

DOKUMENTACIJA

Student:

Aleksandra Marković 8/18

Sadržaj

1. Opis projekta	3
2. Slike stranica i opis funkcionalnosti	3
3. Dizajn baze podataka	12
4. Kodovi.....	12
\App\Http\Controllers	12
\App\Http\Middleware	24
\App\Http\Requests	26
\App\Models	31
\Public\Assets\Js	37
\Resources\Views\Admin	50
\Resources\Views\Fixed	54
\Resources\Views\Layouts	57
\Resources\Views\Pages\Main.....	58
\Resources\Views\Pages\Products	64
\Resources\Views\Pagination	69
\Resources\Views\Partials.....	70
\Routes.....	71

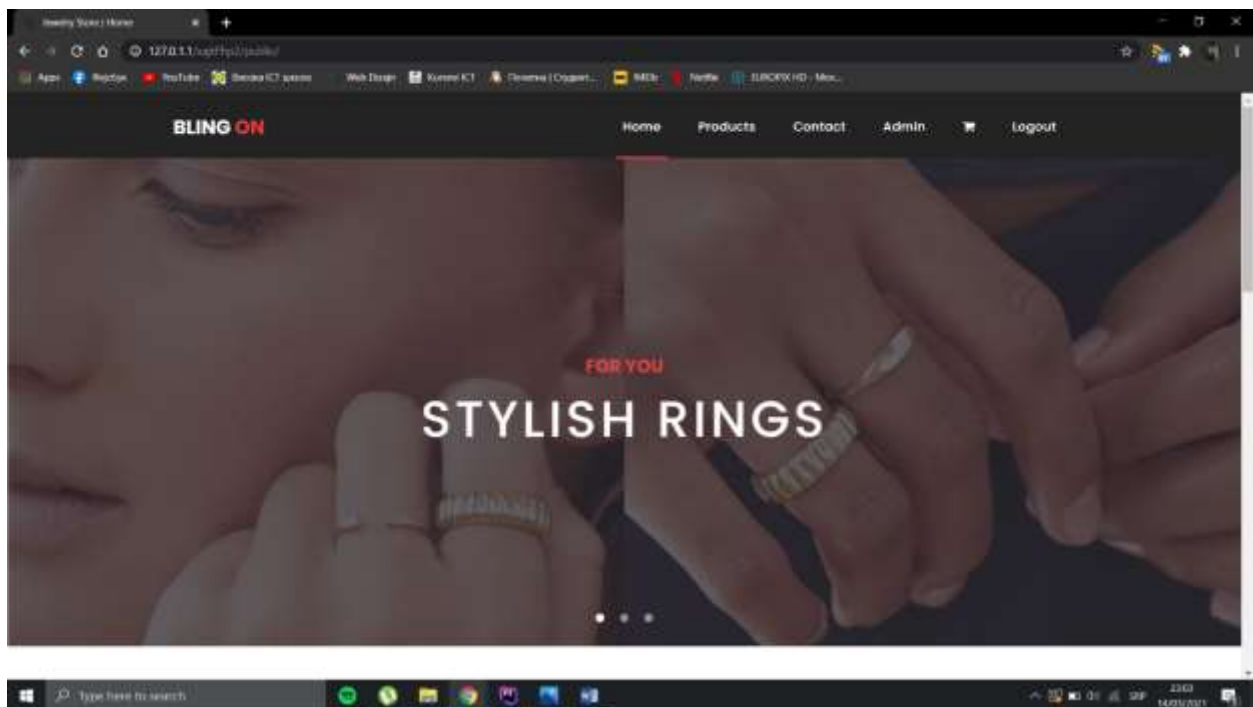
1. Opis projekta

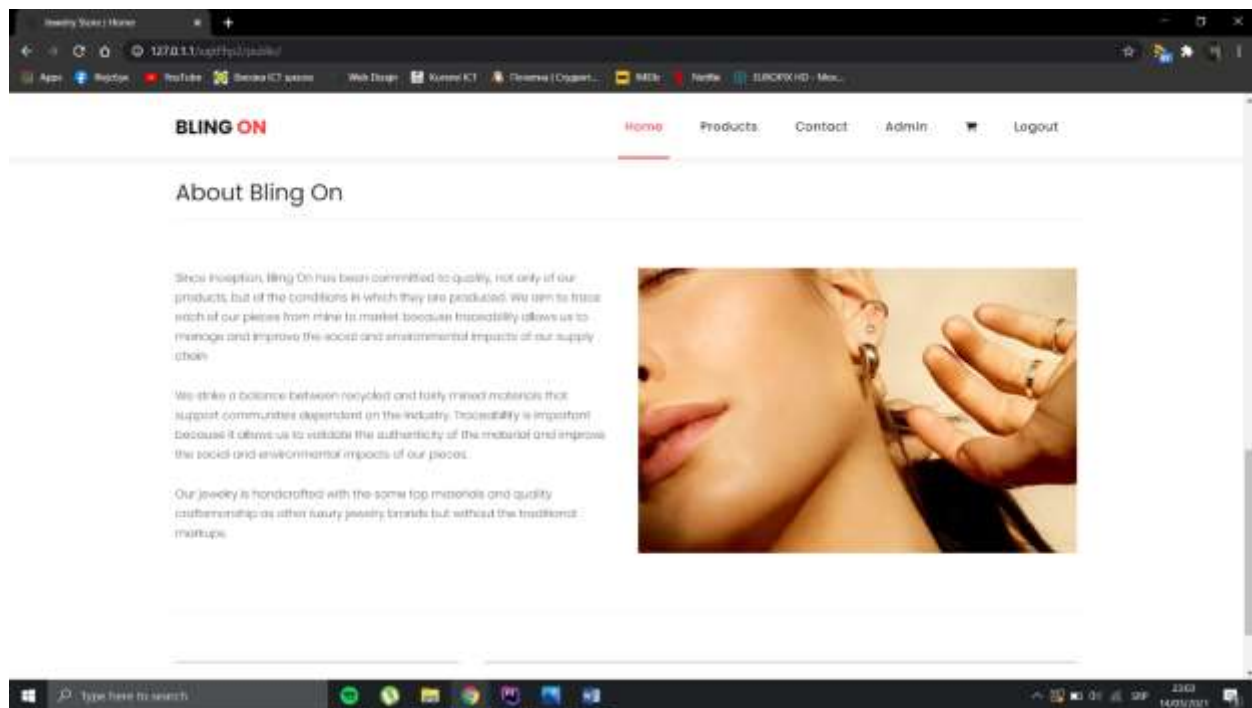
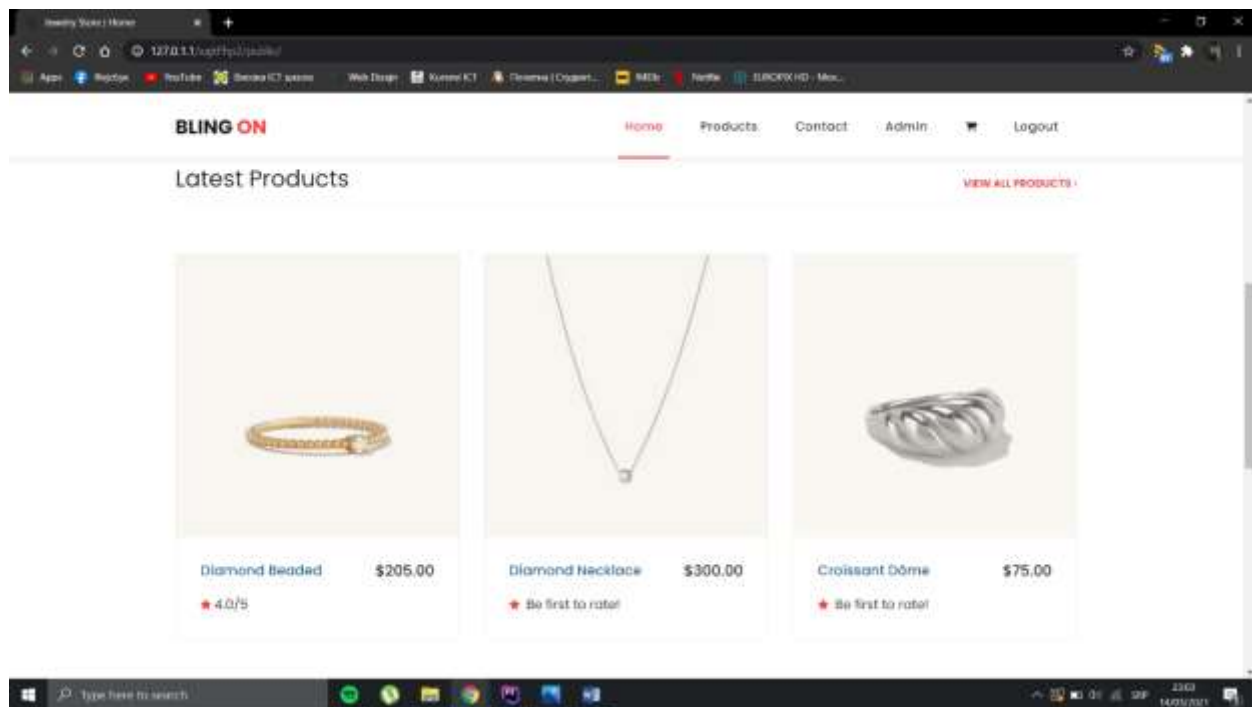
Sajt se bavi prodajom nakita. Svi korisnici mogu da vide proizvode ali ih samo autorizovani korisnici mogu oceniti, dodati u korpu i kupiti. Postoje različite boje, brendovi, tipovi i veličine proizvoda. Korisnici mogu kontaktirati admina pomoću forme koja se nalazi na kontakt stranici. Admin može dodati nove proizvode, izmeniti i obrisati postojeće, takođe može videti sve ključne aktivnosti korisnika na posebnoj stranici kojoj samo on ima pristup.

2. Slike stranica i opis funkcionalnosti

Meni i futer su fiksni. Sadržaj menija se menja na osnovu uloge korisnika koji je ulogovan.

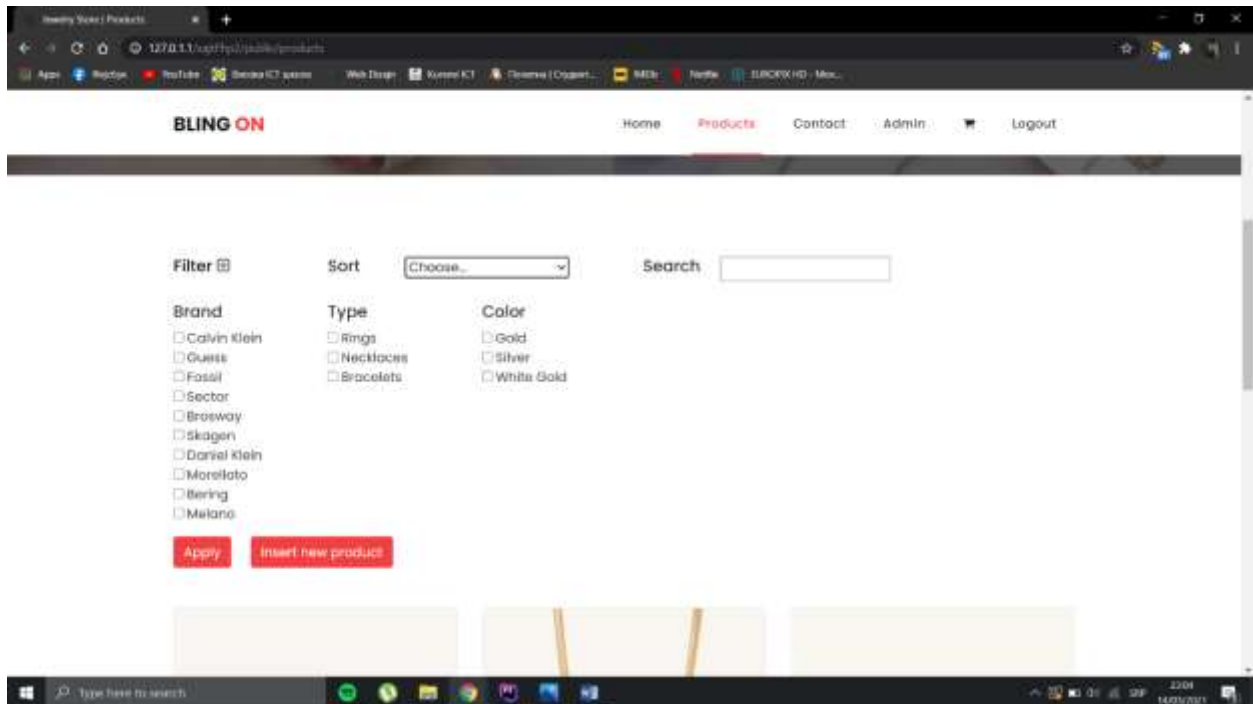
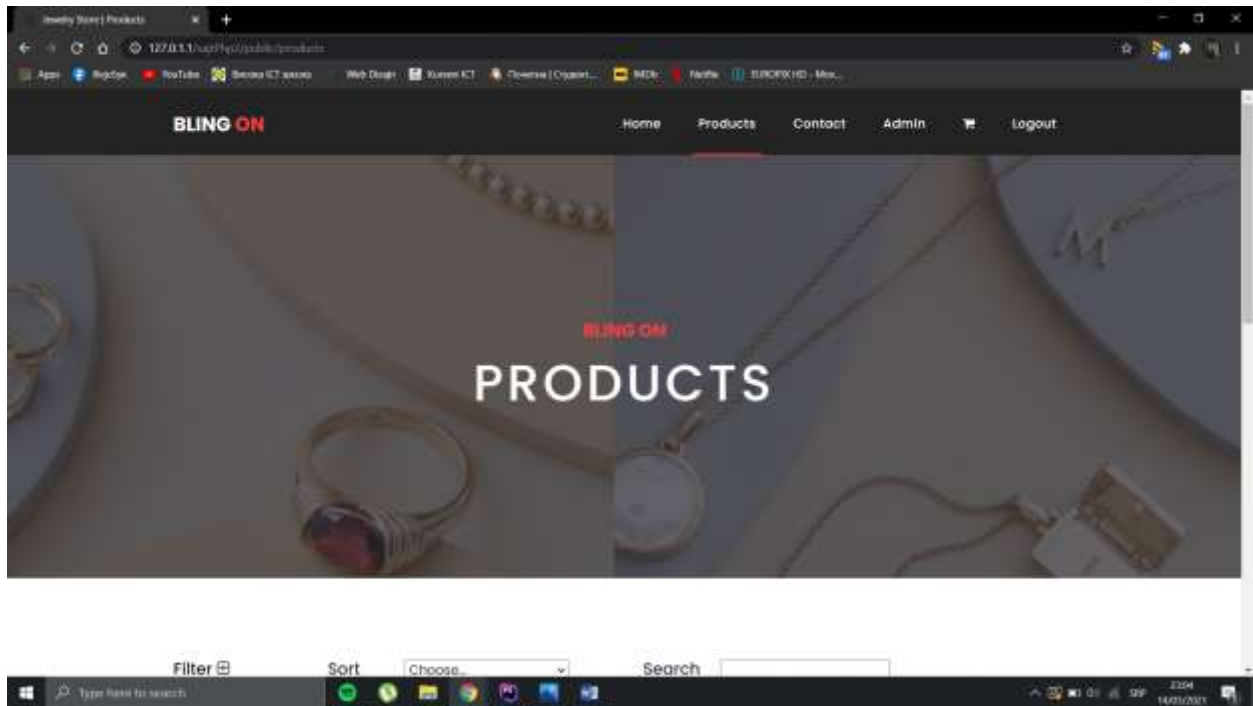
Home

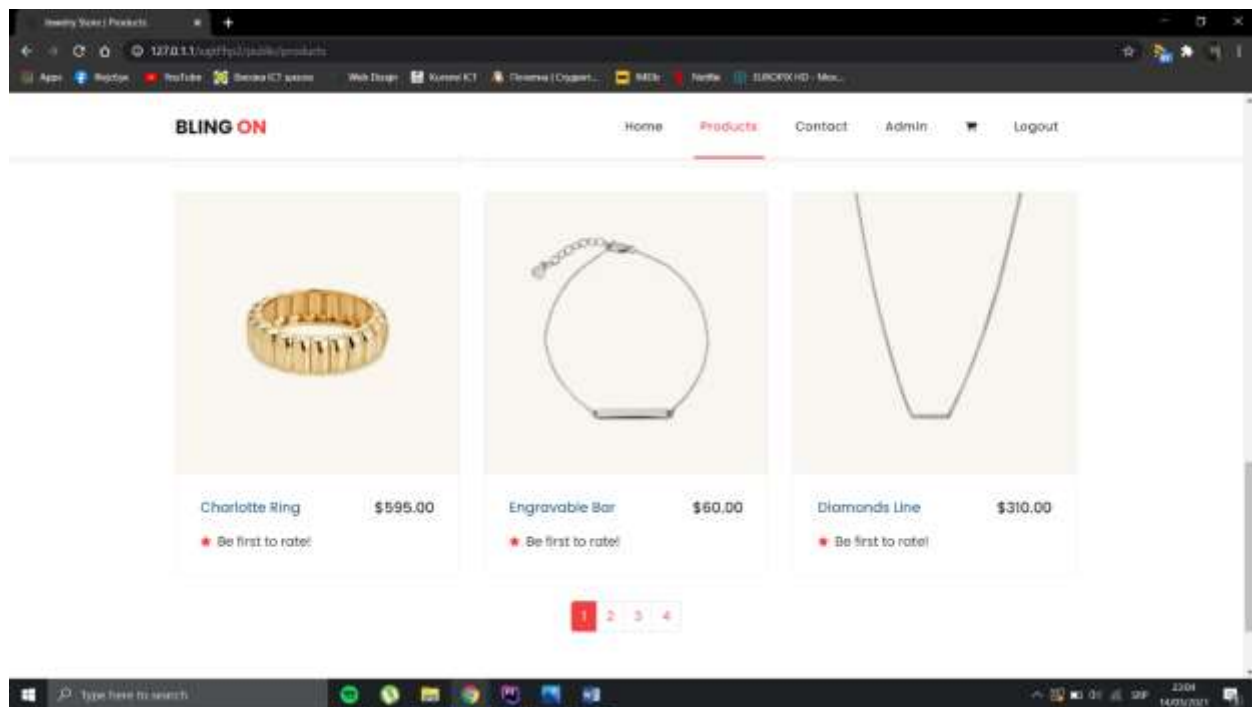




Na home stranici mogu se videti najnovija 3 proizvoda kao i kratak tekst o firmi.

Proizvodi

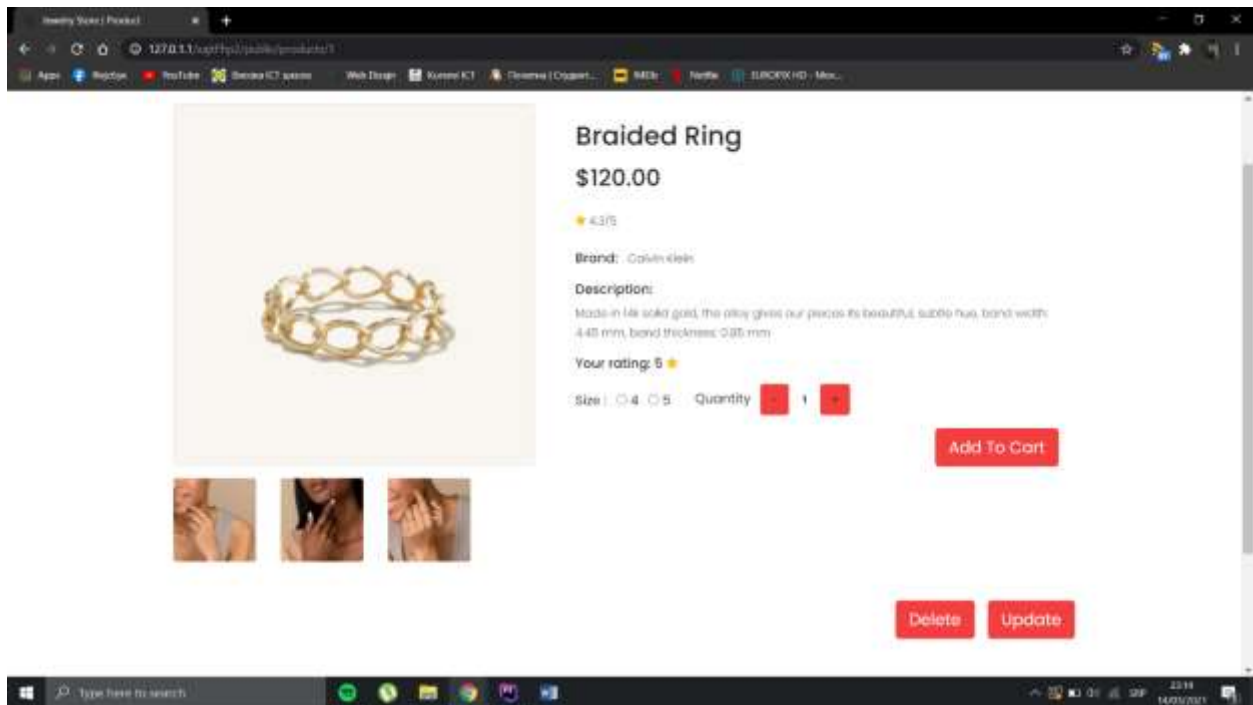




Napomena: Za sajt je korišćen gotov template koji ima bag i dešava se da se ova stranica raspadne ali nakon refreša se vrati u normalu.

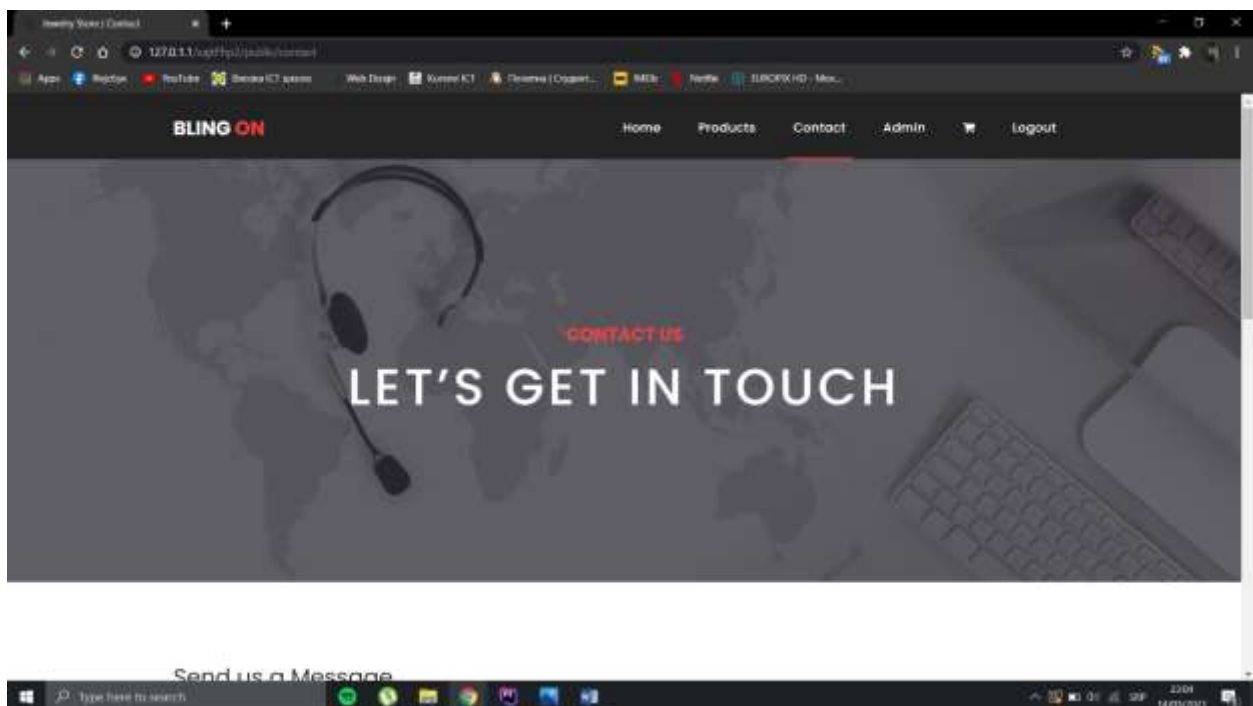
Na ovoj stranici mogu se videti svi proizvodi koji su na raspolaganju i oni se mogu pretraživati po imenu, filtrirati po brendu, boji i tipu i sortirati po ceni i i oceni.

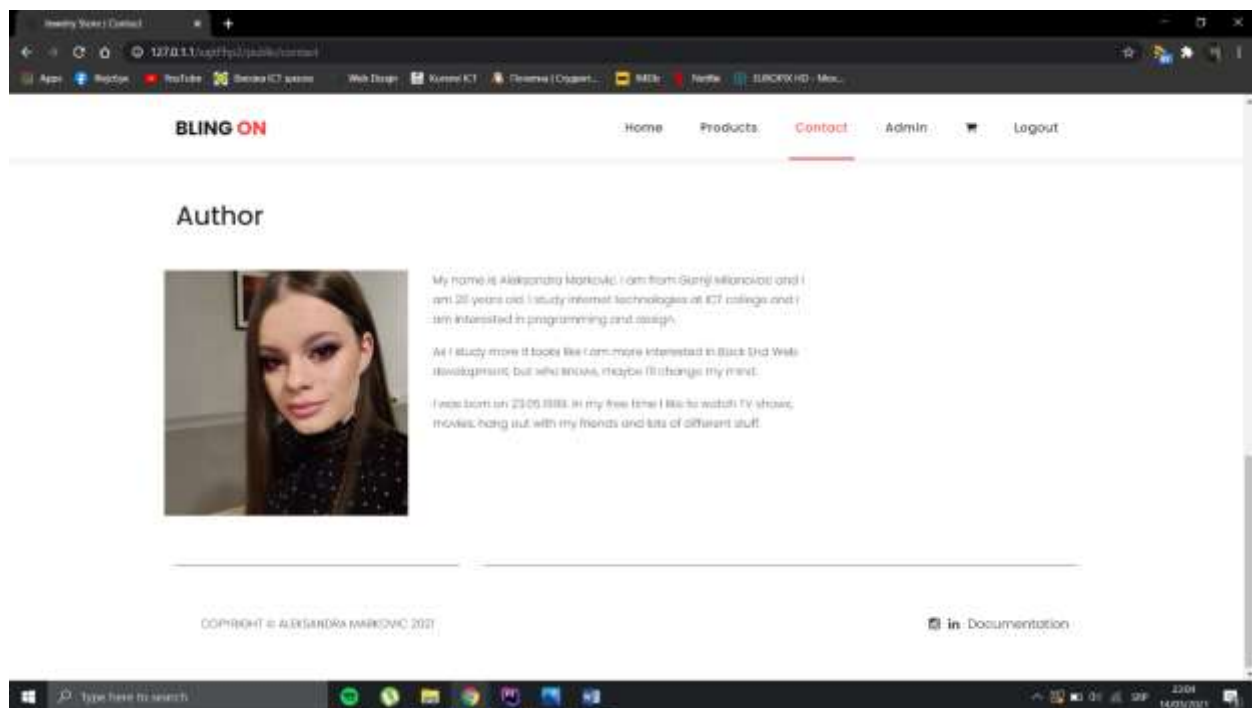
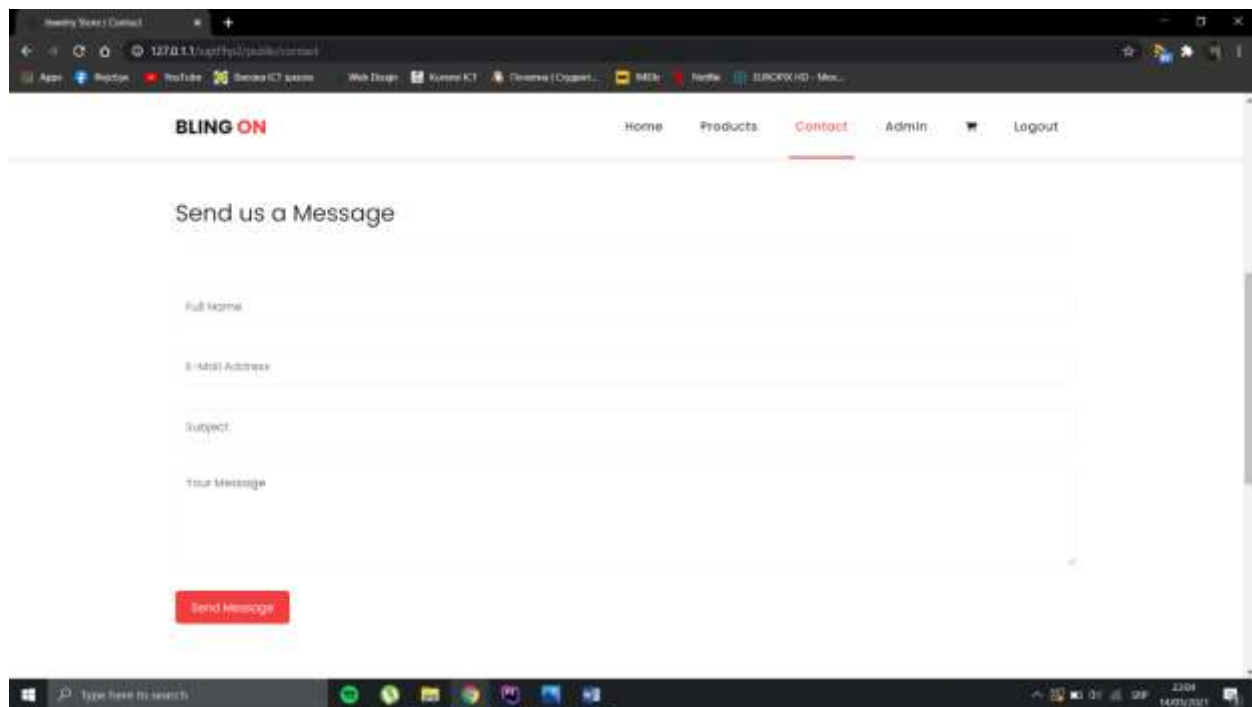
Dugme za insert novog proizvoda vidljivo je samo adminu.



Kada se klikne na ime ili sliku proizvoda odlazi se na pojedinačnu stranicu gde se mogu videti dodatne slike i informacije. Autorizovani korisnici mogu oceniti proizvod i dodati u korpu. Delete i update su vidljivi samo adminu.

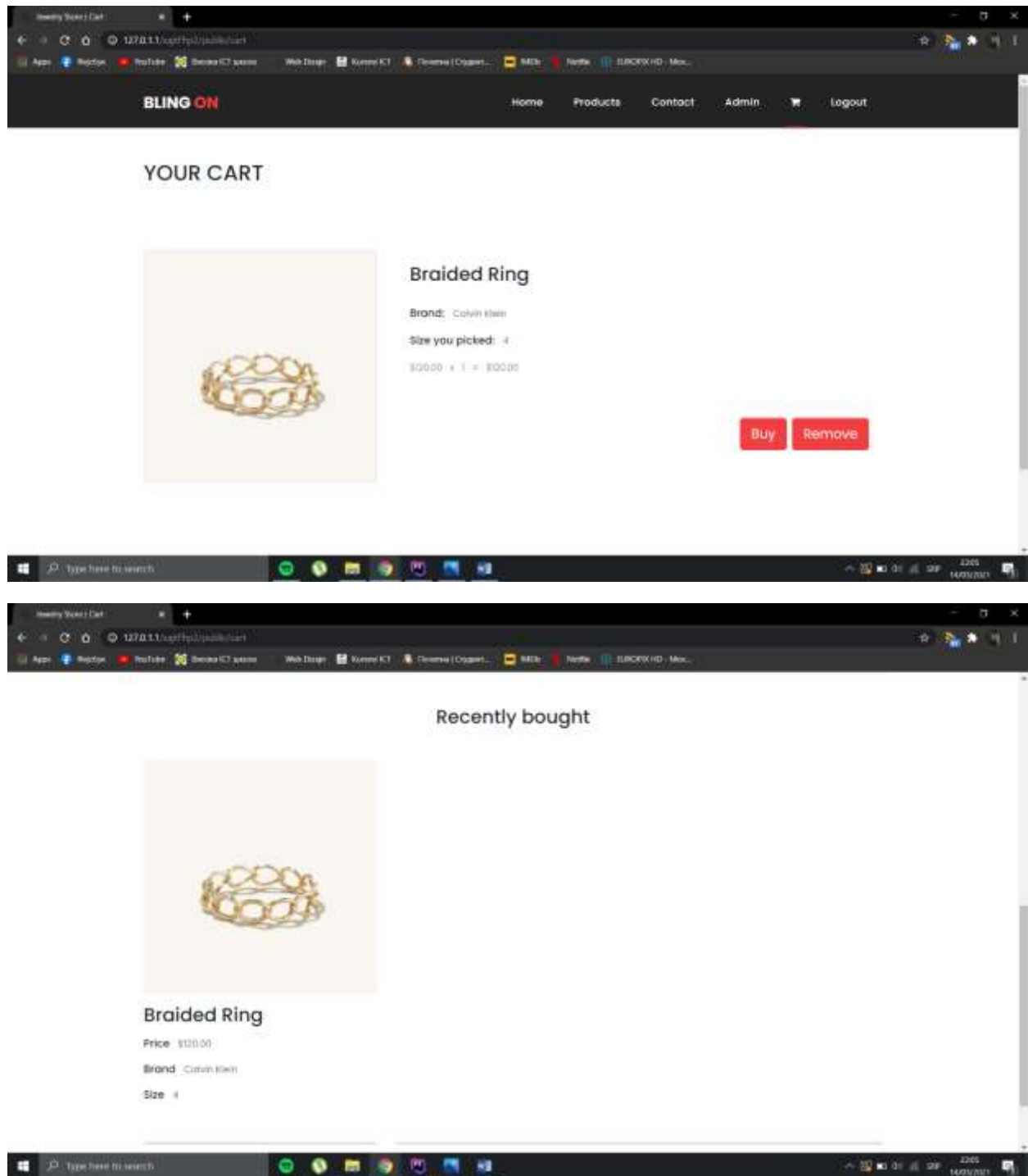
Kontakt stranica





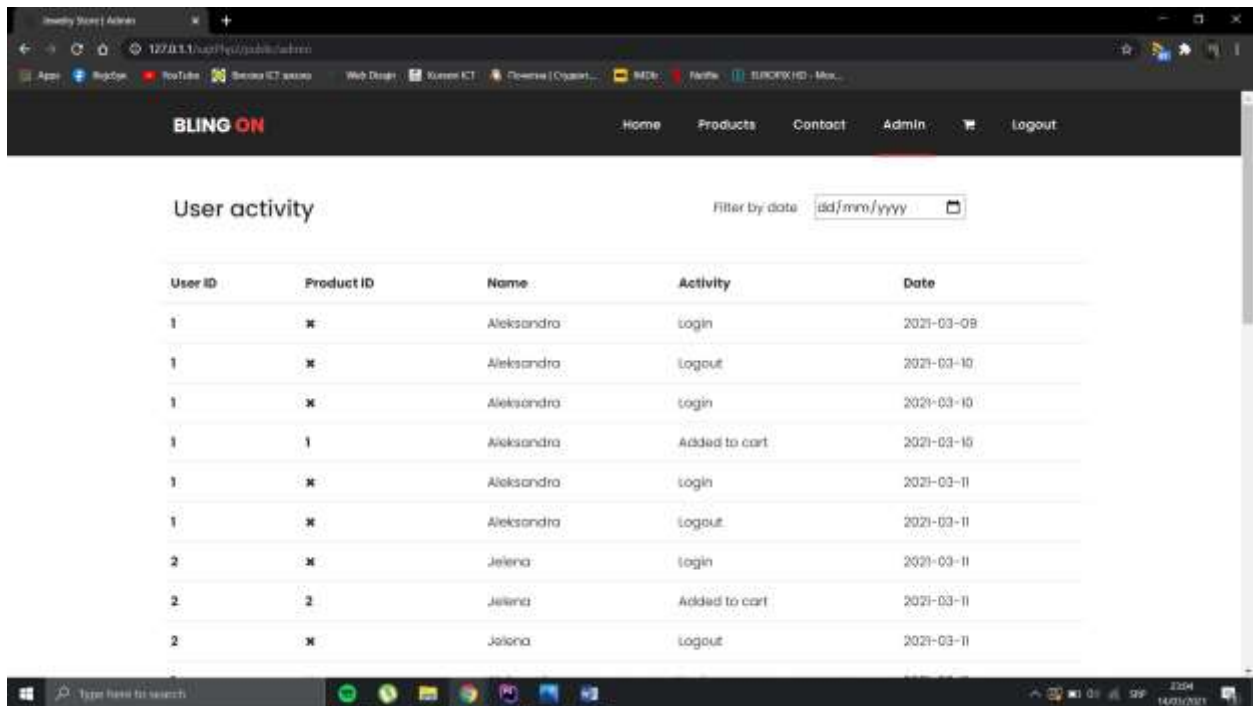
Na stranici kontakt nalazi se forma za kontaktiranje admina i podaci o autoru.

Korpa



U korpi se mogu videti svi dodati proizvodi i kupiti se. Na dnu korpe korisniku se prikazuju poslednja 3 kupljena proizvoda.

Admin



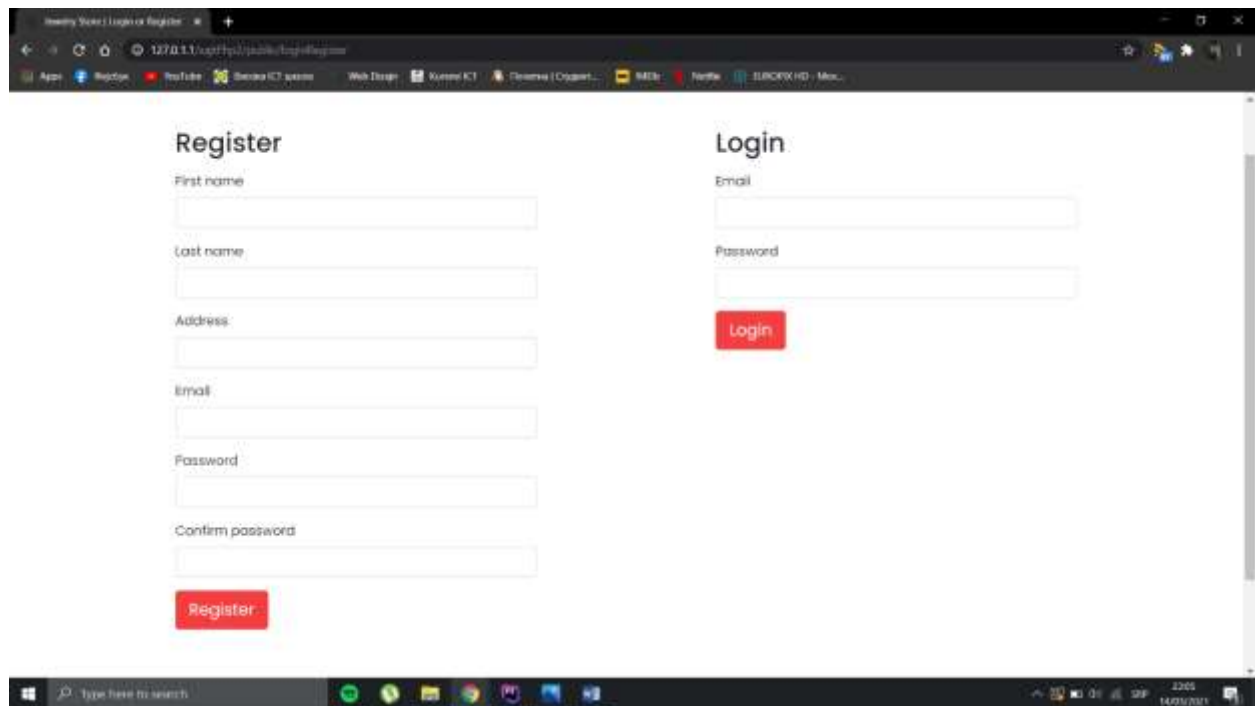
The screenshot shows a web browser window with the URL `127.0.0.1:5000/public/admin`. The page has a dark header with the 'BLING ON' logo and navigation links: Home, Products, Contact, Admin (active), and Logout. Below the header, the 'User activity' section features a date filter set to 'dd/mm/yyyy' and a table of user actions.

User ID	Product ID	Name	Activity	Date
1	*	Aleksandra	login	2021-03-09
1	*	Aleksandra	logout	2021-03-10
1	*	Aleksandra	login	2021-03-10
1	1	Aleksandra	Added to cart	2021-03-10
1	*	Aleksandra	login	2021-03-11
1	*	Aleksandra	logout	2021-03-11
2	*	Jelena	login	2021-03-11
2	2	Jelena	Added to cart	2021-03-11
2	*	Jelena	logout	2021-03-11

Na admin stranici kojoj ima pristup samo admin nalazi se tabela gde se prikazuju sve ključne aktivnosti korisnika.

Može se filtrirati po datumu.

Login i registracija



The screenshot shows a web browser window with a dark theme. The address bar displays the URL `127.0.0.1:54321/public/login/register`. The browser's tab bar shows several tabs, including 'Newly Booked / Login or Register'. The main content area is divided into two sections: 'Register' on the left and 'Login' on the right. The 'Register' section contains input fields for 'First name', 'Last name', 'Address', 'Email', 'Password', and 'Confirm password', followed by a red 'Register' button. The 'Login' section contains input fields for 'Email' and 'Password', followed by a red 'Login' button. The Windows taskbar at the bottom shows the search bar and several application icons.

Register

First name

Last name

Address

Email

Password

Confirm password

Register

Login

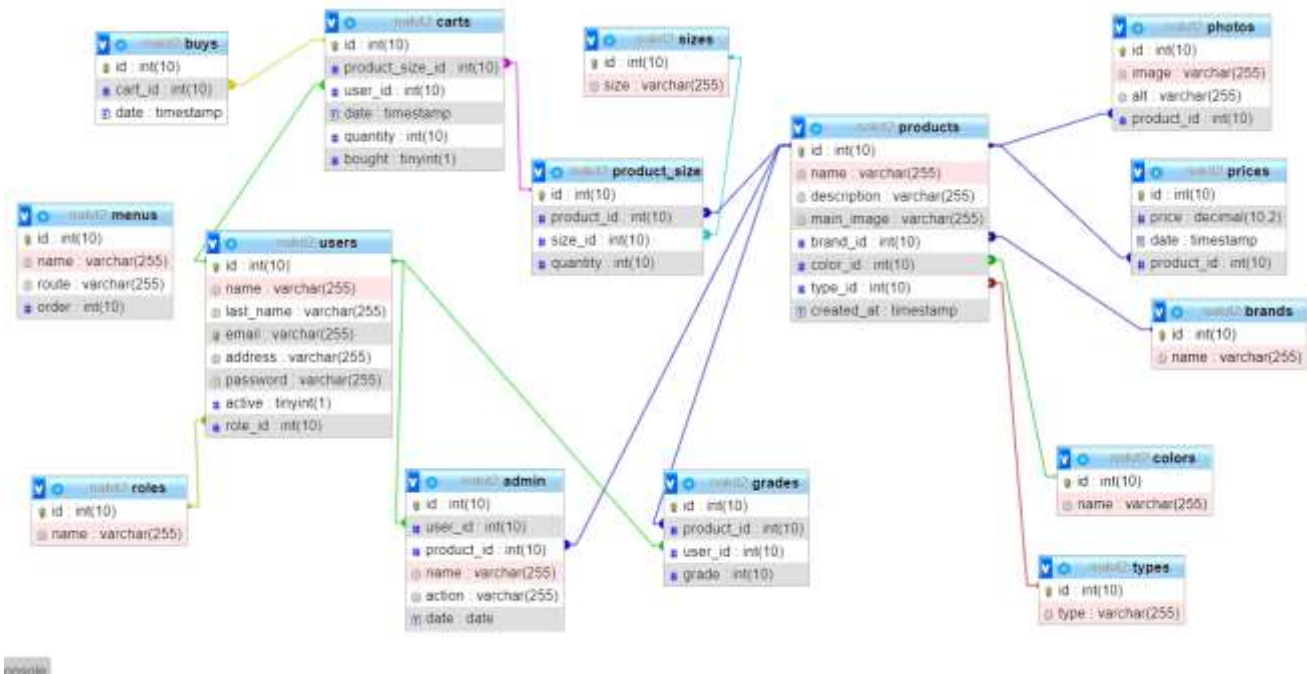
Email

Password

Login

Na ovoj stranici nalaze se forme za login i registraciju korisnika.

3. Dizajn baze podataka



4. Kodovi

\App\Http\Controllers

AdminController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Admin;

class AdminController extends OsnovniController
{
    private $adminModel;
```

```

public function index() {
    $this->adminModel = new Admin();
    $this->data['admins'] = $this->adminModel->getForAdmin();
    return view('admin.admin', $this->data);
}

public function admin() {
    $this->adminModel = new Admin();
    $adminData = $this->adminModel->getForAdmin();
    return $adminData;
}
}

```

CartController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Buy;
use App\Models\Cart;
use App\Models\Product;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Http\Response;
use Illuminate\Support\Facades\Log;

class CartController extends OsnovniController
{
    private $cartModel;
    private $boughtModel;

    public function index() {
        return view('pages.main.cart', $this->data);
    }

    public function addToCart(Request $request) {
        $product_id = $request->get('id');
        $size_id = $request->get('size');
        $quantity = $request->get('quantity');
        if(session()->has('user')) {
            $user_id = session('user')->user_id;
        }

        try {
            $product_size_id = \DB::table('product_size')
                ->select('id', 'quantity')
                ->where('product_id', $product_id)
                ->where('size_id', $size_id)
                ->first();

```

```

        if($product_size_id){
            if($quantity > $product_size_id->quantity){
                return "Not enough in stock!";
            }

            $select = \DB::table('carts')
                ->where('product_size_id', $product_size_id->id)
                ->where('user_id', $user_id)
                ->first();
            if($select){
                return 'Already in cart.';
            }
            else {
                $insert = \DB::table('carts')->insert([
                    'product_size_id' => $product_size_id->id,
                    'user_id' => $user_id,
                    'quantity' => $quantity,
                    'bought' => false
                ]);
                if($insert){
                    \DB::table('admin')->insert([
                        'user_id' => session('user')->user_id,
                        'product_id' => $product_id,
                        'name' => session('user')->user_name,
                        'action' => 'Added to cart',
                        'date' => Carbon::now()->toDate()
                    ]);
                    return 'Successfully added to cart!';
                }
            }
        }
    }
}

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

}

public function showCart(){
    $this->cartModel = new Cart();
    $products = $this->cartModel->getCart();
    return response()->json($products);
}

public function delete(Request $request){
    $id = $request->get('id');
    if(session()->has('user')){
        $user_id = session('user')->user_id;
    }

    $delete = \DB::table('carts')
        ->where('product_size_id', $id)
        ->where('user_id', $user_id)
        ->delete();

    if($delete){

```

```

        $this->cartModel = new Cart();
        $products = $this->cartModel->getCart();
        return response()->json($products);
    }
}

public function buy(Request $request){
    $insert = \DB::table('buys')->insert([
        'cart_id' => $request->get('id'),
        'date' => Carbon::now()->toDateTime()
    ]);

    if($insert){
        $quantity = \DB::table('product_size')
            ->select('quantity')
            ->where('id', $request->get('product_size_id'))
            ->first();

        $quantityCart = \DB::table('carts')
            ->select('quantity')
            ->where('id', $request->get('id'))
            ->first();

        \DB::table('product_size')->where('id', $request-
>get('product_size_id'))
            ->update([
                'quantity' => $quantity->quantity - $quantityCart-
>quantity
            ]);

        \DB::table('carts')->where('id', $request->get('id'))
            ->update([
                'bought' => true
            ]);

        $this->cartModel = new Cart();
        $products = $this->cartModel->getCart();
        return response()->json($products);
    }
}

public function bought(){
    $this->boughtModel = new Buy();
    $products = $this->boughtModel->getBought();
    return response()->json($products);
}
}

```

ContactController.php

```
<?php

namespace App\Http\Controllers;

use App\Mail\SendEmail;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;

class ContactController extends OsnovniController
{
    public function index(){
        return view('pages.main.contact', $this->data);
    }

    public function sendEmail(Request $request){
        $this->validate($request, [
            'name' => 'required',
            'email' => 'required|email',
            'subject' => 'required',
            'message' => 'required'
        ]);

        $data = array(
            'name' => $request->input('name'),
            'email' => $request->input('email'),
            'subject' => $request->input('subject'),
            'message' => $request->input('message')
        );

        Mail::to('markovic749@gmail.com')->send(new SendEmail($data));
        return "Thank you for contacting us.";
    }
}
```

HomeController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Product;
use Illuminate\Http\Request;

class HomeController extends OsnovniController
{
    private $productsModel;

    public function index(){
        $this->productsModel = new Product();
        $this->data['products'] = $this->productsModel->getTopProducts(3);
        return view('pages.main.home', $this->data);
    }
}
```



```
}  
}
```

LoginController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Http\Requests\LoginRequest;  
use Carbon\Carbon;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Log;  
  
class LoginController extends OsnovniController  
{  
    public function index(){  
        return view('pages.main.loginRegister', $this->data);  
    }  
  
    public function login(LoginRequest $request){  
        $email = $request->input('email');  
        $password = md5($request->input('password'));  
  
        try {  
            $user = \DB::table('users')  
                ->select('id as user_id', 'name as user_name', 'role_id')  
                ->where('email', $email)  
                ->where('password', $password)  
                ->first();  
  
            if($user) {  
                $request->session()->put('user', $user);  
                \DB::table('admin')->insert([  
                    'user_id' => session('user')->user_id,  
                    'name' => session('user')->user_name,  
                    'action' => 'Login',  
                    'date' => Carbon::now()->toDate()  
                ]);  
                return 'Successfully logged in!';  
            }  
            else {  
                return response(['error'=>true, 'errorMsg'=>"Please check your  
data and try again."],400);  
            }  
        }  
        catch (\Exception $e) {  
            Log::error($e->getMessage());  
        }  
    }  
  
    public function logout(Request $request){  
        \DB::table('admin')->insert([  
            'user_id' => session('user')->user_id,  
            'name' => session('user')->user_name,  
            'action' => 'Logout',  
            'date' => Carbon::now()->toDate()  
        ]);  
    }  
}
```

```

    ];
    $request->session()->pull("user");
    return redirect()->route("home");
}
}

```

OsnovniController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Menu;
use Illuminate\Http\Request;

class OsnovniController extends Controller
{
    public $data;
    private $menuModel;

    public function __construct() {
        $this->menuModel = new Menu();
        $this->data["menu"] = $this->menuModel->getMenu();
    }
}

```

ProductController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreRequest;
use App\Http\Requests\UpdateRequest;
use App\Models\Brand;
use App\Models\Color;
use App\Models\Grade;
use App\Models\Photo;
use App\Models\Product;
use App\Models\Size;
use App\Models\Type;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Storage;

class ProductController extends OsnovniController
{
    private $productsModel;
}

```

```

private $typesModel;
private $colorsModel;
private $brandsModel;
private $gradeModel;

public function __construct()
{
    parent::__construct();

    $this->productsModel = new Product();
    $this->typesModel = new Type();
    $this->colorsModel = new Color();
    $this->brandsModel = new Brand();
    $this->gradeModel = new Grade();
}

public function index() {
    $this->data['products'] = $this->productsModel->getAllProducts();
    $this->data['types'] = $this->typesModel->getTypes();
    $this->data['colors'] = $this->colorsModel->getColors();
    $this->data['brands'] = $this->brandsModel->getBrands();
    return view('pages.products.products', $this->data);
}

public function single($id) {
    $this->data['product'] = $this->productsModel->getOneProduct($id);
    $this->data['sizes'] = $this->productsModel->getProductSizes($id);
    $this->data['photos'] = $this->productsModel->getProductImages($id);
    $this->data['grades'] = $this->productsModel->getProductGrades($id);
    if(session()->has('user')) {
        $this->data['singleGrade'] = $this->gradeModel->getGrade($id);
    }
    return view('pages.products.single', $this->data);
}

public function sortFilter(Request $request) {
    $types = $request->types;
    $colors = $request->colors;
    $brands = $request->brands;
    $sortValue = $request->sortValue;
    $search = $request->search;
    $productsData = $this->productsModel->filterProducts($types, $colors,
    $brands, $sortValue, $search);
    return response()->json($productsData);
}

public function create() {
    $this->data['types'] = $this->typesModel->getTypes();
    $this->data['colors'] = $this->colorsModel->getColors();
    return view('admin.create', $this->data);
}

public function store(StoreRequest $request) {
    $name = $request->input('productName');
    $description = $request->input('productDescription');

```

```

        $color = $request->input('productColor');
        $type = $request->input('productType');
        $brand = $request->input('brandName');
        $price = $request->input('productPrice');
        $sizes = $request->input('productSize');

        $imageName = $this->productsModel->image('productImage');
        $imageName1 = $this->productsModel->image('image1');
        $imageName2 = $this->productsModel->image('image2');
        $imageName3 = $this->productsModel->image('image3');

        try{
            $selectBrand = \DB::table('brands')
                ->where('name', $brand)
                ->first();

            if($selectBrand){
                $brandId = $selectBrand->id;
            }
            else {
                $brandId = \DB::table('brands')->insertGetId([
                    'name' => $brand
                ]);
            }

            $insertProduct = \DB::table("products")->insertGetId([
                'name' => $name,
                'description' => $description,
                'main_image' => $imageName,
                'brand_id' => $brandId,
                'color_id' => $color,
                'type_id' => $type,
                'created_at' => Carbon::now()->toDateTime()
            ]);

            \DB::table('prices')->insert([
                'price' => $price,
                'date' => Carbon::now()->toDateTime(),
                'product_id' => $insertProduct
            ]);

            $this->productsModel->insertImage($imageName1, $insertProduct);
            $this->productsModel->insertImage($imageName2, $insertProduct);
            $this->productsModel->insertImage($imageName3, $insertProduct);

            $this->productsModel->sizes($sizes, $insertProduct);

            return redirect()->route('products.create')->with('success',
'Product added successfully!');
        }
        catch (\Exception $e){
            Log::error($e->getMessage());
        }
    }
}

```

```

public function edit($id){
    $this->data['product'] = $this->productsModel->getOneProduct($id);
    $this->data['sizes'] = $this->productsModel->getProductSizes($id);
    $this->data['types'] = $this->typesModel->getTypes();
    $this->data['colors'] = $this->colorsModel->getColors();
    return view('admin.edit', $this->data);
}

public function update(UpdateRequest $request, $id){
    $name = $request->input('productName');
    $description = $request->input('productDescription');
    $color = $request->input('productColor');
    $type = $request->input('productType');
    $brand = $request->input('brandName');
    $price = $request->input('productPrice');
    $sizes = $request->input('productSize');

    $imageName = $this->productsModel->image('productImage');
    $imageName1 = $this->productsModel->image('image1');
    $imageName2 = $this->productsModel->image('image2');
    $imageName3 = $this->productsModel->image('image3');

    try {
        $selectBrand = \DB::table('brands')
            ->where('name', $brand)
            ->first();

        if($selectBrand){
            $brandId = $selectBrand->id;
        }
        else {
            $brandId = \DB::table('brands')->insertGetId([
                'name' => $brand
            ]);
        }

        \DB::table("products")->where('id', $id)
            ->update([
                'name' => $name,
                'description' => $description,
                'main_image' => $imageName,
                'brand_id' => $brandId,
                'color_id' => $color,
                'type_id' => $type
            ]);

        \DB::table('prices')->where('product_id', $id)
            ->update([
                'price' => $price,
                'date' => Carbon::now()->toDateTime()
            ]);

        $photoId1 = \DB::table('photos')->select('id')
            ->where('product_id', $id)
            ->first();

        $photoId2 = \DB::table('photos')->select('id')

```

```

        ->where('product_id', $id)
        ->offset(1)
        ->limit(1)
        ->first();

        $photoId3 = \DB::table('photos')->select('id')
        ->where('product_id', $id)
        ->offset(2)
        ->limit(1)
        ->first();

        $this->productsModel->updateImage($imageName1, $photoId1);
        $this->productsModel->updateImage($imageName2, $photoId2);
        $this->productsModel->updateImage($imageName3, $photoId3);

        \DB::table('product_size')->where('product_id', $id)->delete();
        $this->productsModel->sizes($sizes, $id);

        return redirect()->route('products.edit', $id)->with('success',
'Product updated successfully!');
    }
    catch (\Exception $e){
        Log::error($e->getMessage());
    }
}

public function destroy($id){
    \DB::table('products')->where('id', $id)->delete();
    return redirect()->route('products');
}
}
}

```

RateProductController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class RateProductController extends Controller
{
    public function rate(Request $request){
        $rating = $request->get('rating');
        $product_id = $request->get('product_id');
        if(session()->has('user')){
            $user_id = session('user')->user_id;
        }
    }
}

```

```

        try {
            $insert = \DB::table('grades')->insert([
                'product_id' => $product_id,
                'user_id' => $user_id,
                'grade' => $rating
            ]);

            if($insert){
                return 'Thank you for rating our products!';
            }
        }
        catch(\Exception $e){
            Log::error($e->getMessage());
        }
    }
}

```

RegistrationController.php

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\UserRequest;
use Carbon\Carbon;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;

class RegistrationController extends Controller
{
    public function register(UserRequest $request){

        try
        {
            $name = $request->get('name');
            $last_name = $request->get('last_name');
            $address = $request->get('address');
            $email = $request->get('email');
            $password = md5($request->get('password'));
            $active = $request->get('active');
            $role_id = $request->get('role_id');

            $selectEmail = \DB::table('users')
                ->where('email', $email)
                ->first();

            if($selectEmail){
                return response(['error'=>true,'errorMsg'=>"Email already exists!"],400);
            }
        }
    }
}

```

```

        $insert = \DB::table('users')->insertGetId([
            'name' => $name,
            'last_name' => $last_name,
            'address' => $address,
            'email' => $email,
            'password' => $password,
            'active' => $active,
            'role_id' => $role_id
        ]);

        if($insert){
            $user = \DB::table('users')->select('id as user_id', 'name as
user_name')
                ->where('id', $insert)
                ->first();

            \DB::table('admin')->insert([
                'user_id' => $user->user_id,
                'name' => $user->user_name,
                'action' => 'Registration',
                'date' => Carbon::now()->toDate()
            ]);
            return "Successfully registered! You can login now :)";
        }
    }
    catch(\Exception $e)
    {
        Log::error($e->getMessage());
    }
}
}

```

\App\Http\Middleware

AdminMiddleware.php

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

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

```



```

    {
        if($request->session()->has('user')){
            if(session('user')->role_id == 2){
                return redirect('/');
            }
            else{
                return $next($request);
            }
        }
        else{
            return redirect('/');
        }
    }
}

```

UserMiddleware.php

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

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

```

\App\Http\Requests

LoginRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'email' => 'required|email|exists:users,email',
            'password' => 'required|min:8'
        ];
    }

    public function messages()
    {
        return [
            'required' => 'The :attribute field is required.',
            'email.exists' => 'Selected email does not exist in the
database.'
        ];
    }
}
```

StoreRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'productName' => ['required', 'regex:/^([A-Z][a-z]{1,50}\s?)+$/'],
            'productDescription' => 'required',
            'productColor' => 'required|not_in:0',
            'productType' => 'required|not_in:0',
            'brandName' => 'required',
            'productPrice' => 'required|numeric',
            'productSize' => 'required',
            'productImage' => 'required|image',
            'image1' => 'required|image',
            'image2' => 'required|image',
            'image3' => 'required|image',
        ];
    }

    public function messages()
    {
        return [
            'required' => 'The :attribute is required.',
            'productName.regex' => 'Each word must start with a capital letter, only letters allowed',
            'numeric' => 'The :attribute field must have a numeric value.',
            'image' => 'File must :attribute be an image.'
        ];
    }
}
```

UpdateRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'productName' => ['required', 'regex:/^([A-Z][a-z]{1,50}\s?)+$/'],
            'productDescription' => 'required',
            'productColor' => 'required|not_in:0',
            'productType' => 'required|not_in:0',
            'brandName' => 'required',
            'productPrice' => 'required|numeric',
            'productSize' => 'required',
            'productImage' => 'required|image',
            'image1' => 'required|image',
            'image2' => 'required|image',
            'image3' => 'required|image',
        ];
    }

    public function messages()
    {
        return [
            'required' => 'The :attribute is required.',
            'productName.regex' => 'Each word must start with a capital letter, only letters allowed',
            'numeric' => 'The :attribute field must have a numeric value.',
            'image' => 'File must :attribute be an image.'
        ];
    }
}
```

UserRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'name' => ['required', 'regex:/^[A-Z][a-z]{1,13}$/'],
            'last_name' => ['required', 'regex:/^[A-Z][a-z]{1,30}\s?+$/'],
            'email' => 'required|email',
            'address' => 'required',
            'password' => 'required|min:8',
            'active' => 'required',
            'role_id' => 'required|exists:roles,id'
        ];
    }

    public function messages()
    {
        return [
            'required' => 'The :attribute field is required.',
            'name.max' => 'Name must not be longer than :max characters.',
            'name.regex' => 'Name must start with a capital letter!',
            'last_name.regex' => 'Last name must start with a capital
letter!',
            'last_name.max' => 'Last name must not be longer than :max
characters.',
            'password.min' => 'Password must have at least 8 characters',
            'role_id.exists' => 'Selected role does not exist in the
database.'
        ];
    }
}
```

\App\Mail

SendEmail.php

```
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class SendEmail extends Mailable
{
    use Queueable, SerializesModels;
    public $data;
    public $sender;
    public $subject;
    public $message;
    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct($data)
    {
        $this->data = $data;
        $this->sender = $this->data['email'];
        $this->subject = $this->data['subject'];
        $this->subject = $this->data['message'];
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->from($this->sender)->subject($this->subject)->view('partials.email')->with('data', $this->message);
    }
}
```

\App\Models

Admin.php

```
<?php

namespace App\Models;

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

class Admin extends Model
{
    public function getForAdmin() {
        return \DB::table('admin')->get();
    }
}
```

Brand.php

```
<?php

namespace App\Models;

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

class Brand extends Model
{
    public function getBrands() {
        return \DB::table('brands')->get();
    }
}
```

Buy.php

```
<?php

namespace App\Models;

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

class Buy extends Model
{
    public function getBought() {
        return \DB::table('products')->select('products.name',
        'products.main image', 'prices.price', 'sizes.size', 'brands.name as brand')
    }
}
```

```

        ->join('prices', 'products.id', '=', 'prices.product_id')
        ->join('brands', 'products.brand_id', '=', 'brands.id')
        ->join('product_size', 'products.id', '=',
'product_size.product_id')
        ->join('sizes', 'sizes.id', '=', 'product_size.size_id')
        ->join('carts', 'product_size.id', '=', 'carts.product_size_id')
        ->join('buys', 'carts.id', '=', 'buys.cart_id')
        ->where('carts.user_id', session('user')->user_id)
        ->where('carts.bought', true)
        ->orderByDesc('buys.date')
        ->limit(3)
        ->get();
    }
}

```

Cart.php

```

<?php

namespace App\Models;

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

class Cart extends Model
{
    public function getCart(){

        return \DB::table('products')
            ->select('product_size.id', 'products.name',
'products.main_image', 'prices.price', 'brands.name as brand', 'sizes.size',
'carts.quantity', 'carts.id as cart_id')
            ->join('prices', 'products.id', '=', 'prices.product_id')
            ->join('brands', 'products.brand_id', '=', 'brands.id')
            ->join('product_size', 'products.id', '=',
'product_size.product_id')
            ->join('sizes', 'sizes.id', '=', 'product_size.size_id')
            ->join('carts', 'product_size.id', '=', 'carts.product_size_id')
            ->leftJoin('buys', 'carts.id', '=', 'buys.cart_id')
            ->where('carts.user_id', session('user')->user_id)
            ->where('carts.bought', false)
            ->get();
    }
}

```


Color.php

```
<?php

namespace App\Models;

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

class Color extends Model
{
    public function getColors() {
        return \DB::table('colors')->get();
    }
}
```

Grade.php

```
<?php

namespace App\Models;

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

class Grade extends Model
{
    public function getGrade($id) {
        return \DB::table('grades')
            ->where('product_id', $id)
            ->where('user_id', session('user')->user_id)
            ->first();
    }
}
```

Menu.php

```
<?php

namespace App\Models;

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

class Menu extends Model
{
    public function getMenu() {
        return \DB::table('menus')->orderBy('order')->get();
    }
}
```

Product.php

```
<?php

namespace App\Models;

use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;

class Product extends Model
{
    /* ----- UPITI ZA PRIKAZ ----- */

    public function getTopProducts($number) {
        return \DB::table('products')->select('products.id', 'name',
'main_image', 'prices.price', DB::raw('round(AVG(grades.grade), 1) as
grade'))
        ->join('prices', 'products.id', '=', 'prices.product_id')
        ->leftJoin('grades', 'products.id', '=', 'grades.product_id')
        ->orderBy('products.created_at', 'desc')
        ->groupBy('products.id', 'name', 'main_image', 'prices.price')
        ->limit($number)
        ->get();
    }

    public function getAllProducts() {
        return \DB::table('products')->select('products.id', 'products.name',
'main_image', 'prices.price', DB::raw('round(AVG(grades.grade), 1) as
grade'))
        ->join('prices', 'products.id', '=', 'prices.product_id')
        ->leftJoin('grades', 'products.id', '=', 'grades.product_id')
        ->groupBy('products.id', 'name', 'main_image', 'prices.price')
        ->paginate(6);
    }

    public function filterProducts($types, $colors, $brands, $sortValue,
$search) {
        $query = \DB::table('products');

        $query = $query->join('prices', 'products.id', '=',
'prices.product_id');
        $query = $query->join('types', 'products.type_id', '=', 'types.id');
        $query = $query->join('brands', 'products.brand_id', '=',
'brands.id');
        $query = $query->join('colors', 'products.color_id', '=',
'colors.id');
        $query = $query->leftJoin('grades', 'products.id', '=',
'grades.product_id');

        if(is_array($types)) {
```

```

        $query = $query->whereIn('products.type_id', $types);
    }

    if(is_array($brands)) {
        $query = $query->whereIn('products.brand_id', $brands);
    }

    if(is_array($colors)) {
        $query = $query->whereIn('products.color_id', $colors);
    }

    if($search) {
        $query = $query->where("products.name", "like", "%". $search
        ."%")
        ->orWhere("brands.name", "like", "%". $search ."%")
        ->orWhere("types.type", "like", "%". $search ."%");
    }

    if($sortValue) {
        if($sortValue == 'lowToHigh') {
            $query = $query->orderBy("prices.price");
        }

        if($sortValue == 'highToLow'){
            $query = $query->orderByDesc("prices.price");
        }

        if($sortValue == 'highestRating'){
            $query = $query->orderByDesc("grades.grade");
        }
    }

    $query = $query->select('products.id', 'products.name as name',
    'main_image', 'prices.price', DB::raw('round(AVG(grades.grade), 1) as
    grade'));
    $query = $query->groupBy('products.id', 'products.name',
    'main_image', 'prices.price');
    $query = $query->paginate(6);

    return $query;
}

public function getOneProduct($id){
    return \DB::table('products')->select('products.*', 'prices.price',
    'brands.name as brand', 'types.id as type_id', 'types.type', 'colors.id as
    color_id', 'colors.name as color')
    ->join('prices', 'products.id', '=', 'prices.product_id')
    ->join('brands', 'products.brand_id', '=', 'brands.id')
    ->join('types', 'products.type_id', '=', 'types.id')
    ->join('colors', 'products.color_id', '=', 'colors.id')
    ->where('products.id', $id)
    ->first();
}

public function getProductSizes($id){
    return \DB::table('product_size')->select('sizes.*',
    'product_size.quantity')

```

```

        ->join('sizes', 'product_size.size_id', '=', 'sizes.id')
        ->where('product_id', $id)
        ->where('product_size.quantity', '>', 0)
        ->get();
    }

    public function getProductImages($id){
        return \DB::table('photos')->select('image', 'alt')
        ->join('products', 'products.id', '=', 'photos.product_id')
        ->where('product_id', $id)
        ->get();
    }

    public function getProductGrades($id){
        return \DB::table('grades')->select(DB::raw('round(AVG(grade), 1) as
grade'))
        ->join('products', 'products.id', '=', 'grades.product_id')
        ->where('product_id', $id)
        ->get();
    }

    /* ----- KRAJ ----- */

    public static function image($key){
        move_uploaded_file($_FILES[$key]['tmp_name'],
public_path().'/assets/images/'.time() . $_FILES[$key]['name']);
        return time() . $_FILES[$key]['name'];
    }

    public static function insertImage($image, $id){
        \DB::table('photos')->insert([
            'image' => $image,
            'product_id' => $id
        ]);
    }

    public static function updateImage($image, $id){
        \DB::table('photos')->where('id', $id->id)
        ->update([
            'image' => $image
        ]);
    }

    public function sizes($sizes, $id){
        $sizeIds = [];
        for ($i = 0; $i < count(explode(",", $sizes)); $i++) {
            $sizeIds[] = \DB::table('sizes')->insertGetId([
                'size' => explode(",", $sizes)[$i]
            ]);
        }

        for($i = 0; $i < count($sizeIds); $i++) {
            \DB::table('product_size')->insert([
                'product_id' => $id,

```

```

                'size_id' => $sizeIds[$i],
                'quantity' => 10
            ]);
        }
    }
}

```

Type.php

```

<?php

namespace App\Models;

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

class Type extends Model
{
    public function getTypes() {
        return \DB::table('types')->get();
    }
}

```

\Public\Assets\Js

Main.js

```

const token = $('meta[name="csrf-token"]').attr('content');

$.ajaxSetup({
    headers: {
        'X-CSRF-TOKEN': token
    }
})

$(document).ready(function () {
    $('#loginBtn').click(login);
    $('#form-submit').click(contact);
    $('.page-link').click(loadMoreProducts);
    $('#filterBtn').click(sortAndFilter);
    $('#registerBtn').click(register);
    $('#rating').change(rate);
    $('.addToCart').click(addToCart);
    $('#datum').change(filterDate);

    if(window.location.href == "http://127.0.1.1/sajtPhp2/public/cart") {
        showProductsCart();
        showBoughtCart();
    }
}

```

```

}))

var types = [];
var colors = [];
var brands = [];

function loadMoreProducts(e) {
    e.preventDefault();
    types = [];
    $.each($(".input[name='types']:checked"), function() {
        types.push($(this).val());
    });

    colors = [];
    $.each($(".input[name='colors']:checked"), function() {
        colors.push($(this).val());
    });

    brands = [];
    $.each($(".input[name='brands']:checked"), function() {
        brands.push($(this).val());
    });

    let sortValue = $('#sort').val();
    let search = $('#search').val();
    let page = $(this).data("page");
    getProducts(page, types, colors, brands, sortValue, search);
}

function sortAndFilter() {
    types = [];
    $.each($(".input[name='types']:checked"), function() {
        types.push($(this).val());
    });

    colors = [];
    $.each($(".input[name='colors']:checked"), function() {
        colors.push($(this).val());
    });

    brands = [];
    $.each($(".input[name='brands']:checked"), function() {
        brands.push($(this).val());
    });

    let sortValue = $('#sort').val();
    let search = $('#search').val();
    getProducts(1, types, colors, brands, sortValue, search);
}

function getProducts(page, types, colors, brands, sortValue, search) {
    const caller = arguments.callee.caller.name;

    $.ajax({

```

```

url: baseUrl + "/products/filter",
method: "get",
data: {
    page : page,
    sortValue : sortValue,
    search : search,
    types : types,
    colors : colors,
    brands : brands
},
dataType: "json",
success: function (response) {
    if(response.data.length > 0){
        showProducts(response.data);
        if(caller == 'sortAndFilter'){
            changePagination(response.last_page,
response.current_page);
        }
        if(caller == 'loadMoreProducts'){
            changeActivePageLink(response.current_page);
        }
        $(".pagination").show();
    }
    else {
        $("#products").html("<div class='row mb-5'><div class='col-
lg-12 d-flex justify-content-center'><h2>No products in this
category!</h2></div></div>");
        $(".pagination").hide();
    }
},
error: function (xhr, status, err) {
    console.log(xhr.status)
}
})
}

function changePagination(totalLinks, currentPage){
    let html = "";
    for(let i = 1; i <= totalLinks; i++){
        if(i != currentPage){
            html += `<li class="page-item"><a class="page-link" id="link${i}"
data-page="${i}" href="#">${i}</a></li>`;
        }else{
            html += `<li class="page-item active"><a class="page-link" id =
"link${i}" data-page="${i}" href="#">${i}</a></li>`;
        }
    }
    $(".pagination").html(html);
    $(".page-link").click(loadMoreProducts);
}

function changeActivePageLink(currentPage) {
    $('.page-item').removeClass('active');
    $('#link' + currentPage).parent().addClass('active');
}

```

```

function login() {
    var email, password, reEmail;
    var validate = true;

    email = document.getElementById('emailLogin').value;
    password = document.getElementById('passLogin').value;
    reEmail = /^[a-zA-Z0-9_\-\.]+@([a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5})$/;

    if (email == "") {
        document.getElementById("emailLoginError").innerHTML = "Email is required!";
        validate = false;
    } else if (!reEmail.test(email)) {
        document.getElementById("emailLoginError").innerHTML = "Email is invalid!";
        validate = false;
    } else {
        document.getElementById("emailLoginError").innerHTML = "";
    }

    if (password == "") {
        document.getElementById("passLoginError").innerHTML = "Password is required!";
        validate = false;
    } else if (password.length < 8) {
        document.getElementById("passLoginError").innerHTML = "Password must have at least 8 characters!";
        validate = false;
    } else {
        document.getElementById("passLoginError").innerHTML = "";
    }

    if (validate) {
        $.ajax({
            url: baseUrl + "/login",
            method: "POST",
            data: {
                email: email,
                password: password,
                _token : token
            },
            success: function (data) {
                alert(data);
                window.location.replace("http://127.0.1.1/sajtPhp2/public/");
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function contact(){
    var name, email, subject, message;

```



```

var validate = true;
name = $("#name").val();
email = $("#email").val();
subject = $("#subject").val();
message = $("#message").val();

if(name == ''){
    $("#nameContactError").html('Name is required');
    validate = false;
}
else {
    $("#nameContactError").html('');
}

if(email == ''){
    $("#emailContactError").html('Email is required');
    validate = false;
}
else {
    $("#emailContactError").html('');
}

if(subject == ''){
    $("#subjectError").html('Subject is required');
    validate = false;
}
else {
    $("#subjectError").html('');
}

if(message == ''){
    $("#messageContactError").html('Message is required');
    validate = false;
}
else {
    $("#messageContactError").html('');
}

if(validate) {
    $.ajax({
        url: baseUrl + "/sendEmail",
        method: "POST",
        data: {
            name : name,
            email: email,
            subject: subject,
            message : message,
            _token : token
        },
        success: function (data) {
            alert(data);
            window.location.reload();
        },
        error: function (xhr, status, err) {
            console.log(xhr.status)
        }
    })
}

```

```

    }
}

function showProducts(products) {
    let html = '';
    html += `<div class="row">`;
    for(let product of products){
        html += `<div class="col-lg-4 col-md-4 all des">
            <div class="product-item">
                <a href="products/${product.id}"></a>
                <div class="down-content">
                    <a href="products/${product.id}"><h4>${
product.name }</h4></a>
                    <h6>${ product.price }</h6>`
                    if(product.grade) {
                        html += `<ul class="stars">
                            <li>
                                <i class="fa fa-star "></i>
${product.grade}/5
                            </li>
                        </ul>`
                    }
                    else {
                        html += `<ul class="stars">
                            <li><i class="fa fa-star mr-2"></i>Be
first to rate!</li>
                        </ul>`
                    }
                    html += `</div>
                </div>`
            </div>`
        };
    }
    html += `</div>`;
    $("#products").html(html);
}

function register() {
    var name, last_name, address, email, password, confirm_password, active,
role_id, reEmail, reName, reLastName;
    var validate = true;

    name = document.getElementById('registerName').value;
    last_name = document.getElementById('registerLastName').value;
    address = document.getElementById('address').value;
    email = document.getElementById('emailRegister').value;
    password = document.getElementById('passRegister').value;
    confirm_password = document.getElementById('confirmPass').value;
    role_id = document.getElementById('role_id').value;
    active = 1;
    reEmail = /^[a-zA-Z0-9_\-\.]+@([a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5})$/;
    reName = /^[A-Z][a-z]{1,13}$/;
    reLastName = /^[A-Z][a-z]{1,30}\s?+$/;

    if (name == "") {

```

```

        document.getElementById("nameRegError").innerHTML = "Name is
required!";
        validate = false;
    }
    else if(!reName.test(name)){
        document.getElementById("nameRegError").innerHTML = "Name must start
with a capital letter!";
        validate = false;
    }
    else {
        document.getElementById("nameRegError").innerHTML = "";
    }

    if (last_name == "") {
        document.getElementById("lastNameRegError").innerHTML = "Last name is
required!";
        validate = false;
    }
    else if(!reLastName.test(last_name)){
        document.getElementById("lastNameRegError").innerHTML = "Last name
must start with a capital letter!";
        validate = false;
    }
    else {
        document.getElementById("lastNameRegError").innerHTML = "";
    }

    if (address == "") {
        document.getElementById("addressRegError").innerHTML = "Address is
required!";
        validate = false;
    }
    else {
        document.getElementById("addressRegError").innerHTML = "";
    }

    if (email == "") {
        document.getElementById("emailLoginError").innerHTML = "Email is
required!";
        validate = false;
    } else if (!reEmail.test(email)) {
        document.getElementById("emailLoginError").innerHTML = "Email is
invalid!";
        validate = false;
    } else {
        document.getElementById("emailLoginError").innerHTML = "";
    }

    if (password == "") {
        document.getElementById("passRegError").innerHTML = "Password is
required!";
        validate = false;
    } else if (password.length < 8) {
        document.getElementById("passRegError").innerHTML = "Password must
have at least 8 characters!";
        validate = false;
    } else {

```

```

        document.getElementById("passRegError").innerHTML = "";
    }

    if(confirm_password != password) {
        document.getElementById("confPassRegError").innerHTML = "Passwords
are not the same!";
        validate = false;
    }
    else {
        document.getElementById("confPassRegError").innerHTML = "";
    }

    if (validate) {
        $.ajax({
            url: baseUrl + "/register",
            method: "POST",
            data: {
                name : name,
                last_name : last_name,
                address : address,
                email: email,
                password: confirm_password,
                active : active,
                role_id : role_id,
                _token : token
            },
            success: function (data) {
                alert(data);
                window.location.reload();
            },
            error: function (xhr, status, err) {
                if(xhr.status == 400){
                    alert(xhr.responseJSON.errorMsg);
                }
            }
        })
    }
}

function rate(){
    var rating, product_id;
    rating = $("#rating").val();
    product_id = $("#product_id").val();

    $.ajax({
        url: baseUrl + "/rate",
        method: "POST",
        data: {
            rating : rating,
            product_id : product_id,
            _token : token
        },
        success: function (data) {
            alert(data);
            window.location.reload();
        },
        error: function (xhr, status, err) {

```

```

        console.log(xhr.status);
    }
    })
}

function addToCart(){
    let id = this.dataset.id;
    let size = $("input[name='size']:checked").data('idsize');
    let quantity = $("#var-value").html();

    if(id != null && size != null && quantity != null) {
        $.ajax({
            url: "http://127.0.1.1/sajtPhp2/public/cart/addToCart",
            method: "post",
            type: "json",
            data: {
                id : id,
                size : size,
                quantity : quantity
            },
            success: function(data){
                alert(data)
            },
            error: function(xhr, err, status){
                console.log(status);
            }
        })
    }
    else {
        alert("Pick a size please");
    }
}

function showCart(products) {
    let html = "";

    products.forEach(element => {
        html += `<div class="row mt-5">
            <div class="col-lg-4">
                <div class="card">
                    
                </div>
            </div>
            <div class="col-lg-8">
                <div class="card border-0">
                    <div class="card-body">
                        <h3>${element.name}</h3>
                        <ul class="list-inline mt-3">
                            <li class="list-inline-item">
                                <h6 class="mt-3">Brand:</h6>
                            </li>
                            <li class="list-inline-item">
                                <p class="text-muted"><strong>
${element.brand} </strong></p>
                            </li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>`;
    });
}

```

```

        </ul>
        <ul class="list-inline mt-3">
            <li class="list-inline-item">
                <h6>Size you picked:</h6>
            </li>
            <li class="list-inline-item">
                <p class="text-muted"><strong>
${element.size} </strong></p>
            </li>
        </ul>
        <p class="text-muted mt-3 mb-3">${element.price} &nbsp; x &nbsp; ${element.quantity} &nbsp; = &nbsp; ${element.price * element.quantity}.00</p>
        <button type="submit" class="btn btn-lg float-right mt-5 removeCart" data-idproductsize="${element.id}" name="removeCart">Remove</button>
        <button type="button" class="btn btn-lg float-right mt-5 mr-2 buy" data-idcart="${element.cart_id}" data-idproductsize="${element.id}" >Buy</button>
    </div>
</div>
</div>
</div>`
    })

    $("#productsCart").html(html);
    $(".removeCart").click(removeFromCart);
    $(".buy").click(buyProduct);
}

function buyProduct() {
    var id = this.dataset.idcart;
    var product_size_id = this.dataset.idproductsize;

    $.ajax({
        url: baseUrl + "/cart/buy",
        method: "post",
        data: {
            id : id,
            product_size_id : product_size_id,
            _token : token
        },
        success: function(data){
            alert("Thank you for buying our products!");
            if(data.length > 0){
                showCart(data);
                showBoughtCart();
            }
            else {
                $("#productsCart").html("<div class='d-flex align-items-center justify-content-center' id='emptyCart'><h1>Your cart is empty</h1></div>");
                showBoughtCart();
            }
        }
    },

```

```

        error: function(xhr, err, status){
            console.log(status);
        }
    })
}

function showBought(products) {
    let html = '';
    html += `<div class="row mt-5 border-top"><div class="col-lg-12 text-center mt-5"><h3>Recently bought</h3></div></div>
    <div class="row mt-5 d-flex justify-content-evenly">`

    products.forEach(element => {
        html += `<div class="col-lg-4">
            
            <h3 class="mt-3">${element.name}</h3>
            <ul class="list-inline">
                <li class="list-inline-item">
                    <h6 class="mt-3">Price
                </li>
                <li class="list-inline-item">
                    <p class="text-muted"><strong> ${element.price}
</strong></p>
                </li>
            </ul>
            <ul class="list-inline">
                <li class="list-inline-item">
                    <h6 class="mt-3">Brand
                </li>
                <li class="list-inline-item">
                    <p class="text-muted"><strong> ${element.brand}
</strong></p>
                </li>
            </ul>
            <ul class="list-inline">
                <li class="list-inline-item">
                    <h6 class="mt-3">Size
                </li>
                <li class="list-inline-item">
                    <p class="text-muted"><strong> ${element.size}
</strong></p>
                </li>
            </ul>
        </div>`
    })

    html += `</div>`;
    $("#bought").html(html);
}

function showBoughtCart() {
    $.ajax({
        url: baseUrl + "/cart/showBought",
        method: "GET",
        type: "json",
    })
}

```

```

        success: function(data){
            if(data.length > 0){
                showBought(data);
            }
            else {
                $("#bought").html("");
            }
        },
        error: function(xhr, err, status){
            console.log(status);
        }
    })
}

function showProductsCart(){
    $.ajax({
        url: baseUrl + "/showCart",
        method: "GET",
        type: "json",
        success: function(data){
            if(data.length > 0){
                showCart(data);
            }
            else {
                $("#productsCart").html("<div class='d-flex align-items-center justify-content-center' id='emptyCart'><h1>Your cart is empty</h1></div>");
            }
        },
        error: function(xhr, err, status){
            console.log(status);
        }
    })
}

function removeFromCart(){
    let id = this.dataset.idproducts;

    $.ajax({
        url: baseUrl + "/cart/delete",
        method: "delete",
        type: "json",
        data: {
            id : id
        },
        success: function(data){
            alert("Successfully removed from cart.");
            if(data.length > 0){
                showCart(data);
            }
            else {
                $("#productsCart").html("<div class='d-flex align-items-center justify-content-center' id='emptyCart'><h1>Your cart is empty</h1></div>");
            }
        },
        error: function(xhr, err, status){

```



```

        console.log(status);
    }
    })
}

function showAdmin(data){
    let html = '';
    data.forEach(el => {
        html += `<tr>
                <th scope="row">${el.user_id }</th>`
        if(el.product_id != null) {
            html += `<th scope="row">${el.product_id}</th>`
        }
        else {
            html += `<th scope="row"><i class="fa fa-times"
aria-hidden="true"></i></th>`
        }
        html += `<td>${el.name }</td>
                <td>${ el.action }</td>
                <td>${ el.date }</td>
                </tr>`
    });

    $("#tableBody").html(html);
}

function filterDate(){
    let datum = $("#datum").val();

    $.ajax({
        url: baseUrl + "/date",
        method: "get",
        success: function(data){
            data = data.filter(el => {
                if(el.date == datum){
                    return true;
                }
            })

            showAdmin(data);
        },
        error: function(xhr, err, status){
            if(xhr.status == 400){
                console.log(xhr.responseJSON);
            }
        }
    })
}

```

\Resources\Views\Admin

Admin.blade.php

```
@extends('layouts.layout')
@section('title') Admin @endsection
@section('description') Website for selling jewelry @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')
    <div id="prazno"></div>
    <div class="container mt-5">
        <div class="row">
            <div class="col-lg-7">
                <h3>User activity</h3>
            </div>
            <div class="col-lg-5" id="dateFilter">
                <label for="datum" class="form-label mr-3">Filter by
date</label>
                <input type="date" id="datum" name="datum">
            </div>
        </div>
        <div class="row" id="activity">
            <table class="table table-responsive-md mt-5">
                <thead>
                    <tr>
                        <th scope="col">User ID</th>
                        <th scope="col">Product ID</th>
                        <th scope="col">Name</th>
                        <th scope="col">Activity</th>
                        <th scope="col">Date</th>
                    </tr>
                </thead>
                <tbody id="tableBody">
                    @foreach($admins as $admin)
                        <tr>
                            <th scope="row">{{ $admin->user_id }}</th>
                            @if($admin->product_id != null)
                                <th scope="row">{{ $admin->product_id }}</th>
                            @else
                                <th scope="row"><i class="fa fa-times" aria-
hidden="true"></i></th>
                            @endif
                            <td>{{ $admin->name }}</td>
                            <td>{{ $admin->action }}</td>
                            <td>{{ $admin->date }}</td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
@endsection
```

Create.blade.php

```
@extends('layouts.layout')
@section('title') Insert Product @endsection
@section('description') Website for selling jewelry @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')
    <div id="prazno"></div>
    @if(session()->has('success'))
        @include('partials.success')
    @endif
    <div class="container mt-5">
        <div class="col-md-12">
            <h2 class="mb-4">Insert new product</h2>
            @include("admin.form", ["action" => 'products.store'])
        </div>
    </div>
@endsection
```

Edit.blade.php

```
@extends('layouts.layout')
@section('title') Edit Product @endsection
@section('description') Website for selling jewelry @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')
    <div id="prazno"></div>
    @if(session()->has('success'))
        @include('partials.success')
    @endif
    <div class="container mt-5">
        <div class="col-md-12">
            <h2 class="mb-4">Edit a product</h2>
            @include("admin.form", ["action" => "products.update"])
        </div>
    </div>
@endsection
```

Form.blade.php

```
<form action="@if($action == 'products.update') {{ route($action, $product->id) }} @else {{ route($action) }} @endif" method="post"
enctype="multipart/form-data" class="mt-3">
    @csrf
    @if($action == "products.update")
        @method('PUT')
    @endif
    <div class="mb-3">
```

```

        <label for="productName" class="form-label">Product name</label>
        <input type="text" class="form-control" id="productName"
name="productName" value="{{ $product->name ?? old('name') }}" />
        @error('productName')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="productDescription" class="form-label">Product
description</label>
        <textarea name="productDescription" class="form-control"
id="productDescription">{{ $product->description ?? old('description')
}}</textarea>
        @error('productDescription')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="productImage" class="form-label mr-3">Product
image</label>
        <input type="file" id="productImage" name="productImage" />
        @error('productImage')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="productColor" class="form-label mr-4">Product
color</label>
        <select name="productColor" id="productColor">
            <option value="0">Choose...</option>
            @if(isset($product))
                <option value="{{ $product->color_id }}" selected>{{
$product->color }}</option>
            @foreach($colors as $color)
                @if($product->color_id != $color->id)
                    <option value="{{ $color->id }}">{{ $color->name
}}</option>
                @endif
            @endforeach
            @else
                @foreach($colors as $color)
                    <option value="{{ $color->id }}">{{ $color->name
}}</option>
                @endforeach
            @endif
        </select>
        @error('productColor')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </div>

```

```

</div>
<div class="mb-3">
  <label for="productType" class="form-label mr-4">Product type</label>
  <select name="productType" id="productType">
    <option value="0">Choose...</option>
    @if(isset($product))
      <option value="{{ $product->type_id }}" selected>{{ $product-
>type }}</option>
    @foreach($types as $type)
      @if($product->type_id != $type->id)
        <option value="{{ $type->id }}">{{ $type->type
}}</option>
      @endif
    @endforeach
  @else
    @foreach($types as $type)
      <option value="{{ $type->id }}">{{ $type->type
}}</option>
    @endforeach
  @endif
</select>
  @error('productType')
  <div class="text-danger">
    {{ $message }}
  </div>
  @enderror
</div>
<div class="mb-3">
  <label for="brandName" class="form-label">Brand name</label>
  <input type="text" class="form-control" id="brandName"
name="brandName" value="{{ $product->brand ?? old('brand') }}" />
  @error('brandName')
  <div class="text-danger">
    {{ $message }}
  </div>
  @enderror
</div>
<div class="mb-3">
  <label for="productPrice" class="form-label">Product price</label>
  <input type="text" class="form-control" id="productPrice"
name="productPrice" value="{{ $product->price ?? old('price') }}" />
  @error('productPrice')
  <div class="text-danger">
    {{ $message }}
  </div>
  @enderror
</div>
<div class="mb-3">
  <ul>
    <li class="mb-2">
      Additional images
    </li>
    <li>
      <input type="file" id="image1" name="image1" class="mb-2" />
      @error('image1')
      <div class="text-danger">
        {{ $message }}
      </div>
    </li>
  </ul>
</div>

```

```

        </div>
        @enderror
    </li>
    <li>
        <input type="file" id="image2" name="image2" class="mb-2"/>
        @error('image2')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </li>
    <li>
        <input type="file" id="image3" name="image3"/>
        @error('image3')
        <div class="text-danger">
            {{ $message }}
        </div>
        @enderror
    </li>
</ul>
</div>
<div class="mb-3">
    <label for="productSize" class="form-label">Product sizes</label>
    @if(isset($product))
        <input type="text" class="form-control mb-2" id="productSize"
name="productSize" value="@foreach($sizes as $size){ { $size->size } }
@if(!$loop->index == count($sizes) -1),@endif @endforeach"/>
    @else
        <input type="text" class="form-control" id="productSize"
name="productSize" />
    @endif
    <div class="form-text text-muted">If there is multiple sizes please
separate them by comma ( , ).</div>
    @error('productSize')
    <div class="text-danger">
        {{ $message }}
    </div>
    @enderror
</div>
<button type="submit" class="btn btn-lg" id="submitBtn">Submit</button>
</form>

```

\Resources\Views\Fixed

Footer.blade.php

```

<footer>
    <div class="container">
        <div class="row">
            <div class="col-md-4">
                <div class="inner-content">

```


Nav.blade.php

```
<header class="">
    <nav class="navbar navbar-expand-lg">
        <div class="container">
            <a class="navbar-brand" href="{{ route('home') }}"><h2>Bling
<em>On</em></h2></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">
                    @foreach($menu as $link)
                        <li class="nav-item @if(request()->routeIs($link-
>route)) active @endif">
                            <a class="nav-link" href="{{ route($link->route)
}}"> {{ $link->name }} </a>
                        </li>
                    @endforeach
                    @if(session()->has('user') && session('user')->role_id ==
1)
                        <li class="nav-item @if(request()-
>routeIs('admin')) active @endif">
                            <a class="nav-link" href="{{ route('admin')
}}"> Admin </a>
                        </li>
                    @endif
                    @if(session()->has('user'))
                        <li class="nav-item @if(request()-
>routeIs('cart')) active @endif">
                            <a class="nav-link" href="{{ route('cart')
}}" id="cart"> <i class="fa fa-shopping-cart" aria-hidden="true"></i> </a>
                        </li>
                    @endif
                    @if(!session()->has('user'))
                        <li class="nav-item @if(request()-
>routeIs('loginRegister')) active @endif">
                            <a class="nav-link" href="{{
route('loginRegister') }}"> Login | Register </a>
                        </li>
                    @else
                        <li class="nav-item">
                            <a class="nav-link" href="{{ route('logout')
}}"> Logout </a>
                        </li>
                    @endif
                </ul>
            </div>
        </div>
    </nav>
</header>
```


Scripts.blade.php

```
<!-- Bootstrap core JavaScript -->
<script src="{{ asset('vendor/jquery/jquery.min.js') }}"></script>
<script src="{{ asset('vendor/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

<!-- Additional Scripts Template -->
<script src="{{ asset('assets/js/custom.js') }}"></script>
<script src="{{ asset('assets/js/owl.js') }}"></script>
<script src="{{ asset('assets/js/slick.js') }}"></script>
<script src="{{ asset('assets/js/isotope.js') }}"></script>
<script src="{{ asset('assets/js/accordions.js') }}"></script>
<script src="{{ asset('assets/js/templatemo.js') }}"></script>
<!-- End Template -->
<script type="text/javascript">
    const baseUrl = "{{ url('/') }}"
</script>
<script src="{{ asset('assets/js/main.js') }}"></script>

<script language = "text/Javascript">
    cleared[0] = cleared[1] = cleared[2] = 0; //set a cleared flag for each
field
    function clearField(t){                      //declaring the array outside
of the
        if(! cleared[t.id]){                      // function makes it static
and global
            cleared[t.id] = 1;  // you could use true and false, but that's
more typing
            t.value='';           // with more chance of typos
            t.style.color='#fff';
        }
    }
</script>
```

\Resources\Views\Layouts

Layout.blade.php

```
<!DOCTYPE html>
<html lang="en">
    @include('fixed.header')
    <body>
        @include('fixed.nav')
        @yield('content')
        @include('fixed.footer')
        @include('fixed.scripts')
    </body>
</html>
```

\Resources\Views\Pages\Main

Cart.blade.php

```
@extends('layouts.layout')
@section('title') Cart @endsection
@section('description') Products for sale @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')
    <div id="prazno"></div>
    <div class="container mt-5">
        <div class="row">
            <div class="col-lg-12">
                <h2 class="mb-5">YOUR CART</h2>
            </div>
        </div>
    </div>
    <div class="container" id="productsCart">
    </div>
    <div class="container mt-5" id="bought">

    </div>
@endsection
```

Contact.blade.php

```
@extends('layouts.layout')
@section('title') Contact @endsection
@section('description') Contact page @endsection
@section('keywords') jewelry, sell, online, contact @endsection
@section('content')

    <!-- ***** Preloader Start ***** -->
    <div id="preloader">
        <div class="jumper">
            <div></div>
            <div></div>
            <div></div>
        </div>
    </div>
    <!-- ***** Preloader End ***** -->

    <!-- Page Content -->
    <div class="page-heading contact-heading header-text">
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <div class="text-content">
                        <h4>contact us</h4>
                        <h2>let's get in touch</h2>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```

    </div>
  </div>

  @if(session()->has('success'))
    @include('partials.success')
  @endif
  <div class="send-message">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <div class="section-heading">
            <h2>Send us a Message</h2>
          </div>
        </div>
        <div class="col-md-12">
          <div class="contact-form">
            <form id="contact" action="{{ route('sendEmail') }}"
method="post">
              <div class="row">
                <div class="col-lg-12 col-md-12 col-sm-12">
                  <p class="text-danger" id="nameContactError"></p>
                  <fieldset>
                    <input name="name" type="text" class="form-control"
id="name" placeholder="Full Name" required="">
                  </fieldset>
                </div>
                <div class="col-lg-12 col-md-12 col-sm-12">
                  <p class="text-danger" id="emailContactError"></p>
                  <fieldset>
                    <input name="email" type="text" class="form-control"
id="email" placeholder="E-Mail Address" required="">
                  </fieldset>
                </div>
                <div class="col-lg-12 col-md-12 col-sm-12">
                  <p class="text-danger" id="subjectError"></p>
                  <fieldset>
                    <input name="subject" type="text" class="form-control"
id="subject" placeholder="Subject" required="">
                  </fieldset>
                </div>
                <div class="col-lg-12">
                  <p class="text-danger" id="messageContactError"></p>
                  <fieldset>
                    <textarea name="message" rows="6" class="form-control"
id="message" placeholder="Your Message" required=""></textarea>
                  </fieldset>
                </div>
                <div class="col-lg-12">
                  <fieldset>
                    <button type="button" id="form-submit" class="filled-
button">Send Message</button>
                  </fieldset>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>

<div class="container-fluid mt-5" id="autor">
    <div class="row mb-5 mt-5">
        <div class="col-lg-9 mx-auto mt-5">
            <h2>Author</h2>
        </div>
    </div>
    <div class="row d-flex justify-content-start">
        <div class="col-12 col-lg-4 text-right" id="imageAuthor">
            
        </div>
        <div class="col-12 col-lg-4">
            <p class="mb-3">My name is Aleksandra Markovic. I am from Gornji Milanovac and I am 21 years old. I study internet technologies at ICT college and I am interested in programming and design.</p>
            <p class="mb-3">As I study more it looks like I am more interested in Back End Web development, but who knows, maybe I'll change my mind. </p>
            <p>I was born on 23.05.1999. In my free time I like to watch TV shows, movies, hang out with my friends and lots of different stuff. </p>
        </div>
    </div>
</div>

@endsection

```

Home.blade.php

```

@extends('layouts.layout')
@section('title') Home @endsection
@section('description') Website for selling jewelry @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')
    <!-- ***** Preloader Start ***** -->
    <div id="preloader">
        <div class="jumper">
            <div></div>
            <div></div>
            <div></div>
        </div>
    </div>
    <!-- ***** Preloader End ***** -->

    <!-- Page Content -->
    <!-- Banner Starts Here -->
    <div class="banner header-text">
        <div class="owl-banner owl-carousel">
            <div class="banner-item-01">
                <div class="text-content">

```

```

        <h4>For you</h4>
        <h2>Stylish Rings</h2>
    </div>
</div>
<div class="banner-item-02">
    <div class="text-content">
        <h4>We offer</h4>
        <h2>High quality bracelets</h2>
    </div>
</div>
<div class="banner-item-03">
    <div class="text-content">
        <h4>Easy buy</h4>
        <h2>Affordable necklaces</h2>
    </div>
</div>
</div>
</div>
<!-- Banner Ends Here -->

<div class="latest-products">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="section-heading">
                    <h2>Latest Products</h2>
                    <a href="{ route('products') }" >view all products <i
class="fa fa-angle-right"></i></a>
                </div>
            </div>
            @foreach($products as $product)
                <div class="col-md-4">
                    @include('partials.product')
                </div>
            @endforeach
        </div>
    </div>
</div>

<div class="best-features">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="section-heading">
                    <h2>About Bling On</h2>
                </div>
            </div>
            <div class="col-md-6">
                <div class="left-content">
                    <p>Since inception, Bling On has been committed to quality, not
only of our products, but of the conditions in which they are produced. We
aim to trace each of our pieces from mine to market because traceability
allows us to manage and improve the social and environmental impacts of our
supply chain</p>
                    <p>We strike a balance between recycled and fairly mined
materials that support communities dependent on the industry. Traceability is
important because it allows us to validate the authenticity of the material

```

```

and improve the social and environmental impacts of our pieces.</p>
    <p>Our jewelry is handcrafted with the same top materials and
quality craftsmanship as other luxury jewelry brands but without the
traditional markups.</p>
    </div>
</div>
<div class="col-md-6">
    <div class="right-image">
        
    </div>
</div>
</div>
</div>
</div>
@endsection

```

LoginRegister.blade.php

```

@extends('layouts.layout')
@section('title') Login or Register @endsection
@section('description') Products for sale @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')

    <div class="row" id="prazno"></div>
    <div class="container mt-5">
        <div class="row d-flex justify-content-between" id="forms">
            <div class="col-md-5" id="register">
                <h2>Register</h2>
                <form action="{{ route('register') }}" method="post"
class="mt-3">
                    <div class="mb-3">
                        <label for="registerName" class="form-label">First
name</label>
                        <input type="text" class="form-control"
id="registerName" name="registerName"/>
                        <p class="text-danger" id="nameRegError"></p>
                    </div>
                    <div class="mb-3">
                        <label for="registerLastName" class="form-label">Last
name</label>
                        <input type="text" class="form-control"
id="registerLastName" name="registerLastName"/>
                        <p class="text-danger" id="lastNameRegError"></p>
                    </div>
                    <div class="mb-3">
                        <label for="address" class="form-
label">Address</label>
                        <input type="text" class="form-control" id="address"
name="address"/>
                        <p class="text-danger" id="addressRegError"></p>
                    </div>
                    <div class="mb-3">

```

```

        <label for="emailRegister" class="form-
label">Email</label>
        <input type="email" class="form-control"
id="emailRegister" name="emailRegister"/>
        <p class="text-danger" id="emailRegError"></p>
    </div>
    <div class="mb-3">
        <label for="passRegister" class="form-
label">Password</label>
        <input type="password" class="form-control"
id="passRegister" name="passRegister"/>
        <p class="text-danger" id="passRegError"></p>
    </div>
    <div class="mb-3">
        <label for="confirmPass" class="form-label">Confirm
password</label>
        <input type="password" class="form-control"
id="confirmPass" name="confirmPass"/>
        <p class="text-danger" id="confPassRegError"></p>
    </div>
    <input type="hidden" id="role_id" name="role_id"
value="2" />
    <button type="button" class="btn btn-lg"
id="registerBtn">Register</button>
    </form>
</div>
<div class="col-md-5">
    <h2>Login</h2>
    <form action="{{ route('login') }}" method="POST" class="mt-
3">
        <div class="mb-3">
            <label for="emailLogin" class="form-
label">Email</label>
            <input type="email" class="form-control"
id="emailLogin" name="emailLogin" />
            <p class="text-danger" id="emailLoginError"></p>
        </div>
        <div class="mb-3">
            <label for="passLogin" class="form-
label">Password</label>
            <input type="password" class="form-control"
id="passLogin" name="passLogin" />
            <p class="text-danger" id="passLoginError"></p>
        </div>
        <button type="button" class="btn btn-lg"
id="loginBtn">Login</button>
    </form>
</div>
</div>
</div>
@endsection

```

\Resources\Views\Pages\Products

Products.blade.php

```
@extends('layouts.layout')
@section('title') Products @endsection
@section('description') Products for sale @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')

    <!-- ***** Preloader Start ***** -->
    <div id="preloader">
        <div class="jumper">
            <div></div>
            <div></div>
            <div></div>
        </div>
    </div>
    <!-- ***** Preloader End ***** -->

    <!-- Page Content -->
    <div class="page-heading products-heading header-text">
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <div class="text-content">
                        <h4>bling on</h4>
                        <h2>products</h2>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="products">
        <div class="container">
            <form method="get" action="{{ route('products') }}">
                <div class="row">
                    <div class="col-md-12">
                        <div class="row">
                            <div class="col-md-2">
                                <h5><a href="#" id="pokaziFilter">Filter <i
class="fa fa-plus-square-o" aria-hidden="true"></i></a> </h5>
                            </div>
                            <div class="col-md-1">
                                <h5 class="mb-2" id="sortLabel">Sort</h5>
                            </div>
                            <div class="col-md-3">
                                <select id="sort" name="sort">
                                    <option value="0">Choose...</option>
                                    <option value="lowToHigh">Price lowest to
highest</option>
                                    <option value="highToLow">Price highest to
lowest</option>
                                </select>
                            </div>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>

</div>
```



```

                                <option value="highestRating">Best
rated</option>
                                </select>
                                </div>
                                <div class="col-md-1 ml-2">
                                    <h5 class="mb-2 mr-3"
id="searchLabel">Search</h5>
                                </div>
                                <div class="col-md-3">
                                    <input type="text" id="search" name="search" />
                                </div>
                                </div>
                                <div class="row mt-4" id="filter">
                                    <div class="col-md-2">
                                        <h5 class="mb-2">Brand</h5>
                                        <ul>
                                            @foreach($brands as $brand)
                                                <li><input type="checkbox" value="{{
$brand->id }}" name="brands" /> {{ $brand->name }}</li>
                                            @endforeach
                                        </ul>
                                    </div>
                                    <div class="col-md-2">
                                        <h5 class="mb-2">Type</h5>
                                        <ul>
                                            @foreach($types as $type)
                                                <li><input type="checkbox" value="{{
$type->id }}" name="types" /> {{ $type->type }}</li>
                                            @endforeach
                                        </ul>
                                    </div>
                                    <div class="col-md-2">
                                        <h5 class="mb-2">Color</h5>
                                        <ul>
                                            @foreach($colors as $color)
                                                <li><input type="checkbox" value="{{
$color->id }}" name="colors" /> {{ $color->name }}</li>
                                            @endforeach
                                        </ul>
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="col-md-1 mt-3">
                                        <button type="button" class="btn"
id="filterBtn">Apply</button>
                                    </div>
                                    @if(session()->has('user') && session('user')->role_id == 1)
                                        <div class="col-md-1 mt-3">
                                            <a type="button" class="btn"
id="addProduct" href="{{ route('products.create') }}">Insert new product</a>
                                        </div>
                                    @endif
                                </div>
                            </div>
                        </div>
                    </div>
                </form>

```

```

<div class="row d-flex justify-content-between mt-5">
  <div class="col-md-12">
    <div class="filters-content" id="products">
      <div class="row grid" >
        @foreach($products as $product)
          <div class="col-lg-4 col-md-4 all des">
            @include('partials.product')
          </div>
        @endforeach
      </div>
    </div>
  </div>
</div>
{{ $products->links('pagination.bootstrap-4') }}
</div>
</div>
@endsection

```

Single.blade.php

```

@extends('layouts.layout')
@section('title') Product @endsection
@section('description') Website for selling jewelry @endsection
@section('keywords') jewelry, sell, online, rings, bracelet @endsection
@section('content')

  <div class="row" id="prazno"></div>
  <!-- Open Content -->
  <section>
    <div class="container pb-5">
      <div class="row">
        <div class="col-lg-5 mt-5">
          <div class="card mb-3">
            
          </div>
          <div class="row">
            <div id="multi-item-example" class="col-10 carousel
slide carousel-multi-item" data-bs-ride="carousel">
              <div class="carousel-inner product-links-wap"
role="listbox">
                <div class="carousel-item active">
                  <div class="row">
                    @foreach($photos as $photo)
                      <div class="col-4">
                        <a href="#">
                          alt }}">
                        </a>
                      </div>

```

```

                                @endforeach
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    <!-- col end -->
    <div class="col-lg-7 mt-5">
        <div class="card border-0">
            <div class="card-body">
                <h1 class="h2">{{ $product->name }}</h1>
                <p class="h3 py-2">${{ $product->price }}</p>
                @foreach($grades as $grade)
                    @if($grade->grade > 0)
                        <p class="py-2">
                            <i class="fa fa-star text-
warning"></i>
                                <span class="list-inline-item text-
dark"> {{ $grade->grade }}/5</span>
                            </p>
                        @endif
                    @endforeach
                <ul class="list-inline">
                    <li class="list-inline-item">
                        <h6 class="mt-3">Brand:</h6>
                    </li>
                    <li class="list-inline-item">
                        <p class="text-muted"><strong>{{
$product->brand }}</strong></p>
                    </li>
                </ul>

                <h6 class="mt-3 mb-2">Description:</h6>
                <p>{{ $product->description }}</p>
                @if(session()->has('user'))
                    @if(!is_null($singleGrade))
                        <h6 class="mt-3">Your rating:
{{ $singleGrade->grade}} <i class="fa fa-star text-warning mr-1"></i> </h6>
                    @else
                        <ul class="list-inline">
                            <li class="list-inline-item">
                                <h6 class="mt-3">Rate:</h6>
                            </li>
                            <li class="list-inline-item">
                                <select id="rating" name="rating"
@if(!session()->has('user')) disabled="disabled" @endif>
                                    <option value="1">1</option>
                                    <option value="2">2</option>
                                    <option value="3">3</option>
                                    <option value="4">4</option>
                                    <option value="5">5</option>
                                </select>
                            </li>
                        </ul>
                    @endif
                @endif
            </div>
        </div>
    </div>

```

```

        <form action="" method="GET">
            @if(!session()->has('user'))
                <div class="row">
                    <div class="col-auto">
                        <p class="text-danger mt-3"> You
must login to order the product! </p>
                    </div>
                </div>
            @endif
            <div class="row">
                <div class="col-auto">
                    <ul class="list-inline pb-3 mt-4">
                        @if(count($sizes) != null)
                            <li class="list-inline-
item">Size : </li>
                                @foreach($sizes as $size)
                                    @if($size->quantity > 0)
                                        <li class="list-
inline-item"><input type="radio" data-idsize="{{ $size->id }}" value="{{ $size-
>size }}" class="size" name="size" @if(!session()->has('user'))
disabled="disabled" @endif /> {{ $size->size }} </li>
                                    @endif
                                @endforeach
                            @else
                                <li class="list-inline-item
text-danger">Out of stock! </li>
                            @endif
                        </ul>
                    </div>
                    @if(count($sizes) != null)
                        <div class="col-auto">
                            <ul class="list-inline pb-3 mt-3">
                                <li class="list-inline-item text-
right">
                                    Quantity
                                    <input type="hidden"
name="product-quanlity" id="product-quanlity" value="1">
                                </li>
                                <li class="list-inline-
item"><span class="btn" id="btn-minus">-</span></li>
                                <li class="list-inline-
item"><span class="badge" id="var-value">1</span></li>
                                <li class="list-inline-
item"><span class="btn" id="btn-plus">+</span></li>
                            </ul>
                            <input type="hidden" value="{{
$product->id }}" name="product_id" id="product_id">
                        </div>
                    @endif
                </div>
                @if(count($sizes) != null)
                    <div class="row pb-3">
                        <div class="col d-grid">
                            <button type="button" class="btn btn-
lg float-right addToCart" name="submit" data-id="{{ $product->id }}"
@if(!session()->has('user')) disabled="disabled" @endif>Add To Cart</button>
                        </div>

```

```

        </div>
        @endif
    </form>
</div>
</div>
</div>
</div>
</div>
@if(session()->has('user') && session('user')->role_id == 1)
    <div class="row mt-5">
        <div class="col-lg-12">
            <a class="btn btn-lg float-right updateBtn ml-3"
name="updateBtn" href="{ route('products.edit', $product->id) }">Update</a>
            <form action="{ route('products.destroy', $product-
>id) }" method="post">
                @csrf
                @method('DELETE')
                <button type="submit" class="btn btn-lg float-
right deleteBtn" name="deleteBtn" >Delete</button>
            </form>
        </div>
    </div>
</div>
@endif
</div>
</section>
@endsection

```

\Resources\Views\Pagination

Bootstrap-4.blade.php

```

@if ($paginator->hasPages())
    <nav>
        <ul class="pagination justify-content-center">

            {{-- Pagination Elements --}}
            @foreach ($elements as $element)

                {{-- Array Of Links --}}
                @if (is_array($element))
                    @foreach ($element as $page => $url)
                        @if ($page == $paginator->currentPage())
                            <li class="page-item active"><a class="page-link"
id="link{{$page}}" data-page="{{$page}}" href="#">{{$page}}</a></li>
                        @else
                            <li class="page-item"><a class="page-link"
id="link{{$page}}" data-page="{{$page}}" href="#">{{$page}}</a></li>
                        @endif
                    @endforeach
                @endif
            @endforeach

        </ul>
    </nav>
@endif

```

```
</nav>
@endif
```

\Resources\Views\Partials

Email.blade.php

```
<p>Hi, this is {{ $data['name'] }}</p>
<p>I have some questions: {{ $data['message'] }}</p>
```

Product.blade.php

```
<div class="product-item">
    <a href="{{ route('product', ['id' => $product->id]) }}">name
 }}"></a>
    <div class="down-content">
        <a href="{{ route('product', ['id' => $product->id]) }}"><h4>{{
$product->name }}</h4></a>
        <h6>{{ $product->price }}</h6>
        @if($product->grade)
            <ul class="stars">
                <li>
                    <i class="fa fa-star "></i> {{ $product->grade }}/5
                </li>
            </ul>
        @else
            <ul class="stars">
                <li><i class="fa fa-star mr-2"></i>Be first to rate!</li>
            </ul>
        @endif
    </div>
</div>
```

Success.blade.php

```
@if(session()->has('success'))
    <div class="container mt-4">
        <div class="alert alert-success" role="alert">
            {{ session('success') }}
        </div>
    </div>
@endif
```

\Routes

Web.php

```
<?php

use \App\Http\Controllers\HomeController;
use \App\Http\Controllers\ProductController;
use \App\Http\Controllers>ContactController;
use \App\Http\Controllers\LoginController;
use \App\Http\Controllers\RegistrationController;
use \App\Http\Controllers\RateProductController;
use \App\Http\Controllers\CartController;
use \App\Http\Controllers\AdminController;
use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [HomeController::class, 'index'])->name('home');

Route::get('/products', [ProductController::class, 'index'])->
>name('products');
Route::get('/products/create', [ProductController::class, 'create'])->
>name('products.create')->middleware('isAdmin');
Route::post('/products', [ProductController::class, 'store'])->
>name('products.store')->middleware('isAdmin');
Route::get('/products/filter', [ProductController::class, 'sortFilter'])->
>name('sortFilter');
Route::get('/products/{id}', [ProductController::class, 'single'])->
>name('product');
Route::get('/products/{id}/edit', [ProductController::class, 'edit'])->
>name('products.edit')->middleware('isAdmin');
Route::put('/products/{id}', [ProductController::class, 'update'])->
>name('products.update')->middleware('isAdmin');
Route::delete('/products/{id}', [ProductController::class, 'destroy'])->
>name('products.destroy')->middleware('isAdmin');

Route::get('/contact', [ContactController::class, 'index'])->name('contact');
Route::POST('/sendEmail', [ContactController::class, 'sendEmail'])->
>name('sendEmail');

Route::get('/loginRegister', [LoginController::class, 'index'])->
>name('loginRegister');
Route::POST("/login", [LoginController::class, "login"])->name("login");
Route::get("/logout", [LoginController::class, "logout"])->name("logout");

Route::POST("/register", [RegistrationController::class, "register"])-
>name("register");
```

```

Route::POST('/rate', [RateProductController::class, 'rate'])->name('rate')->
>middleware('isUser');

Route::get('/cart', [CartController::class, 'index'])->name('cart')->
>middleware('isUser');
Route::POST('/cart/addToCart', [CartController::class, 'addToCart'])->
>name('addToCart')->middleware('isUser');
Route::get('/showCart', [CartController::class, 'showCart'])->
>name('showCart')->middleware('isUser');
Route::get('/cart/showBought', [CartController::class, 'bought'])->
>name('bought')->middleware('isUser');
Route::post('/cart/buy', [CartController::class, 'buy'])->name('buy')->
>middleware('isUser');
Route::delete('/cart/delete', [CartController::class, 'delete'])->
>name('delete')->middleware('isUser');

Route::get('/admin', [AdminController::class, 'index'])->name('admin')->
>middleware('isAdmin');
Route::get('/date', [AdminController::class, 'admin'])->name('date')->
>middleware('isAdmin');

```