Sadržaj:

1. Uvod

Web aplikacija „Flower Shop” rađena je u Laravelu a namenjena je za online kupovinu cveća. Dolaskom na početnu stranicu neregistrovani korisnik može videti sve proizvode koje su u ponudi, njihovu cenu kao i kratak opis i takođe može da ih doda u korpu.Klikom na dugme „Read more“ može pogledati nešto više o tom proizvodu kao što su detaljan opis proizvoda, cena kao i eventualni popust.Proizvodi se mogu pretražiti po ključnim rečima. Takođe, proizvodi se mogu filtrirati po raznim kategorijama.Klikom na ikonicu za korpu, korisnik odlazi na stranicu za korpu, gde ima uvid u proizvode koje želi kupiti i gde može da ukloni određeni proizvod,smanji ili poveča količinu proizvoda ili završi kupovinu.Osim naslovne stranice i galerije cveća, tu je i kontakt stranica gde nas korisnici mogu kontaktirati.Korisnici mogu i da ostave komentare ispod određenih proizvoda kad i da menjaju i brišu svoje komentare.Na order stranici imaju i pregled svojih naručenih proizvoda.Administrator web aplikacije ima pregled svih korisnika. Osnovne informacije kao i datum registracije. Administrator može ukloniti korisnike kao i njihove komentare.Takođe, može videti kategorije i proizvode. Može ih dodavati, izmeniti ili obrisati. Ima uvid u logove kao i primljene porudbine preko order stranice gde moze da ih briše ili da pošalje mail o poslatoj pošiljci

2. Radno okruženje

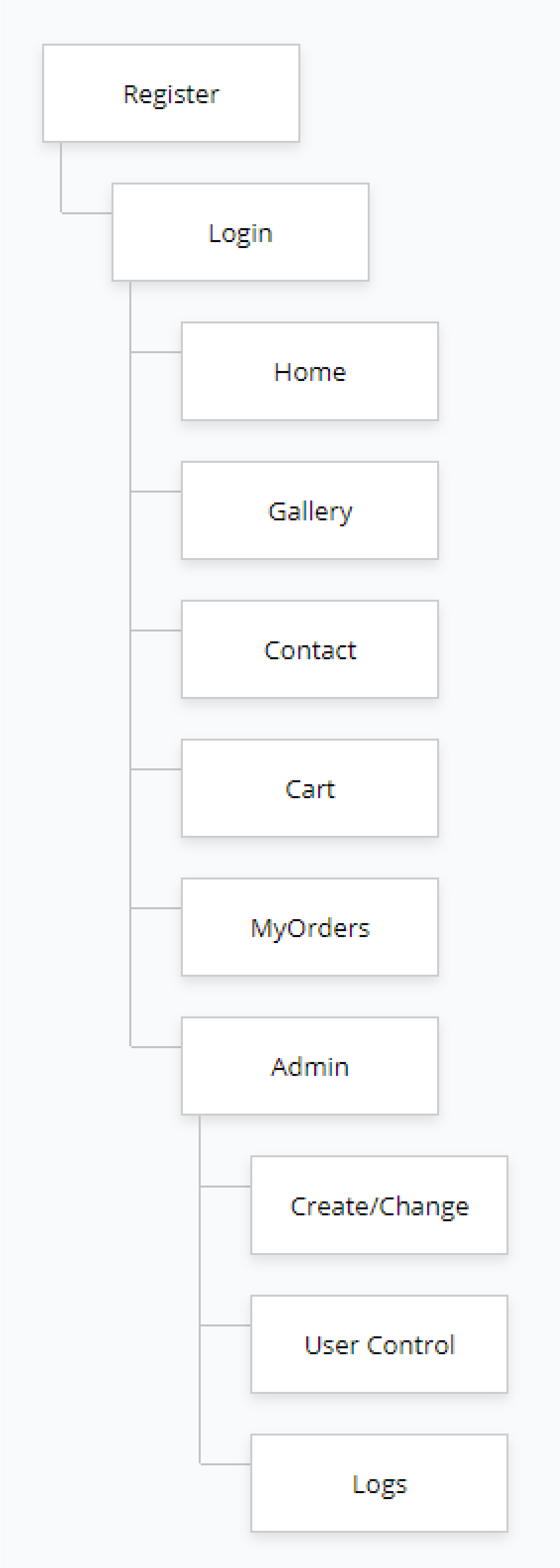
Razvojni alat:

- PhpStorm
- phpMyAdmin
- Xampp

Tehnologije:

- PHP – Laravel Framework
- JavaScript
- HTML
- CSS
- MySql
- JQuery

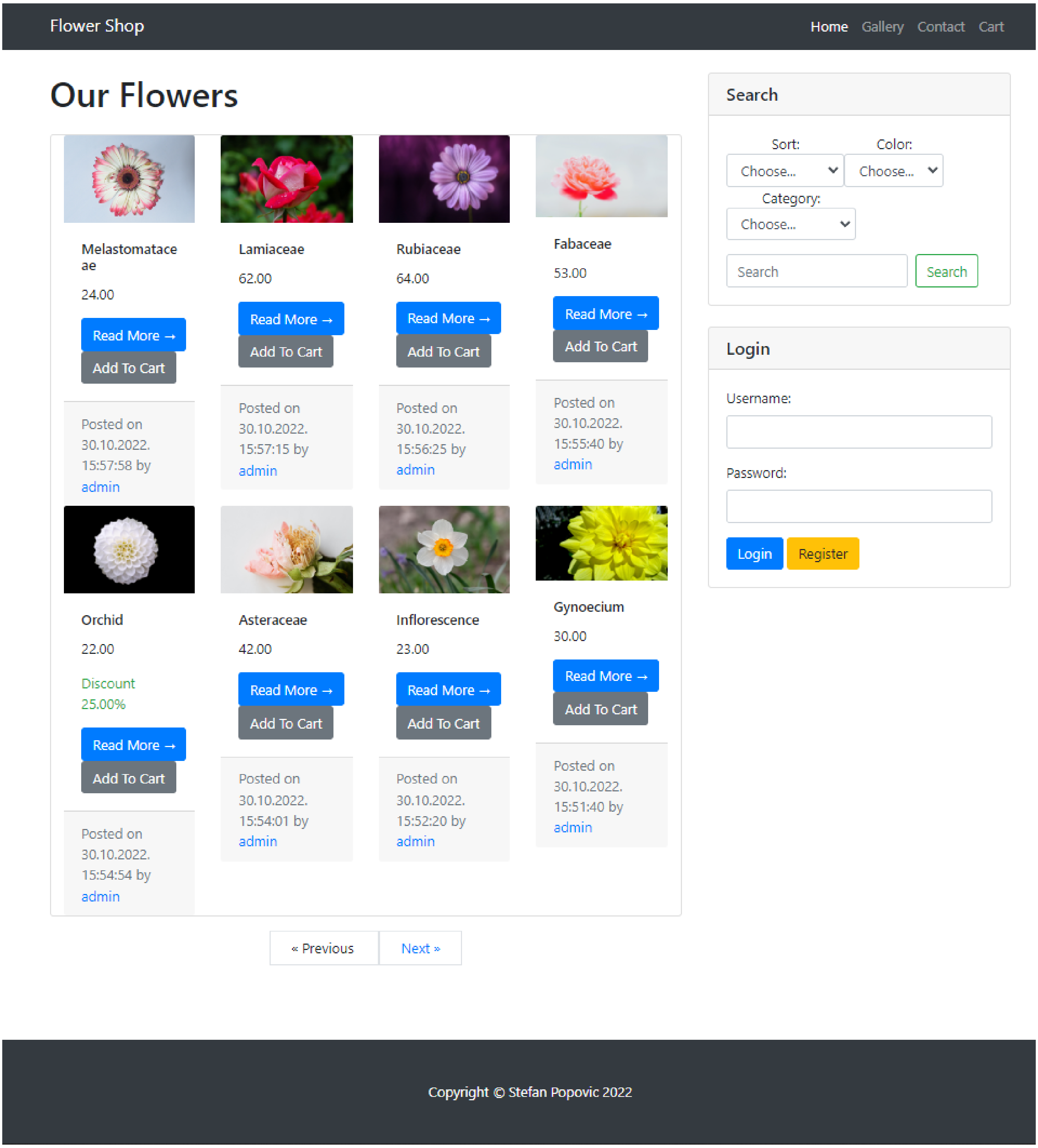
2. Organizacija



Slika 1 – blok šema

3.1 Stranice

3.1.1 Naslovna



Our Flowers

Melastomataceae
24.00
Posted on 30.10.2022. 15:57:58 by admin

Lamiaceae
62.00
Posted on 30.10.2022. 15:57:15 by admin

Rubiaceae
64.00
Posted on 30.10.2022. 15:56:25 by admin

Fabaceae
53.00
Posted on 30.10.2022. 15:55:40 by admin

Orchid
22.00
Discount 25.00%
Posted on 30.10.2022. 15:54:54 by admin

Asteraceae
42.00
Posted on 30.10.2022. 15:54:01 by admin

Inflorescence
23.00
Posted on 30.10.2022. 15:52:20 by admin

Gynoecium
30.00
Posted on 30.10.2022. 15:51:40 by admin

Search

Sort: Choose...
Color: Choose...
Category: Choose...

Search

Login

Username:

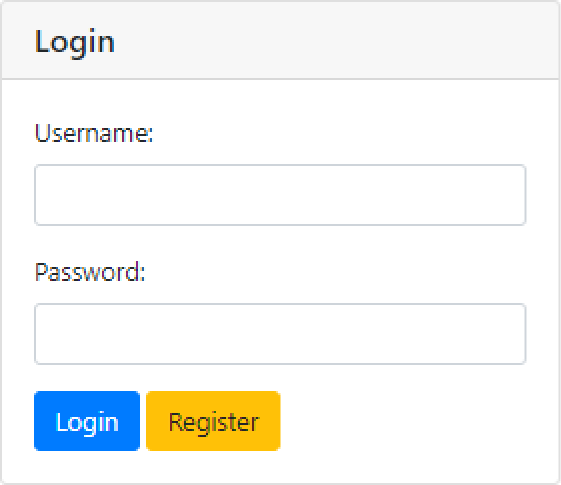
Password:

Copyright © Stefan Popovic 2022

Slika 2 – naslovna strana

Na naslovnoj strani (slika 2) su prikazani svi proizvodi sa cenama i popustom ukoliko ga ima. Proizvodi se mogu pretraživati po ključnim rečima upisom u polje search.
Proizvodi se mogu filtrirati klikom na neku od kategorija.

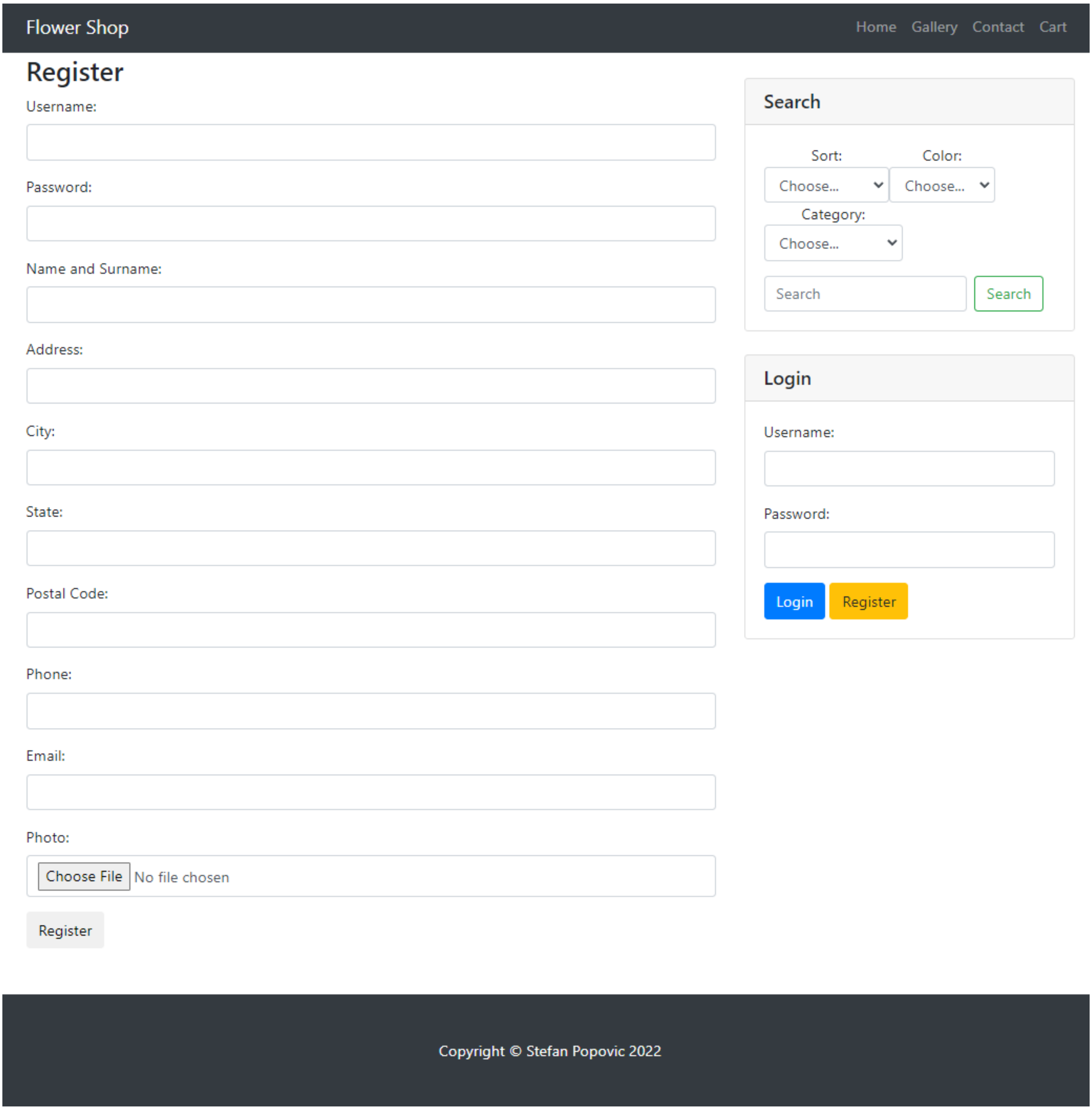
3.1.2 Prijava



Slika 3 – prijava

Prijava je u sklopu svake strane radi lakog logovanja(slika 3). Ovde se korisnik može prijaviti kao korisnik ili admin.

3.1.3 Registracija



Flower Shop

Home Gallery Contact Cart

Register

Username:

Password:

Name and Surname:

Address:

City:

State:

Postal Code:

Phone:

Email:

Photo:

Choose File No file chosen

Register

Search

Sort: Choose...

Color: Choose...

Category: Choose...

Search

Search

Login

Username:

Password:

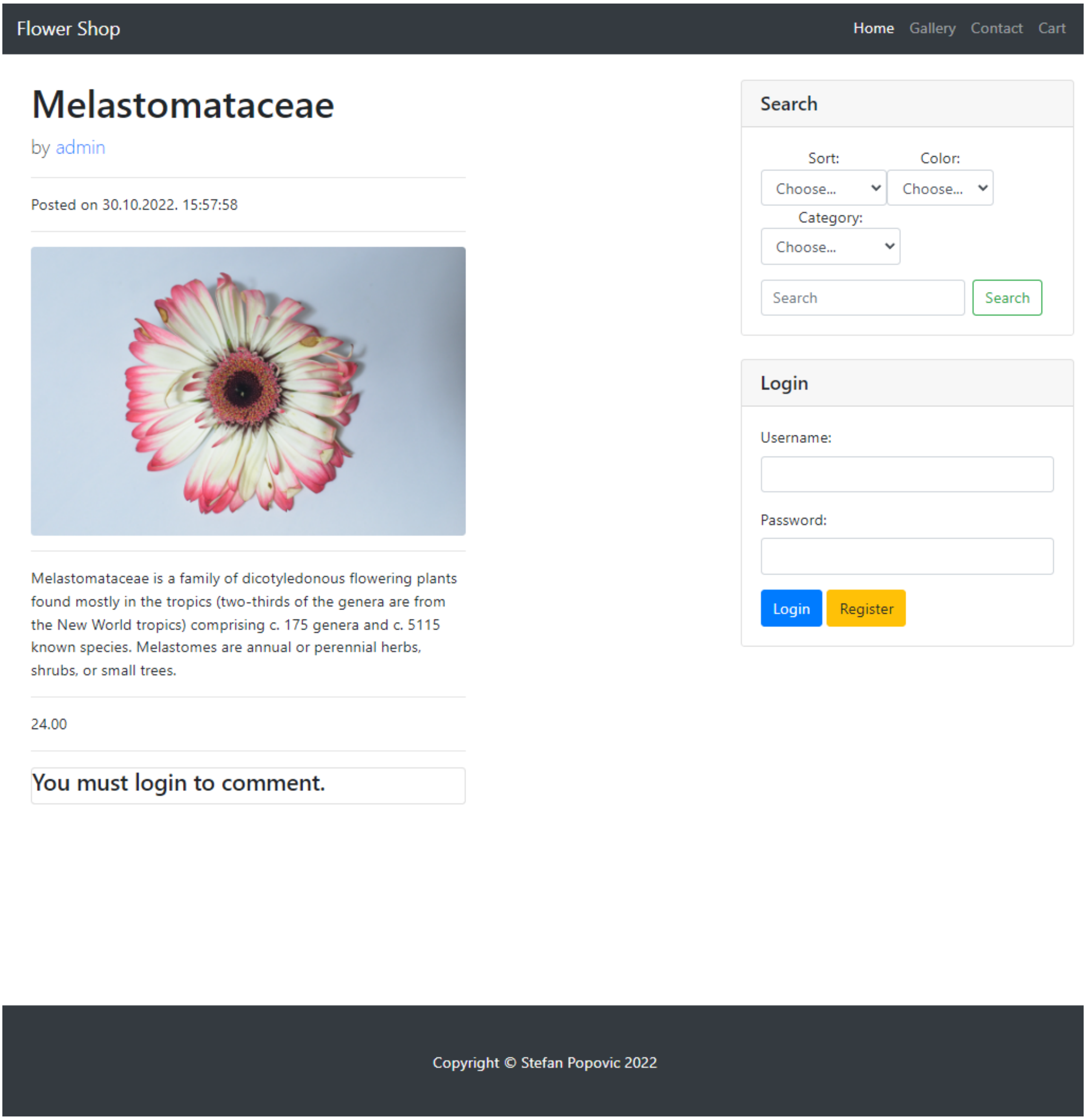
Login Register

Copyright © Stefan Popovic 2022

Slika 4 – stranica za registraciju

Stranica za registraciju korisnika (slika 4).

3.1.4 Stranica o pojedinačnom proizvodu



Melastomataceae

by admin

Posted on 30.10.2022. 15:57:58

Melastomataceae is a family of dicotyledonous flowering plants found mostly in the tropics (two-thirds of the genera are from the New World tropics) comprising c. 175 genera and c. 5115 known species. Melastomes are annual or perennial herbs, shrubs, or small trees.

24.00

You must login to comment.

Search

Sort: Choose...

Color: Choose...

Category: Choose...

Search

Login

Username:

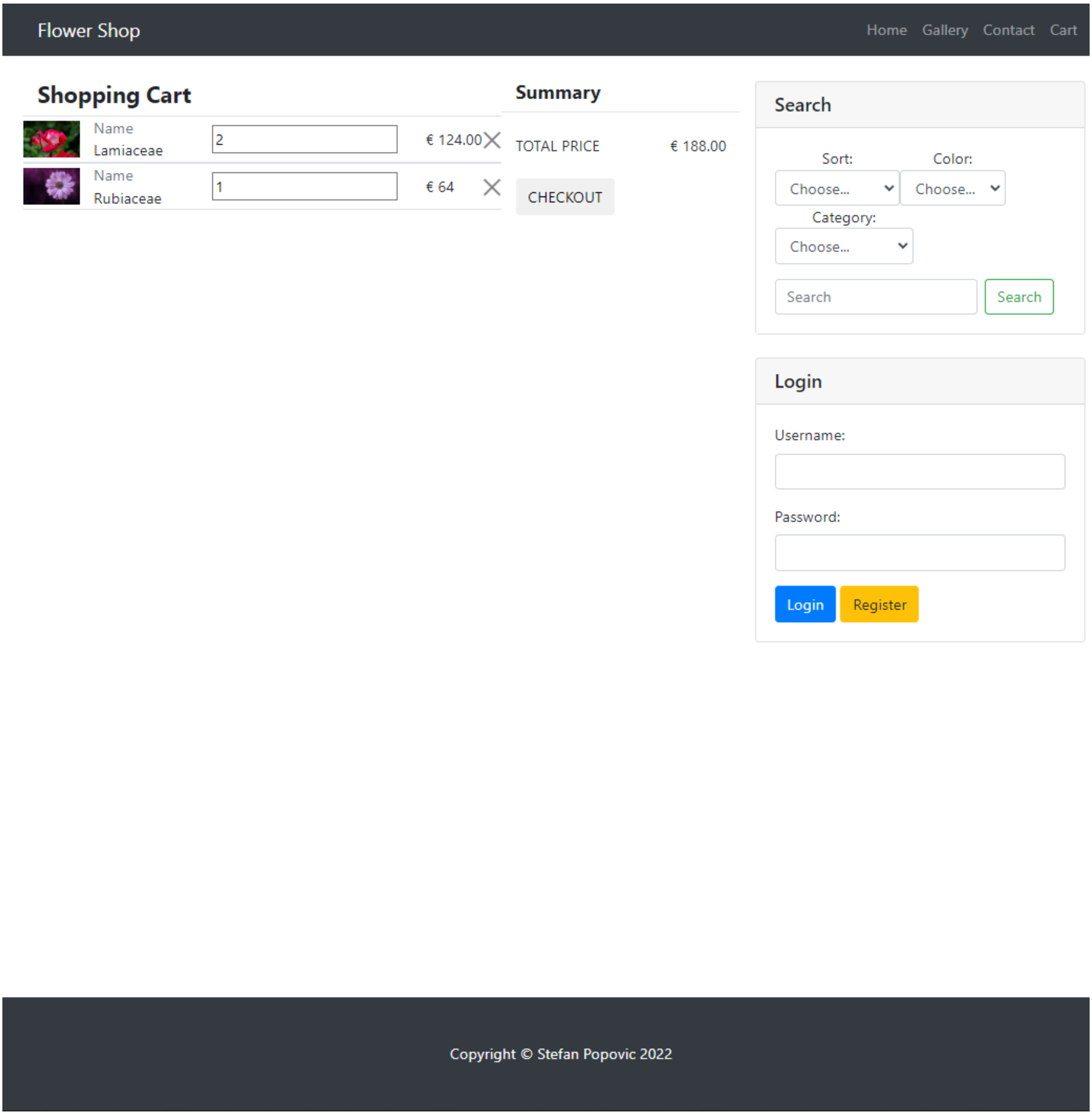
Password:

Copyright © Stefan Popovic 2022

Slika 5 – stranica o pojedinačnom proizvodu

Ovde se mogu videti detalji o proizvodu (slika 5).

3.1.5 Korpa



Flower Shop

Shopping Cart

	Name Lamiaceae	2	€ 124.00	✕
	Name Rubiaceae	1	€ 64	✕

Summary

TOTAL PRICE € 188.00

Search

Sort: Choose...

Color: Choose...

Category: Choose...

Login

Username:

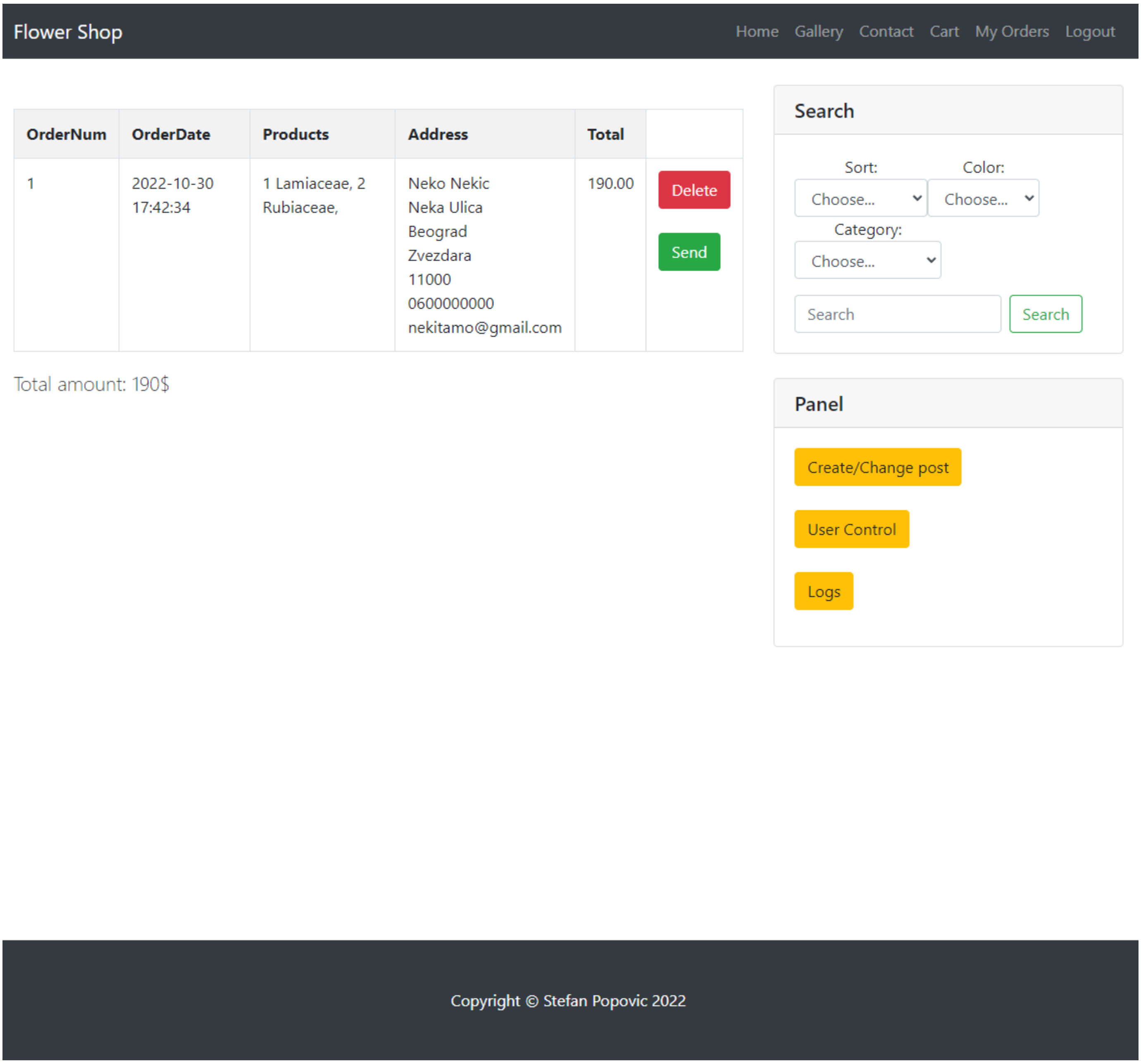
Password:

Copyright © Stefan Popovic 2022

Slika 6 – korpa

Na stranici za korpu (slika 6) možemo videti proizvode koje smo ubacili u korpu a i možemo menjati količinu. Proizvode možemo ukloniti iz korpe kao i kupiti klikom na „Checkout“.

3.1.6 Porudbine



OrderNum	OrderDate	Products	Address	Total	
1	2022-10-30 17:42:34	1 Lamiaceae, 2 Rubiaceae,	Neko Nekic Neka Ulica Beograd Zvezdara 11000 0600000000 nekitamo@gmail.com	190.00	Delete Send

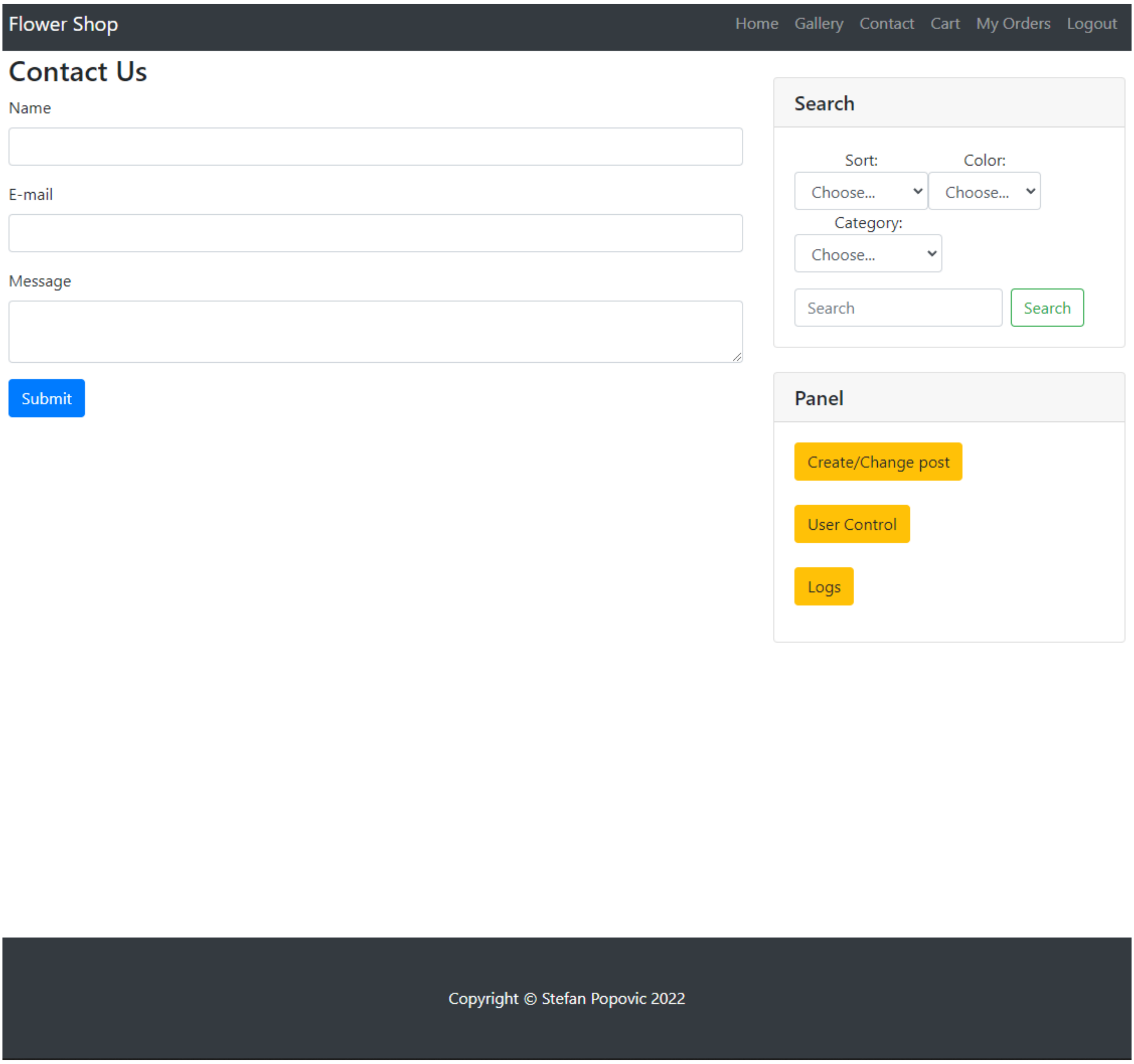
Total amount: 190$

Copyright © Stefan Popovic 2022

Slika 7 – stranica za porudbine

Informacije o porudbinama. Korisnici imaju pregled svojim dok admin ima pregled u sve i takođe može da ih briše i šsalje mail za poslate pošiljke (slika 7).

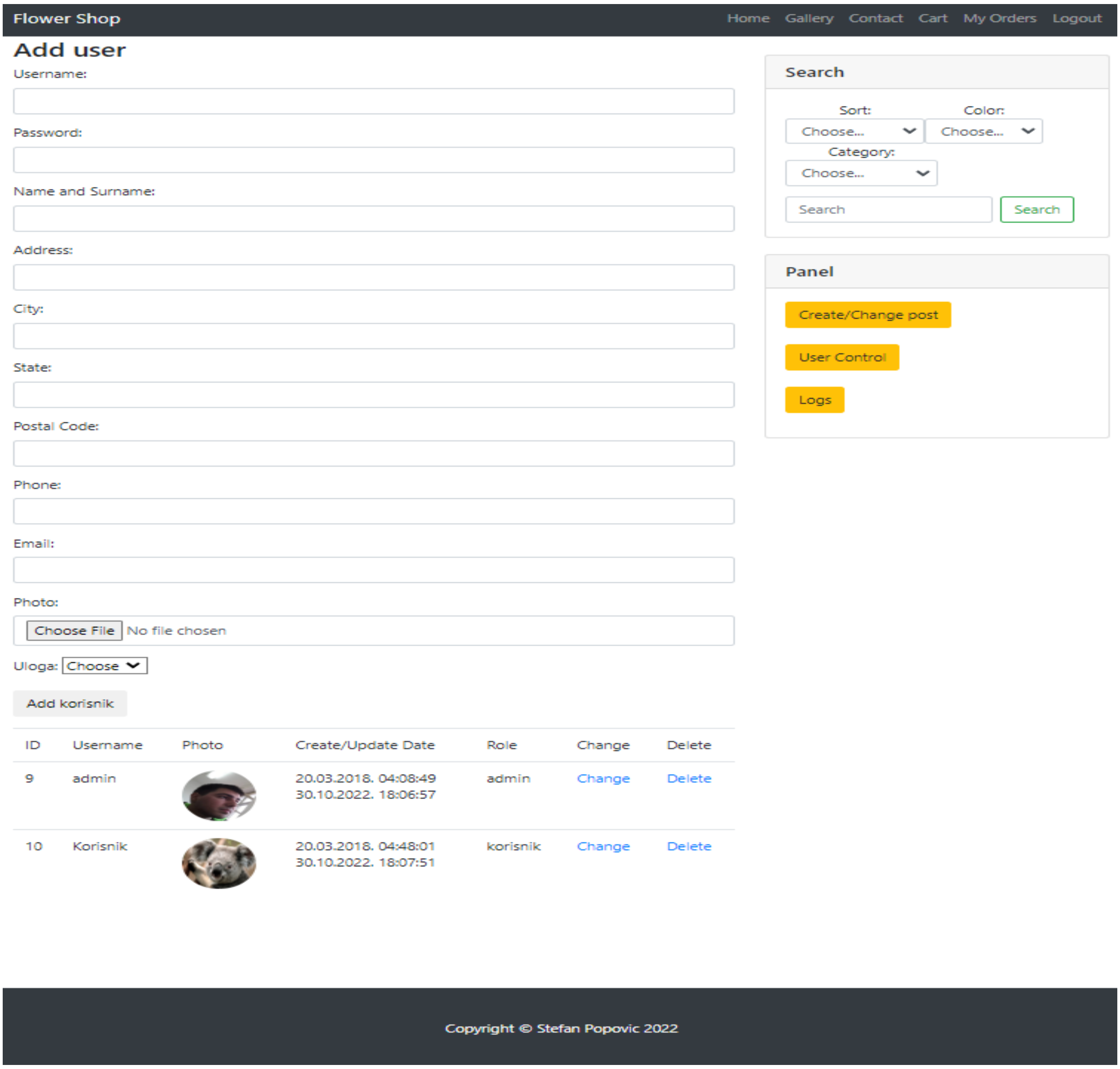
3.1.7 Kontakt



Slika 8 – stranica za kontakt

Kontakt stranica (slika 8).

3.1.8 Dodaj Korisnika



Add user

Username:

Password:

Name and Surname:

Address:

City:

State:

Postal Code:

Phone:

Email:

Photo:

Choose File No file chosen

Uloga: Choose

Add korisnik

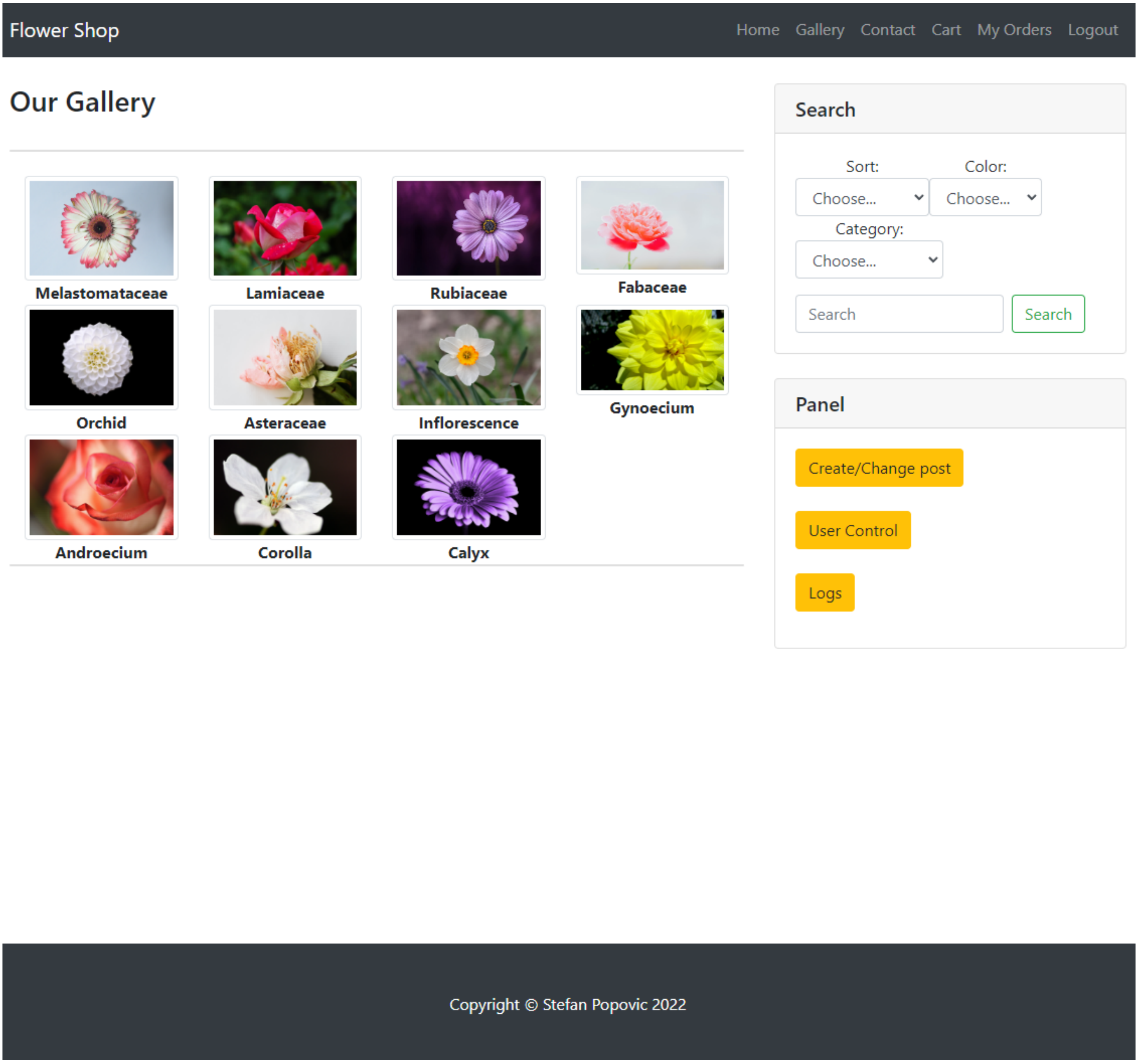
ID	Username	Photo	Create/Update Date	Role	Change	Delete
9	admin		20.03.2018. 04:08:49 30.10.2022. 18:06:57	admin	Change	Delete
10	Korisnik		20.03.2018. 04:48:01 30.10.2022. 18:07:51	korisnik	Change	Delete

Copyright © Stefan Popovic 2022

Slika 9 – admin stranica za korisnika

Admin stranica o korisnicima gde admin može sam dodavati nove korisnike kao i da menja i briše postojeće(slika 9).

3.1.9 Galerija

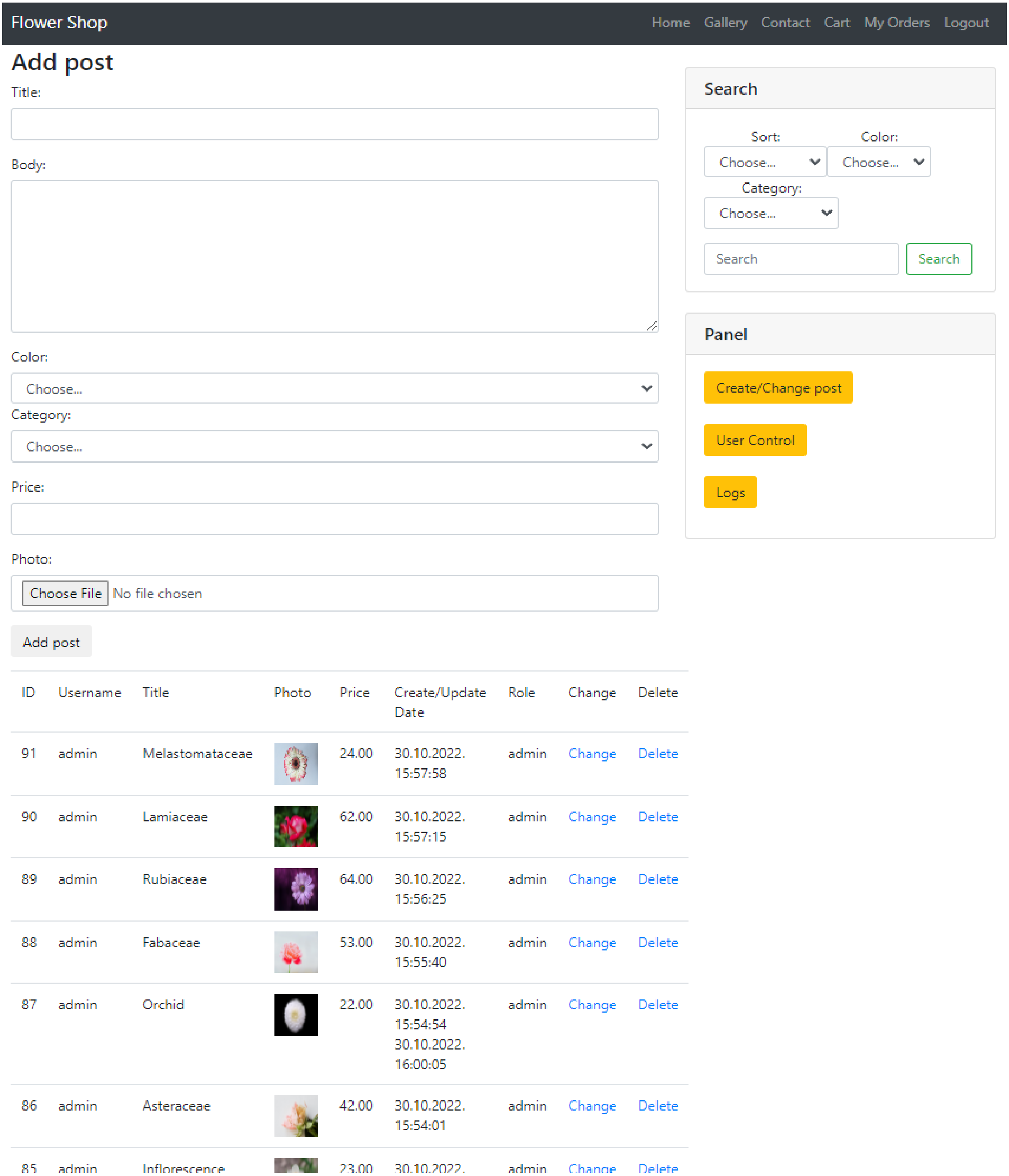


Copyright © Stefan Popovic 2022

Slika 10 – Galerija

Galerija slika (slika 10).

3.1.10 Postovanje

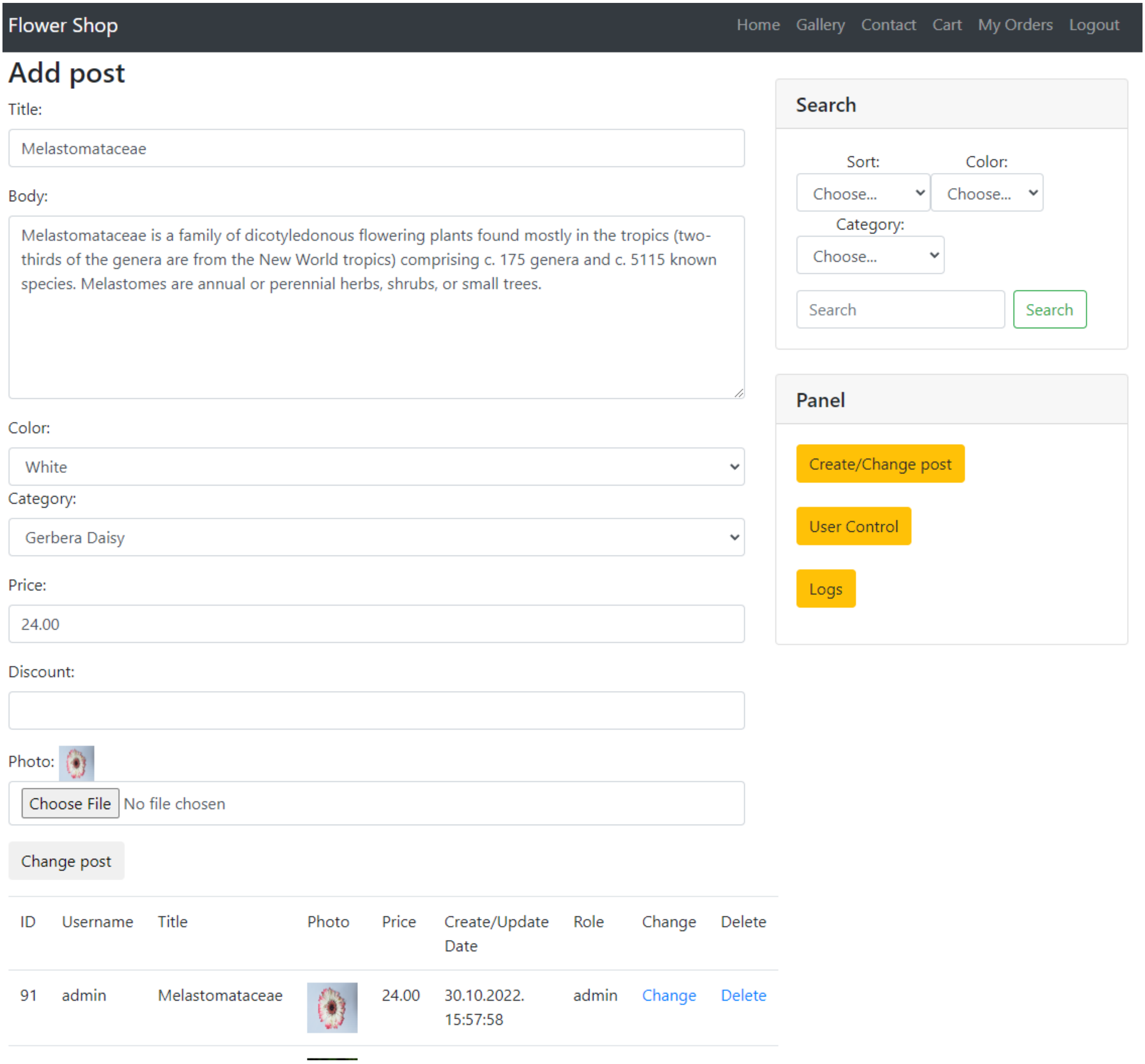


ID	Username	Title	Photo	Price	Create/Update Date	Role	Change	Delete
91	admin	Melastomataceae		24.00	30.10.2022. 15:57:58	admin	Change	Delete
90	admin	Lamiaceae		62.00	30.10.2022. 15:57:15	admin	Change	Delete
89	admin	Rubiaceae		64.00	30.10.2022. 15:56:25	admin	Change	Delete
88	admin	Fabaceae		53.00	30.10.2022. 15:55:40	admin	Change	Delete
87	admin	Orchid		22.00	30.10.2022. 15:54:54 30.10.2022. 16:00:05	admin	Change	Delete
86	admin	Asteraceae		42.00	30.10.2022. 15:54:01	admin	Change	Delete
85	admin	Inflorescence		23.00	30.10.2022.	admin	Change	Delete

Slika 11 – admin stranica za dodavanje postova

Na ovoj stranici (slika 11), admin može dodavati proizvode, izmeniti ako nešto nije u redu kao i obrisati ako više nisu u ponudi. Klikom na dugme „Change“ pored samog proizvoda, odlazi na stranicu gde može izmeniti sve podatke za određeni proizvod.

3.1.11 Izmena proizvoda



Add post

Title: Melastomataceae

Body: Melastomataceae is a family of dicotyledonous flowering plants found mostly in the tropics (two-thirds of the genera are from the New World tropics) comprising c. 175 genera and c. 5115 known species. Melastomes are annual or perennial herbs, shrubs, or small trees.

Color: White

Category: Gerbera Daisy

Price: 24.00

Discount:

Photo:

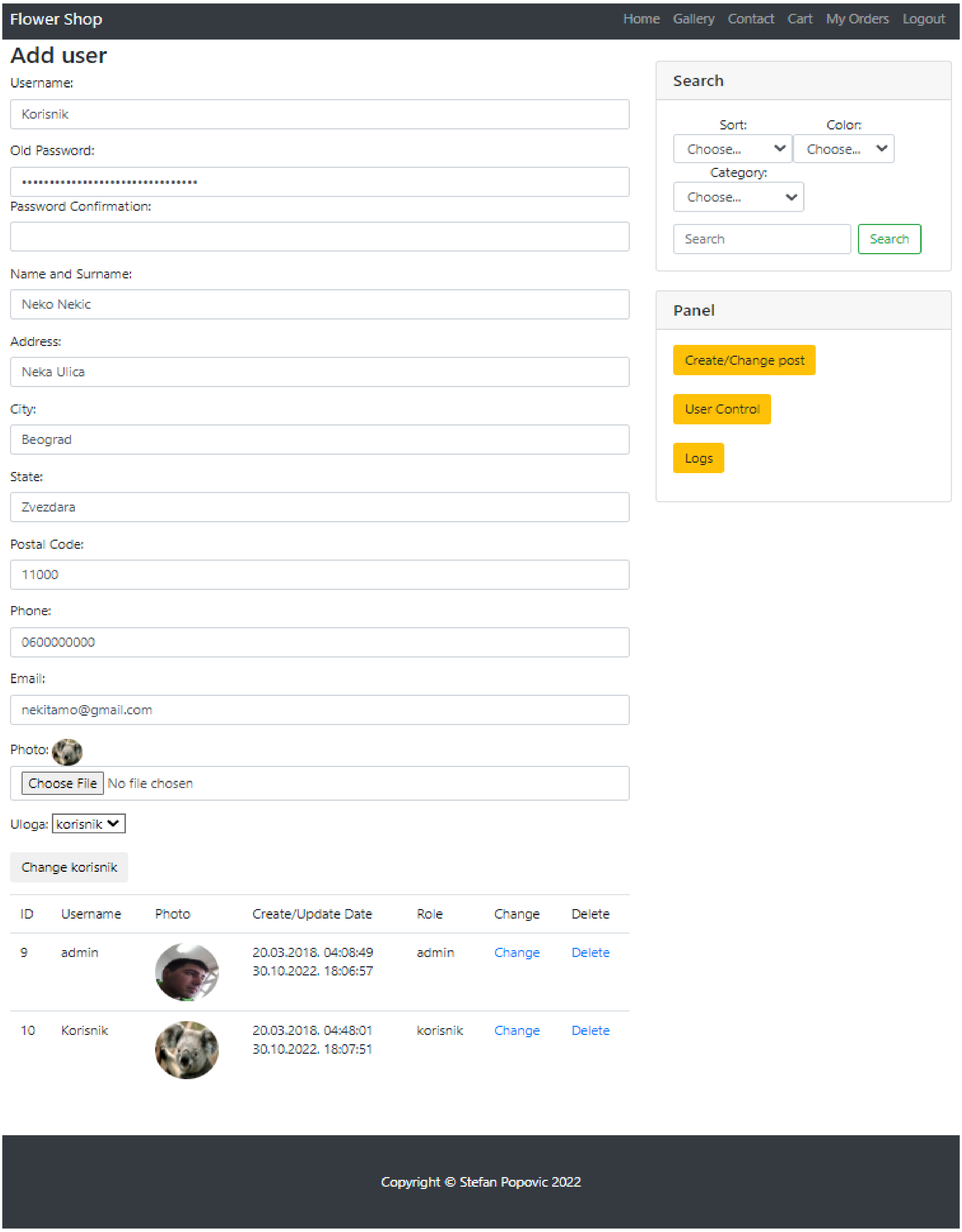
Choose File No file chosen

ID	Username	Title	Photo	Price	Create/Update Date	Role	Change	Delete
91	admin	Melastomataceae		24.00	30.10.2022. 15:57:58	admin	Change	Delete

Slika 12 – admin stranica za izmenu proizvoda

Ovde se mogu izmeniti informacije vezane za određeni proizvod (slika 12). Takođe se može izmeniti slika a i može se dodati ili skinuti popust.

3.1.12 Izmena korisnika



Add user

Username: Korisnik

Old Password: ••••••••••••••••••••••••••••••••

Password Confirmation:

Name and Surname: Neko Nekic

Address: Neka Ulica

City: Beograd

State: Zvezdara

Postal Code: 11000

Phone: 0600000000

Email: nekitamo@gmail.com

Photo:

Choose File No file chosen

Uloga: korisnik

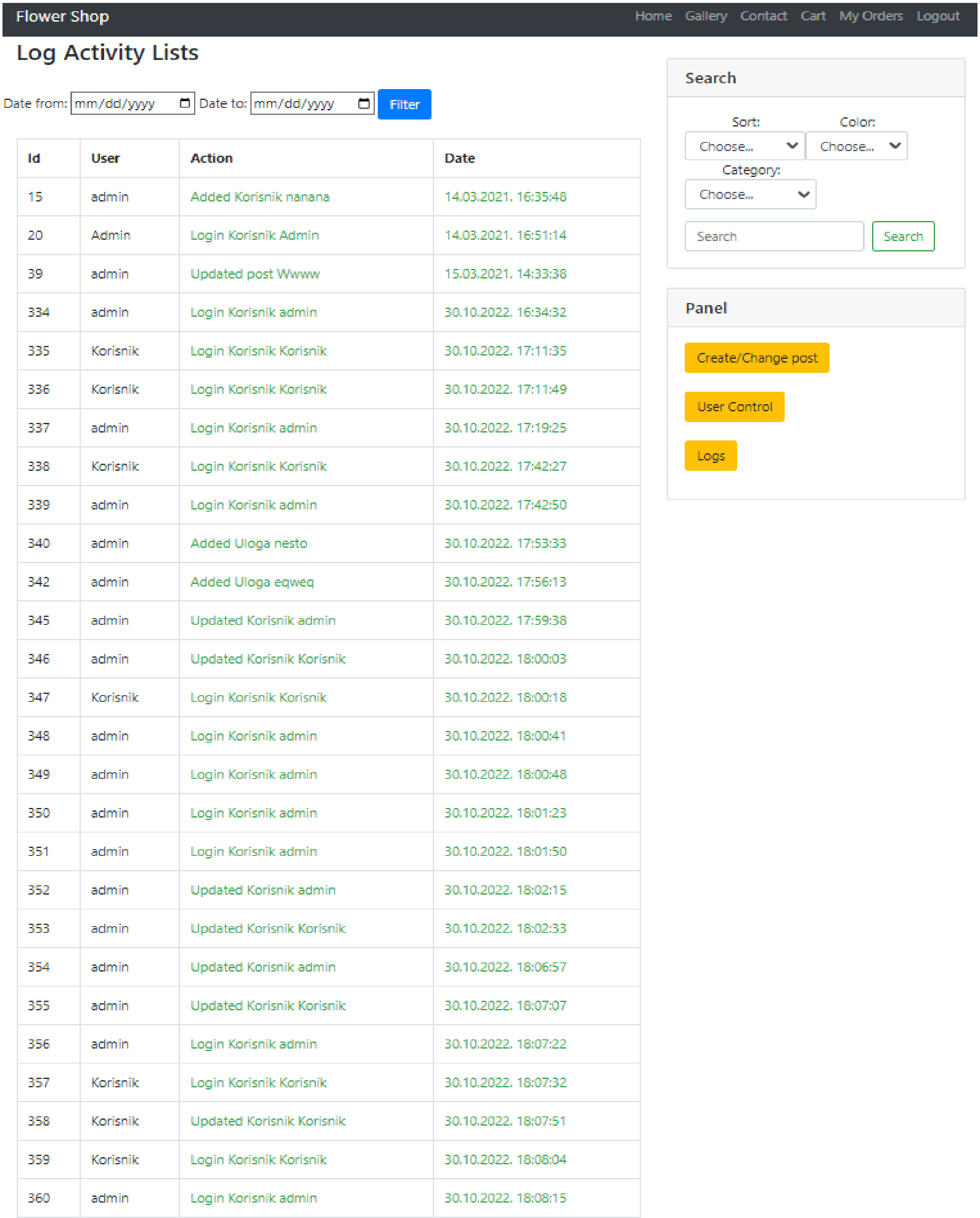
ID	Username	Photo	Create/Update Date	Role	Change	Delete
9	admin		20.03.2018. 04:08:49 30.10.2022. 18:06:57	admin	Change	Delete
10	Korisnik		20.03.2018. 04:48:01 30.10.2022. 18:07:51	korisnik	Change	Delete

Copyright © Stefan Popovic 2022

Slika 13 – admin stranica za izmenu korisnika

Menjanje korisnika i ukoliko ste admin možete postaviti korisnike za admine (slika 13).

3.1.13 Logovi



Log Activity Lists

Date from: mm/dd/yyyy Date to: mm/dd/yyyy

Id	User	Action	Date
15	admin	Added Korisnik nanana	14.03.2021. 16:35:48
20	Admin	Login Korisnik Admin	14.03.2021. 16:51:14
39	admin	Updated post Wwww	15.03.2021. 14:33:38
334	admin	Login Korisnik admin	30.10.2022. 16:34:32
335	Korisnik	Login Korisnik Korisnik	30.10.2022. 17:11:35
336	Korisnik	Login Korisnik Korisnik	30.10.2022. 17:11:49
337	admin	Login Korisnik admin	30.10.2022. 17:19:25
338	Korisnik	Login Korisnik Korisnik	30.10.2022. 17:42:27
339	admin	Login Korisnik admin	30.10.2022. 17:42:50
340	admin	Added Uloga nesto	30.10.2022. 17:53:33
342	admin	Added Uloga eqweq	30.10.2022. 17:56:13
345	admin	Updated Korisnik admin	30.10.2022. 17:59:38
346	admin	Updated Korisnik Korisnik	30.10.2022. 18:00:03
347	Korisnik	Login Korisnik Korisnik	30.10.2022. 18:00:18
348	admin	Login Korisnik admin	30.10.2022. 18:00:41
349	admin	Login Korisnik admin	30.10.2022. 18:00:48
350	admin	Login Korisnik admin	30.10.2022. 18:01:23
351	admin	Login Korisnik admin	30.10.2022. 18:01:50
352	admin	Updated Korisnik admin	30.10.2022. 18:02:15
353	admin	Updated Korisnik Korisnik	30.10.2022. 18:02:33
354	admin	Updated Korisnik admin	30.10.2022. 18:06:57
355	admin	Updated Korisnik Korisnik	30.10.2022. 18:07:07
356	admin	Login Korisnik admin	30.10.2022. 18:07:22
357	Korisnik	Login Korisnik Korisnik	30.10.2022. 18:07:32
358	Korisnik	Updated Korisnik Korisnik	30.10.2022. 18:07:51
359	Korisnik	Login Korisnik Korisnik	30.10.2022. 18:08:04
360	admin	Login Korisnik admin	30.10.2022. 18:08:15

i

Slika 14 – admin stranica za logove

Na stranici za logove (slika 14), admin može videti sve logove kao i filtrirati ih radi lakše pretrage.

3. Kodovi

4.1 Controllers

4.1.1 FrontendController

```
<?php

namespace App\Http\Controllers;
use App\Models\Category;
use App\Models\Color;
use App\Models\CommentsModel;
use App\Models\Meni;
use App\Models\Post;
use App\Models\Logs;
use App\Models\Korisnik;
use App\Models\PostEloquentModel;
use Illuminate\Http\Request;
use DB;

class FrontendController extends Controller {

   private $data = [];

   public function __construct(){
      $meni = new Meni();
      $this->data['menus'] = $meni->getAll();
       $colors = Color::all();
       $categories=Category::all();
       $this->data['colors'] = $colors;
       $this->data['categories'] = $categories;
   }
   public function index(){
      $post = new Post();
       $this->data['posts'] =$post->getAll();

       $users = DB::table('post')->select('*','post.id as postId','post.created_at
as create','post.updated_at as update')->join('slika as
s','post.slika_id','=','s.id')->join('korisnik as
k','post.korisnik_id','=','k.id')->join('uloga as
u','k.uloga_id','=','u.id')->orderBy('post.created_at','desc');
       $this->data['users'] =$users->simplePaginate(8);

       return view('pages.home', $this->data);
   }
    public function galerija(){
       $post = new Post();
       $this->data['posts'] =$post->getAll();
       return view('pages.galerija', $this->data);
   }
   public function showlogs(){
       $logs = new Logs();
      $this->data['logovi'] = $logs->getlogs();
       return view('pages.logovi' , $this->data);
   }
   public function trazi(Request $request) {

       $keyword = $request->keyword;
       $boja=$request->boje;
       $sort=$request->sortBy;
       $kat=$request->kategorije;
           $post = new Post();
          $this->data['posts'] =$post->getAll();

           $users = DB::table('post')->select('*','post.id as
postId','post.created_at as create','post.updated_at as update')->join('slika as
s','post.slika_id','=','s.id')->join('korisnik as
k','post.korisnik_id','=','k.id')->join('uloga as
u','k.uloga_id','=','u.id')->join('color as
c','post.ColorId','=','c.ColorId')->join('category as
g','post.CategoryId','=','g.CategoryId');

           if($sort!=0){
       if (!empty($sort) && $sort == 'Price-desc') {
           $users=$users->orderBy('price','DESC');
       }
       if (!empty($sort) && $sort == 'Price-asc') {
           $users=$users->orderBy('price','ASC');
       }
       if (!empty($sort) && $sort == 'Naslov-asc') {
           $users=$users->orderBy('naslov','ASC');
       }
       if (!empty($sort) && $sort == 'Naslov-desc') {
           $users=$users->orderBy('naslov','DESC');
       }}


       if($boja==0){
           if ($kat==0) {
               $this->data['users'] = $users->where('post.naslov', 'like', '%' .
$keyword . '%')->simplePaginate(8);
           }else
           $this->data['users']
=$users->where('post.naslov','like','%'.$keyword.'%')->where('post.CategoryId','='
,$kat)->simplePaginate(8);
       }elseif ($kat==0){
           $this->data['users']
=$users->where('post.naslov','like','%'.$keyword.'%')->where('post.ColorId','=',$b
oja)->simplePaginate(8);
       }else
     $this->data['users']
=$users->where('post.naslov','like','%'.$keyword.'%')->where('post.CategoryId','='
,$kat)->where('post.ColorId','=',$boja)->simplePaginate(8);



       $users=$this->data['users'];

       return  view("pages.search",
           [
               "users" => $users,
           ],$this->data);
   }
   public function getPostjs($id){
       $output='';
       $post = new Post();
       $this->data['singlePost'] = $post->get($id);

       $commentModel = new CommentsModel();
       $this->data['comments'] = $commentModel->getPostComments($id);
       if($this->data['singlePost']){

           $output .= '<div class="col-md-8">' .
               '<h1 class="mt-4">'.$this->data['singlePost']->naslov.'</h1>' .
               '<p class="lead">' .
               'by&nbsp' .
               '<a href="#">'.$this->data['singlePost']->postKorisnik.'</a>' .
               '</p><hr>' .
               '<p>Posted on '.date("d.m.Y. H:i:s",
$this->data['singlePost']->create).'</p><hr>' .
               '<img class="img-fluid rounded"
src="'.asset("/".$this->data['singlePost']->putanja).'" ><hr>' .
               '<p> '.$this->data['singlePost']->sadrzaj.'</p><hr>' .
               '<p> '.$this->data['singlePost']->price.'</p><hr>';

           if($this->data['singlePost']->popust!=Null &&
$this->data['singlePost']->popust!=0){
               $output .='<p class="card-text">'.
                   '<p class="text-success">Discount
'.$this->data['singlePost']->popust.'%<p>'.
                   '</p>';
           }
           $output .='<div class="card mb-4">';
           if(session()->get('user')){
               $output .= '<div class="card my-4">' .
                   '<h5 class="card-header">Leave a Comment:</h5>' .
                   '<div class="card-body">' .
                   '<form action="#" method="get">' .
                   csrf_field() .
                   '<div class="form-group">'.
                   '<textarea class="form-control" id="psalji" rows="3"
name="content"></textarea></div>' .
                   '<button name="akcija" value="post"
onclick="salji('.$this->data['singlePost']->postId.')" class="btn btn-primary
salji">Submit</button>' .
                   '</form><br></div></div>'.
                   '<div hidden>' .
                   '<form id="pform" action="'.route("editComment", ["postId" =>
$this->data['singlePost']->postId],["commentId" =>"2"]).'" method="post">' .
                   csrf_field() .
                   '<div class="form-group">' .
                   '<textarea class="form-control" id="txta" rows="3"
name="content1"></textarea></div>' .
                   '<button type="submit" name="akcija" value="edit" class="btn
btn-primary">Update</button></form></div>';
               if($this->data['comments']){
                   $output .= '<div class="comments
heading"><h3>Comments</h3><form id="komentform">';
                   foreach ($this->data['comments'] as $comment){
                       $output .= '<div class="media"><div
class="media-body"><table class="table"><tr><td>' .
                           '<h4
class="media-heading">'.$comment->korisnicko_ime.'</h4>' .
                           '<h6><label>'.date("F j, Y",
strtotime($comment->created_at)).'</label></h6>' .
                           '<span id="comment'.$comment->id.'">' .
                           '<span
id="kom'.$comment->id.'">'.$comment->content.'</span></br>';
                       if(session()->get('user')[0]->id == $comment->user_id ||
session()->get('user')[0]->naziv == 'admin'){
                           $output .= '<a href="'.route("deleteComment",
["commentId" => $comment->id]).'"><i class="fa fa-trash">Obrisi</i></a>&nbsp';
                       }
                       if(session()->get('user')[0]->id == $comment->user_id){
                           $output .= '<a href="#comment'.$comment->id.'"><i
class="fa fa-edit"
onclick="editComment('.$comment->id.','.$this->data['singlePost']->postId.')">Izme
ni</i></a>';
                       }
                       $output .= '<span
id="#comments"></span></span></td></tr></table></div><div
class="media-right"></div></div>';
                   }
                   $output .= '</form></div>';
               }
           }else{
               $output .= '<h4>You must login to comment.</h4>';
           }$output .= '</div>';

       }

       return response()->json($output);
   }
   public function getPost($id){
       $post = new Post();
       $this->data['singlePost'] = $post->get($id);

       $commentModel = new CommentsModel();
       $this->data['comments'] = $commentModel->getPostComments($id);

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

}
```

4.1.2 LoginController

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Korisnik;

class LoginController extends Controller {

  public function login(Request $request){
     $request->validate([
        'tbKorisnickoIme' => ['required','alpha'],
     ], [
        'required' => ' :attribute is required!'
     ]);

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

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

     $loginKorisnik = $korisnik->getByUsernameAndPassword();

     if(!empty($loginKorisnik)){
        $request->session()->push('user', $loginKorisnik);
        return redirect()->back()->with('success','Successful Login!');
     }
     return redirect()->back()->with('error','You are not registered!');
  }

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

4.1.3 PostController

```
<?php

namespace App\Http\Controllers;
use App\Models\Category;
use App\Models\Color;
use App\Models\Invoice;
use App\Models\InvoiceLine;
use Validator;
use App\Models\CommentsModel;
use App\Models\Meni;
use App\Models\Post;
use App\Models\Slika;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;



class PostController extends Controller {

  private $data = [];

   public function __construct(){
      $meni = new Meni();
       $colors = Color::all();
       $categories=Category::all();
       $this->data['colors'] = $colors;
       $this->data['categories'] = $categories;
      $this->data['menus'] = $meni->getAll();
   }

  public function create(){

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

  public function store(Request $request) {


     $rules = [
        'title' => 'regex:/^[A-Z][a-z]+(\s[\w\d\-]+)*$/',
        'body' => 'required',
           'pboje' => 'required',
           'pkategorije' => 'required',
        'photo' => 'required|mimes:jpg,jpeg,png,gif|max:3000'
     ];
     $custom_messages = [
        'required' => ' :attribute is required!',
        'title.regex' => 'Title must start with a capital letter!',
        'max' => 'File cant be bigger than :max KB.',
        'mimes' => 'Allowed formats are: :values.'
     ];
     $request->validate($rules, $custom_messages);

     $photo = $request->file('photo');
     $extension = $photo->getClientOriginalExtension();
     $tmp_path = $photo->getPathName();

     $folder = 'images/';
     $file_name = time().".".$extension;
     $new_path = ($folder).$file_name;

     try {


        File::move($tmp_path, $new_path);



        $slika = new Slika();
        $slika->putanja = 'images/'.$file_name;
        $slika_id = $slika->save();


        $post = new Post();
        $post->naslov = $request->get('title');
        $post->sadrzaj = $request->get('body');
           $post->price = $request->get('cena');
           $post->ColorId = $request->get('pboje');
           $post->CategoryId = $request->get('pkategorije');
           $post->korisnik_id = session()->get('user')[0]->id;
        $post->slika_id = $slika_id;
        $post->save();

        return redirect('/')->with('success','Successfully Added!');
     }
     catch(\Illuminate\Database\QueryException $ex){
        \Log::error($ex->getMessage());
        return redirect()->back()->with('error','Database error!');
     }
     catch(\Symfony\Component\HttpFoundation\File\Exception\FileException $ex) {
        \Log::error('Problem sa fajlom!! '.$ex->getMessage());
        return redirect()->back()->with('error','Photo add error!');
     }
     catch(\ErrorException $ex) {
        \Log::error('Problem sa fajlom!! '.$ex->getMessage());
        return redirect()->back()->with('error','Error..');
     }
  }
       public function postshow($id = null){
     $post = new Post();
     $this->data['postovi'] = $post->getAll();

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

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

       public function update(Request $request,$id) {
               $rules = [
        'title' => 'regex:/^[A-Z][a-z]+(\s[\w\d\-]+)*$/',
        'body' => 'required',
                   'pboje' => 'required',
                   'pkategorije' => 'required',
        'photo' => 'mimes:jpg,jpeg,png,gif|max:3000',
     ];
     $custom_messages = [
        'required' => ' :attribute is required!',
        'title.regex' => 'Title is not in the correct format!',
        'max' => 'File cant be bigger than :max KB.',
        'mimes' => 'Allowed formats are: :values.'
     ];
     $request->validate($rules, $custom_messages);
```

```
           $oldPictureId = null;

        $photo = $request->file('photo');
           $post = new post();
           $post->id = $id;
           $post->naslov = $request->get('title');
           $post->sadrzaj = $request->get('body');
           $post->price = $request->get('cena');
           $post->ColorId = $request->get('pboje');
           $post->CategoryId = $request->get('pkategorije');
           $post->popust = $request->get('popust');
        $post->slika_id=$post->get($id)->slika_id;
        $oldPictureId = $post->getPictureId($id);
           $puti=$request->get('puti');
        if (!empty($photo)){

                    try{

                       $rezz=$post->deleted();
                       $folder = ('images/');
                       $extension = $photo->getClientOriginalExtension();
                       $file_name = time().".".$extension;
                       $photo->move($folder, $file_name);

                       $slika = new Slika();
                       $slika->putanja = 'images/' . $file_name;
                       $slika_id = $slika->save();
                       $post->slika_id = $slika_id;


                    }catch (QueryException $e) {
              \Log::error("Update Error: " . $e->getMessage());
           } catch (FileException $e) {
              \Log::error("Photo Update Error: " . $e->getMessage());
           }
           }
               $rez=$post->update();

           if($rez == 1){
           return redirect('/posts')->with("success", "Post successfully
edited!");
        }else {
           return redirect('/posts')->with('error','Update Error!');
        }


           }




       public function destroy($id){
```

```
       $nodel=InvoiceLine::where('PostId','=',$id)->first();
       if(!empty($nodel)){
           return redirect('/posts')->with('error','Product is in a order!');
       }
     $post = new Post();
     $post->id = $id;

               $post->slika_id=$post->get($id)->slika_id;

               $rezz=$post->deleted();


     $rez = $post->delete();
     if($rez == 1 && $rezz==1){
        return redirect('/posts')->with('success','Successful Delete!');
     }
     else {
        return redirect('/posts')->with('error','Delete Error!');
     }
  }
}
```

4.1.4 DateRangeController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Meni;
use App\Models\Logs;
use DB;

class DateRangeController extends Controller
{
   private $data = [];
   public function __construct(){
      $meni = new Meni();
      $this->data['menus'] = $meni->getAll();
   }
   function index(Request $request)
   {
       $logs = new Logs();
     if(!empty($request->fromdate))
     {
       $fromdate=$request->fromdate;
       $todate=$request->todate;
        $this->data['logovi']=$logs->getlog($fromdate, $todate);
     }
     else
     {
      $this->data['logovi'] = $logs->getlogs();
     }

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

?>
```

4.1.5 CartController

```
<?php

namespace App\Http\Controllers;
use App\Models\Category;
use App\Models\Color;
use App\Models\Invoice;
use App\Models\InvoiceLine;
use App\Models\PostEloquentModel;
use App\Models\Post;
use App\Models\Meni;
use App\Models\SlikaEloquent;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Mail;
class CartController extends Controller
{
   private $data = [];

   public function __construct(){
       $meni = new Meni();
       $this->data['menus'] = $meni->getAll();
       $colors = Color::all();
       $categories=Category::all();
       $this->data['colors'] = $colors;
       $this->data['categories'] = $categories;
   }
   public  function index(Request $request) {
       try {
           $postId = $request->get("id");

           if(!$request->session()->has("cart")) {
               $request->session()->put("cart", []);
           }

           $post = PostEloquentModel::with('slika')->find($postId);
           $cart = $request->session()->get("cart");
           $popust=$post->popust;
           $cena=($post->price)-(($post->price)*($popust/100));
           $slike=SlikaEloquent::find($post->slika_id);

           if(isset($cart[$postId])) {
               $alreadyInCart = $cart[$postId];

               $alreadyInCart->quantity++;

               $cart[$postId] = $alreadyInCart;
           } else {
               $item = new \stdClass();
               $item->postId = $postId;
               $item->quantity = 1;
               $item->price =$cena;
               $item->image = $slike->putanja;
               $item->name = $post->naslov;
               $cart[$postId] = $item;
           }

           $request->session()->put("cart", $cart);
       } catch(\Exception $e) {
           echo $e->getMessage();
       }
   }
   public  function adjustQuantity(Request $request) {
       $postId = $request->get("id");
       $quantity = $request->get("quantity");

       $cart = $request->session()->get("cart");

       if(isset($cart[$postId])) {
           $alreadyInCart = $cart[$postId];
```

```
            $alreadyInCart->quantity = $quantity;

            $cart[$postId] = $alreadyInCart;

            $request->session()->put("cart", $cart);
        }

    }

    public function get(Request $request) {
        $cart = $request->session()->get("cart");

        if(!$cart) {
            $cart = [];
        }

        return view("pages.cart", [
            "cartItems" => $cart ], $this->data);
    }
    public  function remove($postId, Request  $request) {

        $cart = $request->session()->get("cart");
        if(isset($cart[$postId])) {
            unset($cart[$postId]);
            $request->session()->put("cart", $cart);
        }
    }
    public function myOrders() {
        $user=session()->get('user')[0]->id;
        if(session()->get('user')[0]->naziv=='admin'){
            $orders = Invoice::with("invoicelines.post")->get();
        }else {
            $orders = Invoice::with("invoicelines.post")->where("CustomerId",
$user)->get();
        }
        return view("pages.orders", [
            "orders" => $orders
        ], $this->data);
    }

    public function destroy($id)
    {
        try {
            $res=Invoice::where('InvoiceId',$id)->delete();
            $res2=InvoiceLine::where('InvoiceId',$id)->delete();
            return redirect()->back()->with("success", "Delete successfull.");
        }catch(\Exception $ex) {
            return redirect()->back()->with("error", $ex->getMessage());
        }
    }
    public function send(Request $request)
    {
        $invoiceid=$request->id;

        $invoice = new Invoice();
```

```
       $voice =Invoice::find($invoiceid);

       $mail=$voice->email;

       $mess="Your Order has been shipped:";


       $data = ['name' => 'Flower Shop' , 'email' => $mail , 'messageBody' =>
$mess ];

       Mail::send('emails.email', $data, function ($message) use ($data)
       {
           $message->from($data['email'], $data['name']);
           $message->to($data['email'], 'Customer')
               ->subject('Order Shipped');
       });

       return redirect()
           ->back()
           ->with('success', 'Email Sent');

   }
}
```

4.1.6 ContactUsController

```
<?php
namespace App\Http\Controllers;
use App\Models\Category;
use App\Models\Color;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Models\Meni;
use Mail;

class ContactUsController extends Controller
{
   public function index()
   {
       $meni = new Meni();
       $colors = Color::all();
       $categories=Category::all();
       $this->data['colors'] = $colors;
       $this->data['categories'] = $categories;
      $this->data['menus'] = $meni->getAll();
       return view('pages.contact',$this->data);
   }

   public function handleForm(Request $request)
   {

       $this->validate($request, ['name' => 'required', 'email' =>
'required|email', 'message_body' => 'required|min:20']);
```

```
       $data = ['name' => $request->get('name') , 'email' =>
$request->get('email') , 'messageBody' => $request->get('message_body') ];

       Mail::send('emails.email', $data, function ($message) use ($data)
       {
           $message->from($data['email'], $data['name']);
           $message->to('androidxx8@gmail.com', 'Admin')
               ->subject('Flower Shop');
       });

       return redirect()
           ->back()
           ->with('success', 'Thank you for your feedback');

   }

}
```

4.1.7 KorisnikController

```
<?php

namespace App\Http\Controllers;
use App\Models\Category;
use App\Models\Color;
use App\Models\Meni;
use App\Models\Uloga;
use App\Models\Korisnik;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;

class KorisnikController extends Controller {

  private $data = [];

  public function __construct() {
     $uloga = new Uloga();
     $this->data['uloge'] = $uloga->getAll();
       $colors = Color::all();
       $categories=Category::all();
       $this->data['colors'] = $colors;
       $this->data['categories'] = $categories;
               $meni = new Meni();
               $this->data['menus'] = $meni->getAll();
  }

  public function show($id = null){
     $korisnik = new Korisnik();
     $this->data['korisnici'] = $korisnik->getAll();

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

     return view('pages.createKorisnik', $this->data);
  }
       public function regshow($id = null){
     $korisnik = new Korisnik();
     $this->data['korisnici'] = $korisnik->getAll();

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

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

  public function store(Request $request){

               $rules=[
        'korisnickoIme' => 'required','unique:korisnik,korisnicko_ime','min:4',
                       'ddlUloga' => 'required','not_in:0',
                       'lozinka' => [
                           'required',
                           'min:6',

'regex:/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{6,}$/'],
        'slika' => 'required','mimes:jpg,jpeg,png,gif',
                   'imeprezime' => 'required',
                   'adresa' => 'required',
                   'grad' => 'required',
                   'opstina' => 'required',
                   'postanski' => 'required',
                   'telefon' => 'required'
     ];

               $messages = [
               "lozinka.regex" => 'Password must contain one uppercase letter and
one number.'];

               $validator = \Validator::make($request->all(), $rules, $messages);
               $validator->validate();


     $korisnicko_ime = $request->get('korisnickoIme');
     $lozinka = $request->get('lozinka');
     $uloga_id = $request->get('ddlUloga');
       if($uloga_id=='0'){
           $uloga_id = "2";
       }
       $ImePrezime= $request->get('imeprezime');
       $BillingAddress= $request->get('adresa');
       $BillingCity= $request->get('grad');
       $BillingState= $request->get('opstina');
       $BillingPostalCode= $request->get('postanski');
       $Phone= $request->get('telefon');
       $email=$request->get('email');

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

     $putanja_server = ($putanja);

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


        $korisnik = new Korisnik();
        $korisnik->korisnicko_ime = $korisnicko_ime;
        $korisnik->lozinka = $lozinka;
        $korisnik->slika = $putanja;
        $korisnik->uloga_id = $uloga_id;
           $korisnik->ImePrezime= $ImePrezime;
           $korisnik->BillingAddress= $BillingAddress;
           $korisnik->BillingCity= $BillingCity;
           $korisnik->BillingState= $BillingState;
           $korisnik->BillingPostalCode= $BillingPostalCode;
           $korisnik->Phone= $Phone;
           $korisnik->email= $email;

        $rez = $korisnik->save();
        if($rez == 1){
           return redirect('/users')->with('success','Success!');
        }
        else {
           return redirect('/users')->with('error','Error!');
        }
     }
     catch (\Exception $ex){
        \Log::error('My Error: '.$ex->getMessage());
     }
  }
       public function regstore(Request $request){

     $rules=[
        'korisnickoIme' => ['required','unique:korisnik,korisnicko_ime','min:4'],
                       'lozinka' => [
                           'required',
                           'min:6',

'regex:/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{6,}$/'],
        'slika' => 'required','mimes:jpg,jpeg,png,gif',
           'imeprezime' => 'required',
           'adresa' => 'required',
           'grad' => 'required',
           'opstina' => 'required',
           'postanski' => 'required',
           'email' =>
['required','unique:korisnik,email','regex:/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@(
[a-z0-9\-]+\.)+[a-z]{2,6}$/ix']
     ];
```

```
               $messages = [
               "lozinka.regex" => 'Password must contain one uppercase letter and
one number.'];

               $validator = \Validator::make($request->all(), $rules, $messages);
               $validator->validate();

     $korisnicko_ime = $request->get('korisnickoIme');
     $lozinka = $request->get('lozinka');
     $uloga_id = "2";
       $ImePrezime= $request->get('imeprezime');
       $BillingAddress= $request->get('adresa');
       $BillingCity= $request->get('grad');
       $BillingState= $request->get('opstina');
       $BillingPostalCode= $request->get('postanski');
       $Phone= $request->get('telefon');
           $email=$request->get('email');
     $slika = $request->file('slika');

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

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


        $korisnik = new Korisnik();
        $korisnik->korisnicko_ime = $korisnicko_ime;
        $korisnik->lozinka = $lozinka;
        $korisnik->slika = $putanja;
        $korisnik->uloga_id = $uloga_id;
           $korisnik->ImePrezime= $ImePrezime;
           $korisnik->BillingAddress= $BillingAddress;
           $korisnik->BillingCity= $BillingCity;
           $korisnik->BillingState= $BillingState;
           $korisnik->BillingPostalCode= $BillingPostalCode;
           $korisnik->Phone= $Phone;
           $korisnik->email= $email;
        $rez = $korisnik->save();

        if($rez == 1){
           return redirect('/register')->with('success','Success!');
        }
        else {
           return redirect('/register')->with('error','Error!');
        }
     }
     catch (\Exception $ex){
        \Log::error('MOJA GRESKA: '.$ex->getMessage());
     }
  }

  public function update($id, Request $request) {
     $korisnicko_ime = $request->get('korisnickoIme');
       $slozinka=$request->get('lozinka1');
     $lozinka = $request->get('lozinka');
       if($request->session()->get('user')[0]->naziv=='admin') {
           $uloga_id = $request->get('ddlUloga');
       }else{
           $uloga_id = 2;
       }
       $ImePrezime= $request->get('imeprezime');
       $BillingAddress= $request->get('adresa');
       $BillingCity= $request->get('grad');
       $BillingState= $request->get('opstina');
       $BillingPostalCode= $request->get('postanski');
       $Phone= $request->get('telefon');
       $email=$request->get('email');
     $slika = $request->file('slika');

     $korisnik = new Korisnik();


     $korisnik->id = $id;
       $olduser=$korisnik->get();
       $oldpass=$olduser->lozinka;
       $korisnik->ImePrezime= $ImePrezime;
       $korisnik->BillingAddress= $BillingAddress;
       $korisnik->BillingCity= $BillingCity;
       $korisnik->BillingState= $BillingState;
       $korisnik->BillingPostalCode= $BillingPostalCode;
       $korisnik->Phone= $Phone;
       $korisnik->email= $email;
       $korisnik->korisnicko_ime = $korisnicko_ime;

       if($request->session()->get('user')[0]->naziv=='admin'){
           $korisnik->lozinka = $slozinka;}else{
           if ($oldpass == md5($lozinka)) {
               $korisnik->lozinka = $lozinka;
           } else {
               return
redirect('/users/'.session()->get('user')[0]->id)->with('error', 'Old password
doesnt match!');
           }
       }
     $korisnik->uloga_id = $uloga_id;

     if(!empty($slika)){


        $korisnik_to_update = $korisnik->get();
        File::delete($korisnik_to_update->slika);


        $tmp_putanja = $slika->getPathName();
        $ime_fajla = time().'.'.$slika->getClientOriginalExtension();
        $putanja = 'images/'.$ime_fajla;
        $putanja_server = ($putanja);
        File::move($tmp_putanja, $putanja_server);

        $korisnik->slika = $putanja;
     }

     $rez = $korisnik->update();

     if($rez == 1){
           if($request->session()->get('user')[0]->naziv=='admin') {
               return redirect('/users')->with('success', 'Successful Update!');
           }else return redirect('/')->with('success', 'Successful Update!');
     }
     else {
        return redirect('/users')->with('error','Update Error');
     }
  }

  public function destroy($id){
     $korisnik = new Korisnik();
     $korisnik->id = $id;

     $korisnik_to_update = $korisnik->get();
     File::delete($korisnik_to_update->slika);

     $rez = $korisnik->delete();
     if($rez == 1){
        return redirect('/users')->with('success','Successful Delete!');
     }
     else {
        return redirect('/users')->with('error','Delete Error!');
     }
  }
}
```

4.1.8 CommentsController

```
<?php

namespace App\Http\Controllers;

use App\Models\CommentsModel;
use App\Models\Post;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;

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

   public function postCommentjs(int $sid,String $keyword)
   {
                   $commentsModel = new CommentsModel();
                   $commentsModel->content = $keyword;
                   $commentsModel->post_id = $sid;
                   try {
                       $commentsModel->save();
                       $output='';
                       $post = new Post();
                       $this->data['singlePost'] = $post->get($sid);

                       $commentModel = new CommentsModel();
                       $this->data['comments'] =
$commentModel->getPostComments($sid);
                       if($this->data['singlePost']){
                           $output .= '<div class="col-md-8">' .
                               '<h1
class="mt-4">'.$this->data['singlePost']->naslov.'</h1>' .
                               '<p class="lead">' .
                               'by&nbsp' .
                               '<a
href="#">'.$this->data['singlePost']->postKorisnik.'</a>' .
                               '</p><hr>' .
                               '<p>Posted on '.date("d.m.Y. H:i:s",
$this->data['singlePost']->create).'</p><hr>' .
                               '<img class="img-fluid rounded"
src="'.asset("/".$this->data['singlePost']->putanja).'"><hr>' .
                               '<p>
'.$this->data['singlePost']->sadrzaj.'</p><hr>' .
                               '<div class="card mb-4">';

                           if(session()->get('user')){
                               $output .= '<div class="card my-4">' .
                                   '<h5 class="card-header">Leave a Comment:</h5>'
.
                                   '<div class="card-body">' .
                                   '<form action="#" method="get">' .
                                   csrf_field() .
                                   '<div class="form-group">'.
                                   '<textarea class="form-control" id="psalji"
rows="3" name="content"></textarea></div>' .
                                   '<button name="akcija" value="post"
onclick="salji('.$this->data['singlePost']->postId.')" class="btn btn-primary
salji">Submit</button>' .
                                   '</form><br></div></div>'.
                                   '<div hidden>' .
                                   '<form id="pform"
action="'.route("editComment", ["postId" =>
$this->data['singlePost']->postId],["commentId" =>"2"]).'" method="post">' .
                                   csrf_field() .
                                   '<div class="form-group">' .
                                   '<textarea class="form-control" id="txta"
rows="3" name="content1"></textarea></div>' .
                                   '<button type="submit" name="akcija"
value="edit" class="btn btn-primary">Update</button></form></div>';
                               if($this->data['comments']){
                                   $output .= '<div class="comments
heading"><h3>Comments</h3><form id="komentform">';
                                   foreach ($this->data['comments'] as $comment){
                                       $output .= '<div class="media"><div
class="media-body"><table class="table"><tr><td>' .
                                          '<h4
class="media-heading">'.$comment->korisnicko_ime.'</h4>' .
                                          '<h6><label>'.date("F j, Y",
strtotime($comment->created_at)).'</label></h6>' .
                                          '<span id="comment'.$comment->id.'">' .
                                          '<span
id="kom'.$comment->id.'">'.$comment->content.'</span></br>';
                                      if(session()->get('user')[0]->id ==
$comment->user_id || session()->get('user')[0]->naziv == 'admin'){
                                          $output .= '<a
href="'.route("deleteComment", ["commentId" => $comment->id]).'"><i class="fa
fa-trash">Obrisi</i></a>&nbsp';
                                      }
                                      if(session()->get('user')[0]->id ==
$comment->user_id){
                                          $output .= '<a
href="#comment'.$comment->id.'"><i class="fa fa-edit"
onclick="editComment('.$comment->id.','.$this->data['singlePost']->postId.')">Izme
ni</i></a>';
                                      }
                                      $output .= '<span
id="#comments"></span></span></td></tr></table></div><div
class="media-right"></div></div>';
                                  }
                                  $output .= '</form></div>';
                              }
                         }else{
                             $output .= '<h4>Da bi komentarisali morate da se
ulogujete.</h4>';
                         }$output .= '</div>';

                     }

                     return response()->json($output);
                 } catch (QueryException $e) {
                     \Log::error("Greska pri dodavanju komentara " .
$e->getMessage());
                     return redirect()->back()->with('warning', "Doslo je do
greske na serveru.");
                 }


   }
   /*public function postKoment(Request $request)
   {
                 $commentsModel = new CommentsModel();
                 $commentsModel->content = $request->get('content');
                 $commentsModel->post_id = $request->get("postid");
                 $id=$request->get("postid");
                 try {
                     $commentsModel->save();
                     $this->data[] = $commentsModel->getPostComments($id)-;
                     return response()->json($this->data);
                 } catch (QueryException $e) {
                     \Log::error("Greska pri dodavanju komentara " .
$e->getMessage());
                       return redirect()->back()->with('warning', "Doslo je do
greske na serveru.");
                   }

   }*/

   public function editComment(Request $request, $commentId)
   {
       switch($request->akcija) {
           case 'edit':
       $commentModel = new CommentsModel();
       try {
           $commentModel->content = $request->get('content1');
           $commentModel->update($commentId);
           return redirect()->back()->with('success', "Komentar uspesno
izmenjen.");
       } catch (\Exception $e) {
           \Log::error("Greska pri izmeni komentara " . $e->getMessage());
           return redirect()->back()->with('warning', "Doslo je do greske na
serveru.");
       }
       break;
   }

   }

   public function deleteComment($commentId)
   {
       $commentModel = new CommentsModel();
       if ($commentModel->find($commentId)) {
           try {
               if ($commentModel->delete($commentId)) {
                   return redirect()->back()->with("success", "Comment
successfully deleted.");
               } else {
                   \Log::critical("Korisnik bez dozvoljenih prava pokusao da
obrise komentar.");
                   return redirect()->back()->with("warning", "An error has
occurred, please try again later.");
               }
           } catch (QueryException $e) {
               \Log::error("Greska pri brisanju komentara  " . $e->getMessage());
               return redirect()->back()->with("warning", "An error has occurred,
please try again later.");
           }
       }
       return redirect()->back();
   }

   public function show($commentId)
   {
       $commentModel = new CommentsModel();
       $this->data['comments'] = $commentModel->getPostComments($commentId);

     if(!empty($commentId)){
        $commentModel->id = $commentId;
        $this->data['comment'] = $commentModel->getPostComments($commentId);
       return view('components.comments', $this->data);

   }

               }
}
```

4.1.9 CheckoutController

```
<?php

namespace App\Http\Controllers;

use App\Models\Invoice;
use App\Models\InvoiceLine;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CheckoutController extends Controller
{
    public function index(Request $request) {
        $cart = $request->session()->get("cart");
        $invoice = new Invoice();
        $invoice->CustomerId = $request->session()->get('user')[0]->id;
        $invoice->InvoiceDate = Carbon::now("UTC");
        $invoice->ImePrezime=$request->session()->get("user")[0]->ImePrezime;
      $invoice->BillingAddress=$request->session()->get("user")[0]->BillingAddress;
        $invoice->BillingCity=$request->session()->get("user")[0]->BillingCity;
        $invoice->BillingState=$request->session()->get("user")[0]->BillingState;
$invoice->BillingPostalCode=$request->session()->get("user")[0]->BillingPostalCod;
        $invoice->Phone=$request->session()->get("user")[0]->Phone;
        $invoice->email=$request->session()->get("user")[0]->email;
        $invoiceId = Invoice::max("InvoiceId")+1;
        $invoice->InvoiceId = $invoiceId;
        $lines = [];
        $total = 0;
        $lineId = InvoiceLine::max("InvoiceLineId") + 1;
        foreach($cart as $c) {
            $line = new InvoiceLine();
            $line->InvoiceId = $invoiceId;
            $line->Quantity = $c->quantity;
            $line->UnitPrice = $c->price;
            $line->PostId = $c->postId;
            $line->InvoiceLineId = $lineId++;
            $total += $c->quantity * $c->price;
            $lines[] = $line;
        }
        $invoice->Total = $total;
        try {
            DB::beginTransaction();
            $invoice->save();
            foreach($lines as $line) {
                $line->save();
            }
            DB::commit();
            $request->session()->remove("cart");
        }catch(\Exception $e) {
            DB::rollBack();
            echo $e->getMessage();
        }
    }
}
```

4.2 Models

4.2.1 Category

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
   use HasFactory;
   protected $table = 'category';
}
```

4.2.2 Color

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Color extends Model
{
   use HasFactory;
   protected $table = 'color';
}
```

4.2.3 CommentsModel

```php
<?php

namespace App\Models;


class CommentsModel
{
   public $content;
   public $post_id;
   private $table = 'comments';

   public function save()
   {
       return \DB::table($this->table)
           ->insert([
              'user_id' => session()->get('user')[0]->id,
              'post_id' => $this->post_id,
               'content' => $this->content,
               'created_at' => date("Y-m-d H:i:s"),
               'updated_at' => date("Y-m-d H:i:s")
           ]);
   }

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

   public function getPostComments($postId)
   {
       return \DB::table($this->table)
           ->join("korisnik", "comments.user_id", "=", "korisnik.id")
           ->where('post_id', $postId)
```

```
            ->select('comments.*', 'korisnik.korisnicko_ime')
            ->orderBy('comments.created_at','desc')
            ->get();
    }

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

    public function getUserComments($userId)
    {
        return \DB::table($this->table)->where('user_id', $userId)->get();
    }

    public function find($id)
    {
        return \DB::table($this->table)
            ->where('id', $id)->get()->first();
    }
    private function canUserDeleteComment($commentId)
    {
        $comment = $this->find($commentId);
        return $comment ? (session()->get('user')[0]->naziv == 'admin') ||
($comment->user_id == session()->get('user')[0]->id) : false;
    }
}
```

4.2.4 Meni

```php
<?php

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Meni {
   //put your code here

   public function getAll(){
       $rezultat = DB::table('meni')->get();
       return $rezultat;
   }
}
```

4.2.5 Logs

```php
<?php

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

class Logs {
   public $fromdate;
   public $todate;

   public function getlogs(){
       $rezultat = DB::table('logs')->get();
       return $rezultat;
   }
   public function getlog($fromdate,$todate){
       $newfromdate=$fromdate.' 00:00:00';
       $newtodate=$todate. '23:59:59';
       $newfrom=\Carbon\Carbon::parse($newfromdate)->timestamp;
       $newto=\Carbon\Carbon::parse($newtodate)->timestamp;
       $rezultat = DB::table('logs')
       ->select('*')
     ->whereBetween('time',array($newfrom,$newto))
       ->get();
       return $rezultat;
   }
}
```

4.2.6 Invoice

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Invoice extends Model
{
   use HasFactory;
   public  $timestamps = false;
   protected $table = "invoice";
   protected  $primaryKey = "InvoiceId";

   public function invoiceLines() {
       return $this->hasMany(InvoiceLine::class, "InvoiceId", "InvoiceId");
   }
}
```

4.2.7 InvoiceLine

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class InvoiceLine extends Model
{
   use HasFactory;
   public  $timestamps = false;
   protected $table = "invoiceline";
   protected  $primaryKey = "InvoiceLineId";

   public  function invoice() {
       return $this->belongsTo(Invoice::class, "InvoiceId", "InvoiceId");
   }
   public function post() {
       return $this->belongsTo(PostEloquentModel::class, "PostId", "id");
   }
}
```

4.2.8 Korisnik

```php
<?php

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

class Korisnik
{
  public $id;
  public $korisnicko_ime;
  public $lozinka;
  public $slika;
  public $uloga_id;
   public $ImePrezime;
   public $BillingAddress;
   public $BillingCity;
   public $BillingState;
   public $BillingPostalCode;
   public $Phone;
   public $email;

  public function getAll(){
     $rezultat = DB::table('korisnik')
              ->select('*', 'korisnik.id AS korisnikId')
              ->join('uloga','uloga.id','=','korisnik.uloga_id')
              ->get();
     return $rezultat;
  }

  public function get() {
     $rezultat = DB::table('korisnik')
              ->select('*')
              ->where('id',$this->id)
              ->first();
     return $rezultat;
  }

  public function getByUsernameAndPassword(){
     $rezultat = DB::table('korisnik')
              ->select('korisnik.*','uloga.naziv')
              ->join('uloga','korisnik.uloga_id', '=', 'uloga.id')
              ->where([
                 'korisnicko_ime' => $this->korisnicko_ime,
                 'lozinka' => md5($this->lozinka)
              ])
              ->first();
     $log = DB::table('logs')->insert([
     'user' =>$this->korisnicko_ime,
     'action' => 'Login Korisnik '.$this->korisnicko_ime,
     'time' => time()]);
     return $rezultat;
  }

  public function save() {
     $rez = DB::table('korisnik')->insert([
        'korisnicko_ime' => $this->korisnicko_ime,
        'lozinka' => md5($this->lozinka),
           'ImePrezime' => $this->ImePrezime,
           'BillingAddress' => $this->BillingAddress,
           'BillingCity' => $this->BillingCity,
           'BillingState' => $this->BillingState,
           'BillingPostalCode' => $this->BillingPostalCode,
           'Phone' => $this->Phone,
           'email' => $this->email,
        'slika' => $this->slika,
        'uloga_id' => $this->uloga_id,
                 'created_at' => time()]);
     $log = DB::table('logs')->insert([
     'user' =>$this->korisnicko_ime,
     'action' => 'Added Korisnik '.$this->korisnicko_ime,
     'time' => time()]);
     return $rez;
  }

  public function update(){
     $data = [
        'korisnicko_ime' => $this->korisnicko_ime,
        'lozinka' => md5($this->lozinka),
           'ImePrezime' => $this->ImePrezime,
           'BillingAddress' => $this->BillingAddress,
           'BillingCity' => $this->BillingCity,
           'BillingState' => $this->BillingState,
           'BillingPostalCode' => $this->BillingPostalCode,
           'Phone' => $this->Phone,
           'email' => $this->email,
        'uloga_id' => $this->uloga_id,
                       'updated_at' => time(),
     ];

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

     $rez = DB::table('korisnik')
     ->where('id',$this->id)
           ->update($data)
           ;
     $log = DB::table('logs')->insert([
     'user' =>session()->get('user')[0]->korisnicko_ime,
     'action' => 'Updated Korisnik '.$this->korisnicko_ime,
     'time' => time()]);
     return $rez;
  }

  public function delete(){
       $user = DB::table('korisnik')
           ->select('*')
           ->where('id',$this->id)
           ->first();
       $username=$user->korisnicko_ime;
     $rezultat = DB::table('korisnik')
              ->where('id', $this->id)
```

```
              ->delete();
     $log = DB::table('logs')->insert([
     'user' =>session()->get('user')[0]->korisnicko_ime,
     'action' => 'Deleted Korisnik '.$username,
     'time' => time()]);
     return $rezultat;
  }
}
```

4.2.9 Post

```
<?php

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

class Post {
  public $id;
  public $naslov;
  public $sadrzaj;
  public $korisnik_id;
  public $slika_id;
   public $price;
   public $ColorId;
   public $CategoryId;
   public $popust;
  private $table = 'post';

  public function getAll(){
     $rezultat = DB::table('post')
              ->select('*','post.id as postId','post.created_at as
create','post.updated_at as update')
              ->join('slika as s','post.slika_id','=','s.id')
              ->join('korisnik as k','post.korisnik_id','=','k.id')
                                   ->join('uloga as
u','k.uloga_id','=','u.id')
              ->orderBy('post.created_at','desc')
              ->get();
     return $rezultat;
  }


  public function save() {
     $rezultat = DB::table('post')->insert([
        'naslov' => $this->naslov,
        'sadrzaj' => $this->sadrzaj,
           'price'=>$this->price,
        'korisnik_id' => $this->korisnik_id,
           'ColorId' => $this->ColorId,
           'CategoryId' => $this->CategoryId,
        'created_at' => time(),
        'slika_id' => $this->slika_id
     ]);
     $log = DB::table('logs')->insert([
        'user' =>session()->get('user')[0]->korisnicko_ime,
        'action' => 'Made new post '.$this->naslov,
        'time' => time()
     ]);
  }


   public function get($id){
       $rezultat =
               DB::table($this->table)
               ->select('*',
                       'post.id AS postId',
                       'post.created_at as create',
                       'post.updated_at as update',
                       'korisnik.korisnicko_ime as postKorisnik',
                       'k.korisnicko_ime as komentarKorisnik',
                       'comments.post_id as comments')
               ->join('slika','slika.id','=','post.slika_id')
               ->join('korisnik','korisnik.id','=','post.korisnik_id')
               ->leftJoin('comments','comments.post_id','=','post.id')
               ->leftJoin('korisnik AS k','k.id','=','comments.user_id')
               ->where('post.id','=',$id )
               ->first();
       return $rezultat;
   }

   public function getp() {
  $rezultat = DB::table('post')
           ->select('*')
           ->where('id',$this->id)
           ->first();
  return $rezultat;
}
   public function update(){
     $data = [
        'naslov' => $this->naslov,
        'sadrzaj' => $this->sadrzaj,
           'price'=>$this->price,
           'ColorId' => $this->ColorId,
           'CategoryId' => $this->CategoryId,
           'popust' => $this->popust,
                       'updated_at' => time(),
     ];

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

     $rez = DB::table('post')
     ->where('id',$this->id)
     ->update($data);
     $log = DB::table('logs')->insert([
        'user' =>session()->get('user')[0]->korisnicko_ime,
        'action' => 'Updated post '.$this->naslov,
        'time' => time()
     ]);
       return $rez;
  }
  public function delete(){
       $post = DB::table('post')
           ->select('*')
           ->where('id',$this->id)
           ->first();
       $postname=$post->naslov;
     $rezultat = DB::table('post')
              ->where('id', $this->id)
              ->delete();
     $log = DB::table('logs')->insert([
     'user' =>session()->get('user')[0]->korisnicko_ime,
     'action' => 'Deleted post '.$postname,
     'time' => time()]);
     return $rezultat;
  }
       public function deleted(){
     $rezultat = DB::table('slika')
              ->where('id', $this->slika_id)
              ->delete();
     return $rezultat;
  }
       public function getPictureId($postId)
   {
       return \DB::table($this->table)
           ->where('id', $postId)
           ->select("slika_id")
           ->get()->first();
   }

}
```

4.2.10 PostEloquentModel

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class PostEloquentModel extends Model
{
   use HasFactory;

   protected $table = 'post';
   protected $primaryKey = 'id';

   public function slika()
   {
       return $this->hasOne(SlikaEloquent::class,'id');
   }
}
```

4.2.11 Slika

```php
<?php

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

class Slika {

  public $id;
  public $putanja;

  public function save(){
     $id = DB::table('slika')
           ->insertGetId([
              'putanja' => $this->putanja
           ]);
     return $id;
  }
       public function find($id)
   {
       return \DB::table($this->table)
           ->where('id', $id)
           ->get()
           ->first();
   }

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

4.2.12 SlikaEloquent

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class SlikaEloquent extends Model
{
   use HasFactory;

   protected $table = 'slika';
   protected $primaryKey = 'id';
}
```

4.2.13 Uloga

```php
<?php

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

```
class Uloga {

  private $table = 'uloga';
  public $id;
  public $naziv;

  public function getAll(){
     $rezultat = DB::table($this->table)->get();
     return $rezultat;
  }

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

  public function save(){
     $rez= DB::table($this->table)
        ->insert([
           'naziv' => $this->naziv
        ]);
        $log = DB::table('logs')->insert([
           'user' =>session()->get('user')[0]->korisnicko_ime,
           'action' => 'Added Uloga '.$this->naziv,
           'time' => time()
        ]);
     return $rez;
  }

  public function update(){
     $rez=DB::table($this->table)
        ->where('id', $this->id)
        ->update([
           'naziv' => $this->naziv
        ]);
        $log = DB::table('logs')->insert([
           'user' =>session()->get('user')[0]->korisnicko_ime,
           'action' => 'Updated Uloga '.$this->naziv,
           'time' => time()
        ]);
     return $rez;
  }
  public function delete(){
     $rez=DB::table($this->table)
        ->where('id', $this->id)
        ->delete();
        $log = DB::table('logs')->insert([
           'user' =>session()->get('user')[0]->korisnicko_ime,
           'action' => 'Deleted Uloga '.$this->id,
           'time' => time()
        ]);
     return $rez;
  }
}
```

4.3 Views

4.3.1 Front.blade.php

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8">
   <link rel="icon" type="image/jpg" href="images/film.png" />
   <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
   <meta name="description" content="flower shop">
   <meta name="keywords" content="flowers" />
   <meta name="author" content="Stefan Popovic">
     <meta name="csrf-token" content="{{ csrf_token() }}" />
   <title> Flower Shop | @yield('title') </title>

   @section('appendCss')
   <link href="{{ asset('/') }}vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
   @show
<style>
   body {
       min-height: 100vh;
       display: flex;
       flex-direction: column;
   }
   footer {
       margin-top: auto;
   }
</style>
<script src="{{ asset('/') }}vendor/jquery/jquery.min.js"></script>
<link rel="stylesheet" href="{{ asset('/') }}js/Popup/dist/magnific-popup.css">
<script src="{{ asset('/') }}js/provera.js"></script>
<script src="{{ asset('/') }}js/Popup/dist/jquery.magnific-popup.min.js"></script>
<script type="text/javascript">
  $(document).ready(function() {

     $('.image-popup-vertical-fit').magnificPopup({
        type: 'image',
        closeOnContentClick: true,
        mainClass: 'mfp-img-mobile',
        image: {
           verticalFit: true
        }

     });

     $('.image-popup-fit-width').magnificPopup({
        type: 'image',
        closeOnContentClick: true,
        image: {
           verticalFit: false
        }
     });

     $('.image-popup-no-margins').magnificPopup({
        type: 'image',
        closeOnContentClick: true,
        closeBtnInside: false,
        fixedContentPos: true,
     mainClass: 'mfp-no-margins mfp-with-zoom',
     image: {
        verticalFit: true
     },
     zoom: {
        enabled: true,
        duration: 300
     }
  });

  });


</script>

 </head>
 <body>

   @include('components.nav')

   <div class="container">

     <div class="row">
       <div class="col-lg-12"></div>
       <div class="col-lg-12">
           <div id="poruke"></div>
         @empty(!session('error'))
           <div class="alert alert-danger">{{ session('error') }}</div>
         @endempty

         @empty(!session('success'))
           <div class="alert alert-success">{{ session('success') }} </div>
         @endempty
       </div>

       @yield('content')

       @include('components.sidebar')

     </div>

   </div>


   @include('components.footer')


   @section('appendJavascript')

       <script>
           $(document).ready(function() {
               correctTotalPrice()
           })
           function changeQuantity(input, price, postId) {
               if(!input.value || input.value < 1) {
                   $('#poruke').html('Not allowed
quantity.').removeClass().addClass("alert alert-danger");
                   input.value = 1;
               } {
                   $("#span_price_" + postId).html((input.value *
price).toFixed(2))
                   correctTotalPrice()

                   $.ajax({
                       method: "POST",
                       url: "/cart/adjustQuantity",
                       data: {
                           trackId : postId,
                           quantity : input.value
                       },
                       success: function() {
                           console.log(data);
                       },
                       error: function(xhr, status, error) {
                           console.log(data);
                       }
                   })
               }
           }
           function correctTotalPrice() {
               var allPrices = document.querySelectorAll(".product_price")

               var totalPrice = 0;

               for(let priceSpan of allPrices) {
                   totalPrice += Number(priceSpan.innerHTML)
               }

               $("#totalPrice").html(totalPrice.toFixed(2))
           }
           function addToCart(id) {
               $.ajax({
                   method: "POST",
                   url: "/cart",
                   data: {
                       id : id
                   },
                   success: function(data) {
                       console.log(data);
                       $('#poruke').html('Product successfully added to
cart.').removeClass().addClass("alert alert-success");
                   },
                   error: function(data) {
                       console.log(data);
                       $('#poruke').html('There was an error processing your
request.').removeClass().addClass("alert alert-danger");
                   }
               })
           }
```

```
           function removeFromCart(postId) {
               $.ajax({
                   method: "DELETE",
                   url: "/cart/" + postId,
                   success: function() {
                       $('#poruke').html('Product successfully removed from the
cart.').removeClass().addClass("alert alert-success");

                       let cartItemDiv = $("#cart_item_" + postId)

                       cartItemDiv.remove()
                       correctTotalPrice()
                   },
                   error: function(xhr, status, error) {
                       $('#poruke').html('There was an error processing your
request.').removeClass().addClass("alert alert-danger");
                   }
               })
           }
           function checkout(btn) {
               $.ajax({
                   method: "POST",
                   url: "/checkout",
                   success: function() {
                       $('#poruke').html('Your order has been
submitted.').removeClass().addClass("alert alert-success");
                       $(btn).attr("disabled", "disabled")
                   },
                   error: function(xhr, status, error) {
                       $('#poruke').html('There was an error processing your
order.').removeClass().addClass("alert alert-danger");
                   }
               })
           }
       </script>
   <script src="{{ asset('/')
}}vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
   <script src="{{ asset("js/ajax.js") }}"></script>
   @show

 </body>

</html>
```

4.3.2 comments.blade.php

```
<div class="comments heading">
   <h3>Comments</h3>
   <form id="komentform">
   @foreach($comments as $comment)
       <div class="media">
           <div class="media-body">
               <table class="table">
                   <tr>
                       <td>
               <h4 class="media-heading"> {{ $comment->korisnicko_ime }}</h4>
               <h6><label>{{ date("F j, Y", strtotime($comment->created_at))
}}</label></h6>
               <span id="comment{{$comment->id}}">
                   <span id="kom{{$comment->id}}">{{ $comment->content
}}</span></br>
                   @if(session('user'))
                       @if(session()->get('user')[0]->id == $comment->user_id ||
session()->get('user')[0]->naziv == 'admin')
                           <a href="{{ route("deleteComment", ['commentId' =>
$comment->id]) }}"><i class="fa fa-trash">Obrisi</i></a>
                       @endif
                       @if(session()->get('user')[0]->id == $comment->user_id)
                                <a href="#comment{{$comment->id}}"><i class="fa
fa-edit" onclick="editComment({{ $comment->id }},{{
$singlePost->postId}})">Izmeni</i></a>
                       @endif
                   @endif
                   <span id="#comments"></span>
               </span>
                       </td>
               </tr>
               </table>
           </div>
           <div class="media-right">

           </div>
       </div>
   @endforeach
   </form>
</div>
```

4.3.3 footer.blade.php

```
<!-- Footer -->
   <footer class="py-5 bg-dark footer">
     <div class="container">
         <p class="m-0 text-center text-white">Copyright &copy; Stefan Popovic
2022</p>
     </div>
   </footer>
<script>
   $.ajaxSetup({
       headers: {
           'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
       }
    });
</script>
  <!--// Footer -->
```

4.3.4 sidebar.blade.php

```
<!-- Desna strana -->
       <div class="col-md-4">
           <div class="card my-4">
               <h5 class="card-header">Search</h5>
               <div class="card-body">
                   <div class="input-group">
           <form method="GET" action="{{ route("searchs") }}" class="form-inline
my-2 my-lg-0">
               <div>
                   <label>Sort:</label>
                   <div class="">
                       <select name="sortBy" class="form-control">
                           <option value="0">Choose...</option>
                           <option value="Price-asc">Price ASC</option>
                           <option value="Price-desc">Price DESC</option>
                           <option value="Naslov-asc">Name ASC</option>
                           <option value="Naslov-desc">Name DESC</option>
                       </select>
                   </div>
               </div>
               <div>
                   <label>Color:</label>
                   <div class="">
                       <select class="form-control" name="boje">
                           <option value="0">Choose...</option>
                           @foreach($colors as $c)

                                   <option value="{{ $c->ColorId }}">{{ $c->Name
}}</option>
                           @endforeach
                       </select>
                   </div>
               </div>

               <div>
                   <label>Category:</label>
                   <div class="">
                       <select class="form-control" name="kategorije">
                           <option value="0">Choose...</option>
                           @foreach($categories as $t)

                               <option value="{{ $t->CategoryId }}">{{ $t->Name
}}</option>
                           @endforeach
                       </select>
                   </div>
               </div>
               <br>
               <input class="form-control mr-sm-2 mt-3" name="keyword"
type="search" placeholder="Search" aria-label="Search">
               <button class="btn btn-outline-success mt-3"
type="submit">Search</button>
           </form>
                   </div>
               </div>
           </div>
           @if(!session()->has('user'))
         <div class="card my-4">
           <h5 class="card-header">Login</h5>
           <div class="card-body">
             <div class="row">
               <div class="col-lg-12">
                 <!-- LOGIN FORM -->

                 <form action="{{ route('login') }}" method="POST">
                   {{ csrf_field() }}
                   <div class="form-group">
                       <label>Username:</label>
                       <input type="text" id="imeprezime" name="tbKorisnickoIme"
class="form-control" onkeyup="proveraKontakt()"/>
                   </div>

                   <div class="form-group">
                     <label>Password:</label>
                     <input type="password" id="sifra" name="tbLozinka"
class="form-control" onkeyup="proveraKontakt()"/>
                   </div>

                   <input type="submit" name="btnLogin" value="Login" class="btn
btn-primary"/>
                 <a href="{{ asset('register') }}" class="btn
btn-warning">Register</a>
                 </form>

                 <!--// LOGIN FORM -->

               </div>
             </div>
           </div>
         </div>
           @endif
         @if(session()->has('user'))
               @if(session()->get('user')[0]->naziv == 'admin')
         <div class="card my-4">
           <h5 class="card-header">Panel</h5>
           <div class="card-body">
             <p>
               <a href="{{ asset('/posts') }}" class="btn
btn-warning">Create/Change post</a>
               <br/> <br/>
               <a href="{{ asset('/users') }}" class="btn btn-warning">User
Control</a>
               <br/> <br/>
               <a href="{{ asset('/logs') }}" class="btn btn-warning">Logs</a>
               @endif
               @if(session()->get('user')[0]->naziv != 'admin')
                   <div class="card my-4">
                       <h5 class="card-header">Panel</h5>
                       <div class="card-body">
                           <p>
                           <td> <a href="{{
asset('/users/'.session()->get('user')[0]->id) }}" class="btn btn-warning">Edit
Profile</a> </td>
                               @endif
             </p>
           </div>
         </div>
         @endif
       </div>
     <!--// Desna strana -->
```

4.3.5 Nav.blade.php

```
<!-- Navigacija -->
   <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
     <div class="container">
       <a class="navbar-brand" href="">Flower Shop</a>
       <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarResponsive" aria-controls="navbarResponsive"
aria-expanded="false" aria-label="Toggle navigation">
         <span class="navbar-toggler-icon"></span>
       </button>
       <div class="collapse navbar-collapse" id="navbarResponsive">
         <ul class="navbar-nav ml-auto">
           @isset($menus)
             @foreach($menus as $menu)
             <li class="nav-item {{ (Request::url().'/' == asset($menu->link))?
'active' : '' }}">
               <a class="nav-link" href="{{ asset($menu->link) }}">
                 {{ $menu->naziv }}
               </a>
             </li>
             @endforeach
           @endisset
               <li>
                   <a class="nav-link" href="{{ route("get-cart")}}">Cart <span
class="sr-only"></span></a>
               </li>
           @if(session()->has('user'))
                   <li>
                       <a class="nav-link" href="{{ route("my-orders")}}">My
Orders <span class="sr-only"></span></a>
                   </li>
             <li>
               <a class="nav-link" href="{{ route('logout') }}">
                 Logout
               </a>
             </li>
           @endif
         </ul>
       </div>
     </div>
   </nav>
  <!--// Navigacija -->
```

4.3.6 contact.blade.php

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

@section('title')
   Contact
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">
           <h3>Contact Us</h3>

           @empty(!session('message'))
             {{ session('message') }}
           @endempty

           @isset($errors)
             @if($errors->any())
               @foreach($errors->all() as $error)
                 <div class="alert alert-danger"> {{ $error }} </div>
               @endforeach
             @endif
           @endisset
           <form action="{{ url('contact-us')}}" method="POST">
             {{ csrf_field() }}
             <div class="form-group">
             <label>Name</label>
               <input type="text" name="name" class="form-control">
             </div>
             <div class="form-group">
             <label>E-mail</label>
               <input type="email" name="email" class="form-control">
             </div>
             <div class="form-group">
             <label>Message</label>
               <textarea name="message_body" class="form-control"></textarea>
             </div>
             <div class="form-group">
             <button type="submit" class="btn btn-primary">Submit</button>
             </div>
           </form>
       </div>
     <!--// Sadrzaj -->
@endsection
```

4.3.7 Cart.blade.php

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

@section('title')
   Contact
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
   <!-- Sadrzaj -->
   <div class="col-md-8">
   <br>
   @if(count($cartItems) == 0)
       <div class="container">
           <p class="lead">Your cart is empty.</p>
       </div>
   @else

           <div class="row">
               <div class="col-md-8 cart">
                   <div class="title">
                       <div class="row">
                           <div class="col">
                               <h4><b>Shopping Cart</b></h4>
                           </div>
                       </div>
                   </div>

                   @foreach($cartItems as $c)
                       <div class="row border-top border-bottom" id="cart_item_{{
$c->postId }}">
                           <div class="row main align-items-center">
                               <div class="col-2"><img class="img-fluid" src="{{
asset('/').$c->image }}"></div>
                               <div class="col">
                                   <div class="row text-muted">Name</div>
                                   <div class="row">{{ $c->name }}</div>
                               </div>
                               <div class="col"><input
onchange="changeQuantity(this,{{ $c->price }}, {{ $c->postId }})" type="number"
value="{{ $c->quantity }}"> </div>
                               <div class="col">&euro; <span class="product_price"
id="span_price_{{ $c->postId }}">{{ $c->price * $c->quantity }}</span> <span
onclick="removeFromCart({{$c->postId}})" class="close">&#10005;</span></div>
                           </div>
                       </div>
                   @endforeach

               </div>
               <div class="col-md-4 summary">
                   <div>
                       <h5><b>Summary</b></h5>
                   </div>
                   <div class="row" style="border-top: 1px solid rgba(0,0,0,.1);
padding: 2vh 0;">
                       <div class="col">TOTAL PRICE</div>
                       <div class="col text-right">&euro; <span id="totalPrice">
</span></div>
                   </div> <button onclick="checkout(this)"
class="btn">CHECKOUT</button>
               </div>
           </div>

   @endif
   <br>
   <br>
   <br>
   </div>
   <!--// Sadrzaj -->
@endsection
```

4.3.8 CreateKorisnik.blade.php

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

@section('title')
   User create
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">
           <h3>Add user</h3>

           @empty(!session('message'))
             {{ session('message') }}
           @endempty

           @isset($errors)
             @if($errors->any())
               @foreach($errors->all() as $error)
                 <div class="alert alert-danger"> {{ $error }} </div>
               @endforeach
             @endif
           @endisset

           <form action="{{ (isset($korisnik))?
asset('/users/update/'.$korisnik->id)  : asset('/users/store') }}" method="POST"
```

```
enctype="multipart/form-data">

             {{ csrf_field() }}

             <div class="form-group">
               <label>Username:</label>
               <input type="text" name="korisnickoIme" class="form-control"
value="{{ (isset($korisnik))? $korisnik->korisnicko_ime : old('korisnickoIme')
}}"/>
             </div>
             <div class="form-group">
                 @if(!isset($korisnik))
               <label>Password:</label>
                 @endif
                 @if(isset($korisnik))
                     <label>Old Password:</label>
                     <input type="password" name="lozinka1" class="form-control"
value="{{ (isset($korisnik))? $korisnik->lozinka : old('lozinka1') }}"/>
                     <label>Password Confirmation:</label>
                     @endif
               <input type="password" name="lozinka" class="form-control"
value="{{ (isset($korisnik))? old('lozinka') : old('lozinka') }}"/>
             </div>
               <div class="form-group">
                   <label>Name and Surname:</label>
                   <input type="text" name="imeprezime" class="form-control"
value="{{ (isset($korisnik))? $korisnik->ImePrezime : old('imeprezime') }}"/>
               </div>
               <div class="form-group">
                   <label>Address:</label>
                   <input type="text" name="adresa" class="form-control" value="{{
(isset($korisnik))? $korisnik->BillingAddress : old('adresa') }}"/>
               </div>
               <div class="form-group">
                   <label>City:</label>
                   <input type="text" name="grad" class="form-control" value="{{
(isset($korisnik))? $korisnik->BillingCity : old('grad') }}"/>
               </div>
               <div class="form-group">
                   <label>State:</label>
                   <input type="text" name="opstina" class="form-control"
value="{{ (isset($korisnik))? $korisnik->BillingState : old('opstina') }}"/>
               </div>
               <div class="form-group">
                   <label>Postal Code:</label>
                   <input type="text" name="postanski" class="form-control"
value="{{ (isset($korisnik))? $korisnik->BillingPostalCode : old('postanski')
}}"/>
               </div>
               <div class="form-group">
                   <label>Phone:</label>
                   <input type="text" name="telefon" class="form-control"
value="{{ (isset($korisnik))? $korisnik->Phone : old('telefon') }}"/>
               </div>
               <div class="form-group">
                   <label>Email:</label>
                   <input type="text" name="email" class="form-control" value="{{
(isset($korisnik))? $korisnik->email : old('email') }}"/>
               </div>
             <div class="form-group">
               <label>Photo:</label>


           @isset($korisnik)
           <img src="{{ asset($korisnik->slika) }}" width="35" height="35"
style="border-radius: 500px; -moz-border-radius: 500px; margin-top: 2px;"/>
               @endisset


               <input type="file" name="slika" class="form-control"  />

             </div>
               @if(session()->has('user'))
                   @if(session()->get('user')[0]->naziv == 'admin')
             <div class="form-group">
               <label>Uloga:</label>
               <select name="ddlUloga">
                 <option value="0">Choose</option>

                 @foreach($uloge as $uloga)
                   <option value="{{ $uloga->id }}" {{ (isset($korisnik) &&
$korisnik->uloga_id == $uloga->id)? 'selected' : ((old('ddlUloga')==$uloga->id)?
'selected' : '' )}} > {{ $uloga->naziv }} </option>
                 @endforeach

               </select>
             </div>
                   @endif
               @endif
             <div class="form-group">
               <input type="submit" name="addKorisnik" value="{{
(isset($korisnik))? 'Change korisnik' : 'Add korisnik' }} " class="btn
btn-default" />
             </div>
           </form>
           @if(session()->has('user'))
               @if(session()->get('user')[0]->naziv == 'admin')
           <table class="table">
               <tr>
                 <td>ID</td>
                 <td>Username</td>
                 <td>Photo</td>
                 <td>Create/Update Date</td>
                 <td>Role</td>
                 <td>Change</td>
                 <td>Delete</td>
               </tr>
               <!-- Prikaz korisnika-->
               @isset($korisnici)
               @foreach($korisnici as $korisnik)
                 <tr>
                   <td> {{ $korisnik->korisnikId }} </td>
                   <td> {{ $korisnik->korisnicko_ime }} </td>
                   <td> <img src="{{ asset($korisnik->slika) }}" width="75"
height="75" style="border-radius: 500px; -moz-border-radius: 500px;" /> </td>
                   <td> {{ date("d.m.Y. H:i:s", $korisnik->created_at)
}}</br>@if($korisnik->updated_at!=null){{ date("d.m.Y. H:i:s",
$korisnik->updated_at) }}@endif </td>
                   <td> {{ $korisnik->naziv }} </td>
                   <td> <a href="{{ asset('/users/'.$korisnik->korisnikId)
}}">Change</a> </td>
                   <td> <a href="{{ asset('/users/destroy/'.$korisnik->korisnikId)
}}">Delete</a> </td>
                 </tr>
               @endforeach
               @endisset
           </table>
               @endif
               @endif
       </div>
     <!--// Sadrzaj -->
@endsection
```

4.3.9 CreatePost.blade.php

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

@section('title')
   Post create
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">
           <h3>Add post</h3>

           @isset($errors)
             @if($errors->any())
               @foreach($errors->all() as $error)
                 <div class="alert alert-danger"> {{ $error }} </div>
               @endforeach
             @endif
           @endisset
           <form action="{{ (isset($post))? asset('/posts/update/'.$post->postId)
: asset('/posts/store') }}" method="POST" enctype="multipart/form-data">
               @csrf
               @method('PUT')
             <div class="form-group">
               <label>Title:</label>
               <input type="text" name="title" class="form-control" value="{{
(isset($post))? $post->naslov : old('naslov') }}"/>
             </div>
             <div class="form-group">
               <label>Body:</label>
               <textarea name="body" class="form-control" rows="7">{{
(isset($post))? $post->sadrzaj : old('sadrzaj') }}</textarea>
             </div>
               <div class="form-group">
                   <div>
                       <label>Color:</label>
                       <div class="">
                           <select class="form-control" name="pboje">
                               <option value="">Choose...</option>
                               @foreach($colors as $c)
                                   <option value="{{ $c->ColorId}}" {{
(isset($post) && $post->ColorId == $c->ColorId)? 'selected' :
((old('pboje')==$c->ColorId)? 'selected' : '' )}} > {{ $c->Name }} </option>
                               @endforeach
                           </select>
                       </div>
                   </div>

                   <div>
                       <label>Category:</label>
                       <div class="">
                           <select class="form-control" name="pkategorije">
                               <option value="">Choose...</option>
                               @foreach($categories as $t)
                                   <option value="{{ $t->CategoryId}}" {{
(isset($post) && $post->CategoryId == $t->CategoryId)? 'selected' :
((old('pkategorije')==$t->CategoryId)? 'selected' : '' )}} > {{ $t->Name }}
</option>
                               @endforeach
                           </select>
                       </div>
                   </div>
               </div>
               <div class="form-group">
                   <label>Price:</label>
                   <input type="text" name="cena" class="form-control" value="{{
(isset($post))? $post->price : old('price') }}"/>
               </div>
               @isset($post)
               <div class="form-group">
                   <label>Discount: </label>
                   <input type="text" name="popust" class="form-control" value="{{
(isset($post))? $post->popust : old('price') }}"/>
               </div>
               @endisset
             <div class="form-group">
               <label>Photo:
                   @isset($post)
                   @endisset
               </label>
               @isset($post)
               <img src="{{ asset($post->putanja) }}" width="35" height="35"/>
               @endisset
               <input type="file" name="photo" class="form-control"  />
             </div>
             <div class="form-group">
               <input type="submit" name="addPost" value="{{ (isset($post))?
'Change post' : 'Add post' }}" class="btn btn-default" />
             </div>
           </form>

           <table class="table">
               <tr>
                 <td>ID</td>
                 <td>Username</td>
                 <td>Title</td>
                 <td>Photo</td>
                   <td>Price</td>
                 <td>Create/Update Date</td>
                 <td>Role</td>
                 <td>Change</td>
                 <td>Delete</td>
               </tr>
               <!-- Prikaz postova-->
               @isset($postovi)
               @if(session()->get('user')[0]->naziv == 'admin')
               @foreach($postovi as $post)
                 <tr>
                   <td> {{ $post->postId }} </td>
                   <td> {{ $post->korisnicko_ime }} </td>
                   <td> {{ $post->naslov }} </td>
                   <td> <img src="{{ asset($post->putanja) }}" width="50"
height="50" /> </td>
                     <td> {{ $post->price }} </td>
                   <td> {{ date("d.m.Y. H:i:s", $post->create)
}}</br>@if($post->update!=null){{ date("d.m.Y. H:i:s", $post->update) }}@endif
</td>
                   <td> {{ $post->naziv }} </td>
                   <td> <a href="{{ asset('/posts/'.$post->postId) }}">Change</a>
</td>
                   <td> <a href="{{ asset('/posts/destroy/'.$post->postId)
}}">Delete</a> </td>
                 </tr>
               @endforeach
               @endif
               @if(session()->get('user')[0]->naziv != 'admin')
               @foreach($postovi->where("korisnik_id", "=",
session()->get('user')[0]->id) as $post)
                 <tr>
                   <td> {{ $post->postId }} </td>
                   <td> {{ $post->korisnicko_ime }} </td>
                   <td> {{ $post->naslov }} </td>
                   <td> <img src="{{ asset($post->putanja) }}" width="50"
height="50" /> </td>
                     <td> {{ $post->price }} </td>
                   <td> {{ date("d.m.Y. H:i:s", $post->create)
}}</br>@if($post->update!=null){{ date("d.m.Y. H:i:s", $post->update) }}@endif
</td>
                   <td> {{ $post->naziv }} </td>
                   <td> <a href="{{ asset('/posts/'.$post->postId) }}">Change</a>
```

```
</td>
                  <td> <a href="{{ asset('/posts/destroy/'.$post->postId)
}}">Delete</a> </td>
                </tr>
              @endforeach
              @endif
              @endisset
          </table>

      </div>
    <!--// Sadrzaj -->
@endsection
```

4.3.10 Galerija.blade.php

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

@section('title')
   Galerija
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->

              <div class="col-md-8" id="rezultati">
          <div class="gallery-top heading"><br>
              <h3>Our Gallery</h3><br>
          </div>
              <div class="card mb-4"></div>
          <div class="container">
          <div class="row">
          @isset($posts)
          @foreach($posts as $post)
          <div class="col-sm-3" >
          <a class="image-popup-no-margins" href="{{ asset('/').$post->putanja
}}"><img src="{{ asset('/').$post->putanja }}" class="img img-thumbnail"></a>
          <br>
          <center><strong>{{ $post->naslov }}</strong></center>
     </div>
          @endforeach
          @endisset
          </div>
          </div>
              <div class="card mb-4"></div>
      </div>
@endsection
```

4.3.11 Home.blade.php

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

@section('title')
   Home page
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">

         <h1 class="my-4">Our Flowers</h1>
         @isset($users)
         {{-- var_dump($cart) --}}
               <div class="card mb-4"><div class="container">
                       <div class="row">
         @foreach($users as $post)
           {{-- var_dump($post) --}}
         <!-- Blog Post --><div class="col-sm-3" >

           <img class="card-img-top" src="{{ asset('/').$post->putanja }}">
           <div class="card-body">
             <h6 class="card-title">{{ $post->naslov }}</h6>
             <p class="card-text">
                 {{ $post->price }}
             </p>
               @if($post->popust!=Null && $post->popust!=0)
               <p class="card-text">
                   <p class="text-success">Discount {{ $post->popust}}%<p>
               </p>
               @endif
               <a href="#post/{{$post->postId}}" class="btn btn-primary vise"
onclick="showpost({{ $post->postId }})">Read More &rarr;</a>
               <div class="btn-group">
                   <button onclick="addToCart({{$post->postId}})" type="button"
class="btn btn-secondary">Add To Cart</button>
               </div>
           </div>
           <div class="card-footer text-muted">
             Posted on {{ date("d.m.Y. H:i:s", $post->create) }} by
             <a href="#">{{ $post->korisnicko_ime }}</a>
           </div>

     <!--// Blog Post --></div>
               @endforeach
               @endisset
                       </div>
                   </div>             </div>
         <!-- Pagination -->
        <ul class="pagination justify-content-center mb-4">
```

```
          <li class="page-item">
           {{ $users->appends(['search' => request('search')])->links() }}
         </li>
        </ul>
       </div>
     <!--// Sadrzaj -->
@endsection
```

4.3.12 Logovi.blade.php

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

@section('title')
   Logovi
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">
           <h3>Log Activity Lists</h3>

           @empty(!session('message'))
             {{ session('message') }}
           @endempty

           @isset($errors)
             @if($errors->any())
               @foreach($errors->all() as $error)
                 <div class="alert alert-danger"> {{ $error }} </div>
               @endforeach
             @endif
           @endisset

            <br />
           <div class="row input-daterange">
               <form action="{{ asset('/flogs') }}" method="GET">
                   <label>Date from:</label>
                   <input type="date" id="from_date" name="fromdate">
                   <label>Date to:</label>
                   <input type="date" id="to_date" name="todate">
                   <input type="submit" class="btn btn-primary" value="Filter">
               </form>
           </div>
           <br />

           <table class="table table-bordered">
     <tr>
        <th>Id</th>
        <th>User</th>
        <th>Action</th>
        <th>Date</th>
     </tr>
           @isset($logovi)
           @foreach($logovi as $log)
        <tr>
           <td>{{ $log->id }}</td>
           <td>{{ $log->user }}</td>
           <td class="text-success">{{ $log->action }}</td>
               <td class="text-success">{{ date("d.m.Y. H:i:s", $log->time)
}}</td>
        </tr>
           @endforeach
           @endisset

  </table>
       </div>
     <!--// Sadrzaj -->
@endsection
```

4.3.13 Orders.blade.php

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

@section('title')
   Contact
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
   <!-- Sadrzaj -->
   <div class="col-md-8">
       <br><br>
       <table class="table table-bordered table-striped table-condensed">
           <tr>
               <th>OrderNum</th>
               <th>OrderDate</th>
               <th>Products</th>
               <th>Address</th>
               <th>Total</th>
           </tr>
           @php
               $total = 0;
           @endphp
           @foreach($orders as $o)
               <tr>
                   <td>{{ $o->InvoiceId }}</td>
                   <td>{{ $o->InvoiceDate }}</td>
                   <td>
                       @foreach($o->invoiceLines as $il)
                           {{ $il->Quantity." ".$il->post->naslov . "," }}
                       @endforeach
                   </td>
                   @if(session()->get('user')[0]->naziv == 'admin')
```

```
                    <td>
                        {{ $o->ImePrezime }}
                        <br>{{ $o->BillingAddress }}
                        <br>{{ $o->BillingCity }}
                        <br>{{ $o->BillingState }}
                        <br>{{ $o->BillingPostalCode }}
                        <br>{{ $o->Phone }}
                        <br>{{ $o->email }}
                    </td>
                    @endif
                    <td>
                        {{ $o->Total }}
                        @php $total += $o->Total; @endphp
                    </td>
                    @if(session()->has('user'))
                        @if(session()->get('user')[0]->naziv == 'admin')
                    <td> <a href="{{ asset('/order/destroy/'.$o->InvoiceId) }}"
class="btn btn-danger">Delete</a>
                        <br><br>
                     <a href="{{ asset('/order/send/'.$o->InvoiceId) }}" class="btn
btn-success">Send</a> </td>
                        @endif
                        @endif
                </tr>
            @endforeach
        </table>

        <p class="lead">Total amount: {{ $total }}$</p>

   </div>
   <!--// Sadrzaj -->
@endsection
```

4.3.14 Post.blade.php

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

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection
@section('content')

@isset($singlePost)
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">


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

         <!-- Author -->
         <p class="lead">
           by
           <a href="#">{{ $singlePost->postKorisnik }}</a>
         </p>
         <hr>

         <!-- Date/Time -->
         <p>Posted on {{ date("d.m.Y. H:i:s", $singlePost->create) }}</p>

         <hr>

         <!-- Preview Image -->
         <img class="img-fluid rounded" src="{{
asset('/'.$singlePost->putanja)}}">

         <hr>

         <!-- Post Content -->
         <p> {{ $singlePost->sadrzaj }}</p>

       <hr>
           <p> {{ $singlePost->price }}</p>

           <hr>
           @if($singlePost->popust!=Null && $singlePost->popust!=0)
               <p class="card-text">
               <p class="text-success">Discount {{ $singlePost->popust}}%<p>
               </p>
           @endif
         <!-- Comments Form -->
         @if(session()->get('user'))


         <div class="card my-4">
           <h5 class="card-header">Leave a Comment:</h5>
           <div class="card-body">
               <form action="#" method="get">
                   {{ csrf_field() }}
                   <div class="form-group">
                       <textarea class="form-control" id="psalji" rows="3"
name="content"></textarea>
                   </div>
                   <button name="akcija" value="post"
onclick="salji({{$singlePost->postId}})" class="btn btn-primary
salji">Submit</button>
               </form>

               <br>
               @if(session('success'))
                   <div class="alert alert-success">
                   {{ session('success') }}
                   </div>
               @endif
               @if(session('warning'))
                   <div class="alert alert-warning">
                   {{ session('warning') }}
                   </div>
               @endif
           </div>
         </div>
         @else
         <h4>Da bi komentarisali morate da se ulogujete.</h4>
         @endif
         <div hidden>
         <form id="pform" action="{{ route("editComment", ['postId' =>
$singlePost->postId],['commentId' =>"2"]) }}" method="post">
                 {{ csrf_field() }}
               <div class="form-group">
                 <textarea class="form-control" id="txta" rows="3"
name="content1"></textarea>
               </div>
               <button type="submit" name="akcija" value="edit" class="btn
btn-primary">Update</button>
             </form>
         </div>
       <!--// Comments Form -->

         <!-- Single Comment -->

              @isset($comments)
               @include('components.comments')
              @endisset



     <!--// Single Comment -->

       </div>
     <!--// Sadrzaj -->
@endisset
@endsection
```

4.3.15 Register.blade.php

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

@section('title')
   Korisnik create
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">
           <h3>Register</h3>

           @empty(!session('message'))
             {{ session('message') }}
           @endempty

           @isset($errors)
```

```
             @if($errors->any())
               @foreach($errors->all() as $error)
                 <div class="alert alert-danger"> {{ $error }} </div>
               @endforeach
             @endif
           @endisset

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

             {{ csrf_field() }}

             <div class="form-group">
               <label>Username:</label>
               <input type="text" name="korisnickoIme" class="form-control"
value="{{ (isset($korisnik))? $korisnik->korisnicko_ime : old('korisnickoIme')
}}"/>
             </div>
             <div class="form-group">
               <label>Password:</label>
               <input type="password" name="lozinka" class="form-control"
value="{{ (isset($korisnik))? $korisnik->lozinka : old('lozinka') }}"/>
             </div>
               <div class="form-group">
                   <label>Name and Surname:</label>
                   <input type="text" name="imeprezime" class="form-control"
value="{{ (isset($korisnik))? $korisnik->ImePrezime : old('imeprezime') }}"/>
               </div>
               <div class="form-group">
                   <label>Address:</label>
                   <input type="text" name="adresa" class="form-control" value="{{
(isset($korisnik))? $korisnik->BillingAddress : old('adresa') }}"/>
               </div>
               <div class="form-group">
                   <label>City:</label>
                   <input type="text" name="grad" class="form-control" value="{{
(isset($korisnik))? $korisnik->BillingCity : old('grad') }}"/>
               </div>
               <div class="form-group">
                   <label>State:</label>
                   <input type="text" name="opstina" class="form-control"
value="{{ (isset($korisnik))? $korisnik->BillingState : old('opstina') }}"/>
               </div>
               <div class="form-group">
                   <label>Postal Code:</label>
                   <input type="text" name="postanski" class="form-control"
value="{{ (isset($korisnik))? $korisnik->BillingPostalCode : old('postanski')
}}"/>
               </div>
               <div class="form-group">
                   <label>Phone:</label>
                   <input type="text" name="telefon" class="form-control"
value="{{ (isset($korisnik))? $korisnik->Phone : old('telefon') }}"/>
               </div>
               <div class="form-group">
                   <label>Email:</label>
```

```
                    <input type="text" name="email" class="form-control" value="{{
(isset($korisnik))? $korisnik->email : old('email') }}"/>
                </div>
              <div class="form-group">
                <label>Photo:</label>


            @isset($korisnik)
                  <img src="{{ asset($korisnik->slika) }}" width="35" height="35"
style="border-radius: 500px; -moz-border-radius: 500px; margin-top: 2px;"/>
                @endisset


                <input type="file" name="slika" class="form-control"  />

              </div>
              <div class="form-group">
                <input type="submit" name="addKorisnik" value="Register" class="btn
btn-default" />
              </div>
            </form>
        </div>
      <!--// Sadrzaj -->
@endsection
```

4.3.16 Search.blade.php

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

@section('title')
   Home page
@endsection

@section('appendCss')
   @parent
   <!-- Custom styles for this template -->
   <link href="{{ asset('/') }}css/blog-home.css" rel="stylesheet"/>
@endsection

@section('content')
<!-- Sadrzaj -->
       <div class="col-md-8" id="rezultati">

         <h1 class="my-4">Film Artikli

         </h1>

         @isset($users)
         {{-- var_dump($posts) --}}
               <div class="card mb-4"><div class="container">
                       <div class="row">
         @foreach($users as $post)
           {{-- var_dump($post) --}}
         <!-- Blog Post --><div class="col-sm-3" >
           <img class="card-img-top" src="{{ asset('/').$post->putanja }}">
           <div class="card-body">
```

```
             <h6 class="card-title">{{ $post->naslov }}</h6>
             <p class="card-text">
                 {{ $post->price }}
             </p>
             <a href="{{ asset('/post/'.$post->postId )}}" class="btn
btn-primary">Read More &rarr;</a>
               <div class="btn-group">
                   <button onclick="addToCart({{$post->postId}})" type="button"
class="btn btn-secondary">Add To Cart</button>
               </div>
           </div>
           <div class="card-footer text-muted">
             Posted on {{ date("d.m.Y. H:i:s", $post->create) }} by
             <a href="#">{{ $post->korisnicko_ime }}</a>
           </div>
               </div>
     <!--// Blog Post -->
               @endforeach
               @endisset
                       </div>
                   </div>             </div>
         <!-- Pagination -->
         <ul class="pagination justify-content-center mb-4">
           <li class="page-item">
             {{ $users->appends(['search' => request('search')])->links() }}
           </li>
         </ul>
       </div>
     <!--// Sadrzaj -->
@endsection
```

4.4 Routes

4.4.1 Web.php

```
<?php
use App\Http\Controllers\FrontendController;
use App\Http\Controllers\KorisnikController;
use App\Http\Controllers\DateRangeController;
use App\Http\Controllers\ContactUsController;
use App\Http\Controllers\PostController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\CommentsController;
use App\Http\Controllers\CartController;
use Illuminate\Support\Facades\Route;

Route::get('/',[FrontendController::class, 'index']);
Route::get('/gallery',[FrontendController::class, 'galerija']);
Route::get('/searchjs', [FrontendController::class,
'searchjs'])->name('searchjs');

Route::group(['middleware' => 'admin'], function() {

   Route::get('/order/destroy/{id}',[CartController::class, 'destroy']);

   Route::get('/order/send/{id}',[CartController::class, 'send'])->name('salji');
   // Upravljanje korisnicima


   Route::post('/users/store', [KorisnikController::class, 'store']);
   Route::get('/logs', [FrontendController::class, 'showlogs']);
   Route::get('/flogs', [DateRangeController::class, 'index']);

   Route::get('/users/destroy/{id}',[KorisnikController::class, 'destroy']);


});

Route::group(['middleware' => 'user'], function() {
   Route::get('/users/{id?}', [KorisnikController::class, 'show']);
   Route::post('/users/update/{id}',[KorisnikController::class, 'update']);
   Route::get('/posts/{id?}', [PostController::class, 'postshow']);
   Route::put('/posts/store', [PostController::class, 'store']);
   Route::put('/posts/update/{id}',[PostController::class, 'update']);
   Route::get('/posts/destroy/{id}',[PostController::class, 'destroy']);

   /*
   Rute za upravljanje komentarima
*/
   Route::get("/commentsjs/{sid}/{keyword}", [CommentsController::class,
'postCommentjs'])->name("postCommentjs");
   Route::post("/comments/{postId}", [CommentsController::class,
'postComment'])->name("postComment");
   Route::post("/post/{postId}/comments/{commentId?}", [CommentsController::class,
'editComment'])->name("editComment");
   Route::get("/comments/{commentId}/delete", [CommentsController::class,
'deleteComment'])->name("deleteComment");
   Route::get('/logout', [LoginController::class, 'logout'])->name('logout');
});
Route::get("/search", [\App\Http\Controllers\FrontendController::class,
"trazi"])->name("searchs");

Route::post("/cart", [\App\Http\Controllers\CartController::class, "index"]);
Route::get("/cart", [\App\Http\Controllers\CartController::class,
"get"])->name("get-cart");
Route::post("/cart/adjustQuantity", [\App\Http\Controllers\CartController::class,
"adjustQuantity"]);
Route::delete("/cart/{postId}", [\App\Http\Controllers\CartController::class,
"remove"]);
Route::post("/checkout", [\App\Http\Controllers\CheckoutController::class,
"index"]);
Route::get("/my-orders", [\App\Http\Controllers\CartController::class,
"myOrders"])->name("my-orders");

Route::get('contact-us',[ContactUsController::class, 'index']);
Route::post('contact-us',[ContactUsController::class, 'handleForm']);




// Logovanje

Route::post('/login', [LoginController::class, 'login'])->name('login');
Route::get('/register', [KorisnikController::class, 'regshow']);
Route::post('/register/store', [KorisnikController::class, 'regstore']);

Route::get('/postjs/{id}',[FrontendController::class, 'getPostjs']);

Route::get('/post/{id}',[FrontendController::class, 'getPost']);
```

4.5 Middleware

4.5.1 CheckRole.php

```
<?php
namespace App\Http\Middleware;
use Closure;
class CheckRole
{
   /**
    * Handle an incoming request.
    *
    * @param  \Illuminate\Http\Request  $request
    * @param  \Closure  $next
    * @return mixed
    */
   public function handle($request, Closure $next)
   {
       if($request->session()->has('user')){

               return $next($request);
       }
       return redirect()->back()->with('error','You dont have access to this
page!');
   }
}
```

4.5.2 CheckUserRole.php

```
<?php
namespace App\Http\Middleware;
use Closure;
class CheckUserRole
{
   /**
    * Handle an incoming request.
    *
    * @param  \Illuminate\Http\Request  $request
    * @param  \Closure  $next
    * @return mixed
    */
   public function handle($request, Closure $next)
   {
       if($request->session()->has('user')){
           $korisnik = $request->session()->get('user')[0];
           if($korisnik->naziv == 'admin'){
               return $next($request);
           } else {
               return redirect()->back()->with('error','You dont have access to
this page!');
           }
       }

       return redirect()->back()->with('error','You dont have access to this
page!');
   }
}
```

4.6 JavaScript

4.6.1 ajax.js

```
function editComment(id,pid) {
       var content=$('#kom'+id).text();
       document.getElementById("txta").value=content;
       var url=document.getElementById("pform").action;
       var res = url.replace(pid, id);
       var form = document.getElementById('pform');
       form.action=res;
       $.ajax({
           method : "get",
           processData: false, contentType: false,
           data: {
               'id' : id,
               'content' : content,
           },
           type : 'json',
           success : function(data) {
               showComment(id,content,form);
           },
           error : function(error)
           {
               console.log("Something went wrong....");
           }
       });
}

function showComment(id,content,form) {
   var url="/editComment/" + id + "/edit";
   var editForm = form;
   $("#comment"+id).html(editForm);
}

function updateComment(id) {
   var editedData = $("#commentContent").val();
   $.ajaxSetup({
       headers: {
           'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
       }
   });
   $.ajax({
       "url" :"/comments/" + id + "/edit",
       method : "post",
       data : {
           'content' : editedData,
           '_method' : "post",
           'csrf-token' : X-CSRF-TOKEN
       },
       type : 'json',
       success : function(data) {
           hideAndReset(id, editedData);
       },
       error : function(error)
       {
           console.log("Something went wrong....");
       }
   });
}

function hideAndReset(id, editedData)
{
   console.log(editedData);
   $("#comment" + id).html(editedData);
   $("#comments").html("<div class='alert alert-info'>Comment successfully
edited!</div>");
}

$(".trazi").click(function(event){
   event.preventDefault();
   var keyword = $('#search').val();
   _token: $('meta[name="csrf-token"]').attr('content'),

   $.ajax({
       url: "/searchjs",
       type:"GET",
       dataType: "json",
       data:{
           'keyword':keyword
       },
       success:function(data){
           $("#rezultati").html(data);
           console.log(data);
       },
       error: function(error) {
           console.log(error);
       }
   });
});

function salji(sid){
   event.preventDefault();
   var keyword = $('#psalji').val();
   _token: $('meta[name="csrf-token"]').attr('content'),

       $.ajax({
           url: "/commentsjs/" + sid+"/"+keyword,
           type:"GET",
           dataType: "json",
           data:{
               'keyword':keyword,
               'sid':sid
           },
           success:function(data){
               $("#rezultati").html(data);
               console.log(data);
           },
           error: function(error) {
               console.log(error);
           }
       });
};
```

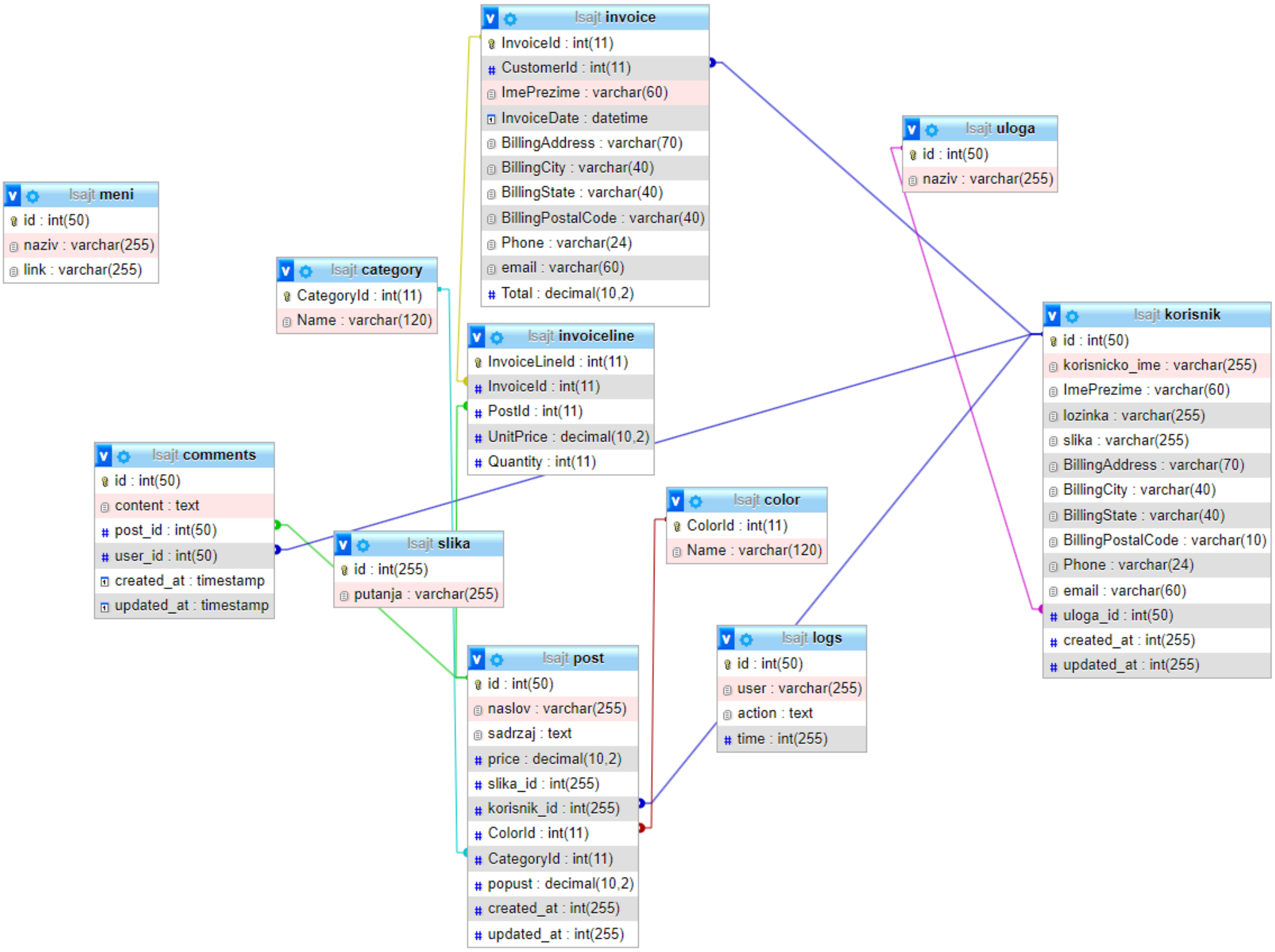
```
function showpost(id) {
   event.preventDefault();
       $.ajax({
           url: "/postjs/" + id,
           type:"GET",
           dataType: "json",
           data:{
               'id':id
           },
           success:function(data){
               $("#rezultati").html(data);
               console.log(data);
           },
           error: function(error) {
               console.log(error);
           }
   });
}
```

4.6.2 provera.js

```
function proveraKontakt(){

  var ime = document.getElementById("imeprezime").value;

  var regIme = /^[A-Za-z\d]{4,}$/;

  if(regIme.test(ime)){

     document.getElementById("imeprezime").style.borderColor = "green";

  }

  else {

     document.getElementById("imeprezime").style.borderColor = "red";

  }

  var password = document.getElementById("sifra").value;

  var regpassword = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{6,}$/;

  if(regpassword.test(password)){

     document.getElementById("sifra").style.borderColor = "green";

  }

  else {

     document.getElementById("sifra").style.borderColor = "red";

  }
}
```

5. Baza

Aplikacija koristi MySQL bazu podataka. Na slici 14, se može videti struktura tabela.



Slika 14 – baza

korisnik – sadrži podatke o registrovanim korisnicima
uloga– sadrži korisničke uloge,
logs – sadrži logove,
meni– sadrži podatke o meniju
comments-sadrži podatke o komentarima,
slika– sadrži podatke o slikama,
invoice– tabela koja sadrži informacije o korpi,
invoceline-tabela koja sadrži informacije o korpi,
color-sadrži podatke o bojama,
category – sadrži podatke o kategorijama,
uloga– sadrži podatke o ulogama,

6. Zaključak

Web aplikacija „Flower Shop“ za online kupovinu cveca korisnik je odmah na početnoj stranici pregledao proizvode iz ponude. Korisnik je filtrirao proizvode po kategorijama kako bi video šta je u ponudi od alata koje je tražio. Pretragu je izvšrio po ključnoj reči.Ukoliko je želeo da naruči neke proizvode, mogao se ulogovati ili registrovati ako nema nalog.Na stranici za korpu imao je uvid u proizvode koje je izabrao.Na svakom postu mogao je komentarisati.Na stranici contact, mogao je poslati upit, njegove poruke, pohvale i kritike.Administrator aplikacije je imao uvid u korisnike, osnovne informacije među kojima i datum registracije. Korisnike je mogao da ukloni ukoliko je za to imao potrebe ili da menja.Pošto postoji potreba dodavanja novih proizvoda i komentara, admin je dodavao, menjao i brisao komentare i proizvodeNa stranici za porudbine, pregledao je sve primljene. Tu je mogao da ih briše ili potvrdi.Takođe, ima uvid u logove.