co | Images by: Pexels

# Admin strana:



LIFE IN BLOOM

MENU ≡

KORISNICI | POST | NAVIGACIJA | ULOGE | ANKETA

← 1 of 3 →

| ID | Kor. ime | Ime | Prezime | Email | Avatar | Registrovan | Menjao profil | Uloga | Brisi | Menjaj |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | admin | Milos | Simic | milos.simic.113.13@ict.edu.rs | | 13.02.2018 | 11.03.2018 | admin | Brisi | Menjaj |
| 2 | user | Milos | Simic | milos.simic002@gmail.com | | 16.02.2018 | nije menjan profil | user | Brisi | Menjaj |
| 3 | tinker | Milos | Simic | milos.simic523@gmail.com | | 26.02.2018 | nije menjan profil | user | Brisi | Menjaj |
| 13 | Balanar | Balac | Balic | bala@gmail.com | | 05.03.2018 | nije menjan profil | user | Brisi | Menjaj |
| 14 | Somy | Milos | Simic | milos.simic.113.13@ict.edu.rs | | 05.03.2018 | nije menjan profil | user | Brisi | Menjaj |

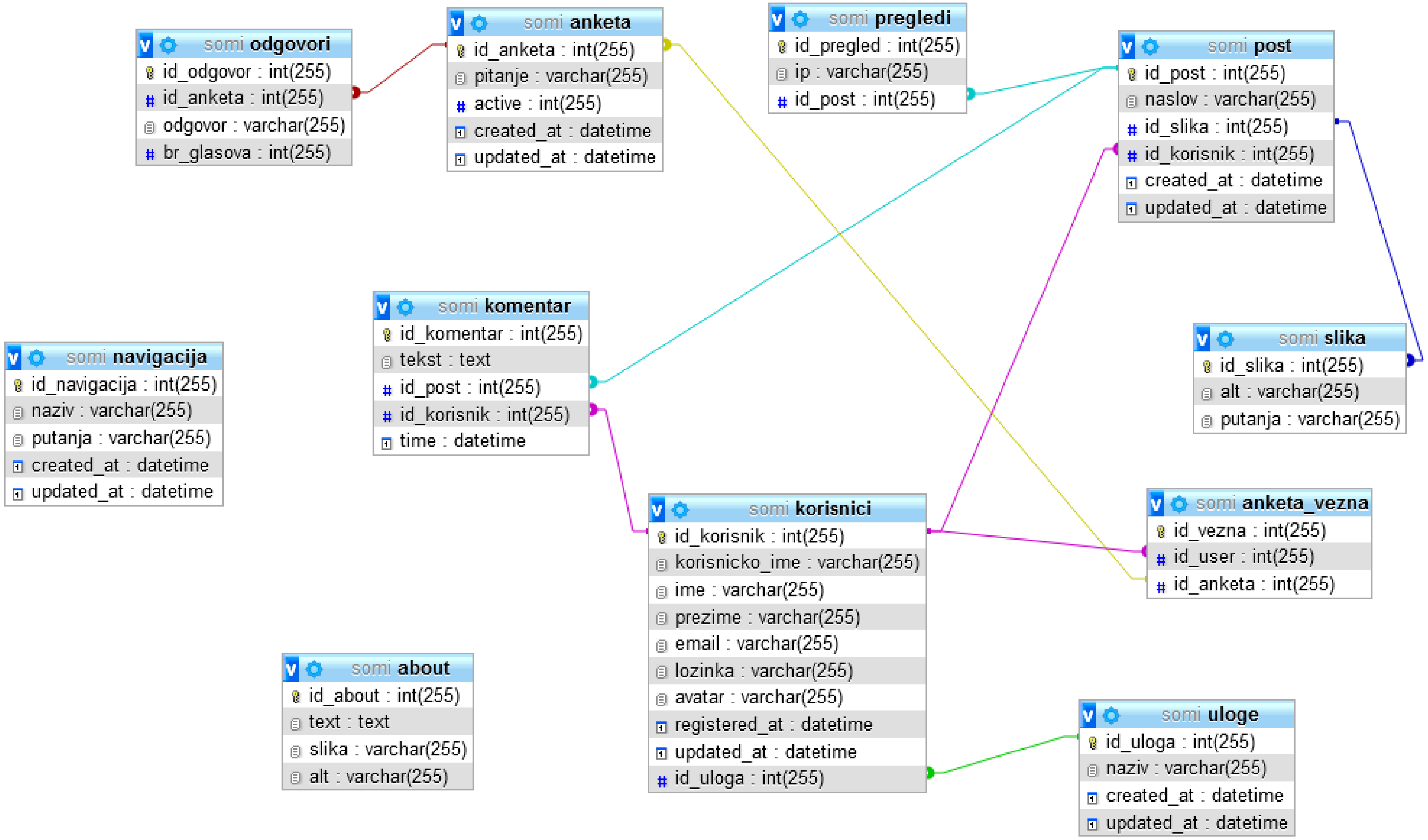
UNESI KORISNIKA

© Hydrogen Free HTML5 Template. All Rights Reserved.

Designed by: FREEHTML5.co | Images by: Pexels

Milos Simic 113/13

## 3. Dijagram baze podataka



## 4. MVC Organizacija

### Kontroleri

| AdminController |
|---|
| __construct() |
| getUsers() |
| getPosts() |
| getNavigations() |
| getRoles() |
| getPoll() |
| showFormUser() |
| showFormPost() |
| showFormNav() |
| showFormRole() |
| showFormPoll() |
| showUserForm($id) |
| showPostForm($id) |
| showNavForm($id) |
| showRoleForm($id) |
| showPollForm($id) |
| getGallery() |

| AnketaController |
|---|
| storeVote(Request $request) |
| insertPoll(Request $request) |
| update(Request $request, $id) |
| delete($id) |
| active($id) |

| CommentController |
|---|
| insertComment($id, Request $request) |
| deleteComment($id, $cId) |
| updateComment(Request $request, $id, $cId) |

| FrontEndController |
|---|
| __construct() |
| index() |
| getPost(Request $request, $id, $cId = null) |
| getRegister() |
| about(Request $request) |
| user() |
| showUser($id) |
| showAnketa() |
| download() |

| NavigationController |
|---|
| delete($id) |
| updateNav($id, Request $request) |
| insertNav(Request $request) |

| PostController |
|---|
| insertPost(Request $request) |
| insertAdminPost(Request $request) |
| updatePost($id, Request $request) |
| deletePost($id) |
| delete($id) |

| RoleController |
|---|
| delete($id) |
| updateRole($id, Request $request) |
| insertRole(Request $request) |

| UserController |
|---|
| register(Request $request) |
| login(Request $request) |
| logout(Request $request) |
| updateUser($id, Request $request) |
| delete($id) |
| insertUser(Request $request) |
| update($id, Request $request) |

## Modeli

| Anketa |
|---|
| show() |
| showAnketa() |
| findUser() |
| insertVote() |
| incResponse() |
| insertResponse() |
| insertResponse1() |
| insertPoll() |
| updatePoll() |
| getPoll() |
| updateResponse() |
| updateResponse1() |
| delete() |
| activate() |
| deactivate() |

| About |
|---|
| get() |

| **Comment** |
|---|
| insertComment() |
| getAll($id) |
| get() |
| deleteComment() |
| updateComment() |

| **Navigation** |
|---|
| all() |
| getAll() |
| getUser() |
| getAdmin() |
| getNav($id) |
| updateNav($id) |
| deleteNav() |
| insertNav() |

| **Post** |
|---|
| getAll() |
| getPost($id) |
| Post($id) |
| insertPost() |
| updatePost() |
| deletePost() |
| views($id, $ip) |

| **Registracija** |
|---|
| reg() |
| log() |

| **Role** |
|---|
| getAll() |
| getRole($id) |
| deleteRole() |
| insertRole() |
| updateRole($id) |

| **Slika** |
|---|
| getSlika() |
| insert() |
| updatePostSlika() |
| deleteSlika() |

| **User** |
|---|
| getUsers() |
| getUser($id) |
| deleteUser() |
| user($id) |
| userAbout($id) |
| updateUser($id) |
| updateAdminUser($id) |
| insertUser() |

## View

| admin |
|---|
| admin_insert_nav.blade.php |
| admin_insert_poll.blade.php |
| admin_insert_post.blade.php |
| admin_insert_role.blade.php |
| admin_insert_user.blade.php |
| admin_link.blade.php |
| admin_nav.blade.php |
| admin_poll.blade.php |
| admin_posts.blade.php |
| admin_roles.blade.php |
| admin_update_nav.blade.php |
| admin_update_poll.blade.php |
| admin_update_post.blade.php |
| admin_update_role.blade.php |
| admin_update_user.blade.php |
| admin_users.blade.php |

| components |
|---|
| footer.blade.php |
| header.blade.php |
| navigation.blade.php |

| layout |
|---|
| frontLayout.blade.php |

| pages |
|---|
| about.blade.php |
| gallery.blade.php |
| index.blade.php |
| onePost.blade.php |
| register.blade.php |
| user.blade.php |
| userUpdate.blade.php |

| pages/admin |
|---|
| admin.blade.php |

## 5. javaScript kod

```
function proveraUsername()
{
    var username = document.getElementById('tbUsername').value;
    var regUsername = /^[\w\d\s]{2,20}$/;

    if(!regUsername.test(username))
    {
       document.getElementById('tbUsername').style.border = '2px solid red';
        document.getElementById('tbUsername').style.color = 'red';
    }
    else
    {
        document.getElementById('tbUsername').style.border = '2px solid green';
        document.getElementById('tbUsername').style.color = 'black';
    }
}
function proveraFirstName()
{
    var fName = document.getElementById('tbFirstName').value;
    var regFName = /^[A-Z][a-z]{2,20}$/;

    if(!regFName.test(fName))
    {
       document.getElementById('tbFirstName').style.border = '2px solid red';
        document.getElementById('tbFirstName').style.color = 'red';
    }
    else
    {
        document.getElementById('tbFirstName').style.border = '2px solid green';
        document.getElementById('tbFirstName').style.color = 'black';
    }
}
function proveraLastName()
{
    var lName = document.getElementById('tbLastName').value;
    var regLName = /^[A-Z][a-z]{3,25}$/;

    if(!regLName.test(lName))
    {
       document.getElementById('tbLastName').style.border = '2px solid red';
        document.getElementById('tbLastName').style.color = 'red';
    }
    else
    {
        document.getElementById('tbLastName').style.border = '2px solid green';
        document.getElementById('tbLastName').style.color = 'black';
    }
}
function proveraEmail()
{
    var email = document.getElementById('tbEmail').value;
    var regEmail = /^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$/;

    if(!regEmail.test(email))
    {
      document.getElementById('tbEmail').style.border = '2px solid red';
        document.getElementById('tbEmail').style.color = 'red';
    }
    else
    {
        document.getElementById('tbEmail').style.border = '2px solid green';
        document.getElementById('tbEmail').style.color = 'black';
    }
}
function proveraPassword()
{
    var password = document.getElementById('tbPassword').value;
    var regPassword = /^[\w\d\s]{3,25}$/;

    if(!regPassword.test(password))
    {
      document.getElementById('tbPassword').style.border = '2px solid red';
        document.getElementById('tbPassword').style.color = 'red';
    }
    else
    {
        document.getElementById('tbPassword').style.border = '2px solid green';
        document.getElementById('tbPassword').style.color = 'black';
    }
}
function proveraPassword2()
{
    var password = document.getElementById('tbPassword').value;
    var password2 = document.getElementById('tbPassword2').value;
```

```
      if(!password.match(password2))
      {
        document.getElementById('tbPassword2').style.border = '2px solid red';
            document.getElementById('tbPassword2').style.color = 'red';
      }
      else
      {
            document.getElementById('tbPassword2').style.border = '2px solid green';
            document.getElementById('tbPassword2').style.color = 'black';
      }
}
function ValidateFileUpload()
{
    var fuData = document.getElementById('fileChooser');
    var FileUploadPath = fuData.value;

    if (FileUploadPath === '') {
      alert("Please upload an image");

    } else {
      var Extension = FileUploadPath.substring(
          FileUploadPath.lastIndexOf('.') + 1).toLowerCase();

if (Extension === "gif" || Extension === "png" || Extension === "bmp"
          || Extension === "jpeg" || Extension === "jpg") {

        if (fuData.files && fuData.files[0]) {
          var reader = new FileReader();

          reader.onload = function(e) {
            $('#blah').attr('src', e.target.result);
          }

          reader.readAsDataURL(fuData.files[0]);
        }

      }
else {
        alert("Photo only allows file types of GIF, PNG, JPG, JPEG and BMP. ");

      }
    }
  }
```

# 6. PHP kod

## AdminController.php

```
<?php

namespace App\Http\Controllers;
use App\Models\Navigation;
use App\Models\User;
use App\Models\Post;
use App\Models\Comment;
use App\Models\Role;
use App\Models\Anketa;

class AdminController extends Controller
{
    public $data;

    public function __construct() {
        $nav = new Navigation();
        $this->data['allNav'] = $nav->all();
        $this->data['all'] = $nav->getAll();
        $this->data['user'] = $nav->getUser();
        $this->data['admin'] = $nav->getAdmin();
    }

    public function getUsers()
    {
        try
        {
            $users = new User();
            $this->data['users'] = $users->getUsers();
            return view('admin.admin_users', $this->data);

        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
        }
    }

    public function getPosts()
```

```
{
   try
   {
      $post = new Post();
      $this->data['posts'] = $post->getAll();
      return view('admin.admin_posts', $this->data);

   } catch (\Exception $ex) {
      \Log::error($ex->getMessage());
   }
}

public function getNavigations()
{
   try
   {
      return view('admin.admin_nav', $this->data);

   } catch (\Exception $ex) {
      \Log::error($ex->getMessage());
   }
}

public function getRoles()
{
   try
   {
      $role = new Role();
      $this->data['roles'] = $role->getAll();
      return view('admin.admin_roles', $this->data);

   } catch (\Exception $ex) {
      \Log::error($ex->getMessage());
   }
}

public function getPoll()
{
   try
   {
      $poll = new Anketa();
      $this->data['polls'] = $poll->showAnketa();
      return view('admin.admin_poll', $this->data);
```

```
    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showFormUser()
{
    try
    {
        $role = new Role();
        $this->data['roles'] = $role->getAll();
        return view('admin.admin_insert_user', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showFormPost()
{
    try
    {
        $getUser = new User();
        $this->data['allUsers'] = $getUser->getUsers();
        return view('admin.admin_insert_post', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showFormNav()
{
    try
    {
        return view('admin.admin_insert_nav', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showFormRole()
{
    try
    {
        return view('admin.admin_insert_role', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showFormPoll()
{
    try
    {
        return view('admin.admin_insert_poll', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showUserForm($id)
{
    try
    {
        $s = new User();
        $t = new Role();
        $this->data['showKor'] = $s->getUser($id);
        $this->data['roles'] = $t->getAll();
        return view('admin.admin_update_user', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showPostForm($id)
{
    try
    {
        $p = new Post();
        $this->data['showPost'] = $p->Post($id);
        return view('admin.admin_update_post', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showNavForm($id)
{
    try
    {
        $n = new Navigation();
        $this->data['showNav'] = $n->getNav($id);
        return view('admin.admin_update_nav', $this->data);
    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showRoleForm($id)
{
    try
    {
        $r = new Role();
        $this->data['showRole'] = $r->getRole($id);
        return view('admin.admin_update_role', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showPollForm($id)
{
    try
    {
        $p = new Anketa();
        $p->id_anketa = $id;
        $this->data['showPoll'] = $p->getPoll();
        return view('admin.admin_update_poll', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function getGallery()
```

```
    {
        try
        {
            $galerija = new Post();
            $this->data['slike'] = $galerija->getAll();
            return view('pages.gallery', $this->data);

        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
        }
    }
}
```

## AnketaController.php

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Anketa;
use DateTime;

class AnketaController extends Controller
{
    public function storeVote(Request $request)
    {
        $anketa = new Anketa();
        $anketa->id_user = $request->get('id_user');
        $anketa->id_anketa = $request->get('id_anketa');
        $anketa->id_odgovor = $request->get('id_odgovor');

        try {
            $rez = $anketa->insertVote();
            $anketa->incResponse();
            return response($rez, 200);
         } catch (\Exception $ex) {
            \Log::error('Moja greska: '.$ex->getMessage());
            return response(500);
        }
    }
```

```
public function insertPoll(Request $request)
{
    $request->validate([
        'question' => 'required',
        'response' => 'required',
        'response1' => 'required'
    ],[
        'question.required' => 'Pitanje je obavezno!',
        'response.required' => 'Odgovor 1 je obavezan!',
        'response1.required' => 'Odgovor 2 je obavezan!',
    ]);

    $question = $request->get('question');
    $response = $request->get('response');

    $response1 = $request->get('response1');

    $date = new DateTime();
    $date->format('d.m.Y');

    try {
        $anketa = new Anketa();
        $anketa->pitanje = $question;
        $anketa->odgovor = $response;
        $anketa->odgovor1 = $response1;
        $anketa->created_at = $date;
        $anketa->updated_at = null;
        $anketa->id_anketa = $anketa->insertPoll();
        $anketa->insertResponse();
        $anketa->insertResponse1();

        return redirect('/admin/poll')->with('success', 'Uspesno ste dodali anketu.');

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri dodavanju ankete!');
    }
}

public function update(Request $request, $id)
{
    $request->validate([
        'question' => 'required',
        'response' => 'required',
        'response1' => 'required'
    ],[
       'question.required' => 'Pitanje je obavezno!',
       'response.required' => 'Odgovor 1 je obavezan!',
       'response1.required' => 'Odgovor 2 je obavezan!',
    ]);

    $question = $request->get('question');
    $response = $request->get('response');
    $response1 = $request->get('response1');
    $idRes = $request->get('odg');
    $idRes1 = $request->get('odg1');

    $date = new DateTime();
    $date->format('d.m.Y');

    try {
        $anketa = new Anketa();
        $anketa->id_anketa = $id;
        $anketa->pitanje = $question;
        $anketa->updated_at = $date;
        $anketa->id_response = $idRes;
        $anketa->id_response1 = $idRes1;
        $anketa->odgovor = $response;
        $anketa->odgovor1 = $response1;

        $anketa->updateResponse();
        $anketa->updateResponse1();

        $anketa->updatePoll();


        return redirect('/admin/poll')->with('success', 'Anketa uspesno promenjena.');

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri menjanju ankete!');
    }
}

public function delete($id)
{
    $anketa = new Anketa();
    $anketa->id_anketa = $id;
```

```
        $anketa->delete();

       try{
           return redirect('/admin/poll')->with('success', 'Uspesno brisanje ankete.');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska prilikom brisanja ankete!');
        }
    }

    public function active($id)
    {
        try {

            $anketa = new Anketa();
            $anketa->id_anketa = $id;
            $anketa->deactivate();
            $anketa->activate();

            return redirect('/admin/poll')->with('success', 'Aktivirana anketa.');

        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska!');
        }
    }
}
```

## CommentController.php

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Comment;
use DateTime;

class CommentController extends Controller
{
    public function insertComment($id, Request $request)
    {
        $rules = ([
           'comment' => 'required'
        ]);

        $messages = ([
           'comment.required' => 'Post some comment!'
        ]);

        $request->validate($rules, $messages);

        $comment = $request->get('comment');

        if(session()->has('user'))
        {
            $id_korisnik = session()->get('user')[0]->id_korisnik;
        }

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            $komentar = new Comment();
            $komentar->tekst = $comment;
            $komentar->id_post = $id;
            $komentar->id_korisnik = $id_korisnik;
            $komentar->time = $date;

            $komentar->insertComment();

            return redirect('/post/'.$komentar->id_post)->with('success', 'Uspesno ste dodali
komentar.');

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error','Greska pri unosu komentara u bazu!');
        }
        catch(\ErrorException $ex) {
            \Log::error('Greska '.$ex->getMessage());
            return redirect()->back()->with('error','Desila se greska..');
        }
    }
```

```
public function deleteComment($id, $cId)
{
    try
    {
        $komentar = new Comment();

        $komentar->id_komentar =$cId;
        $komentar->deleteComment();

        return redirect()->back()->with('success', 'Uspesno ste izbrisali komentar.');

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri brisanju komentara iz baze!');
    }
    catch(\ErrorException $ex) {
        \Log::error('Greska '.$ex->getMessage());
        return redirect()->back()->with('error', 'Desila se greska..');
    }
}

public function updateComment(Request $request, $id, $cId)
{
    try
    {
        $komentar = new Comment();

        $tekst = $request->get('comment');
        $komentar->id_komentar = $cId;
        $komentar->tekst = $tekst;

        $komentar->updateComment();

        return redirect('/post/'.$id)->with('success','Uspesan update komentara! ');

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri menjanju komentara u bazi!');
    }
    catch(\ErrorException $ex) {
        \Log::error('Greska '.$ex->getMessage());
        return redirect()->back()->with('error', 'Desila se greska..');
    }
    }
}
```

### FrontEndController.php

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Navigation;
use App\Models\Post;
use App\Models\About;
use App\Models\User;
use App\Models\Comment;
use App\Models\Anketa;

class FrontEndController extends Controller {

    public $data;

    public function __construct() {
        $nav = new Navigation();
        $this->data['all'] = $nav->getAll();
        $this->data['user'] = $nav->getUser();
        $this->data['admin'] = $nav->getAdmin();
    }
    public function index()
    {
        try
        {
            $post = new Post();
            $this->data['posts'] = $post->getAll();
            return view('pages.index', $this->data);

        } catch (\Exception $ex) {
            \Log::error($ex->getMessage());
        }

    }
    public function getPost(Request $request, $id, $cId = null)
    {
```

```
    try
    {
        $post = new Post();
        $comm = new Comment();
        $comm->id_komentar = $cId;
        $post->views($id, $request->ip());
        $this->data['singlePost'] = $post->getPost($id);
        $this->data['onePost'] = $post->Post($id);
        $this->data['singleComm'] = $comm->getAll($id);
        $this->data['selectedComm'] = $comm->get();
        return view('pages.onePost', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}
public function getRegister()
{
    try
    {
        return view('pages.register', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}
public function about(Request $request)
{
    if($request->session()->has('user'))
    {
        $idUser = $request->session()->get('user')[0]->id_korisnik;
        $anketa = new Anketa();
        $anketa->id_user = $idUser;
        $rez = $anketa->findUser();
        if($rez)
        {
            $this->data['rez'] = $rez;
        }
        else
        {
            $this->data['rez'] = $rez;
        }
    }
    try
    {
        $about = new About();
        $this->data['me'] = $about->get();
        return view('pages.about', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}
public function user()
{
    try
    {
        $user = new User();
        if(session()->has('user'))
        {
            $id = session()->get('user')[0]->id_korisnik;
            $this->data['showKor'] = $user->user($id);
            $this->data['showAbout'] = $user->userAbout($id);
            return view('pages.user', $this->data);
        }
    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}
public function showUser($id)
{
    try
    {
        $u = new User();
        $this->data['showUser'] = $u->getUser($id);
        return view('pages.userUpdate', $this->data);

    } catch (\Exception $ex) {
        \Log::error($ex->getMessage());
    }
}

public function showAnketa()
{
    try{
        $anketa = new Anketa();
        $mojaAnketa = $anketa->show();
        return response($mojaAnketa, 200);
```

```
            }
            catch(\Exception $ex){
            \Log::error('Greska: ' . $ex->getMessage());
            return response(null, 500);
            }
    }

    public function download(){
        $headers = array(
          'Content-Type: application/pdf',
        );
        return response()->download(public_path('Dokumentacija.pdf'), 'Dokumentacija.pdf',
$headers);
    }
}
```

## NavigationController.php

```
<?php

namespace App\Http\Controllers;
use App\Models\Navigation;
use Illuminate\Http\Request;
use DateTime;

class NavigationController extends Controller {

    public function delete($id)
    {
        try
        {
            $nav = new Navigation();

            $nav->id = $id;
            $nav->deleteNav();

            return redirect()->back()->with('success', 'Uspesno ste izbrisali meni u navigaciji.');

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri brisanju menija iz baze!');
        }
        catch(\ErrorException $ex) {
            \Log::error('Greska '.$ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function updateNav($id, Request $request)
    {
        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            $naziv = $request->get('naziv');
            $putanja = $request->get('putanja');
            $navigation = new Navigation();

            $navigation->id = $id;
            $navigation->naziv = $naziv;
            $navigation->putanja = $putanja;
            $navigation->updated_at = $date;

            $navigation->updateNav($id);

            return redirect('/admin/navigation')->with('success', 'Uspesno ste promenili meni
'.$navigation->naziv);

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska sa bazom pri promeni menija!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function insertNav(Request $request)
    {
        $request->validate([
            'naziv' => 'required',
            'putanja' => 'required'
        ],[
            'required' => 'Polje :attribute je obavezno.'
        ]);

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            $nav = new Navigation();
            $nav->naziv = $request->get('naziv');
            $nav->putanja = $request->get('putanja');
            $nav->created_at = $date;
            $nav->updated_at = null;

            $nav->insertNav();

            return redirect('/admin/navigation')->with('success','Uspesno ste dodali meni u
navigaciji '.$nav->naziv);

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri unosu menija u bazu!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }
}
```

## PostController.php

```
<?php

namespace App\Http\Controllers;
use App\Models\Post;
use App\Models\Slika;
use Illuminate\Support\Facades\File;
use Illuminate\Http\Request;
use DateTime;

class PostController extends COntroller {

    public function insertPost(Request $request)
    {

        $request->validate([
            'slika' => 'required|mimes:jpg,jpeg,png,gif|max:4000',
            'naziv' => 'required'
        ],[
            'mimes' => 'Format nije dozvoljen!',
            'max' => 'Fajl ne sme biti veci od :max KB.',
            'required' => 'Polje :attribute je obavezno.'
        ]);

        $photo = $request->file('slika');

        $tmp_putanja = $photo->getPathName();
        $ekstenzija = $photo->getClientOriginalExtension();
        $ime_fajla = time().'.'.$ekstenzija;
        $putanja = 'images/'.$ime_fajla;

        $putanja_server = public_path($putanja);

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            File::move($tmp_putanja, $putanja_server);

            $slika = new Slika();
            $slika->alt = $ime_fajla;
            $slika->putanja = $putanja;
            $slika_id = $slika->insert();

            $post = new Post();
            $post->naslov = $request->get('naziv');
            $post->id_slika = $slika_id;
            $post->id_korisnik = session()->get('user')[0]->id_korisnik;
            $post->created_at = $date;
            $post->insertPost();
            return redirect('/user')->with('success','Uspesno ste dodali post '.$post->naslov);

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri unosu posta u bazu!');
        }
        catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
            \Log::error('Problem sa file-om '.$ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri dodavanju slike!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function insertAdminPost(Request $request)
    {
        $request->validate([
            'slika' => 'required|mimes:jpg,jpeg,png,gif|max:4000',
            'ddlKorisnik' => 'required|not_in:0',
            'naziv' => 'required'
        ],[
            'slika.mimes' => 'Format nije dozvoljen!',
            'max' => 'Fajl ne sme biti veci od :max KB.',
            'required' => 'Polje :attribute je obavezno.',
            'ddlKorisnik.not_in' => 'Izaberite korisnika!'
        ]);


        $photo = $request->file('slika');

        $tmp_putanja = $photo->getPathName();
        $ekstenzija = $photo->getClientOriginalExtension();
        $ime_fajla = time().'.'.$ekstenzija;
        $putanja = 'images/'.$ime_fajla;

        $putanja_server = public_path($putanja);

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            File::move($tmp_putanja, $putanja_server);

            $slika = new Slika();
            $slika->alt = $ime_fajla;
            $slika->putanja = $putanja;
            $slika_id = $slika->insert();

            $post = new Post();
            $post->naslov = $request->get('naziv');
            $post->id_slika = $slika_id;
            $post->id_korisnik = $request->get('ddlKorisnik');
            $post->created_at = $date;
            $post->updated_at = null;
            $post->insertPost();

            return redirect('/admin/post')->with('success','Uspesno ste dodali post '.$post->naslov);

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri unosu posta u bazu!');
        }
        catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
            \Log::error('Problem sa file-om '.$ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri dodavanju slike!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function updatePost($id, Request $request)
    {
        $request->validate([
            'slika' => 'mimes:jpg,jpeg,png,gif|max:4000'
        ],[
            'slika.mimes' => 'Format nije dozvoljen!',
            'max' => 'Fajl ne sme biti veci od :max KB.'
        ]);

        $slikaPutanja = $request->file('slika');
        $date = new DateTime();
        $date->format('d.m.Y');
```

```
try
{
   if(!empty($slikaPutanja))
   {
      $slika = new Slika();
      $slika->id_pos = $id;

      $slika_to_update = $slika->getSlika();
      File::delete($slika_to_update->putanja);

      $tmp_putanja = $slikaPutanja->getPathName();
      $ime_fajla = time().'.'.$slikaPutanja->getClientOriginalExtension();
      $putanja = 'images/'.$ime_fajla;
      $putanja_server = public_path($putanja);

      File::move($tmp_putanja, $putanja_server);

      $slika->putanja = $putanja;
      $slika->alt = $ime_fajla;

      $slika->updatePostSlika();
   }

   $naslov = $request->get('naslov');
   $new = new Post();

   $new->id = $id;
   $new->naslov = $naslov;
   $new->updated_at = $date;

   $new->updatePost();

   return redirect('/admin/post')->with('success','Uspesan update post-a! '.$new->naslov);

} catch (\Illuminate\Database\QueryException $ex) {
   \Log::error($ex->getMessage());
   return redirect()->back()->with('error', 'Greska sa bazom pri menjanju post-a!');
}
catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
   \Log::error('Problem sa file-om '.$ex->getMessage());
   return redirect()->back()->with('error', 'Greska pri menjanju slike!');
}
catch (\Exception $ex) {
   \Log::error($ex->getMessage());
```

```
        return redirect()->back()->with('error', 'Desila se greska..');
    }
}

public function deletePost($id)
{
    $postUser = new Post();
    $postUser->id = $id;

    $slikaUser = new Slika();
    $slikaUser->id_pos = $id;

    try
    {
        $post_to_delete = $slikaUser->getSlika();
        File::delete($post_to_delete->putanja);

        $slikaUser->deleteSlika();
        $postUser->deletePost();

        return redirect('/user')->with('success', 'Uspesno ste izbrisali post.');

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri brisanju slike iz baze!');
    }
    catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
        \Log::error('Problem sa file-om '.$ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri brisanju slike!');
    }
    catch (\Exception $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Desila se greska..');
    }
}

public function delete($id)
{
    $post = new Post();
    $post->id = $id;

    $slika = new Slika();
    $slika->id_pos = $id;
```

```
        try{
           $post_to_delete = $slika->getSlika();
            File::delete($post_to_delete->putanja);

            $slika->deleteSlika();
            $post->deletePost();

            return redirect('/admin/post')->with('success', 'Uspesno ste izbrisali post ');

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri brisanju slike iz baze!');
        }
        catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
            \Log::error('Problem sa file-om '.$ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri brisanju slike!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }
}
```

## RoleController.php

```
<?php

namespace App\Http\Controllers;
use App\Models\Role;
use Illuminate\Http\Request;
use DateTime;

class RoleController extends Controller{

    public function delete($id)
    {
        try
        {
            $role = new Role();
            $role->id = $id;

            $role->deleteRole();

            return redirect('/admin/role')->with('success', 'Uspesno ste izbrisali ulogu.');

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error','Greska pri brisanju uloga iz baze!');
        }
        catch(\ErrorException $ex) {
            \Log::error('Greska '.$ex->getMessage());
            return redirect()->back()->with('error','Desila se greska..');
        }
    }

    public function updateRole($id, Request $request)
    {
        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            $naziv = $request->get('naziv');
            $role = new Role();

            $role->id = $id;
            $role->naziv = $naziv;
            $role->updated_at = $date;

            $role->updateRole($id);

            return redirect('/admin/role')->with('success', 'Uspesno ste promenili ulogu '.$role->naziv);

        } catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska sa bazom pri promeni uloge!');
        }
        catch(\ErrorException $ex) {
            \Log::error('Greska '.$ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function insertRole(Request $request)
    {
        $request->validate([
            'naziv' => 'required'
        ],[
            'required' => 'Polje :attribute je obavezno.'
        ]);

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            $role = new Role();
            $role->naziv = $request->get('naziv');
            $role->created_at = $date;
            $role->updated_at = null;
            $rez = $role->insertRole();

            if($rez == 1)
            {
                return redirect('/admin/role')->with('success','Uspesno ste dodali ulogu '.$role-
>naziv);
            }
            else
            {
                return redirect('/admin/show/insert/role')->with('error','Niste uspeli da dodate ulogu
'.$role->naziv);
            }
        }
        catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri unosu!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }
}
```

## UserController.php

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;
use App\Models\Registracija;
use App\Models\User;
use DateTime;

class UserController extends Controller {

    public function register(Request $request)
    {
        $rules = ([
            'korisnickoIme' => 'regex:/^[\w\d\s]{2,20}$/|unique:korisnici,korisnicko_ime',
            'ime' => 'regex:/^[A-z]{3,20}$/',
            'prezime' => 'regex:/^[A-z]{4,25}$/',
            'email' => 'regex:/^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$/',
            'lozinka' => 'regex:/^[\w\d\s]{3,15}$/|confirmed',
            'avatar' => 'mimes:jpg,jpeg,png,gif|max:4000'
        ]);

        $custom_messages = ([
            'max' => 'Fajl ne sme biti veci od :max KB',
            'mimes' => 'Dozvoljeni formati su: :values',
            'korisnickoIme.regex' => 'Polje :attribute nije u ispravnom formatu',
            'ime.regex' => 'Polje :attribute nije u ispravnom formatu',
            'prezime.regex' => 'Polje :attribute nije u ispravnom formatu',
            'email.regex' => 'Polje :attribute nije u ispravnom formatu',
            'lozinka.regex' => 'Polje :attribute nije u ispravnom formatu',
            'korisnickoIme.unique' => 'Korisnicko ime je zauzeto'
        ]);

        $request->validate($rules, $custom_messages);

        $avatar = $request->file('slika');

        $tmp_putanja = $avatar->getPathName();
        $ekstenzija = $avatar->getClientOriginalExtension();
        $ime_fajla = time().'.'.$ekstenzija;
        $putanja = 'images/avatar/'.$ime_fajla;

        $putanja_server = public_path($putanja);

        $date = new DateTime();
        $date->format('d.m.Y');

        try
        {
            File::move($tmp_putanja, $putanja_server);

            $korisnik = new Registracija();
            $korisnik->korisnicko_ime = $request->get('korisnickoIme');
            $korisnik->ime = $request->get('ime');
            $korisnik->prezime = $request->get('prezime');
            $korisnik->email = $request->get('email');
            $korisnik->lozinka = $request->get('lozinka');
            $korisnik->avatar = $putanja;
            $korisnik->registered_at = $date;
            $korisnik->updated_at = null;

            $korisnik->reg();

            return redirect('/')->with('success', 'Uspesno ste se registrovali '.$korisnik-
>korisnicko_ime);
        }
        catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska sa bazom pri registraciji!');
        }
        catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
            \Log::error('Problem sa file-om '.$ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri dodavanju avatara!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function login(Request $request)
    {
        $request->validate([
            'korisnickoIme' => ['required','alpha_num'],
            'lozinka' => ['required']
            ], [
                'required' => 'Polje :attribute je obavezno!'
        ]);

        $korisnickoIme = $request->get('korisnickoIme');
        $lozinka = $request->get('lozinka');

        $korisnik = new Registracija();
        $korisnik->korisnicko_ime = $korisnickoIme;
        $korisnik->lozinka = $lozinka;

        $korisnikLogin = $korisnik->log();

        if(!empty($korisnikLogin))
        {
            $request->session()->push('user', $korisnikLogin);
            if(session()->get('user')[0]->naziv == 'admin')
            {
                return redirect('/admin')->with('success', 'Uspesno ste se ulogovali
'.$korisnickoIme);
            }
            else
            {
                return redirect('/user')->with('success', 'Uspesno ste se ulogovali '.$korisnickoIme);
            }
        }
        return redirect('/register')->with('error', 'Pogresan username ili password!');
    }

    public function logout(Request $request)
    {
        $request->session()->forget('user');
        $request->session()->flush();
        return redirect('/');
    }

    public function updateUser($id, Request $request)
    {
        $request->validate([
            'slika' => 'mimes:jpg,jpeg,png,gif|max:4000'
        ],[
            'slika.mimes' => 'Format nije dozvoljen!',
            'max' => 'Fajl ne sme biti veci od :max KB.'
```

```
]);

$korisnicko_ime = $request->get('korisnickoIme');
$email = $request->get('email');

$pass = $request->get('password');

if(strlen($pass) == 32)
{
   $password = $request->get('password');
}
else
{
   $password = md5($request->get('password'));
}

$avatar = $request->file('slika');
$date = new DateTime();
$date->format('d.m.Y');

try{
   $kor = new User();
   $kor->id = $id;
   $kor->username = $korisnicko_ime;
   $kor->email = $email;
   $kor->password = $password;
   $kor->updated_at = $date;

   if(!empty($avatar))
   {
      $korisnik_to_update = $kor->getUser($id);
      File::delete($korisnik_to_update->avatar);

      $tmp_putanja = $avatar->getPathName();
      $ime_fajla = time().'.'.$avatar->getClientOriginalExtension();
      $putanja = 'images/avatar/'.$ime_fajla;
      $putanja_server = public_path($putanja);

      File::move($tmp_putanja, $putanja_server);

      $kor->avatar = $putanja;
   }

      $kor->updateUser($id);
```

```
        return redirect('/user')->with('success','Uspesan update!');

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri menjanju korisnika u bazi!');
    }
    catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
        \Log::error('Problem sa file-om '.$ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri menjanju avatara!');
    }
    catch (\Exception $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Desila se greska..');
    }
}

public function delete($id)
{
    $user = new User();
    $user->id = $id;

    try{
        $korisnik_to_delete = $user->getUser($id);
        File::delete($korisnik_to_delete->avatar);

        $user->deleteUser();

        return redirect('/admin')->with('success', 'Uspesno ste izbrisali korisnika ');

    } catch (\Illuminate\Database\QueryException $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri bisanju korisnika iz baze!');
    }
    catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
        \Log::error('Problem sa file-om '.$ex->getMessage());
        return redirect()->back()->with('error', 'Greska pri brisanju avatara!');
    }
    catch (\Exception $ex) {
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error', 'Desila se greska..');
    }
}
```

```
public function insertUser(Request $request)
{

    $request->validate([
        'korisnickoIme' => 'unique:korisnici,korisnicko_ime',
        'ddlUloga' => 'required|not_in:0',
        'slika' => 'required|mimes:jpg,jpeg,png,gif'
    ],[
        'slika.mimes' => 'Format nije dozvoljen!',
        'max' => 'Fajl ne sme biti veci od :max KB.',
        'required' => 'Polje :attribute je obavezno.',
        'ddlUloga.not_in' => 'Izaberite ulogu.',
        'korisnickoIme.unique' => 'Korisnicko ime je zauzeto'
    ]);

    $avatar = $request->file('slika');

    $tmp_putanja = $avatar->getPathName();
    $ekstenzija = $avatar->getClientOriginalExtension();
    $ime_fajla = time().'.'.$ekstenzija;
    $putanja = 'images/avatar/'.$ime_fajla;

    $putanja_server = public_path($putanja);

    $date = new DateTime();
    $date->format('d.m.Y');

    try
    {
        File::move($tmp_putanja, $putanja_server);

        $korisnik = new User();
        $korisnik->username = $request->get('korisnickoIme');
        $korisnik->ime = $request->get('ime');
        $korisnik->prezime = $request->get('prezime');
        $korisnik->email = $request->get('email');
        $korisnik->password = $request->get('lozinka');
        $korisnik->avatar = $putanja;
        $korisnik->registered_at = $date;
        $korisnik->updated_at = null;
        $korisnik->id_uloga = $request->get('ddlUloga');

        $korisnik->insertUser();
```

```
            return redirect('/admin')->with('success', 'Uspesno ste uneli korisnika '.$korisnik->username);

        }
        catch (\Illuminate\Database\QueryException $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri unosu korisnika!');
        }
        catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
            \Log::error('Problem sa file-om '.$ex->getMessage());
            return redirect()->back()->with('error', 'Greska pri dodavanju avatara!');
        }
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }

    public function update($id, Request $request)
    {
        $request->validate([
          'ddlUloga' => 'not_in:0',
          'avatar' => 'mimes:jpg,jpeg,png,gif|max:4000'
        ],[
          'ddlUloga.not_in' => 'Uloga mora biti izabrana.',
          'max' => 'Fajl ne sme biti veci od :max KB',
          'mimes' => 'Dozvoljeni formati su: :values'
        ]);

        $korisnicko_ime = $request->get('korisnickoIme');
        $ime = $request->get('ime');
        $prezime = $request->get('prezime');
        $email = $request->get('email');
        $id_uloga = $request->get('ddlUloga');

        $pass = $request->get('lozinka');
        if(strlen($pass) == 32)
        {
            $password = $pass;
        }
        else
        {
            $password = md5($pass);
        }
```

```
$avatar = $request->file('avatar');
$date = new DateTime();
$date->format('d.m.Y');

try
{
    $kor = new User();
    $kor->id = $id;
    $kor->username = $korisnicko_ime;
    $kor->ime = $ime;
    $kor->prezime = $prezime;
    $kor->email = $email;
    $kor->password = $password;
    $kor->updated_at = $date;
    $kor->id_uloga = $id_uloga;

    if(!empty($avatar))
    {
        $korisnik_to_update = $kor->getUser($id);
        File::delete($korisnik_to_update->avatar);

        $tmp_putanja = $avatar->getPathName();
        $ime_fajla = time().'.'.$avatar->getClientOriginalExtension();
        $putanja = 'images/avatar/'.$ime_fajla;
        $putanja_server = public_path($putanja);

        File::move($tmp_putanja, $putanja_server);

        $kor->avatar = $putanja;
    }

    $kor->updateAdminUser($id);

    return redirect('/admin')->with('success','Uspesan update korisnika! '.$kor->username);

} catch (\Illuminate\Database\QueryException $ex) {
    \Log::error($ex->getMessage());
    return redirect()->back()->with('error', 'Greska sa bazom pri menjanju korisnika!');
}
catch (\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
    \Log::error('Problem sa file-om '.$ex->getMessage());
    return redirect()->back()->with('error', 'Greska pri menjanju avatara!');
}
```

```
        catch (\Exception $ex) {
            \Log::error($ex->getMessage());
            return redirect()->back()->with('error', 'Desila se greska..');
        }
    }
}
```

## AdminMiddleware.php

```
<?php

namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
use function abort;

class AdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {

        if($request->session()->has('user'))
        {
            $korisnik = $request->session()->get('user')[0];
            if($korisnik->naziv == 'admin')
            {
                return $next($request);
            }
            abort(404);
        }
        abort(404);
        \Log::error('Pokusaj upada preko ' . $request->get()->ip() . ' adrese');
    }
}
```

## UserMiddleware.php

```
<?php

namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
use function abort;

class UserMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {

        if($request->session()->has('user'))
        {
            return $next($request);
        }
        abort(404);
        \Log::error('Pokusaj upada preko ' . $request->get()->ip() . ' adrese');
    }
}
```

## Models/About.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class About {

    public function get()
    {
        return DB::table('about')
                ->first();
    }
}
```

## Models/Anketa.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Anketa {

    public $id_user;
    public $id_anketa;
    public $id_odgovor;
    public $odgovor;
    public $odgovor1;
    public $response;
    public $response1;
    public $pitanje;
    public $created_at;
    public $updated_at;

    public function show()
    {
        return DB::table('anketa')
                ->select('*', 'anketa.id_anketa as idAnketa','odgovori.id_odgovor as idOdgovor')
```

```
            ->join('odgovori', 'odgovori.id_anketa', '=', 'anketa.id_anketa')
            ->where('active', '=', 1)
            ->get();
}

public function showAnketa(){
    return DB::table('anketa')
            ->get();
}

public function findUser()
{
    return DB::table('anketa_vezna')
            ->join('anketa', 'anketa.id_anketa', '=', 'anketa_vezna.id_anketa')
            ->where([
                ['anketa_vezna.id_user', '=', $this->id_user],
                ['anketa.active', '=', 1]
            ])
            ->first();
}

public function insertVote()
{
    return DB::table('anketa_vezna')
            ->insertGetId([
                'id_anketa' => $this->id_anketa,
                'id_user' => $this->id_user
            ]);
}

public function incResponse()
{
    return DB::table('odgovori')
            ->where('id_odgovor', '=', $this->id_odgovor)
            ->increment('br_glasova', 1);
}

public function insertResponse()
{
    return DB::table('odgovori')
            ->insert([
                'id_anketa' => $this->id_anketa,
                'odgovor' => $this->odgovor,
                'br_glasova' => 0
```

```
        ]);
}

public function insertResponse1()
{
    return DB::table('odgovori')
            ->insert([
                'id_anketa' => $this->id_anketa,
                'odgovor' => $this->odgovor1,
                'br_glasova' => 0

        ]);
}

 public function insertPoll()
{
    return DB::table('anketa')
            ->insertGetId([
                'pitanje' => $this->pitanje,
                'active' => 0,
                'created_at' => $this->created_at,
                'updated_at' => $this->updated_at
        ]);
}

public function updatePoll()
{
    return DB::table('anketa')
            ->where('id_anketa', '=', $this->id_anketa)
            ->update([
                'pitanje' => $this->pitanje,
                'updated_at' => $this->updated_at
        ]);
}


public function getPoll()
{
    return DB::table('anketa')
            ->select('*', 'anketa.id_anketa as idAnketa', 'odgovori.id_odgovor as idOdg')
            ->join('odgovori', 'odgovori.id_anketa', '=', 'anketa.id_anketa')
            ->where('anketa.id_anketa', '=', $this->id_anketa)
            ->get();
}
```

```
public function updateResponse()
{
    return DB::table('odgovori')
            ->where([
                'id_anketa' => $this->id_anketa,
                'id_odgovor' => $this->id_response
            ])
            ->update([
               'odgovor' => $this->odgovor
            ]);
}

public function updateResponse1()
{
    return DB::table('odgovori')
            ->where([
                'id_anketa' => $this->id_anketa,
                'id_odgovor' => $this->id_response1
            ])
             ->update([
               'odgovor' => $this->odgovor1
            ]);
}

public function delete(){
    return DB::table('anketa')
            ->where('id_anketa', $this->id_anketa)
            ->delete();
}

public function activate()
{
    return DB::table('anketa')
            ->where('id_anketa', '=', $this->id_anketa)
            ->update([
                'active' => 1
            ]);
}

public function deactivate()
{
    return DB::table('anketa')
            ->update([
                'active' => 0
            ]);
    }
}
```

## Models/Comment.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Comment {

    public $id_komentar;
    public $tekst;
    public $id_post;
    public $id_korisnik;
    public $time;

    public function insertComment()
    {
        return DB::table('komentar')
                ->insert([
                   'tekst' => $this->tekst,
                   'id_post' => $this->id_post,
                   'id_korisnik' => $this->id_korisnik,
                   'time' => $this->time
                ]);
    }

    public function getAll($id)
    {
        return DB::table('komentar')
            ->select('*')
            ->join('korisnici', 'komentar.id_korisnik', '=', 'korisnici.id_korisnik')
            ->join('post AS p', 'p.id_post', '=', 'komentar.id_post')
            ->join('slika', 'slika.id_slika', '=', 'p.id_slika')
            ->where('p.id_post', '=', $id)
            ->orderBy('time', 'desc')
            ->get();
    }

    public function get()
    {
        return DB::table('komentar')
                ->select('*')
                ->where('id_komentar', $this->id_komentar)
                ->first();
    }

    public function deleteComment()
    {
        return DB::table('komentar')
                ->where('id_komentar', $this->id_komentar)
                ->delete();
    }

    public function updateComment()
    {
        $rezultat = DB::table('komentar')
                ->where('id_komentar', $this->id_komentar)
                ->update([
                    'tekst' => $this->tekst
                ]);
        return $rezultat;
    }
}
```

## Models/Navigation.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;


class Navigation {

    public $id;
```

```
}

public $naziv;
public $putanja;
public $created_at;
public $updated_at;

public function all()
{
   return DB::table('navigacija')
           ->get();
}

public function getAll()
{
   return DB::table('navigacija')
           ->whereNotIn('id_navigacija', [6,7,8])
           ->get();
}

public function getUser()
{
   return DB::table('navigacija')
           ->whereNotIn('id_navigacija', [5,7])
           ->get();
}

public function getAdmin()
{
   return DB::table('navigacija')
           ->whereNotIn('id_navigacija', [5])
           ->get();
}

public function getNav($id)
{
   $rezultat = DB::table('navigacija')
           ->select('*')
           ->where('id_navigacija', $id)
           ->first();
   return $rezultat;
}

public function updateNav($id)
{
   $data = ([
           'naziv' => $this->naziv,
           'putanja' => $this->putanja,
           'updated_at' => $this->updated_at
        ]);

        $rezultat = DB::table('navigacija')
                ->where('id_navigacija', $id)
                ->update($data);
        return $rezultat;
    }

    public function deleteNav()
    {
        $rezultat = DB::table('navigacija')
                ->where('id_navigacija', $this->id)
                ->delete();
        return $rezultat;
    }

    public function insertNav()
    {
        $rezultat = DB::table('navigacija')
                ->insert([
                   'naziv' =>  $this->naziv,
                   'putanja' => $this->putanja,
                   'created_at' => $this->created_at,
                   'updated_at' => $this->updated_at
                ]);
        return $rezultat;
    }
}
```

## Models/Post.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;
```

```
class Post {

    public $id;
    public $naslov;
    public $id_korisnik;
    public $id_slika;
    public $created_at;
    public $updated_at;
    public $putanja;
    public $ip;

    public function getAll()
    {
        $rezultat = DB::table('post')
                ->select('*', 'post.created_at as postCreated', 'post.updated_at as postUpdated')
                ->join('slika', 'slika.id_slika', '=', 'post.id_slika')
                ->join('korisnici', 'korisnici.id_korisnik', '=', 'post.id_korisnik')
                ->orderBy('created_at', 'desc')
                ->get();
        return $rezultat;
    }

    public function getPost($id)
    {
        $rezultat = DB::table('post')
                ->select('*',
                        'korisnici.korisnicko_ime as postKorisnik',
                        'k.korisnicko_ime as komentarKorisnik',
                        \DB::raw("(SELECT count(id_pregled) FROM pregledi WHERE id_post =
$id) as views"))
                ->join('slika', 'slika.id_slika', '=', 'post.id_slika')
                ->join('korisnici', 'korisnici.id_korisnik', '=', 'post.id_korisnik')
                ->leftJoin('komentar', 'post.id_post', '=', 'komentar.id_post')
                ->leftJoin('korisnici AS k', 'k.id_korisnik', '=', 'komentar.id_korisnik')
                ->where('post.id_post', $id)
                ->first();
        return $rezultat;
    }

    public function Post($id)
    {
        $rezultat = DB::table('post')
                ->select('*')
                ->join('slika', 'slika.id_slika', '=', 'post.id_slika')
```

```
            ->where('post.id_post', $id)
            ->first();
    return $rezultat;
}

public function insertPost()
{
    $rezultat = DB::table('post')
            ->insert([
               'naslov' =>  $this->naslov,
               'id_slika' => $this->id_slika,
               'id_korisnik' => $this->id_korisnik,
               'created_at' => $this->created_at,
               'updated_at' => $this->updated_at
            ]);
    return $rezultat;
}

public function updatePost()
{
    $data['naslov'] = $this->naslov;
    $data['updated_at'] = $this->updated_at;

    $rezultat = DB::table('post')
            ->where('id_post', $this->id)
            ->update($data);
    return $rezultat;
}

public function deletePost()
{
    $rezultat = DB::table('post')
            ->where('id_post', $this->id)
            ->delete();
    return $rezultat;
}

public function views($id, $ip)
{
    return DB::table('pregledi')
        ->insert([
            'ip' => $ip,
            'id_post' => $id
        ]);
    }
}
```

## Models/Registracija.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Registracija {

    public $korisnicko_ime;
    public $ime;
    public $prezime;
    public $email;
    public $lozinka;
    public $avatar;
    public $registered_at;
    public $updated_at;

    public function reg()
    {
        $rezultat = DB::table('korisnici')
                ->insert([
                    'korisnicko_ime' => $this->korisnicko_ime,
                    'ime' => $this->ime,
                    'prezime' => $this->prezime,
                    'email' => $this->email,
                    'lozinka' => md5($this->lozinka),
                    'avatar' => $this->avatar,
                    'id_uloga' => 2,
                    'registered_at' => $this->registered_at,
                    'updated_at' => $this->updated_at
                ]);
        return $rezultat;
    }

    public function log()
    {
        $rezultat = DB::table('korisnici')
```

```
                ->select('korisnici.*', 'uloge.naziv')
                ->join('uloge', 'uloge.id_uloga', '=', 'korisnici.id_uloga')
                ->where([
                    'korisnicko_ime' => $this->korisnicko_ime,
                    'lozinka' => md5($this->lozinka)
                ])
                ->first();
        return $rezultat;
    }
}
```

## Models/Role.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Role {

    public $id;
    public $naziv;
    public $created_at;
    public $updated_at;

    public function getAll()
    {
        return DB::table('uloge')
                ->get();
    }

    public function getRole($id)
    {
        $rezultat = DB::table('uloge')
                ->select('*')
                ->where('id_uloga', $id)
                ->first();
        return $rezultat;
    }

    public function deleteRole()
    {
        $rezultat = DB::table('uloge')
                ->where('id_uloga', $this->id)
                ->delete();
        return $rezultat;
    }

    public function insertRole()
    {
        $rezultat = DB::table('uloge')
                ->insert([
                    'naziv' =>  $this->naziv,
                    'created_at' => $this->created_at,
                    'updated_at' => $this->updated_at
                ]);
        return $rezultat;
    }

    public function updateRole($id)
    {
        $data = ([
            'naziv' => $this->naziv,
            'updated_at' => $this->updated_at
        ]);

        $rezultat = DB::table('uloge')
                ->where('id_uloga', $id)
                ->update($data);
        return $rezultat;
    }
}
```

## Models/Slika.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Slika {
```

```
public $id;
public $alt;
public $putanja;
public $id_pos;

public function getSlika()
{
    $rezultat = DB::table('slika')
            ->select('*')
            ->join('post', 'post.id_slika', '=', 'slika.id_slika')
            ->where('post.id_post', $this->id_pos)
            ->first();
    return $rezultat;
}

public function insert()
{
    $id = DB::table('slika')
            ->insertGetId([
                'alt' => $this->alt,
                'putanja' => $this->putanja
            ]);
    return $id;
}

public function updatePostSlika()
{
    if(!empty($this->putanja))
    {
        $data['slika.putanja'] = $this->putanja;
    }
    $data['slika.alt'] = $this->alt;
    $rezultat = DB::table('slika')
            ->join('post', 'post.id_slika', '=', 'slika.id_slika')
            ->where('post.id_post', $this->id_pos)
            ->update($data);
    return $rezultat;
}

public function deleteSlika()
{
    return DB::table('slika')
            ->join('post', 'post.id_slika', '=', 'slika.id_slika')
            ->where('post.id_post', '=', $this->id_pos)
            ->delete();
    }
}
```

## Models/User.php

```
<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;


class User {

    public $id;
    public $avatar;
    public $username;
    public $ime;
    public $prezime;
    public $email;
    public $password;
    public $registered_at;
    public $updated_at;
    public $id_uloga;

    public function getUsers()
    {
        return DB::table('korisnici')
            ->join('uloge', 'uloge.id_uloga', '=', 'korisnici.id_uloga')
            ->get();
    }

    public function getUser($id)
    {
        $rezultat = DB::table('korisnici')
            ->select('*')
            ->join('uloge', 'uloge.id_uloga', '=', 'korisnici.id_uloga')
            ->where('id_korisnik', $id)
            ->first();
        return $rezultat;
    }
```

```
public function deleteUser()
{
    $rezultat = DB::table('korisnici')
            ->where('id_korisnik', $this->id)
            ->delete();
    return $rezultat;
}

public function user($id)
{
    $rezultat = DB::table('post')
            ->select('*', 'korisnici.updated_at as profileChanged')
            ->join('slika', 'slika.id_slika', '=', 'post.id_slika')
            ->join('korisnici', 'korisnici.id_korisnik', '=', 'post.id_korisnik')
            ->where('post.id_korisnik', '=', $id)
            ->orderBy('created_at', 'desc')
            ->get();
    return $rezultat;
}

public function userAbout($id)
{
    $rezultat = DB::table('korisnici')
            ->where('id_korisnik', '=', $id)
            ->first();
    return $rezultat;
}

public function updateUser($id)
{
    $data = ([
       'korisnicko_ime' => $this->username,
       'email' => $this->email,
       'lozinka' => $this->password,
       'updated_at' => $this->updated_at
    ]);

    if(!empty($this->avatar))
    {
        $data['avatar'] = $this->avatar;
    }

    $rezultat = DB::table('korisnici')
```

```
                ->where('id_korisnik', $id)
                ->update($data);
        return $rezultat;
    }

    public function updateAdminUser($id)
    {
        $data = ([
            'korisnicko_ime' => $this->username,
            'ime' => $this->ime,
            'prezime' => $this->prezime,
            'email' => $this->email,
            'lozinka' => $this->password,
            'updated_at' => $this->updated_at,
            'id_uloga' => $this->id_uloga
        ]);

        if(!empty($this->avatar))
        {
            $data['avatar'] = $this->avatar;
        }

        $rezultat = DB::table('korisnici')
                ->where('id_korisnik', $id)
                ->update($data);
        return $rezultat;
    }

    public function insertUser()
    {
        return DB::table('korisnici')
                ->insert([
                    'korisnicko_ime' => $this->username,
                    'ime' => $this->ime,
                    'prezime' => $this->prezime,
                    'email' => $this->email,
                    'lozinka' => md5($this->password),
                    'avatar' => $this->avatar,
                    'registered_at' => $this->registered_at,
                    'updated_at' => $this->updated_at,
                    'id_uloga' => $this->id_uloga
                ]);
    }
}
```

## admin/ admin_insert_nav.blade.php

```
@extends('pages.admin.admin')

@section('admin')

        <form method="post" action="{{asset('/admin/nav/insert')}}">
            {{csrf_field()}}
@endsection
```

## admin/ admin_insert_poll.blade.php

```
@extends('pages.admin.admin')

@section('admin')

        <form method="post" action="{{asset('/admin/poll/insert')}}">
            {{csrf_field()}}
@endsection
```

## admin/ admin_insert_post.blade.php

```
@extends('pages.admin.admin')

@section('admin')

        <form method="post" action="{{asset('/admin/post/insert')}}">
            {{csrf_field()}}

            @foreach($allUsers as $oneUser)
                            <option value="{{ $oneUser->id_korisnik }}"> {{ $oneUser-
            >korisnicko_ime }} </option>
             @endforeach
@endsection
```

## admin/ admin_insert_post.blade.php

```
@extends('pages.admin.admin')

@section('admin')

        <form method="post" action="{{asset('/admin/role/insert')}}">
            {{csrf_field()}}
@endsection
```

## admin/ admin_insert_user.blade.php

```
@extends('pages.admin.admin')

@section('admin')

        <form method="post" action="{{asset('/admin/user/insert')}}">
            {{csrf_field()}}

            @foreach($roles as $role)
                        <option value="{{ $role->id_uloga }}"> {{ $role->naziv }} </option>
            @endforeach
@endsection
```

## admin/ admin_link.blade.php

```
<ul id="buttons">
    <li><a href="{{route('admin_users')}}" class="btn btn-info btn-md">Korisnici</a></li>
    <li><a href="{{route('admin_posts')}}" class="btn btn-info btn-md">Post</a></li>
    <li><a href="{{route('admin_nav')}}" class="btn btn-info btn-md">Navigacija</a></li>
    <li><a href="{{route('admin_role')}}" class="btn btn-info btn-md">Uloge</a></li>
    <li><a href="{{route('admin_poll')}}" class="btn btn-info btn-md">Anketa</a></li>
</ul>
```

## admin/ admin_nav.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    <div class='table-wrapper'>
            @if(count($allNav) > 5)

            @endif
    <table class="table">
            @foreach($allNav as $n)
            <tr class="info">
                <td>{{$n->id_navigacija}}</td>
                <td>{{$n->naziv}}</td>
                <td>{{$n->putanja}}</td>
                <td>{{date('d.m.Y', strtotime($n->created_at))}}</td>
                <td>{{(isset($n->updated_at)) ? date('d.m.Y', strtotime($n->updated_at)) : 'nije
menjan meni'}}</td>
                <td><li><a href="{{asset('/admin/nav/'.$n->id_navigacija)}}">Brisi</a></li></td>
                <td><li><a href="{{asset('/admin/show/update/nav/'.$n-
>id_navigacija)}}">Menjaj</a></li></td>
            </tr>
            @endforeach
        </tbody>
    </table>
    <li id="buttons"><a href="{{asset('/admin/show/insert/navigation')}}" class="btn btn-info
btn-block">Unesi novi meni</a></li>
    </div>
@endsection
```

## admin/ admin_poll.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    <div class='table-wrapper'>
            @if(count($polls) > 5)

            @endif
    <table class="table">
            @foreach($polls as $p)
            <tr class="info">
                <td>{{$p->id_anketa}}</td>
                <td>{{$p->pitanje}}</td>
                <td>@if($p->active == 1)<p style="color:green;">ACTIVE</p> @else <a 
href="{{asset('/admin/poll/active/'.$p->id_anketa)}}">ACTIVATE</a>@endif</td>
                <td>{{$p->active}}</td>
                <td>{{date('d.m.Y', strtotime($p->created_at))}}</td>
                <td>{{(isset($p->updated_at)) ? date('d.m.Y', strtotime($p->updated_at)) : 'nije 
menjana anketa'}}</td>
                <td><li><a href="{{asset('/admin/poll/'.$p->id_anketa)}}">Brisi</a></li></td>
                <td><li><a href="{{asset('/admin/show/update/poll/'.$p-
>id_anketa)}}">Menjaj</a></li></td>
            </tr>
            @endforeach
        </tbody>
    </table>
    <li id="buttons"><a href="{{asset('/admin/show/insert/poll')}}" class="btn btn-info btn-
block">Unesi novu anketu</a></li>
    </div>
@endsection
```

## admin/ admin_posts.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    <div class='table-wrapper'>
            @if(count($posts) > 5)

            @endif
    <table class="table">
            @foreach($posts as $post)
            <tr class="info">
                <td>{{$post->id_post}}</td>
                <td>{{$post->naslov}}</td>
                <td><img src="{{asset($post->putanja)}}" width="150px" height="150px"></td>
                <td>{{$post->korisnicko_ime}}</td>
                <td>{{date('d.m.Y H:i:s', strtotime($post->postCreated))}}</td>
                <td>{{(isset($post->postUpdated)) ? date('d.m.Y', strtotime($post->postUpdated)) : 
'nije menjan post'}}</td>
                <td><li><a href="{{asset('/admin/post/'.$post->id_post)}}">Brisi</a></li></td>
                <td><li><a href="{{asset('/admin/show/update/post/'.$post->id_post)}}">Menjaj</a></li></td>
            </tr>
        </tbody>
        @endforeach
    </table>
    <li id="buttons"><a href="{{asset('/admin/show/insert/post')}}" class="btn btn-info btn-block">Unesi post</a></li>
    </div>
@endsection
```

## admin/ admin_roles.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    <div class='table-wrapper'>
            @if(count($roles) > 5)

            @endif
    <table class="table">

            @foreach($roles as $role)
            <tr class="info">
                <td>{{$role->id_uloga}}</td>
                <td>{{$role->naziv}}</td>
                <td>{{date('d.m.Y', strtotime($role->created_at))}}</td>
                <td>{{(isset($role->updated_at)) ? date('d.m.Y', strtotime($role->updated_at)) : 'nije menjana uloga'}}</td>
                <td><li><a href="{{asset('/admin/role/'.$role->id_uloga)}}">Brisi</a></li></td>
                <td><li><a href="{{asset('/admin/show/update/role/'.$role->id_uloga)}}">Menjaj</a></li></td>
            </tr>
            @endforeach
        </tbody>
    </table>
    <li id="buttons"><a href="{{asset('/admin/show/insert/role')}}" class="btn btn-info btn-block">Unesi novu ulogu</a></li>
    </div>
@endsection
```

## admin/ admin_update_nav.blade.php

```
@extends('pages.admin.admin')

@section('admin')
        @isset($showNav)
        <form method="post" action="{{asset('/admin/nav/update/'.$showNav-
>id_navigacija)}}">
            {{csrf_field()}}
                <input type="text" name="naziv" class="form-control" value="{{$showNav-
>naziv}}"/>
                <input type="text" name="putanja" class="form-control" value="{{$showNav-
>putanja}}"/>
                <a href="{{asset('/admin/navigation')}}" class="btn btn-danger">Cancel</a>
        @endisset
@endsection
```

## admin/ admin_update_poll.blade.php

```
@extends('pages.admin.admin')

@section('admin')
        @isset($showPoll)
        <form method="post" action="{{asset('/admin/poll/update/'.$showPoll[0]-
>idAnketa)}}">
            {{csrf_field()}}
                <input type="text" class="form-control" name='question' value='{{ $showPoll[0]-
>pitanje }}'>
                <input type="text" class="form-control" name='response' value='{{ $showPoll[0]-
>odgovor }}'>
                <input type="text" class="form-control" value="{{$showPoll[0]->br_glasova}}"
disabled=""/>
                <input type="hidden" name="odg" value="{{$showPoll[0]->idOdg}}"/>
                <input type="text" class="form-control" name='response1' value='{{ $showPoll[1]-
>odgovor }}'>
                <input type="text" class="form-control" name='num1' value="{{$showPoll[1]-
>br_glasova}}" disabled=""/>
                <input type="hidden" name="odg1" value="{{$showPoll[1]->idOdg}}"/>
                <a href="{{asset('/admin/poll')}}" class="btn btn-danger">Cancel</a>
        @endisset
@endsection
```

## admin/ admin_update_post.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    @isset($showPost)
    <form method="POST" action="{{asset('/admin/post/update/'.$showPost->id_post)}}" enctype="multipart/form-data">
      {{csrf_field()}}
        <input type="text" name="naslov" class="form-control" value="{{$showPost->naslov}}"/>
        <img src="{{asset($showPost->putanja)}}" width="150px" height="150px">
        <a href="{{asset('/admin/post')}}" class="btn btn-danger">Cancel</a>
    @endisset
@endsection
```

## admin/ admin_update_role.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    @isset($showRole)
    <form method="post" action="{{asset('/admin/role/update/'.$showRole->id_uloga)}}">
      {{csrf_field()}}
        <input type="text" name="naziv" class="form-control" value="{{$showRole->naziv}}"/>
        <a href="{{asset('/admin/role')}}" class="btn btn-danger">Cancel</a>
    @endisset
@endsection
```

## admin/ admin_update_user.blade.php

```
@extends('pages.admin.admin')

@section('admin')
    @isset($showKor)
    <form method="post" action="{{asset('/admin/user/update/'.$showKor->id_korisnik)}}"
enctype="multipart/form-data">
      {{csrf_field()}}
        <label>Trenutna uloga: {{$showKor->naziv}}</label> <br>
          @foreach($roles as $role)
            <option value="{{ $role->id_uloga }}"> {{ $role->naziv }} </option>
          @endforeach
value="{{$showKor->korisnicko_ime}}"/>
        <input type="text" name="ime" class="form-control" value="{{$showKor-
>ime}}"/>
        <input type="text" name="prezime" class="form-control" value="{{$showKor-
>prezime}}"/>
        <input type="email" name="email" class="form-control" value="{{$showKor-
>email}}"/>
        <input type="password" name="lozinka" class="form-control"
value="{{$showKor->lozinka}}"/>
        <img src="{{asset($showKor->avatar)}}" width="150px" height="150px">
        <input type="submit" name="updateUser" class="btn btn-info" value="Promeni
korisnika"/>
        <a href="{{asset('/admin')}}" class="btn btn-danger">Cancel</a>
    @endisset
@endsection
```

## admin/ admin_users.blade.php

```
@extends('pages.admin.admin')

@section('admin')
  <div class='table-wrapper'>
    @if(count($users) > 5)
    <div class='wrapper-paging'>
    @endif
  <table class="table">
      @foreach($users as $user)
       <td>{{$user->id_korisnik}}</td>
```

```
                <td>{{$user->korisnicko_ime}}</td>
                <td>{{$user->ime}}</td>
                <td>{{$user->prezime}}</td>
                <td>{{$user->email}}</td>
                <td><img src="{{asset($user->avatar)}}" width="150px" height="150px"></td>
                <td>{{date('d.m.Y', strtotime($user->registered_at))}}</td>
                <td>{{(isset($user->updated_at)) ? date('d.m.Y', strtotime($user->updated_at)) : 'nije menjan profil'}}</td>
                <td>{{$user->naziv}}</td>
                <td><li><a href="{{asset('/admin/user/'.$user->id_korisnik)}}">Brisi</a></li></td>
                <td><li><a href="{{asset('/admin/show/update/user/'.$user->id_korisnik)}}">Menjaj</a></li></td>
            </tr>
            @endforeach
    </table>
    <li id="buttons"><a href="{{asset('/admin/show/insert/user')}}" class="btn btn-info btn-block">Unesi korisnika</a></li>
@endsection
```

### components/ footer.blade.php

```
@section('ajax')
            <!-- jQuery -->
            <script src="{{asset('/')}}js/jquery.min.js"></script>
            <!-- jQuery Easing -->
            <script src="{{asset('/')}}js/jquery.easing.1.3.js"></script>
            <!-- Bootstrap -->
            <script src="{{asset('/')}}js/bootstrap.min.js"></script>
            <!-- Waypoints -->
            <script src="{{asset('/')}}js/jquery.waypoints.min.js"></script>
            <!-- Magnific Popup -->
            <script src="{{asset('/')}}js/jquery.magnific-popup.min.js"></script>
            <!-- Salvattore -->
            <script src="{{asset('/')}}js/salvattore.min.js"></script>
            <!-- Main JS -->
            <script src="{{asset('/')}}js/main.js"></script>
        <script src="{{asset('/')}}js/mine.js"></script>
        <script>
                const baseUrl = '{{asset("/")}}';
                const token = '{{csrf_token()}}';
```

```
    </script>
    @show
```

## components/ header.blade.php

```
<title>@yield('title')| Life in bloom</title>
<link rel="icon" href="{{asset('/')}}images/about/favicon.png">
    @section('css')
        <!-- Google Webfonts -->
        <link href='http://fonts.googleapis.com/css?family=Roboto:400,300,100,500'
rel='stylesheet' type='text/css'>
        <link href='https://fonts.googleapis.com/css?family=Montserrat:400,700'
rel='stylesheet' type='text/css'>

        <!-- Animate.css -->
        <link rel="stylesheet" href="{{asset('/')}}css/animate.css">
        <!-- Icomoon Icon Fonts-->
        <link rel="stylesheet" href="{{asset('/')}}css/icomoon.css">
        <!-- Magnific Popup -->
        <link rel="stylesheet" href="{{asset('/')}}css/magnific-popup.css">
        <!-- Salvattore -->
        <link rel="stylesheet" href="{{asset('/')}}css/salvattore.css">
        <!-- Theme Style -->
        <link rel="stylesheet" href="{{asset('/')}}css/style.css">
        <link rel="stylesheet" href="{{asset('/')}}css/moj.css">
        <!-- Modernizr JS -->
        <script src="{{asset('/')}}js/modernizr-2.6.2.min.js"></script>
    @show
```

## components/ navigation.blade.php

```
        @if(session()->has('user'))
            @if(session()->get('user')[0]->naziv == 'admin')
                @isset($admin)
                    @foreach($admin as $adm)
                <li class="{{ Request::is($adm->putanja) ? 'active' : null }}"><a
href="{{url($adm->putanja)}}">{{$adm->naziv}}</a></li>
                    @endforeach
```

```
          @endisset
        @elseif(session()->get('user')[0]->naziv == 'user')
          @isset($user)
            @foreach($user as $u)
          <li class="{{ Request::is($u->putanja) ? 'active' : null }}"><a href="{{url($u-
>putanja)}}">{{$u->naziv}}</a></li>
            @endforeach
          @endisset
        @endif
        @else
          @isset($all)
            @foreach($all as $a)
            <li class="{{ Request::is($a->putanja) ? 'active' : null }}"><a href="{{url($a-
>putanja)}}">{{$a->naziv}}</a></li>
            @endforeach
          @endisset
      @endif
            </ul>
        @if(session()->has('user'))
          Logged in as:&nbsp<strong>{{session()->get('user')[0]-
>korisnicko_ime}}</strong>
        @endif

        <a href="{{asset('/download')}}"><h4>Dokumentacija</h4></a>
```

## layout/ frontLayout.blade.php

```
  @include('components.header')
  <body>
    @include('components.navigation')
    @empty(!session('success'))
              <div class="alert alert-success fade in"> <a href="#" class="close" data-
dismiss="alert" aria-label="close">&times;</a> <strong> {{ session('success') }} </strong>
</div>
              @endempty

              @empty(!session('error'))
              <div class="alert alert-danger fade in"> <a href="#" class="close" data-
dismiss="alert" aria-label="close">&times;</a> <strong> {{ session('error') }} </strong>
</div>
              @endempty
```

```
                        @isset($errors)

                        <div class="errors">
                           @if($errors->any())
                              @foreach($errors->all() as $error)
                                 <div class="alert alert-danger fade in"> <a href="#" class="close"
data-dismiss="alert" aria-label="close">&times;</a> <strong> {{ $error }} </strong> </div>
                              @endforeach
                           @endif
                        </div>

                        @endisset

                        @yield('content')

        @include('components.footer')
```

## pages/ admin/admin.blade.php

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

@section('title')
   Admin
@endsection

@section('content')

@section('css')
   @parent
   <link rel="stylesheet" href="{{asset('/')}}css/paginacija.css">
@endsection

@include('admin.admin_link')

@yield('admin')

@endsection
```

## pages/about.blade.php

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

@section('title')
   About
@endsection

@section('content')
   @isset($me)
      <p><img src="{{asset($me->slika)}}" alt="{{$me->alt}}" class="img-rounded img-
responsive"></p><hr>
      <p>{{$me->text}}</p><hr>
      @if(session()->has('user'))
      <input type='hidden' value='{{session()->get('user')[0]->id_korisnik}}'
id='idUser'></input>
      @isset($rez)
         <input type='hidden' value='1' id='postoji'></input>
      @else
         <input type='hidden' value='0' id='postoji'></input>
      @endisset
      @endif
   </div>
   @endisset

   @section('ajax')
      @parent
      <script src="{{asset('/')}}js/ajax.js"></script>
   @endsection
@endsection
```

## pages/gallery.blade.php

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

@section('title')
   Gallery
@endsection

@section('content')

   @isset($slike)
```

```
        @foreach($slike as $slika)
        <a href="{{asset($slika->putanja)}}" class="image-popup fh5co-board-img"><img
src="{{asset($slika->putanja)}}" alt="{{$slika->alt}}" width="350px" height="300px"></a>
        @endforeach
   @endisset
@endsection
```

## pages/index.blade.php

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

@section('title')
   Home
@endsection

@section('content')
   @isset($posts)
      @foreach($posts as $post)
               <img src="{{asset($post->avatar)}}" width="20" height="15" alt="">
               <span>{{$post->korisnicko_ime}}</span>
                  <div class="animate-box">
                          <a href="{{asset('/post/'.$post->id_post)}}"><img src="{{asset($post-
>putanja)}}" alt="{{$post->alt}}"></a>
                  </div>
                  <div class="fh5co-desc">{{$post->naslov}}</div>
                  <div class="fh5co-desc">{{ date("F d, Y", strtotime($post->created_at))
}}</div>
      @endforeach
   @endisset
@endsection
```

## pages/onePost.blade.php

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

@section('title')
   Post
@endsection
```

```
@section('content')

        <h1 class="mt-4">{{ $singlePost->naslov }}</h1>

          by
          <strong>{{ $singlePost->postKorisnik }}</strong>

        <p>Posted on {{ date("F d, Y  H:i", strtotime($singlePost->created_at)) }}</p>

        <p>Views: {{$singlePost->views}}</p>

        <img class="img-fluid rounded" src="{{ asset($singlePost->putanja)}}" alt="{{
$singlePost->alt}}" width="750px">
        @foreach($singleComm as $comm)
                            <img src="{{asset($comm->avatar)}}" width="15px"/>
                            <p>{{ $comm->tekst }}</p> <span class="date sub-text">on {{date("F d,
Y  H:i", strtotime($comm->time))}}</span>
            @if(session()->has('user'))
                @if(session()->get('user')[0]->korisnicko_ime == $comm->korisnicko_ime )
                <a href="{{asset('/post/'.$onePost->id_post.'/delete/'.$comm->id_komentar)}}"
class="del"><span class="glyphicon glyphicon-trash"></span></a>
                <a href="{{asset('/post/'.$onePost->id_post.'/'.$comm->id_komentar)}}"
class="del"><span class="glyphicon glyphicon-edit"></span></a>
                @elseif(session()->get('user')[0]->id_uloga == 1)
                    <a href="{{asset('/post/'.$onePost->id_post.'/delete/'.$comm->id_komentar)}}"
class="del"><span class="glyphicon glyphicon-trash"></span></a>
                    <a href="{{asset('/post/'.$onePost->id_post.'/'.$comm->id_komentar)}}"
class="del"><span class="glyphicon glyphicon-edit"></span></a>
                @endif
            @endif
        @endforeach
        @if(session()->has('user'))

                <h5 class="card-header">{{isset($selectedComm) ? 'Change comment:' : 'Leave a
Comment:'}}</h5>
                    <form method="post" action="{{ isset($selectedComm) ? asset('/post/'.$onePost-
>id_post.'/'.$selectedComm->id_komentar) : asset('/post/'.$onePost->id_post.'/comment')}}">
                        {{csrf_field()}}
                          <textarea class="form-control" rows="3" name="comment">{{
isset($selectedComm) ? $selectedComm->tekst : old('comment')}}</textarea>
                        </div>
                        <button type="submit" class="btn btn-info">{{isset($selectedComm) ?
'Update comment' : 'Post comment'}}</button>
                        @if(isset($selectedComm))
```

```
                            <a href="{{asset('post/'.$onePost->id_post)}}" class="btn btn-danger">Cancel</a>
                        @endif
                    </form>
        @endif
@endsection
```

## pages/register.blade.php

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

@section('title')
    Register
@endsection

@section('content')

    @section('css')
        @parent
        <link rel="stylesheet" href="{{asset('/')}}css/forma.css">
    @endsection
                    <form id="login-form" action="{{ asset('/register/login') }}" method="post" role="form" style="display: block;">
                        {{csrf_field()}}

                    <form id="register-form" action="{{ asset('/register/reg') }}" method="post" role="form" style="display: none;" enctype="multipart/form-data">
                        {{csrf_field()}}
                                <input type="text" name="korisnickoIme" id="tbUsername" tabindex="1" class="form-control" placeholder="Username" onKeyUp="proveraUsername();">
                                <input type="text" name="ime" id="tbFirstName" tabindex="1" class="form-control" placeholder="Name" onKeyUp="proveraFirstName();">
                                <input type="text" name="prezime" id="tbLastName" tabindex="1" class="form-control" placeholder="Prezime" onKeyUp="proveraLastName();">
                                <input type="text" name="email" id="tbEmail" tabindex="2" class="form-control" placeholder="E-mail" onKeyUp="proveraEmail();">
                                <input type="password" name="lozinka" id="tbPassword" tabindex="2" class="form-control" placeholder="Password" onKeyUp="proveraPassword();">
                                <input type="password" name="lozinka_confirmation" id="tbPassword2" tabindex="2" class="form-control" placeholder="Repeat password" onKeyUp="proveraPassword2();">
                                <input type="file" name="slika" id="fileChooser" tabindex="2"
class="form-control" style="display: none;" onchange="return
ValidateFileUpload()"></label>

    @section('ajax')
        @parent
        <script src="{{asset('/')}}js/provera.js"></script>
    @endsection

@endsection
```

## pages/user.blade.php

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

@section('title')
    User
@endsection

@section('content')

    @section('css')
        @parent
        <link rel="stylesheet" href="{{asset('/')}}css/hoverImage.css">

    @endsection

        @isset($showKor)
            @if(count($showKor) == 0)
                <h3>You haven't posted yet.</h3>
            @else
                @foreach($showKor as $show)
                        <img src="{{$show->putanja}}" alt="{{$show->alt}}" width="280px"
height="180px" class="image"/>
                        <li class="middle"><a href="{{asset('/user/delete/'.$show-
>id_post)}}"><h2>Delete</h2></a></li>
                    </div>
                @endforeach
            @endif
        @endisset
    @isset($showAbout)
      <table class="table">
            <td><img src="{{asset($showAbout->avatar)}}" width="150px"
height="130px"/></td>
            <td>{{$showAbout->korisnicko_ime}}</td>
            <td>{{$showAbout->email}}</td>
            <td>{{(isset($showAbout->updated_at)) ? date('d.m.Y',
strtotime($showAbout->updated_at)) : 'nije menjan profil'}}</td>
            <td><li><a href="{{asset('/user/show/'.$showAbout-
>id_korisnik)}}">Menjaj</a></li></td>
    @endisset
      <form method="post" action="{{asset('/user/insert')}}" enctype="multipart/form-
data">
        {{csrf_field()}}
@endsection
```

## pages/userUpdate.blade.php

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

@section('title')
  User update
@endsection

@section('content')

    <h2>Change profile:</h2>
    @isset($showUser)
      <form method="post" action="{{asset('/user/update/'.$showUser->id_korisnik)}}"
enctype="multipart/form-data">
        {{csrf_field()}}
    @endisset
@endsection
```

## routes/web.php

```
Route::get('/', 'FrontEndController@index');
Route::get('/post/{id}/{cId?}', 'FrontEndController@getPost');
Route::get('/register', 'FrontEndController@getRegister');
Route::post('/register/reg', 'UserController@register');
Route::post('/register/login', 'UserController@login');
Route::get('/about', 'FrontEndController@about');
Route::get('/gallery', 'AdminController@getGallery');
Route::get('/ajax/show', 'FrontEndController@showAnketa');
Route::post('/glasovi/insert', 'AnketaController@storeVote');
Route::get('/logout', 'UserController@logout')->name('logout');
Route::get('/download','FrontEndController@download');


Route::group(['middleware' => 'user'], function()
{
   Route::get('/user', 'FrontEndController@user');
   Route::get('/user/show/{id}', 'FrontEndController@showUser');
   Route::post('/user/update/{id}', 'UserController@updateUser');
   Route::post('/user/insert', 'PostController@insertPost');
   Route::get('/user/delete/{id}', 'PostController@deletePost');
   Route::post('/post/{id}/comment', 'CommentController@insertComment');
   Route::get('/post/{id}/delete/{cId}', 'CommentController@deleteComment');
   Route::post('/post/{id}/{cId}', 'CommentController@updateComment');
});

Route::group(['middleware' => 'admin'], function()
{
   Route::get('/admin', 'AdminController@getUsers')->name('admin_users');
   Route::get('/admin/post', 'AdminController@getPosts')->name('admin_posts');
   Route::get('/admin/navigation', 'AdminController@getNavigations')->name('admin_nav');
   Route::get('/admin/role', 'AdminController@getRoles')->name('admin_role');
   Route::get('/admin/poll', 'AdminController@getPoll')->name('admin_poll');

   Route::get('/admin/show/insert/user', 'AdminController@showFormUser');
   Route::post('/admin/user/insert', 'UserController@insertUser');
   Route::get('/admin/show/insert/post', 'AdminController@showFormPost');
   Route::post('/admin/post/insert', 'PostController@insertAdminPost');
   Route::get('/admin/show/insert/navigation', 'AdminController@showFormNav');
   Route::post('/admin/nav/insert', 'NavigationController@insertNav');
   Route::get('/admin/show/insert/role', 'AdminController@showFormRole');
```

```
    Route::post('/admin/role/insert', 'RoleController@insertRole');
    Route::get('/admin/show/insert/poll', 'AdminController@showFormPoll');
    Route::post('/admin/poll/insert', 'AnketaController@insertPoll');

    Route::get('/admin/show/update/user/{id}', 'AdminController@showUserForm');
    Route::post('/admin/user/update/{id}', 'UserController@update');
    Route::get('/admin/show/update/post/{id}', 'AdminController@showPostForm');
    Route::post('/admin/post/update/{id}', 'PostController@updatePost');
    Route::get('/admin/show/update/nav/{id}', 'AdminController@showNavForm');
    Route::post('/admin/nav/update/{id}', 'NavigationController@updateNav');
    Route::get('/admin/show/update/role/{id}', 'AdminController@showRoleForm');
    Route::post('/admin/role/update/{id}', 'RoleController@updateRole');
    Route::get('/admin/show/update/poll/{id}', 'AdminController@showPollForm');
    Route::post('/admin/poll/update/{id}', 'AnketaController@update');

    Route::get('/admin/user/{id}', 'UserController@delete');
    Route::get('/admin/post/{id}', 'PostController@delete');
    Route::get('/admin/nav/{id}', 'NavigationController@delete');
    Route::get('/admin/role/{id}', 'RoleController@delete');
    Route::get('/admin/poll/{id}', 'AnketaController@delete');

    Route::get('/admin/poll/active/{id}', 'AnketaController@active');
});
```