

Akademija tehničko-umetničkih strukovnih studija Beograd
Odsek Visoka škola za informacione i komunikacione tehnologije

Dokumentacija za sajt
Clocky Shop

Predmet: Web
programiranje PHP 2
Student: Stevan Zubovic
33/18

Table of Contents

Uvod.....	3
Korišćene tehnologije.....	3
Opis funkcionalnosti.....	4
Baza podataka.....	5
Strane sajta I njihov opis.....	6
Home strana.....	7
Product strana.....	9
Admin strana.....	9
Login strana.....	11
Register strana.....	12
Cart strana.....	12
Kodovi.....	13
AdminController.php.....	13
Controller.php.....	16
HomeController.php.....	17
LoginController.php.....	17
ProductController.php.....	18
PurchaseController.php.....	19
RegistrationController.php.....	20
UserController.php.....	21
isAdmin.php.....	21
IsAuth.php.....	22
Product.php.....	23
Price.php.....	23
Image.php.....	23
Discount.php.....	24
Home.blade.php.....	24
Admin.php.....	26
Login.blade.php.....	27
SingleProduct.blade.php.....	28
layout.blade.php.....	28
formPageLayout.blade.php.....	29
head.blade.php.....	29
menu.blade.php.....	29
register.blade.php.....	30
scripts.blade.php.....	31
web.php.....	31
main.js.....	32
style.css.....	44

Uvod

Korišćene tehnologije

- HTML5
- CSS3

- PHP
- AJAX
- SQL
- GIMP
- LARAVEL
- ELOQUENT

Opis funkcionalnosti

- Mogućnosti registracije, autorizacije i autentifikacije korisnika
- Dinamičko ispisivanje svih relevantnih delova sajta iz baze podataka
- Strana za administratore pomoću koje se može upravljati bazom podataka
- Provera podataka koji dolaze od strane korisnika i izveštavanje o eventualnim greskama
- Mogućnost stvaranja narudžbine sa proizvoljnim proizvodima i njeno čuvanje u bazi podataka

Baza podataka

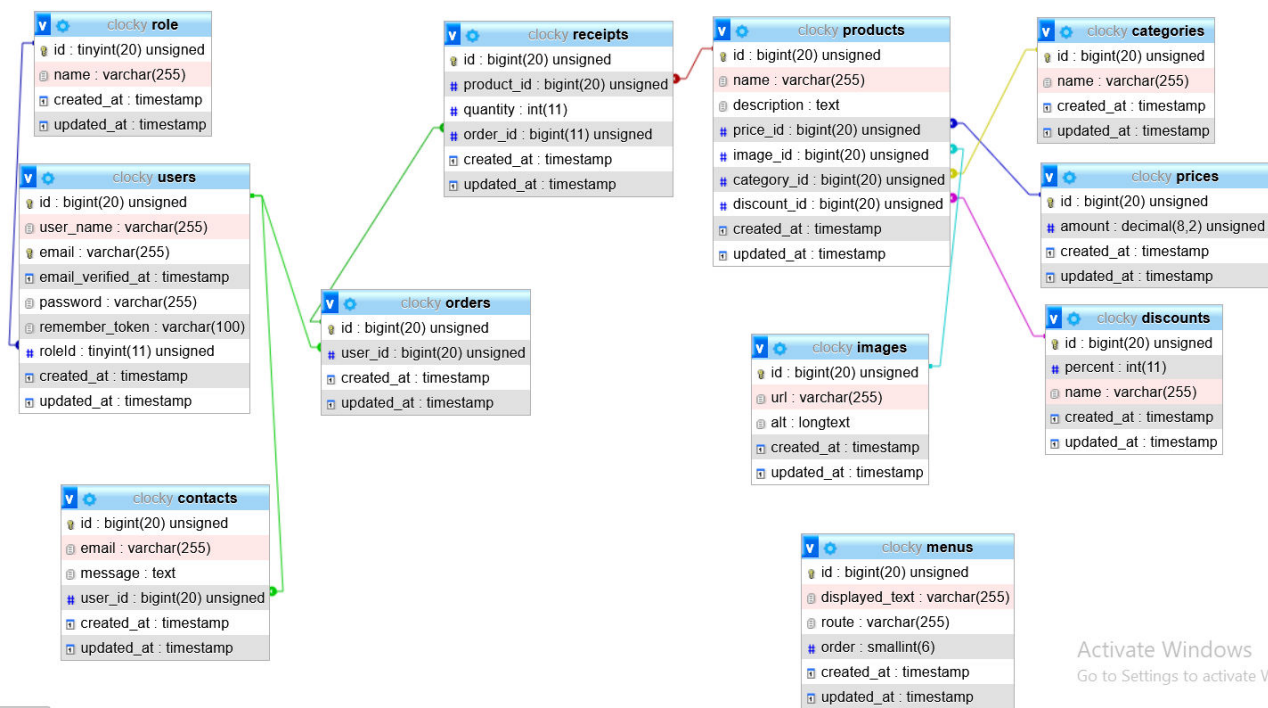


Figure 1: Dizajn baze podataka

Strane sajta I njihov opis

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://Clocky-shop.com/home</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>daily</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>http://Clocky-shop.com/Login</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.6</priority>
  </url>
  <url>
    <loc>http://Clocky-shop.com/register</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.6</priority>
  </url>
  <url>
    <loc>http://Clocky-shop.com/panel</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.9</priority>
  </url>
  <url>
    <loc>http://Clocky-shop.com/product</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.9</priority>
  </url>
  <url>
    <loc>http://Clocky-shop.com/cart</loc>
    <lastmod>2022-05-06</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.9</priority>
  </url>
</urlset>
```

Home strana

Na home strani se nalaze proizvodi dinamički učitani iz baze podataka. Postoji mogućnost pretraživanja proizvoda. Takođe sadrži formu pomoću koji korisnik može da stupi u kontakt sa administratorina sajta.

Clocky shop

One stop shop for all your clock needs

☐ Cuckoo
☐ Wall
 Sort by price
 Order by price ▼
 Search



Clock Peddler 8 Day
Mechanical Chalet
Cuckoo Clock 27cm
By TRENKLE

620.00-€ 496 €



4 Leaf & Bird 1 Day
Mechanical Carved
Cuckoo Clock 20cm
By SCHNEIDER

997.00 €



Heidi House Battery
Chalet Cuckoo Clock
With Edelweiss
Flowers 21cm By
TRENKLE

380.00 €



1/4 Hour 1 Day
Mechanical Chalet
Cuckoo Clock 20cm
By ENGSTLER

620.00-€ 496 €



5 Leaf & Bird 1 Day
Mechanical Carved
Cuckoo Clock 20cm
By HÖNES

380.00 €



Dog & Water Trough
1 Day Mechanical
Chalet Cuckoo Clock
22cm By TRENKLE

315.00 €



Alphorn Battery
Chalet Cuckoo Clock
19cm By ENGSTLER

620.00 €



Bears Battery Chalet
Cuckoo Clock 25cm
By ENGSTLER

330.00 €



Black Cube 8 Day
Mechanical Modern
Cuckoo Clock With
Clear Front 26cm By
ROMBA

970.00-€ 824.5 €



12.5cm Black
Nautical Quartz
Clock By FISCHER

620.00 €



18cm Brass Ship's
Bell Nautical Quartz
Clock By HERMLE

490.00 €



26cm Alma Pink
Children's Time
Teaching Wall Clock
By ACCTIM

550.00 €



29cm Black & White
Square Glass Wall
Clock By AMS

240.00 €



30cm Black Silent
Modern Wall Clock
By BRAUN

620.00 €



34cm Beige Owl
Gufino Modern Wall
Clock By ARTI E
MESTIERI

240.00-€ 192 €



43cm Stella Silver
Starburst Wall Clock
By ACCTIM

380.00-€ 323 €

Email Enquiry Form

Fill out this form to email us your enquiry

Email

admin@admin.com

Message

Message 1

Submit

Product strana

Na product strani nalaze se informacije o pojedinačnom proizvodu i moguće ga je dodati u „korpu“.

[Home](#) [Logout](#) [Admin](#)



Clock Peddler 8 Day Mechanical Chalet Cuckoo Clock
27cm By TRENKLE

620.00 €

Add to cart

Paying Homage To Artisans From A Time Long Since Past This lovely 8 Day Mechanical Cuckoo Clock represents a genuine, high quality hand-crafted Black Forest cuckoo clock, produced by artisan clock makers TRENKLE from Simonswald in the Black Forest region of Germany. Measuring 27cm in height*, this beautiful clock portrays a delightful scene in the bavarian countryside, featuring a chalet house with a unique facade consisting of both brick and wood. Out the front of the chalet includes many features hand carved out of timber, including a dog and a water trough, as well as a pine tree. One unique feature of this clock is a clock peddler who features on the right hand side of the clock. The story of the clock peddler dates back hundreds of years where the artisans would spend long winters crafting cuckoo clocks, then travel far and wide in the spring to sell their unique creations. Today the clock peddler is celebrated as a prominent symbol of the successes that the cuckoo clock industry has seen throughout the years, and are often symbolised in many cuckoo clock designs, paying homage to artisans from a time long since past. This fascinating scene also includes a wooden clock dial in the centre of the chalet, and hand painted details including red window shutters and flowers underneath the windowsills. Chalet style cuckoo clocks are the most famous and common type of cuckoo clock, and are often featured in the shape of alpine houses. They usually feature carvings that represent life in the mountain regions of Europe (mainly around Southern Germany, The Black Forest, Switzerland and Austria). Chalet style cuckoo clocks can range from simple arrangements of carved features to very detailed scenes with many moving parts (known as automaton). This clock features both hand crafted and hand carved details to provide a truly unique timepiece that is sure to spark a sense of joy. The in-built cuckoo of this clock calls the time each hour and once at half past through a door that opens, with a manual shut-off mode. This particular clock needs to be wound once a week and does not include music. This beautifully designed clock by TRENKLE is a charming timepiece, suitable for those looking for a quality traditional cuckoo clock using carving methods that have been practised for hundreds of years.

Admin strana

Na admin strani korisnici sa odgovarajućim privilegijama mogu upravljati proizvodima, njihovim kategorijama, cenama, popustima kao i korisnicima sajta.

[Products](#)[Categories](#)[Discounts](#)[Users](#)[New product](#)

Product Id	Product name	Price	Delete	Edit //TO DO
13	29cm Black & White Square Glass Wall Clock By AMS	240.00 €	Delete	Edit
15	34cm Beige Owl Gufino Modern Wall Clock By ARTI E MESTIERI	240.00 €	Delete	Edit
6	Dog & Water Trough 1 Day Mechanical Chalet Cuckoo Clock 22cm By TRENKLE	315.00 €	Delete	Edit
8	Bears Battery Chalet Cuckoo Clock 25cm By ENGSTLER	330.00 €	Delete	Edit
3	Heidi House Battery Chalet Cuckoo Clock With Edelweiss Flowers 21cm By TRENKLE	380.00 €	Delete	Edit
5	5 Leaf & Bird 1 Day Mechanical Carved Cuckoo Clock 20cm By HÖNES	380.00 €	Delete	Edit
16	43cm Stella Silver Starburst Wall Clock By ACCTIM	380.00 €	Delete	Edit
11	18cm Brass Ship's Bell Nautical Quartz Clock By HERMLE	490.00 €	Delete	Edit
12	26cm Alma Pink Children's Time Teaching Wall Clock By ACCTIM	550.00 €	Delete	Edit
1	Clock Peddler 8 Day Mechanical Chalet Cuckoo Clock 27cm By TRENKLE	620.00 €	Delete	Edit
4	1/4 Hour 1 Day Mechanical Chalet Cuckoo Clock 20cm By ENGSTLER	620.00 €	Delete	Edit
7	Alphorn Battery Chalet Cuckoo Clock 19cm By ENGSTLER	620.00 €	Delete	Edit
10	12.5cm Black Nautical Quartz Clock By FISCHER	620.00 €	Delete	Edit
14	30cm Black Silent Modern Wall Clock By BRAUN	620.00 €	Delete	Edit
9	Black Cube 8 Day Mechanical Modern Cuckoo Clock With Clear Front 26cm By ROMBA	970.00 €	Delete	Edit
2	4 Leaf & Bird 1 Day Mechanical Carved Cuckoo Clock 20cm By SCHNEIDER	997.00 €	Delete	Edit

Login strana

Na login strani ranije registrovani korisnici se mogu autentifikovati.

[Home](#) [Login](#) [Register](#)

User name

Password

Submit

Register strana

Strana za registraciju novih korisnika

[Home](#) [Login](#) [Register](#)

User name

Email

Password

Submit

Cart strana

Na kart strani korisnik može videti proizvode koje je u nju ranije dodao. Klikom na checkout dugme korpa se prazni i porudžbina se čuva u bazi podataka.

Home Logout Author Cart Docs				
Product name	Quantity	Total	Discount	Delete
Heidi House Battery Chalet Cuckoo Clock With Edelweiss Flowers 21cm By TRENKLE	1	380 €	/	Remove
26cm Alma Pink Children's Time Teaching Wall Clock By ACCTIM	1	550 €	/	Remove
29cm Black & White Square Glass Wall Clock By AMS	1	240 €	/	Remove
34cm Beige Owl Gufino Modern Wall Clock By ARTI E MESTIERI	2	384 €	mayday - 20%	Remove

Checkout

Kodovi

AdminController.php

```
<?php
namespace App\Http\Controllers;

use App\Models\Category;
use App\Models\Discount;
use App\Models\Price;
use App\Models\Product;
```

```
use App\Models\User;
use http\Client\Response;
use Illuminate\Http\Request;
use phpDocumentor\Reflection\Exception\PcreException;
```

```
class AdminController extends Controller
```

```
{
    public function index() {

        $this->data['products'] = Product::all();
        $this->data['categories'] = Category::all();
        $this->data['prices'] = Price::all();
        $this->data['discounts'] = Discount::all();
        return view('admin', $this->data);
    }
}
```

```
public function getDiscounts() {
    return json_encode(Discount::all());
}
```

```
public function getUsers() {
    return json_encode(User::all());
}
```

```
public function getCategories() {
    return json_encode(Category::all());
}
```

```
public function deleteCategory(Request $request) {
    try {
        $id = $request->get('id');
        $category = Category::find($id);
        if($category->delete()){
            return ['message' => 'Category deleted'];
        }
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Category could not be deleted'],403);
    }
}
```

```
public function deleteUser(Request $request) {
    try {
        $user = User::find($request->get('id'));
        $user->delete();
        return response()->json(['message' => 'User deleted'],200);
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'User could not be deleted'],403);
    }
}
```

```
}
```

```
public function tryCreateNewProduct(Request $request) {
    try {
        $product = new Product();
```

```
        $product->name = $request->get('name');
        $product->discount_id = $request->get('discount');
```

```
$product->price_id = $request->get('price');
$product->description = $request->get('description');
$product->category_id = $request->get('category');
$product->image_id = 1;
```

```
if($product->save()) {
    return ['message' => 'Product created successfully'];
} else {
    //TODO redirect to error page
}
```

```
} catch (\PDOException $exception) {
```

```
}
```

```
public function createCategory(Request $request) {
    try {
        $category = new Category();
        $category->name = $request->get('name');
        return $category->save();
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Category could not be created'],503);
    }
}
```

```
public function tryDeleteDiscount(Request $request) {
    try {
        $id = $request->get('id');
        $discount = Discount::find($id);
        if($discount->delete()) {
            return response()->json(['message' => 'Discount deleted'],200);
        }
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Discount could not be deleted'],403);
    }
}
```

```
public function createDiscount(Request $request) {
    try {
        $discount = new Discount();
        $discount->name = $request->get('name');
        $discount->percent = $request->get('percent');
        $discount->save();
        return response()->json(['message' => 'Discount created'],200);
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Discount could not be created'],503);
        //return $exception;
    }
}
```

```
public function editDiscount(Request $request) {
    try {
        $discount = Discount::find($request->get('id'));
        $discount->name = $request->get('name');
        $discount->percent = $request->get('percent');
        $discount->save();
        return response()->json(['message' => 'Discount edited'],200);
    }
}
```

```

    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Discount could not be edited'],503);
    }
}

```

```

public function editProduct(Request $request) {
    try {
        $product = Product::find($request->get('id'));
        $product->name = $request->get('name');
        $product->category_id = $request->get('category');
        $product->price_id = $request->get('price');
        $product->discount_id = $request->get('discount');
        $product->description = $request->get('description');
        $product->save();
        return response()->json(['message' => 'Product edited'],200);
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Could not edit product'],503);
    }
}

```

```

public function editCategory(Request $request) {
    try {
        $category = Category::find($request->get('id'));
        $category->name = $request->get('name');
        $category->save();
        return response()->json(['message' => 'Category edited'],200);
    } catch (\PDOException $exception) {
        return response()->json(['message' => 'Could not edit category'],503);
    }
}

```

```

    }
}

```

Controller.php

```
<?php
```

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

```

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;
use App\Models\Menu;

```

```

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;

```

```
    protected array $data;
```

```
    function __construct() {
```



```

        $this->data['menu'] = Menu::all();
    }
}

```

HomeController.php

```
<?php
```

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

```

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

```

```

class HomeController extends Controller
{
    public function index() {

```

```

        $categories = Category::all();
        $products = Product::with('discount')->with('price')->with('image')->get();

```

```

        $this->data['products'] = $products;
        $this->data['categories'] = $categories;
        return view('home', $this->data);
    }
}

```

LoginController.php

```
<?php
```

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

```

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

```

```

class LoginController extends Controller
{
    public function index() {
        return view('login', $this->data);
    }

```

```

    public function logout() {
        session()->forget('user');
        return redirect()->route('home');
    }

```

```

    public function tryLogin(Request $request) {
        $username = $request->get('username');
        $password = $request->get('password');

```

```

        $user = User::all()->where('user_name', '=', $username)->first();

```

```

        if($user) {
            if (password_verify($password, $user->password)) {
                session()->put('user', $user);
                return redirect()->route('home');
            } else {
                return redirect()->route('login')->with('userOrPassError', 'Username or password incorrect');
            }
        }
        //dd('nije uspeo');
    }
}
}

```

ProductController.php

```
<?php
```

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

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

```
class ProductController extends Controller
```

```

{
    public function filteredProducts(Request $request) {
        try {
            $categoryIds = $request->get('categoryIds');
            $sortType = $request->get('sortType');
            $search = $request->get('searchTerm');
            if($sortType !== 'asc' && $sortType !== 'desc') {
                $sortType = 'asc';
            }
            $products = Product::select('products.*', 'prices.amount as amount')
                ->join('prices', 'products.price_id', '=', 'prices.id')
                ->with('image')
                ->with('discount')
                ->when($categoryIds, function($query, $categoryIds) {
                    $query->whereIn('category_id', $categoryIds);
                })
                ->when($search, function($query, $search) {
                    $query->where('name', 'LIKE', "%{$search}%");
                })
                ->orderBy('amount', $sortType)
                ->get();

```

```
        return json_encode($products);
```

```

    } catch(\PDOException $exception) {
        //TODO
        return $exception;
    }

```

```
}
```

```

public function index($id) {
    try {
        $product = Product::find($id);
        $this->data['product'] = $product;
        return view('singleProduct', $this->data);
    } catch (\PDOException $exception) {

    }
}

```

```

public function tryDeleteProduct(Request $request) {
    try {
        $id = $request->get('id');
        $product = Product::find($id);
        $success = $product->delete();
        if($success) {
            return ['message' => 'success'];
        } else {
            http_response_code(500);
        }
    } catch (\PDOException $exception) {

    }
}

```

PurchaseController.php

```
<?php
```

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

```

use App\Models\Order;
use App\Models\Receipt;
use Illuminate\Http\Request;

```

```

class PurchaseController extends Controller
{
    public function createPurchase(Request $request) {
        try {
            if (session()->has('user')) {
                $userId = session()->get('user')->id;
                $order = new Order();
                $order->user_id = $userId;
                $order->save();
                $products = $request->get('data');
                foreach ($products as $product) {
                    $receipt = new Receipt();
                    $receipt->product_id = $product['id'];
                    $receipt->quantity = $product['quantity'];
                    $receipt->order_id = $order->id;
                    $receipt->save();
                }
                return response()->json(['message' => 'Thank you for shopping with us'],200);
            }
        }
    }
}

```

```

    }
    return response()->json(['message' => 'Thank you for shopping with us'],200);
} catch(PDOException $exception) {
    return response()->json(['message' => 'Thank you for shopping with us'],500);
    //return $exception;
}
}
}

```

RegistrationController.php

```
<?php
```

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

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

```
use App\Models\User;
```

```
class RegistrationController extends Controller
```

```
{
```

```
    public function index() {
```

```
        return view('fixed.register', $this->data);
```

```
    }
```

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

```
    {
```

```
        try {
```

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

```
            $userName = $request->get('username');
```

```
            $email = $request->get('email');
```

```
            $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
```

```
            $RegErrors = [];
```

```
            if (User::all()->where('email', '=', $email)->first()) {
```

```
                $RegErrors['email'] = 'Given address already exists';
```

```
            }
```

```
            if (User::all()->where('user_name', '=', $userName)->first()) {
```

```
                $RegErrors['userName'] = 'Given user name already exists';
```

```
            }
```

```
            if (count($RegErrors)) {
```

```
                return redirect()->back()->with($RegErrors);
```

```
            }
```

```
            $user = new User();
```

```
            $user->user_name = $userName;
```

```
            $user->email = $email;
```

```
            $user->password = $hashedPassword;
```

```
            $user->roleId = 1;
```

```

        $success = $user->save();
        if ($success) {
            return redirect()->route('home');
        }
    } catch (\PDOException $exception) {
        return redirect()->route('error');
    }
}

```

```

        //TO DO
    }
}

```

UserController.php

```
<?php
```

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

```
use App\Models\Contact;
use Illuminate\Http\Request;
```

```
class UserController extends Controller
{
    public function index() {
        return view('profile');
    }
}

```

```

    public function insertContact(Request $request) {
        try {
            $contact = new Contact();

```

```

            $contact->message = $request->get('message');
            $contact->email = $request->get('email');
            if(session()->has('user')) {
                $contact->user_id = session()->get('user')->id;
            }
            $contact->save();
            return response()->json(['message' => 'Thank you for contact us'],200);
        } catch (\PDOException $exception) {
            return response()->json(['message' => 'User could not be deleted'],403);
            //return $exception;
        }
    }
}

```

isAdmin.php

```
<?php
```

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

```

use Closure;
use Illuminate\Http\Request;

class IsAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(session()->get('user')->roleId == 2) {
            return $next($request);
        } else {
            return redirect()->route('home');
        }
    }
}

```

IsAuth.php

```
<?php
```

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

```

use Closure;
use Illuminate\Http\Request;

```

```

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

```

Product.php

```
<?php

namespace App\Models;

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

class Product extends Model
{
    public function image() {
        return $this->belongsTo(Image::class);
    }

    public function price() {
        return $this->belongsTo(Price::class);
    }

    public function discount() {
        return $this->belongsTo(Discount::class);
    }
    use HasFactory;
}
```

Price.php

```
<?php

namespace App\Models;

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

class Price extends Model
{
    public function product() {
        $this->belongsToMany(Product::class);
    }
    use HasFactory;
}
```

Image.php

```
<?php

namespace App\Models;

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

```

class Image extends Model
{
    public function product() {
        return $this->belongsTo(Product::class);
    }
    use HasFactory;
}

```

Discount.php

```
<?php
```

```
namespace App\Models;
```

```

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

```

```

class Discount extends Model
{

    public function product() {
        return $this->belongsTo(Product::class);
    }
    use HasFactory;
}

```

Home.blade.php

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

```

@section('title')
    Clocky shop - Home
@endsection

```

```
@section('main')
```

```

<div class="container">
    <div class="row text-lg-center top-gap bottom-gap">
        <h1>Clocky shop</h1>
        <h3>One stop shop for all your clock needs</h3>
    </div>
    <div class="row">
        <div class="filters col-2">
            <form action="">
                @csrf
            </div>

```

```

        @foreach($categories as $category)
            <div class="form-check">
                <input type="checkbox" class="form-check-input category" id="{{ $category->id }}" value="{{ $category->id }}" />
                <label for="{{ $category->id }}" class="form-check-label">{{ ucfirst($category->name) }}</label>
            </div>

```



```

    @endforeach
  </div>
  <div class="form-group">
    <label for="sort">Sort by price</label>
    <select name="sort-type" id="sort" class="form-select">
      <option value="default">Order by price</option>
      <option value="asc">Ascending</option>
      <option value="desc">Descending</option>
    </select>
  </div>
  <div class="form-group">
    <label for="search">Search</label>
    <input type="text" id="search" name="search">
  </div>
  {{-- <label for="customRange1" class="form-label">Example range</label>--}}
  {{-- <input type="range" class="form-range" id="customRange1">--}}
</form>
</div>

<div class="col-8" id="products">
  <div class="row">
    @foreach($products as $product)
      <div class="col-xl-3 col-lg-4 col-md-6 mb-4">
        <div class="bg-white rounded shadow-sm">
          image->alt }}"
class="img-fluid card-img-top">
          <div class="p-4">
            <h6 class="product-card"> <a href="{{ route('singleProduct', $product->id) }}" class="">{{ $product-
>name }}</a></h6>
            <div class="row">
              @if($product->discount)
                <div class="col-6">
                  <s class="price text-nowrap">{{ $product->price->amount }} &euro;</s>
                </div>
                <div class="col-6">
                  <p class="price text-nowrap">{{ $product->price->amount - (($product->price->amount / 100) *
$product->discount->percent) }} &euro;</p>
                </div>
              @else
                <div class="col-6">
                  <p class="price text-nowrap">{{ $product->price->amount }} &euro;</p>
                </div>
              @endif
            </div>
          </div>
        </div>
      </div>
    @endforeach
    {{ $product->links }}
  </div>
</div>
</div>

```

```
@endsection
```

Admin.php

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

```
@section('title')
```

```
    Clocky shop - Panel
```

```
@endsection
```

```
@section('main')
```

```
    @csrf
```

```
    <div class="popup">
```

```
        <div id="product-edit">
```

```
            <button id="close">&times;</button>
```

```
            <from id="input-form">
```

```
                <input type="text" class="form-control" placeholder="Enter new name...">
```

```
                <select name="category" id="category" class="form-control">
```

```
                    <option value="" selected>Select Category</option>
```

```
                    @foreach($categories as $category)
```

```
                        <option value="{{ $category->id }}">{{ ucfirst($category->name) }}</option>
```

```
                    @endforeach
```

```
                </select>
```

```
                <select name="price" id="price" class="form-control">
```

```
                    <option value="">Select Price</option>
```

```
                    @foreach($prices as $price)
```

```
                        <option value="{{ $price->id }}">{{ $price->amount }} &euro;</option>
```

```
                    @endforeach
```

```
                </select>
```

```
                <select name="discount" id="discount" class="form-control">
```

```
                    <option value="">Select Discount</option>
```

```
                    @foreach($discounts as $discount)
```

```
                        <option value="{{ $discount->id }}">{{ ucfirst($discount->name) }} - {{ $discount->percent }}%</option>
```

```
                    @endforeach
```

```
                </select>
```

```
                <textarea name="description" id="description" cols="30" rows="10" class="form-control" placeholder="New description..."></textarea>
```

```
                <button class="btn btn-success">Submit</button>
```

```
            </from>
```

```
        </div>
```

```
    </div>
```

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

```
        <div class="col-2">
```

```
            <nav class="navbar navbar-expand-lg navbar-light bg-dark flex-column align-items-baseline">
```

```
                <ul id="admin-nav">
```

```
                    <li class="nav-link link-light" id="products">Products</li>
```

```
                    <li class="nav-link link-light" id="categories">Categories</li>
```

```
                    <li class="nav-link link-light" id="discounts">Discounts</li>
```

```
                    <li class="nav-link link-light" id="users">Users</li>
```

```
                </ul>
```

```
            </nav>
```

```
        </div>
```

```
        <div class="col-1" id="insert-new"></div>
```

```
        <div class="col-8" id="admin-display">
```

```
    </div>
```

```

</div>
<div id="info" class="hidden">
  <select id="categories-hidden" class="form-control">
    <option value="" selected>Select Category</option>
    @foreach($categories as $category)
      <option value="{{ $category->id }}">{{ ucfirst($category->name) }}</option>
    @endforeach
  </select>
  <select id="prices-hidden" class="form-control">
    <option value="">Select Price</option>
    @foreach($prices as $price)
      <option value="{{ $price->id }}">{{ $price->amount }} &euro;</option>
    @endforeach
  </select>
  <select id="discounts-hidden" class="form-control">
    <option value="">Select Discount</option>
    @foreach($discounts as $discount)
      <option value="{{ $discount->id }}">{{ ucfirst($discount->name) }} - {{ $discount->percent }}%</option>
    @endforeach
  </select>
</div>
@endsection

```

Login.blade.php

```

@extends('layouts.formPageLayout')

@section('title')
  Clocky shop - Login
@endsection

@section('main')
  <form action="" method="POST" class="col-md-5 container lone-form">
    @csrf
    <div class="form-group">
      <label for="username" class="form-label">User name</label>
      <input type="text" name="username" id="username" class="form-control">
      <div class="invalid-feedback">
        Error text
      </div>
    </div>

    <div class="form-group">
      <label for="password" class="form-label">Password</label>
      <input type="password" name="password" id="password" class="form-control @if(session()-
>has('userOrPassError')) is-invalid @endif">
      <div class="invalid-feedback">
        @if(session()->has('userOrPassError'))
          {{ session('userOrPassError') }}
        @endif
      </div>
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
@endsection

```

SingleProduct.blade.php

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

@section('title')
    Clocky shop - {{ $product->name }}
@endsection

@section('main')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-4 ">
                <figure>
                    image->alt }}">
                    <figcaption>{{ $product->name }}</figcaption>
                </figure>
            </div>
            <div class="col-6 align-self-center">
                <div id="purchase-options" class="row justify-content-between">
                    <input type="number" min="0" value="1" class="col-4">
                    <div class="price col-4">{{ $product->price->amount }} &euro; </div>
                    <button class="btn btn-lg blue col-4">Add to cart</button>
                </div>
            </div>
        </div>
        <div class="row border-top">
            <p>
                {{ $product->description }}
            </p>
        </div>
    </div>
@endsection
```

layout.blade.php

```
@include('fixed.head')

<body>
@include('fixed.menu')

@yield('main')

@include('fixed.footer')

@include('fixed.scripts')
</body>
```

formPageLayout.blade.php

```
@include('fixed.head')

<body>
@include('fixed.menu')

@yield('main')

@include('fixed.scripts')
</body>
```

footer.blade.php

```
<footer class="bg-dark text-center text-lg-start">
    <!-- Copyright -->
    <div class="text-center p-3" style="background-color: rgba(1, 1, 1, 0.7);">
        © 2020 Copyright:
        <a class="link-light" href="{{ route('home') }}">Clocky shop</a>
    </div>
    <!-- Copyright -->
</footer>
```

head.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" type="image/jpg" href="{{ asset('icon.ico') }}" />
    <link href="{{ asset('assets/vendor/bootstrap/css/bootstrap.min.css') }}" rel="stylesheet">
    <link href="{{ asset('assets/styles/style.css') }}" rel="stylesheet">
    <script src="https://kit.fontawesome.com/7878f726d4.js" crossorigin="anonymous"></script>
    <title>@yield('title')</title>
</head>
```

menu.blade.php

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark" role="navigation">
    @foreach($menu as $item)
        @if(session()->has('user') && $item->displayed_text == 'user')
            <a href="{{ route($item->route) }}" class="nav-link link-light flex"> <i class="fa-solid fa-circle-user"></i> </a>
        @elseif(!session()->has('user') && $item->displayed_text == 'user')
            @continue
        @elseif(session()->has('user') && $item->displayed_text == 'login')
            @continue
```

```

    @elseif(session()->has('user') && $item->displayed_text == 'register')
        @continue
    @elseif(session()->has('user') && $item->displayed_text == 'logout')
        <a href="{{ route($item->route) }}" class="nav-link link-light">{{ ucfirst($item->displayed_text) }}</a>
    @elseif(!session()->has('user') && $item->displayed_text == 'logout')
        @continue
    @elseif( (!session()->has('user') || session()->get('user')->roleId !== 2) && $item->displayed_text == 'admin')
        @continue
    @else
        <a href="{{ route($item->route) }}" class="nav-link link-light">{{ ucfirst($item->displayed_text) }}</a>
    @endif
@endforeach

```

```
</nav>
```

register.blade.php

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

```

@section('title')
    Clocky shop - Register
@endsection

```

```

@section('main')
    {{ session('userName') }}
    <form action="" method="POST" class="col-md-5 container lone-form">
        @csrf
        <div class="form-group">
            <label for="username" class="form-label">User name</label>
            <input type="text" name="username" id="username" class="form-control @if(session('userName')) is-invalid @else
@endif">
            <div class="invalid-feedback">
                {{ session('userName') }}
            </div>
        </div>

        <div class="form-group">
            <label for="email" class="form-label">Email</label>
            <input type="text" name="email" id="email" class="form-control @if(session('email')) is-invalid @endif">
            <div class="invalid-feedback">
                {{ session('email') }}
            </div>
        </div>

        <div class="form-group">
            <label for="password" class="form-label">Password</label>
            <input type="text" name="password" id="password" class="form-control">
            <div class="invalid-feedback">
                Error text
            </div>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>

```

@endsection

scripts.blade.php

```
<script src="{{ asset('assets/vendor/jquery/jquery-3.6.0.min.js') }}"></script>
<script src="{{ asset('assets/vendor/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<script src="{{ asset('assets/js/main.js') }}"></script>
```

web.php

<?php

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\RegistrationController;
use App\Http\Controllers>LoginController;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\UserController;
```

```
/*
|-----
| 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('/login', [LoginController::class, 'index'])->name('login');
Route::post('/login', [LoginController::class, 'tryLogin'])->name('tryLogin');
Route::get('/logout', [LoginController::class, 'logout'])->name('logout');
```

```
Route::group(['middleware' => 'isAuth', 'isAdmin'], function() {
    Route::get('/admin', [App\Http\Controllers\AdminController::class, 'index'])->name('adminPanel');
    Route::delete('deleteProduct', [App\Http\Controllers\ProductController::class, 'tryDeleteProduct']);
    Route::post('createProduct', [App\Http\Controllers\AdminController::class, 'tryCreateNewProduct']);
    Route::post('editProduct', [App\Http\Controllers\AdminController::class, 'editProduct']);
    Route::post('getCategories', [App\Http\Controllers\AdminController::class, 'getCategories']);
    Route::delete('deleteCategory', [App\Http\Controllers\AdminController::class, 'deleteCategory']);
    Route::post('createCategory', [App\Http\Controllers\AdminController::class, 'createCategory']);
    Route::post('editCategory', [App\Http\Controllers\AdminController::class, 'editCategory']);
    Route::post('getUsers', [App\Http\Controllers\AdminController::class, 'getUsers']);
    Route::delete('deleteUser', [App\Http\Controllers\AdminController::class, 'deleteUser']);
    Route::post('getDiscounts', [App\Http\Controllers\AdminController::class, 'getDiscounts']);
    Route::delete('deleteDiscount', [App\Http\Controllers\AdminController::class, 'tryDeleteDiscount']);
    Route::post('createDiscount', [App\Http\Controllers\AdminController::class, 'createDiscount']);
    Route::post('editDiscount', [App\Http\Controllers\AdminController::class, 'editDiscount']);
});
```

```
Route::get('/register', [RegistrationController::class, 'index'])->name('register');
Route::post('/register', [RegistrationController::class, 'tryRegister'])->name('tryRegister');
```

```

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

Route::get('/profile', [UserController::class, 'index'])->name('profile')->middleware('isAuth');

Route::get('/error', [\App\Http\Controllers\ErrorController::class, 'index'])->name('error');

Route::post('/products', [\App\Http\Controllers\ProductController::class, 'filteredProducts'])->name('getProducts');

Route::get('/product/{id}', [\App\Http\Controllers\ProductController::class, 'index'])->name('singleProduct');

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

Route::get('/cart', [\App\Http\Controllers\CartController::class, 'index'])->name('cart');
Route::post('/getProductsByIds', [\App\Http\Controllers\ProductController::class, 'productsByIds']);
Route::post('/receipt', [\App\Http\Controllers\PurchaseController::class, 'createPurchase']);
Route::post('/contact', [UserController::class, 'insertContact']);

```

main.js

```

window.onload = function () {

    var categories = [];
    var discounts = [];

    if(location.pathname == '/') {
        $('.category').change(getProducts);
        $('#sort').change(getProducts);
        $('#search').blur(getProducts);
    }

    if(location.pathname == '/admin') {

        sendAjaxRequest('/products', 'POST', "", productsAdminDisplay, empty);
        $('#products').click(function () {sendAjaxRequest('/products', 'POST', "", productsAdminDisplay)});
        $('#close').click(function () {
            $('.popup').css('display', 'none ');
        })
        $('#categories').click(function () {
            sendAjaxRequest('/getCategories','POST', "", categoriesAdminDisplay, empty)
        });
        $('#discounts').click(function () {
            sendAjaxRequest('/getDiscounts', 'POST', "", discountsAdminDisplay, empty)
        });
        $('#users').click(function () {
            sendAjaxRequest('/getUsers', 'POST',"", userAdminDisplay, empty)
        });
        $('#display-msg').click(function () {

```



```

        $(this).css('display', 'none');
    })

}

if(location.pathname == '/cart') {

    if ((getFromLocalStorage('cart') == null) || getFromLocalStorage('cart').length == 0)
    {
        showEmptyCart()
    } else
    {
        let productIds = [];
        let products = getFromLocalStorage('cart');
        products.forEach(p => {
            productIds.push(p.id)
        })
        let data = {
            'ids':productIds
        }
        sendAjaxRequest('getProductsByIds','POST', data, populateCart, empty);
    }
}

$('#add-to-cart').click(addProductToCart);
$('#add-to-cart').click(function() {
    $('.cart-info').css('display', 'block');
});
$('.close').click(function(){
    $('.cart-info').css('display', 'none');
});

$('#submit-contact-form').click(function() {

    let contact = {
        'message':$('#message').val(),
        'email':$('#contact-form-email').val()
    }
    sendAjaxRequest('/contact', 'POST', contact, displayContactFormMessage);
});

}

function displayContactFormMessage(message) {
    let messageDiv = $('#contact-form-message')
    messageDiv.html(message.message);
    messageDiv.css('display', 'block');
    setTimeout(() => {messageDiv.css('display', 'none')}, 5000)
}

function showEmptyCart() {
    let result = `
<div id="cart-error" class="row justify-content-center text-center">
    <p class="">No products currently in cart</p>
</div>`;
    $('#cart-container').html(result);
    $('#checkout').css('display', 'none');
}

```

```

function addProductToCart() {
  let productId = $(this).data('id');
  var numberOfProducts = $('#number-of-products').val();
  var productsInCart = getFromLocalStorage('cart');

  if (productsInCart)
  {
    if (productAlreadyInCart(productId))
    {
      incrementProductInCart()
    } else
    {
      addProductToCartFirstTime()
    }
  } else
  {
    firstProductToCart();
  }

  function firstProductToCart() {
    let data = [];
    data[0] = {
      id: productId,
      quantity: numberOfProducts
    }
    setToLocalStorage(data, 'cart')
  }

  function incrementProductInCart() {

    for (let product of productsInCart)
    {
      if (product.id == productId)
      {
        product.quantity = parseInt(product.quantity) + parseInt(numberOfProducts);
        break
      }
    }
    setToLocalStorage(productsInCart, 'cart');
  }

  function addProductToCartFirstTime() {

    let data = {
      id: productId,
      quantity: parseInt(numberOfProducts)
    }
    productsInCart.push(data);
    setToLocalStorage(productsInCart, 'cart');
  }

  function productAlreadyInCart(id) {
    return productsInCart.filter(x => x.id == productId).length
  }
}

```

```

function populateCart(data) {

    let productsInStorage = getFromLocalStorage('cart');
    data.forEach(element => {
        for(let product of productsInStorage) {
            if(element.id == product.id) {
                element.quantity = product.quantity;
            }
        }
    });
    let result = `<table class="table">
    <thead>
        <tr>
            <!-- <th>Image</th>-->
            <th>Product name</th>
            <th>Quantity</th>
            <th>Total</th>
            <th>Discount</th>
            <th>Delete</th>
        </tr>
    </thead>
    <tbody>`;

    data.forEach(product => {
        result += `
        <tr>
            <!-- <td></td>-->
            <td>${product.name}</td>
            <td>${product.quantity}</td>
            <td class="text-nowrap">${product.discount ? (product.price.amount - ((product.price.amount / 100) *
product.discount.percent))*product.quantity : (product.price.amount)*product.quantity} &euro;</td>
            <td>${product.discount ? product.discount.name + ' - ' + product.discount.percent + '%' : ''}</td>
            <td><input type="button" data-id="${product.id}" class="btn btn-danger remove-from-cart tm-btn-right"
            value="Remove"/>
            </td>`;
    });
    $('#cart-products').html(result);
    $('.remove-from-cart').click(removeFromLocalStorage);
    $('.remove-from-cart').click(updateCart);
    $('#checkout').click(function () {
        let products = getFromLocalStorage('cart');
        sendAjaxRequest('receipt', 'POST', {'data':products}, cartMessage);
        localStorage.clear();
    })
}

function cartMessage(message) {
    // console.log(message);
    let result = `
    <div id="cart-error" class="row justify-content-center text-center">
        <p class="">${message.message}</p>
    </div>`;
    $('#cart-container').html(result);
    $('#checkout').css('display', 'none');
}

```

```

function updateCart() {
  $(this).parent().parent().remove();
  if(getFromLocalStorage('cart').length == 0) {
    showEmptyCart()
  }
}

function removeFromLocalStorage() {
  let products = getFromLocalStorage('cart');
  let filtered = products.filter(p => $(this).data('id') != p.id);
  setToLocalStorage(filtered, 'cart');
}

function setToLocalStorage(data, id) {
  localStorage.setItem(id, JSON.stringify(data));
}

function getFromLocalStorage(id) {
  return JSON.parse(localStorage.getItem(id))
}

function sendAjaxRequest(url, method, data, success, failure = empty) {
  $.ajaxSetup({
    headers:{
      'X-CSRF-TOKEN':$("input[name='_token']").attr('value')
    }
  });

  $.ajax({
    url:url,
    data:data,
    dataType:'JSON',
    method:method,
    success:function(response) {
      success(response);
    },
    error:function(xhr) {
      failure(xhr);
    }
  })
}

function empty(response) {

}

function getProducts() {
  let categories = $('#category:checked');
  let sortType = $('#sort').val();
  let searchTerm = $('#search').val();
  let categoryIds = [];

  for (const category of categories) {

```

```

        categoryIds.push(category.value)
    }
    let data = {
        'searchTerm':searchTerm,
        'categoryIds':categoryIds,
        'sortType':sortType
    }
    sendAjaxRequest('/products', 'POST', data, productsFill, empty);
}

function productsFill(response) {
    //console.log(response);
    //let products = JSON.parse(response);
    let result = `<div class="row">`;
    console.log(response);
    response.forEach(product => {
        result += `
            <div class="col-xl-3 col-lg-4 col-md-6 mb-4">
                <div class="bg-white rounded shadow-sm">
                    
                    <div class="p-4">
                        <h6 class="product-card"> <a href="/product/${product.id}" class="">${product.name}</a></h6>
                        <div class="row">
                            ${printPrice(product.amount, product.discount)}
                        </div>
                    </div>
                </div>
            </div>`;
    });
    result += `</div>`;
    $('#products').html(result);
}

function printPrice(price, discount) {
    if(discount) {
        return `
            <div class="col-6">
                <s class="price text-nowrap">${price} &euro;</s>
            </div>
            <div class="col-6">
                <p class="price text-nowrap">${price - ((price / 100) * discount.percent)} &euro;</p>
            </div>`;
    } else {
        return `<p class="price text-nowrap">${price} &euro;</p>`;
    }
}

function productsAdminDisplay(response) {
    let result = `<table class="table">
    <thead>
        <tr>
            <th>Product Id</th>
            <th>Product name</th>
            <th>Price</th>
            <th>Discount</th>

```

```

        <th>Category</th>
        <th>Delete</th>
        <th>Edit</th>
    </tr>
</thead>
<tbody>`;

response.forEach(product => {
    result += `
    <tr>
    <td>${product.id}</td>
    <td>${product.name}</td>
    <td class="text-nowrap">${product.discount ? (product.amount - ((product.amount / 100) *
product.discount.percent)) : (product.amount)} &euro;</td>
    <td class="text-nowrap">${product.discount ? product.discount.name : ''}</td>
    <td>${product.category.name}</td>
    <td><input type="button" data-id="${product.id}" class="btn btn-danger delete-product tm-btn-right"
value="Delete"/>
    <td><input type="button" data-id="${product.id}" class="btn btn-primary edit-product tm-btn-right" value="Edit"/>
    </td>`;
    })
    // ${price - ((price / 100) * discount.percent)} &euro;
    $('#admin-display').html(result);
    $('#insert-new').html(`<button id="insert-new" class="btn btn-success top-gap">New product</button>`);
    $('.delete-product').click(function() {
        let id = $(this).data('id');
        let data = {
            'id': id
        }

        sendAjaxRequest('/deleteProduct', 'DELETE', data, refreshAdminProducts, empty);
    });

    $('#insert-new').click(function() {
        let formHtml = `<input type="text" id="product-name" class="form-control" placeholder="Enter new name...">
<div class="form-group">
    <label for="product-category">Select a category</label>
    <select name="product-category" id="product-category" class="form-control"></select>
</div>
<div class="form-group">
<label for="product-price">Select price</label>
<select name="product-price" id="product-price" class="form-control"></select>
</div>
<div class="form-group">
<label for="product-discount">Select discount</label>
<select name="product-discount" id="product-discount" class="form-control"></select>
</div>
<textarea name="description" id="product-description" cols="30" rows="10" class="form-control" placeholder="New
description..."></textarea>
    <button class="btn btn-success" id="new-product">Submit</button>`;

        $('#input-form').html(formHtml);
        $('#product-category').html($('#categories-hidden').html());
        $('#product-price').html($('#prices-hidden').html());
        $('#product-discount').html($('#discounts-hidden').html());
        $('#popup').css('display', 'block');
        $('#new-product').click(function() {

```

```

let productName = $('#product-name').val();
let category = $('#product-category').val();
let price = $('#product-price').val();
let discount = $('#product-discount').val();
let description = $('#product-description').val();
console.log(productName);
let data = {
  'name':productName,
  'category':category,
  'price':price,
  'discount':discount,
  'description':description
}
sendAjaxRequest('/createProduct', 'POST', data, refreshPage, empty);
});
});

```

```

$('#edit-product').click(function() {
let id = $(this).attr('data-id');
let product = response.filter(x => x.id == id);
console.log(product);
let formHtml = `<input type="text" id="product-name" class="form-control" value="${product[0].name}">
<div class="form-group">
  <label for="product-category">Select a category</label>
  <select name="product-category" id="product-category" class="form-control"></select>
</div>
<div class="form-group">
<label for="product-price">Select price</label>
<select name="product-price" id="product-price" class="form-control"></select>
</div>
<div class="form-group">
<label for="product-discount">Select discount</label>
<select name="product-discount" id="product-discount" class="form-control"></select>
</div>
<div class="form-group">
<label for="product-description">Description</label>
<textarea name="description" id="product-description" cols="30" rows="10" class="form-control"> $
{product[0].description}</textarea>
</div>
  <button class="btn btn-success" data-id="${product[0].id}" id="edit-product">Submit</button>`;

```

```

$('#input-form').html(formHtml);
$('#product-category').html($('#categories-hidden').html());
$('#product-price').html($('#prices-hidden').html());
$('#product-discount').html($('#discounts-hidden').html());
$('.popup').css('display', 'block');
$('#edit-product').click(function() {
let productName = $('#product-name').val();
let category = $('#product-category').val();
let price = $('#product-price').val();
let discount = $('#product-discount').val();
let description = $('#product-description').val();
console.log(productName);
let data = {
  'id':$(this).attr('data-id'),
  'name':productName,
  'category':category,
  'price':price,

```

```

        'discount':discount,
        'description':description
    }
    sendAjaxRequest('/editProduct', 'POST', data, refreshPage, displayAdminMessage);
});
$('.popup').css('display', 'block');
});
}

function refreshPage() {
    location.reload();
}

function categoriesAdminDisplay(data) {
    let result = `<table class="table">
    <thead>
        <tr>
            <th>Category Id</th>
            <th>Category name</th>
            <th>Created </th>
            <th>Updated</th>
            <th>Delete</th>
            <th>Edit</th>
        </tr>
    </thead>
    <tbody>`;

    data.forEach(category => {
        result += `
        <tr>
            <td>${category.id}</td>
            <td>${category.name}</td>
            <td>${category.created_at}</td>
            <td>${category.updated_at ? category.updated_at : ''}</td>
            <td><input type="button" data-id="${category.id}" class="btn btn-danger delete-category" value="Delete"/>
            <td><input type="button" data-id="${category.id}" class="btn btn-primary edit-category" value="Edit"/>
            </td>`;
    });

    $('#admin-display').html(result);
    $('#insert-new').html(`<button id="insert-new" class="btn btn-success top-gap">New category</button>`);
    $('#insert-new').click(function() {
        let formHTML = `
        <div class="form-group">
            <label for="category-name">Category name</label>
            <input type="text" class="form-control" id="category-name">
        </div>
        <button class="btn btn-success" id="new-category">Submit</button>`;
        $('#input-form').html(formHTML);
        $('.popup').css('display', 'block');
        $('#new-category').click(function() {
            //let id = $('#category-name').val();
            let data = {
                'name':$('#category-name').val()
            }
            sendAjaxRequest('/createCategory', 'POST', data, refreshPage, displayAdminMessage);
        });
    });
}

```



```

});
$('.delete-category').click(function() {
  let id = $(this).data('id');
  let data = {
    'id':id
  }
  sendAjaxRequest('/deleteCategory', 'DELETE', data, refreshAdminCategories, displayAdminMessage);
});
$('.edit-category').click(function() {
  let id = $(this).attr('data-id');
  let category = data.filter(x => x.id == id);
  let formHtml = `
    <div class="form-group">
      <label for="category-name">Category name</label>
      <input type="text" id="category-name" class="form-control" value="${category[0].name}">
    </div>
    <button class="btn btn-success" data-id="${category[0].id}" id="edit-category">Submit</button>`;
  $('#input-form').html(formHtml);
  $('.popup').css('display', 'block');
  $('#edit-category').click(function() {
    let catData = {
      'id':category[0].id,
      'name':$('#category-name').val()
    }
    sendAjaxRequest('/editCategory', 'POST', catData, refreshPage, displayAdminMessage);
  })
});
}

```

```

function displayAdminMessage(text) {
  // console.log(text.responseJSON);
  $('.popup').css('display', 'none');
  let div = $('#display-msg');
  div.html(text.responseJSON.message);
  div.css('display', 'block');
}

```

```

function discountsAdminDisplay(data) {
  let result = `<table class="table">
<thead>
  <tr>
    <th>Discount Id</th>
    <th>Discount name</th>
    <th>Percent</th>
    <th>Created </th>
    <th>Updated</th>
    <th>Delete</th>
    <th>Edit</th>
  </tr>
</thead>
<tbody>`;

  data.forEach(discount => {
    result += `
    <tr>

```

```

<td>${discount.id}</td>
<td>${(discount.name)}</td>
<td>${discount.percent}%</td>
<td>${discount.created_at}</td>
<td>${discount.updated_at ? discount.updated_at : ''}</td>
<td><input type="button" data-id="${discount.id}" class="btn btn-danger delete-discount" value="Delete"/>
<td><input type="button" data-id="${discount.id}" class="btn btn-primary edit-discount" value="Edit"/>
</td>`;
});

$('#admin-display').html(result);
$('#insert-new').html(`<button id="insert-new" class="btn btn-success top-gap">New discount</button>`);
$('.delete-discount').click(function() {
    let id = $(this).data('id');
    let data = {
        'id':id
    }
    sendAjaxRequest('/deleteDiscount', 'DELETE', data, refreshAdminDiscounts, displayAdminMessage);
});
$('#insert-new').click(function() {
    let formHtml = `
        <div class="form-group">
            <label for="discount-name">Discount name</label>
            <input type="text" id="discount-name" class="form-control" placeholder="Enter new name...">
        </div>
        <div class="form-group">
            <label for="discount-percent">Discount percent</label>
            <input type="text" id="discount-percent" class="form-control" placeholder="Enter a percentage...">
        </div>
        <button class="btn btn-success" id="new-discount">Submit</button>`;
    $('#input-form').html(formHtml);
    $('#new-discount').click(function() {
        let data = {
            'name':$('#discount-name').val(),
            'percent':$('#discount-percent').val()
        }
        sendAjaxRequest('/createDiscount', 'POST', data, refreshPage, displayAdminMessage);
    });
});
$('.edit-discount').click(function() {
    let id = $(this).attr('data-id');
    let discount = data.filter(x => x.id == id);
    let formHtml = `
        <div class="form-group">
            <label for="discount-name">Discount name</label>
            <input type="text" id="discount-name" class="form-control" value="${discount[0].name}">
        </div>
        <div class="form-group">
            <label for="discount-percent">Discount percent</label>
            <input type="text" id="discount-percent" class="form-control" value="${discount[0].percent}">
        </div>
        <button class="btn btn-success" data-id="${discount[0].id}" id="edit-discount">Submit</button>`;
    $('#input-form').html(formHtml);
    $('.popup').css('display', 'block');
    $('#edit-discount').click(function() {
        let discountData = {
            'id':$('#edit-discount').attr('data-id'),
            'name':$('#discount-name').val(),

```

```

        'percent':$('#discount-percent').val()
    }
    sendAjaxRequest('/editDiscount', 'POST', discountData, refreshPage, displayAdminMessage);
});
});
}

function userAdminDisplay(data) {
    $('#insert-new').html('');
    let result = `<table class="table">
<thead>
<tr>
<th>User Id</th>
<th>User name</th>
<th>Email</th>
<th>Registered </th>
<th>Delete</th>
</tr>
</thead>
<tbody>`;

    data.forEach(user => {
        result += `
<tr>
<td>${user.id}</td>
<td>${user.user_name}</td>
<td>${user.email}</td>
<td>${user.created_at}</td>
<td><input type="button" data-id="${user.id}" class="btn btn-danger delete-user" value="Delete"/>
</td>`;
    });

    $('#admin-display').html(result);
    $('.delete-user').click(function() {
        let id = $(this).attr('data-id');
        let userId = {
            'id':id
        }
        sendAjaxRequest('deleteUser', 'DELETE', userId, refreshAdminUsers)
    });
}

function refreshAdminProducts() {
    sendAjaxRequest('/products', 'POST', "", productsAdminDisplay);
}

function refreshAdminCategories() {
    sendAjaxRequest('/getCategories', 'POST', "", categoriesAdminDisplay);
}

function refreshAdminDiscounts() {
    sendAjaxRequest('/getDiscounts', 'POST', "", discountsAdminDisplay);
}

function refreshAdminUsers() {
    sendAjaxRequest('/getUsers', 'POST', "", userAdminDisplay);
}

```

syle.css

```
a {
  text-decoration: none;
  color: black;
}

a:hover {
  color: #9e9bff;
  /*text-decoration: underline;*/
}

.blue {
  background-color: #0b6568;
  color: white;
}

.price {
  font-size: 1.1rem;
}

#purchase-options {
}

#admin-nav li:hover {
  background-color: #0b6568;
}

#admin-nav li {
  font-size: 1.7rem;
}

#admin-nav {
  width: 100%;
  padding-left: 0;
  /*height: 100vh;*/
}

#admin-nav li {
  cursor: pointer;
}

#product-edit {
  position: absolute;
  left: 50%;
```

```
top: 50%;
transform: translate(-50%,-50%);
z-index: 999;
width: 30%;
}
```

```
.popup {
  position: fixed;
  z-index: 999;
  top: 0;
  left: 0;
  background-color: white;
  width: 100%;
  height: 100%;
  display: none;
}
```

```
#close{
  display: block;
  margin: 0 0 20px auto;
  background-color: transparent;
  font-size: 30px;
  color: #ffffff;
  background: #212529;
  border-radius: 100%;
  width: 40px;
  height: 40px;
  border: none;
  outline: none;
  cursor: pointer;
}
```

```
.top-gap {
  margin-top: 30px;
}
```

```
.bottom-gap {
  margin-bottom: 40px;
}
```

```
.product-card {
  min-height: 96px;
}
```

```
.lone-form {
  margin-top: 40px;
}
```

```
main {
  min-height: 100vh;
}
```

```
#edit-form {
  width: 30%;
}
```

```
textarea {
```

```
    resize: none;  
}
```

```
.hidden {  
    display: none;  
}
```

```
#display-msg {  
    position: fixed;  
    min-width: 200px;  
    min-height: 100px;  
    padding: 0.5em;  
    border-radius: 10px;  
    background-color: rgba(111,111,111,0.5);  
    color: white;  
    z-index: 997;  
    left: 50%;  
    top: 50%;  
    transform: translate(-50%,-50%);  
    font-size: 2rem;  
    cursor: pointer;  
}
```