

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Dokumentácia  
KVP - ESHOP  
Základy webových technológií

Adam Jurčišin  
Andrej Stuchlý

Dátum : 12. 5. 2024

# Obsah

<b>Zadanie.....</b>	<b>3</b>
<b>Diagram fyzického dátového modelu.....</b>	<b>4</b>
<b>Návrhové rozhodnutie.....</b>	<b>5</b>
<b>Programové prostredie.....</b>	<b>6</b>
<b>Opis implementácie vybraných prípadov použitia.....</b>	<b>7</b>
Prihlásenie a registrácia.....	8
Aplikovanie filtrov.....	9
Stránkovanie produktov.....	10
Pridanie nového produktu.....	10
Zmena množstva produktu v košíku.....	11
<b>Snímky obrazoviek.....</b>	<b>12</b>
Klient rozhranie.....	12
Admin rozhranie.....	18

# Zadanie

Vytvorte webovú aplikáciu - eshop, ktorá komplexne rieši nižšie definované prípady použitia vo vami zvolenej doméne (napr. elektro, oblečenie, obuv, nábytok). Presný rozsah a konkretizáciu prípadov použitia si dohodnete s vašim vyučujúcim.

**Aplikácia musí realizovať tieto prípady použitia:**

## Klientská časť

- - zobrazenie prehľadu všetkých produktov z vybranej kategórie používateľom
  - základné filtrovanie (aspoň podľa 3 atribútov, napr. rozsah cena od-do, značka, farba)
  - stránkovanie
  - preusporiadanie produktov (napr. podľa ceny vzostupne/zostupne)
- - zobrazenie konkrétneho produktu - detail produktu
  - prídanie produktu do košíka (ľubovoľné množstvo)
- - plnotextové vyhľadávanie nad katalógom produktov
- - zobrazenie nákupného košíka
  - zmena množstva pre daný produkt
  - odobratie produktu
  - výber dopravy
  - výber platby
  - zadanie dodacích údajov
  - dokončenie objednávky
  - umožnenie nákupu bez prihlásenia
  - prenositeľnosť nákupného košíka v prípade prihláseného používateľa
- - registrácia používateľa/zákazníka
- - prihlásenie používateľa/zákazníka
- - odhlásenie zákazníka

## Administrátorská časť

- - prihlásenie administrátora do administrátorského rozhrania eshopu
- - odhlásenie administrátora z administrátorského rozhrania
- - vytvorenie nového produktu administrátorom cez administrátorské rozhranie
  - produkt musí obsahovať minimálne názov, opis, aspoň 2 fotografie
- - upravenie/vymazanie existujúceho produktu administrátorom cez administrátorské rozhranie

# Diagram fyzického dátového modelu

Pôvodný návrh fyzického dátového modelu:

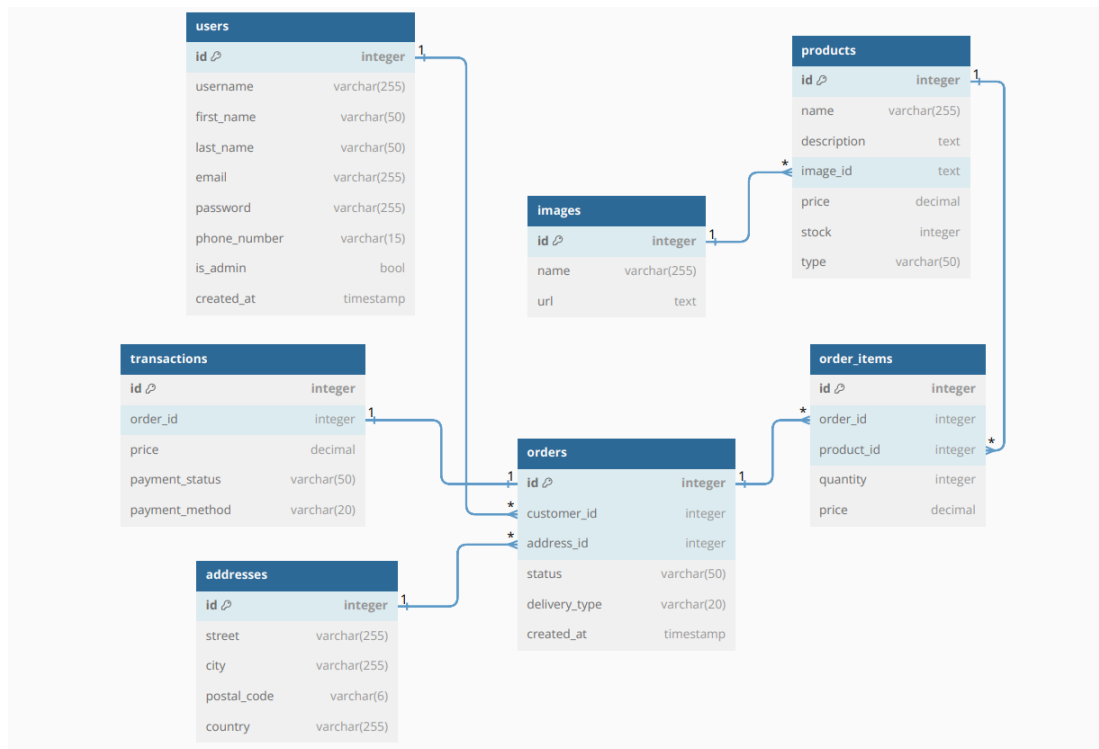


Diagram 1 - Pôvodný návrh

Finálne použitý fyzický dátový model:

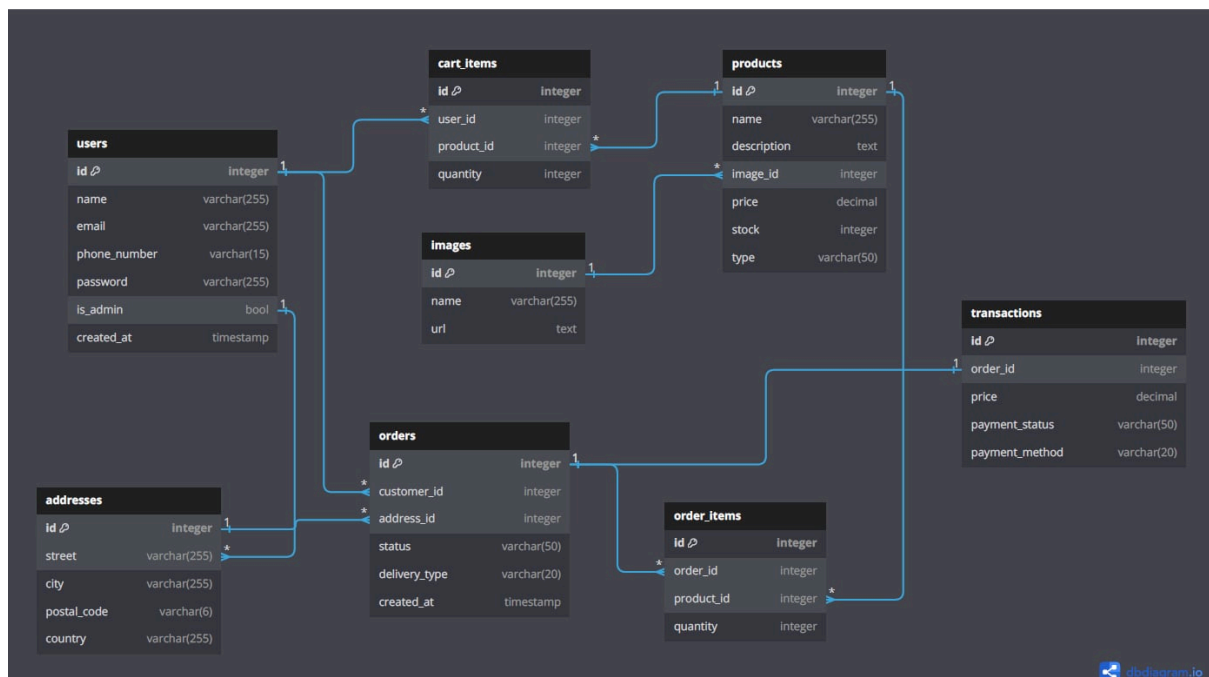


Diagram 2 - Finálny návrh

Oproti návrhu sme aplikovali nasledujúce zmeny:

Tabuľka users: Odstránenie stĺpcov firstname, lastname a username. Pridanie stĺpca name.

Dôvod: Po pridaní Laravel Breeze ukladanie mena ako jeden string.

Tabuľka cart\_items: Odstránenie stĺpca type.

Dôvod: Nepotrebný stĺpec.

Tabuľka orders: Pridanie stĺpca email, name a phone\_number.

Dôvod: Ukladanie ďalších dôležitých informácií o objednávke.

Tabuľka order\_items: Odstránenie stĺpca price.

Dôvod: Nepotrebný stĺpec.

Následne sa v databáze použili tabuľky, ktoré automaticky vytvoril framework Laravel, tieto tabuľky zabezpečujú napríklad prihlásenie, registrovanie.

## Návrhové rozhodnutie

Pre náš e-shop sme sa rozhodli použiť Laravel framework pre backend vývoj a Bootstrap framework pre frontend. Autentifikáciu sme zabezpečili pomocou Laravel Breeze a ako databázový systém sme zvolili PostgreSQL. Implementovali sme jednoduchú autentifikáciu, používateľ môže byť bežný zákazník alebo administrátor .

Pri návrhu sme upravili pôvodný databázový model, odstránili niektoré stĺpce a pridali nové, aby sme lepšie zodpovedali potrebám aplikácie. Zaviedli sme možnosť aplikovania filtrov a stránkovanie produktov na zlepšenie užívateľského zážitku pri prehliadaní produktov.

Pri vyplňaní formulárov sme zaviedli validáciu údajov, aby nám nevznikli poškodené dáta a neočakávané správanie webu. Konkrétne riešenia a rozhodnutie je možné vidieť v jednotlivých častiach dokumentácie.

# Programové prostredie

Pre implementáciu e-shopu sme využili technológie, nástroje a vývojové prostredia spomenuté nižšie.

Použité Frontend technológie:

- CSS,
- HTML,
- framework Bootstrap.

Použité Backend technológie:

- PHP,
- framework Laravel,
- Laravel Breeze,
- PostgreSQL.

Použité vývojové prostredia a nástroje:

- Artisan,
- pgAdmin 4,
- diagrams.net,
- VS Code,
- PHPStorm,
- Google Chrome.

## Opis implementácie vybraných prípadov použitia

V tejto kapitole máme znázornené a opísané všetky zaujímavé prípady vyskytujúce sa v našom eshope.

Na nasledujúcom obrázku môžeme vidieť nami implementované endpointy REST API volaní, ktoré zabezpečujú chod nášho eshopu. Tieto endpointy volajú funkcie, ktoré spracujú volania a vrátia vizualizáciu stránky s potrebnými dátami.

```
Route::get('/dashboard', function () {
    return view('dashboard');
})->middleware(['auth', 'verified'])->name('dashboard');

Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
});

Route::get('/', [PageController::class, 'landingPage'])->name('landing_page');

Route::get('/detail', [PageController::class, 'detail'])->name('detail');
Route::post('/detail', [PageController::class, 'editCartAmount'])->name('cart-items.detail');
Route::post('/detail-session', [PageController::class, 'addToSession'])->name('cart-items.detail-session');

Route::get('/detail/{id}/edit', [PageController::class, 'editProduct'])->name('detail.edit');
Route::put('/pdetail/{id}', [PageController::class, 'updateProduct'])->name('product.update');

Route::get('/new_product_admin', [PageController::class, 'newProductAdmin'])->name('new_product_admin');
Route::post('/new_product_admin', [NewProductController::class, 'newProductAdminPost'])->name('new_product_admin');

Route::get('/payment', [PageController::class, 'payment'])->name('payment');
Route::post('/order_submit', [PageController::class, 'submitOrder'])->name('payment.submit');

Route::match(['get', 'post'], '/order', [PageController::class, 'order'])->name('order');

Route::get('/products', [PageController::class, 'products'])->name('products');

Route::get('/shopping_cart', [PageController::class, 'shoppingCart'])->name('shopping_cart');
Route::put('/cart-items/{id}', [PageController::class, 'updateCartItem'])->name('cart-items.update');
Route::delete('/cart-items/{id}/remove', [PageController::class, 'removeCartItem'])->name('cart-items.remove');

require __DIR__.'/auth.php';
```

Obrázok 1 - Smerovanie REST API volaní

Na nasledujúcom obrázku môžeme vidieť smerovanie REST API volaní, ktoré boli vytvorené pomocou Laravel Breeze. Tieto volania zabezpečujú autentifikáciu používateľa. Pomocou týchto volaní sa užívateľ prihlási, registruje, vytvorí si svoju session a podobne.

```

Route::middleware('guest')->group(function () {
    Route::get('register', [RegisteredUserController::class, 'create'])
        ->name('register');

    Route::post('register', [RegisteredUserController::class, 'store']);

    Route::get('login', [AuthenticatedSessionController::class, 'create'])
        ->name('login');

    Route::post('login', [AuthenticatedSessionController::class, 'store']);

    Route::get('forgot-password', [PasswordResetLinkController::class, 'create'])
        ->name('password.request');

    Route::post('forgot-password', [PasswordResetLinkController::class, 'store'])
        ->name('password.email');

    Route::get('reset-password/{token}', [NewPasswordController::class, 'create'])
        ->name('password.reset');

    Route::post('reset-password', [NewPasswordController::class, 'store'])
        ->name('password.store');
});

```

Obrázok 2 - Smerovanie Laravel Breeze autentifikačných REST API volaní

## Prihlásenie a registrácia

Prihlásenie a registrácia je realizovaná pomocou Laravel Breeze. Proces spracovania údajov je znázornený na nasledujúcom obrázku. Prijaté dáta, ktoré zadá užívateľ sa overia aby splnili všetky podmienky a následne sa vytvorí nový profil, ktorý sa uloží do databázy.

```

public function store(Request $request): RedirectResponse
{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:'.User::class],
        'password' => ['required', 'confirmed', Rules\Password::defaults()->min(4)],
    ]);

    $user = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    event(new Registered($user));

    Auth::login($user);

    return redirect(route('home', absolute: false));
}

```

Obrázok 3 - Funkcia pre registráciu



## Aplikovanie filtrov

Na nasledujúcom obrázku môžeme vidieť spôsob aplikácie filtrov, pokiaľ používateľ aplikuje filtre, tie sú zaslané pomocou formulára ako GET požiadavka. Tieto filtre prijmem a z databázy z tabuľky products vyberieme všetky produkty, ktoré tieto filtre spĺňajú. Aplikované filtre sú vyhľadávanie podľa textu, kde vyhľadávaný text hľadáme ako podreťazec v názve alebo popise produktu. Zoradenie produktov zoraduje produkty podľa dátumu od najnovšieho alebo podľa ceny zostupne a vzostupne. Taktiež môžeme filtrovať produkty podľa kategórie.

```
public function products(Request $request)
{
    $search = $request->input('search');
    $sort = $request->input('sort', 'najnovsie');
    $categories = $request->input('category', ['Kúrenie', 'Voda', 'Plyn']);
    $min_price = $request->input('min_price');
    $max_price = $request->input('max_price');

    $productsQuery = Product::query();

    if ($sort === 'najnovsie') {
        $productsQuery->orderBy('created_at', 'desc');
    } elseif ($sort === 'najmensie') {
        $productsQuery->orderBy('price');
    } elseif ($sort === 'najvacsie') {
        $productsQuery->orderBy('price', 'desc');
    }
    if (!empty($categories)) {
        $productsQuery->whereIn('type', $categories);
    }
    if (!empty($min_price)) {
        $productsQuery->where('price', '>=', $min_price);
    }

    if (!empty($max_price)) {
        $productsQuery->where('price', '<=', $max_price);
    }

    if (!empty($search)) {
        $productsQuery->where(function ($query) use ($search) {
            $query->where('name', 'like', "%$search%")
                ->orWhere('description', 'like', "%$search%");
        });
    }

    $products = $productsQuery->paginate(9)->withQueryString();

    return view('products', compact('products'));
```

Obrázok 4 - Funkcia aplikovania filtrov

## Stránkovanie produktov

Na stránkovanie produktov sú na frontende aplikované tlačidlá, ktoré vykonávajú GET požiadavky pre načítanie nasledujúcich produktov. Na backende ako bolo znázornené na poslednom obrázku sa tieto požiadavky spracujú a pomocou možnosti paginate kde je zvolený počet produktov, ktoré sa na stránke znázornia sa tieto produkty postupne načítavajú z databázy.

## Pridanie nového produktu

Pre pridanie nového produktu sa zavolá endpoint, ktorý zavolá funkciu znázornenú na obrázku nižšie. Táto funkcia prvotne vykoná validáciu vložených údajov. Vložené obrázky musia byť obrázkových formátov s maximálnou veľkosťou 2048 megabajtov. Pokiaľ validácia prebehne úspešne, vytvorí sa nový produkt s vloženými údajmi.

```
public function newProductAdminPost(Request $request)
{
    $images = $request->file('images');

    foreach($images as $image) {
        $validator = Validator::make(['image' => $image], [
            'image' => 'required|image|mimes:jpeg,png,jpg,gif|max:2048',
        ]);
    }

    if (!$validator->fails()) {
        $newProduct = new Product();
        $newProduct->name = $request->input("name");
        $newProduct->price = $request->input("price");
        $newProduct->description = $request->input("info");
        $newProduct->stock = $request->input("count");
        $newProduct->type = $request->input("type");
        $newProduct->created_at = Carbon::now();
        $newProduct->updated_at = Carbon::now();
        $newProduct->save();
    }
}
```

Obrázok 5 - Funkcia pridania nového produktu - validácia

Po vložení nového produktu sa vykoná uloženie obrázkov, ktoré môžeme vidieť na nasledujúcom obrázku. Každý vložený obrázok sa ukladá postupne, pri ukladaní obrázkov sa kontroluje či už obrázok s rovnakým názvom neexistuje, pokiaľ áno, názov obrázku sa zmení pridaním nasledujúceho voľného čísla v zátvorkách.

```

foreach($images as $image) {
    $imageName = $image->getClientOriginalName();
    $extension = $image->getClientOriginalExtension();
    $publicImagesFolder = public_path('images');

    $counter = 1;
    $imageNameNew = $imageName;
    while (file_exists($publicImagesFolder . '/' . $imageNameNew)) {
        $imageNameNew = pathinfo($imageName, PATHINFO_FILENAME) . "($counter).$extension";
        $counter++;
    }

    $image->move($publicImagesFolder, $imageNameNew);

    $newImage = new Image();
    $newImage->name = $imageNameNew;
    $newImage->product_id = $newProduct->id;
    $newImage->url = 'http://localhost:8000/images/' . $imageName;
    $newImage->created_at = Carbon::now();
    $newImage->updated_at = Carbon::now();
    $newImage->save();
}
return Redirect::route('detail', ['id' => $newProduct->id]);
}
return Redirect::route('products');

```

Obrázok 6 - Funkcia pridania nového produktu - uloženie obrázkov

## Zmena množstva produktu v košíku

Zmena množstva produktu v košíku sa rieši dvoma spôsobmi - priamo v košíku a na stránke detailu daného produktu. V oboch prípadoch sa postupuje podobne, získajú sa existujúce záznamy položiek v košíku, ak sa daný produkt už v košíku vyskytuje, upraví sa jeho počet, inak sa vytvorí nový záznam.

```

public function editCartAmount(Request $request)
{
    $productId = $request->input('product_id');
    $userId = $request->input('user_id');
    $quantity = $request->input('quantity');

    $existingCartItem = CartItem::where('product_id', $productId)
        ->where('user_id', $userId)
        ->first();

    if ($quantity == 0 && $existingCartItem) {
        $existingCartItem->delete();
    } elseif ($existingCartItem) {
        $existingCartItem->quantity = $quantity;
        $existingCartItem->save();
    } elseif ($quantity > 0) {
        CartItem::create([
            'product_id' => $productId,
            'user_id' => $userId,
            'quantity' => $quantity,
        ]);
    }
    return redirect()->route('detail', ['id' => $productId])->with('success', 'Product updated successfully.');
```

Obrázok 7 - Funkcia zmeny množstva v košíku

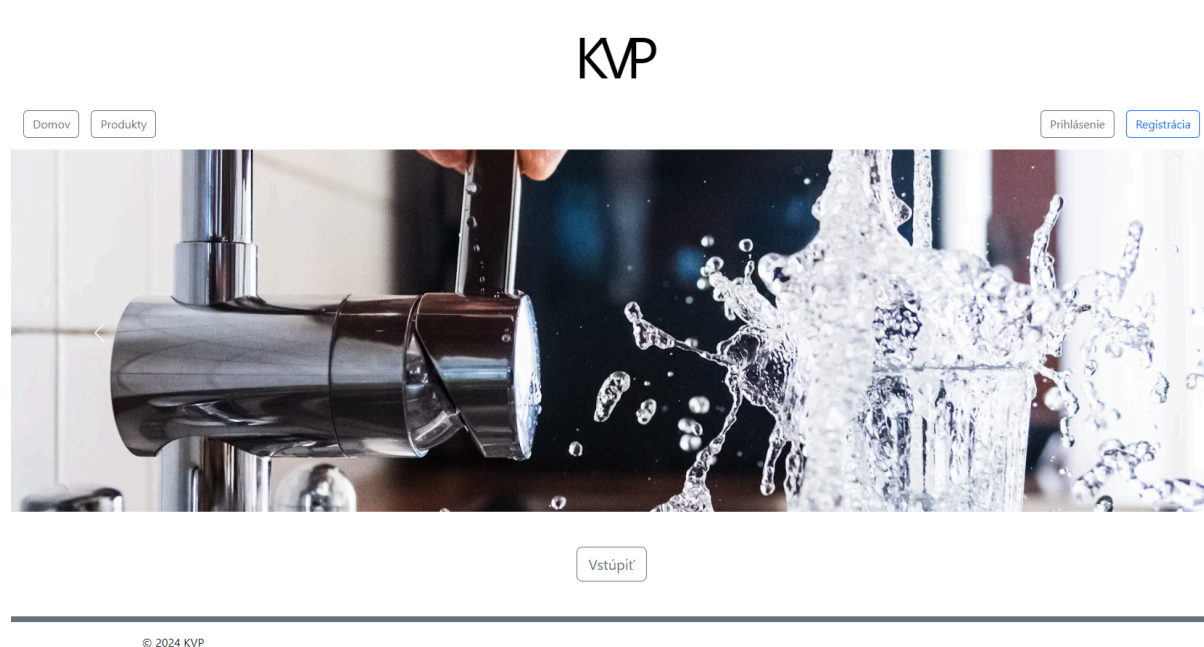
# Snímky obrazoviek

V tejto kapitole sú znázornené všetky typy obrazoviek, ktoré sa v našej implementácii e-shopu využívajú. Snímky máme rozdelené podľa typu užívateľa na Klient rozhranie a Admin rozhranie.

## Klient rozhranie

### Úvodná obrazovka

Na nasledujúcom obrázku môžeme vidieť úvodnú obrazovku nášho eshopu. Na tejto obrazovke sa nachádza carousel kde sa posúvajú obrázky s tematikou kúrenie, voda a plyn.



Obrázok 8 - Úvodná obrazovka

## Obrazovka prihlásenia

Na nasledujúcom obrázku môžeme vidieť obrazovku prihlásenia. Na tejto obrazovke sa využíva prihlasovanie pomocou Laravel Breeze.

KVP

Prihlásenie

Email

Password

☐ Zapamätat si ma

[Zabudol si heslo?](#) **PRIHLÁSIŤ SA**

[Ešte nemám účet](#)

© 2024 KVP

Obrázok 9 - Obrazovka prihlásenia

## Obrazovka registrácie

Na nasledujúcom obrázku môžeme vidieť obrazovku registrácie. Na tejto obrazovke sa využíva registrácia pomocou Laravel Breeze.

KVP

Registrácia

Meno

Email

Heslo

Zopakujte heslo

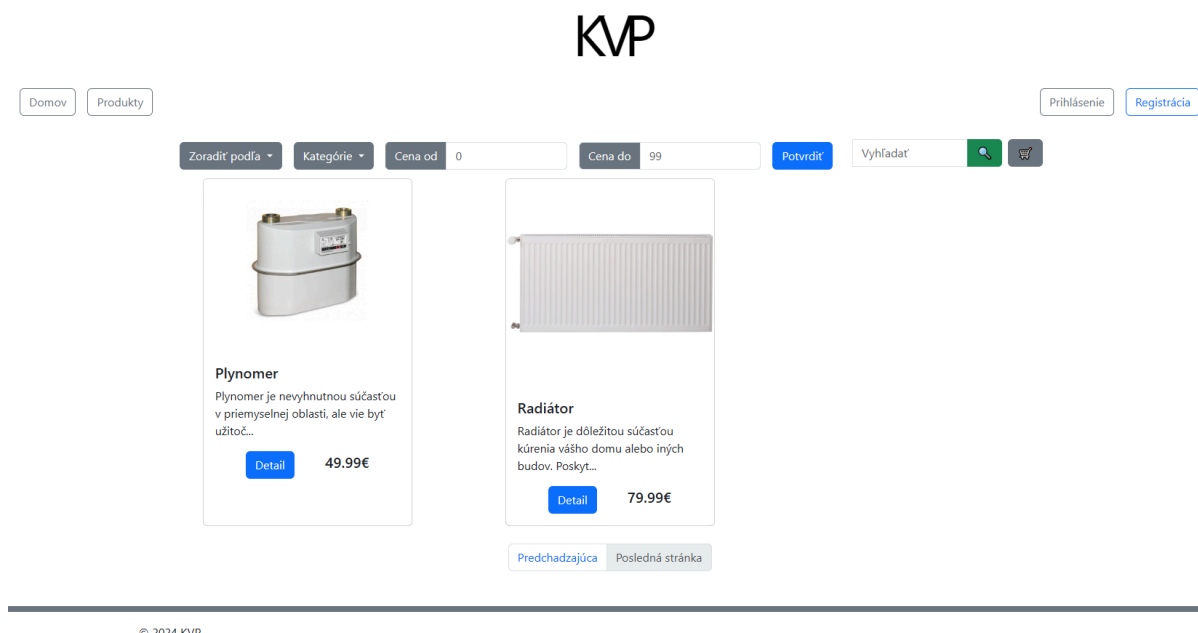
**REGISTROVAŤ**

[Už mám účet](#)

Obrázok 10 - Obrazovka registrácie

## Obrazovka všetkých produktov

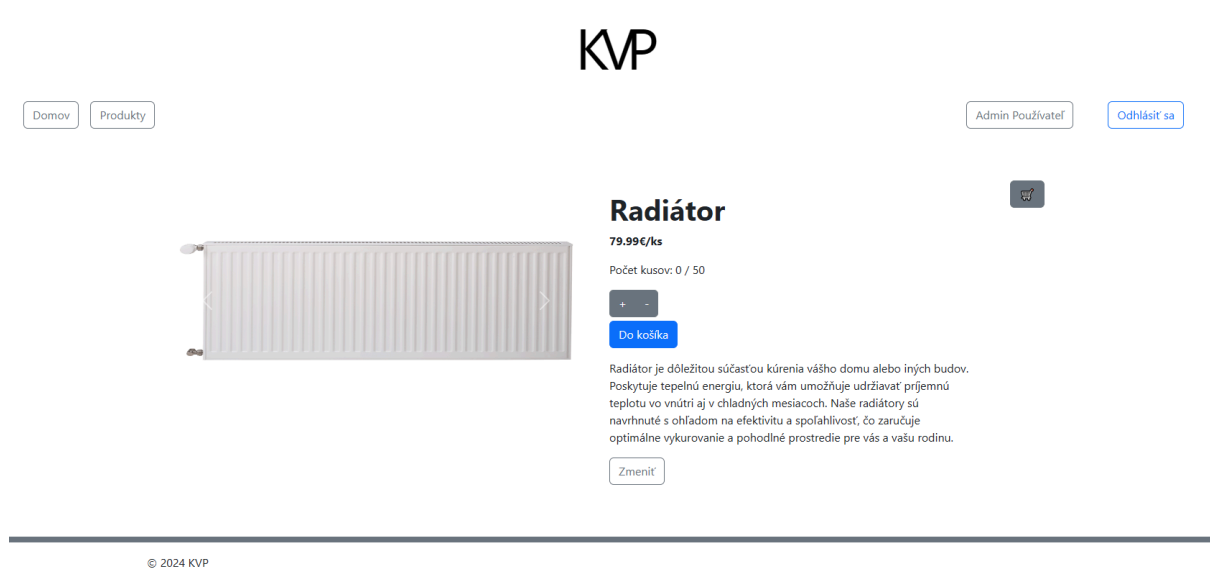
Na nasledujúcom obrázku môžeme vidieť obrazovku všetkých produktov. Na tejto obrazovke je možné aplikovať filtre pre zúženie vyhľadávania produktov. Použitím filtrov si vieme zoradiť produkty podľa ceny alebo dátumu pridania, výber kategórie, minimálnej a maximálnej ceny. Taktiež môžeme zadať text, ktorý vyhladá produkty, ktoré majú tento text v názve alebo popise.



Obrázok 11 - Obrazovka všetkých produktov

## Detail produktu

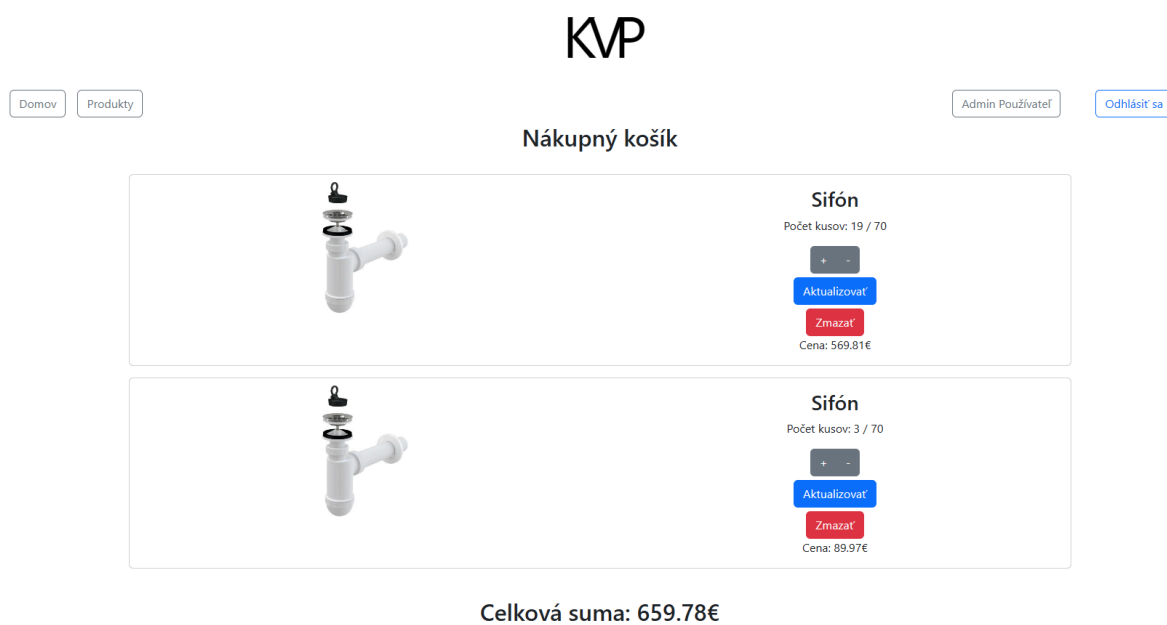
Na nasledujúcom obrázku môžeme vidieť obrazovku detailu produktu. Na tejto obrazovke si vie návštevník pomocou carouselu pozrieť obrázky produktu, prečítať informácie o produkte, zvoliť množstvo a pridať ho do košíku. Administrátor sa vie prekliknúť a zmeniť informácie o produkte alebo produkt vymeniť.



Obrázok 12 - Detail produktu

## Nákupný košík

Na nasledujúcom obrázku môžeme vidieť obrazovku nákupného košíku. Na tejto obrazovke vidí návštevník produkty, ktoré vložil do košíka. Návštevník vie zmeniť počet kusov v košíku alebo produkt odstrániť.



Obrázok 13 - Nákupný košík



## Obrazovka platby

Na nasledujúcom obrázku môžeme vidieť obrazovku platby. Na tejto obrazovke zadá návštevník svoje údaje o platbe a doručení, tieto údaje a objednávka sa uložia do databázy po kliknutí na Dokončiť objednávku.

PSČ:

123 45

Krajina:

Slovensko

Telefónne číslo:

+421900123456

Spôsob dopravy:

- ☐ Kuriér na adresu  
☐ Packeta ZBOX

Spôsob platby:

- ☒ Kartou online  
☐ Dobierka

Číslo karty:

XXXX XXXX XXXX XXXX

Dátum: MM/YY

CVC: XXX

Dokončiť objednávku

Obrázok 14 - Obrazovka platby

# Admin rozhranie

## Pridanie nového produktu

Na nasledujúcom obrázku môžeme vidieť obrazovku pridania nového produktu. Na tejto obrazovke vie prihlásený administrátor zadať údaje o novom produkte, ktorý chce vložiť.

The screenshot shows the 'Nový produkt' (New Product) form in the KVP admin interface. The form is centered on the page and includes the following fields and elements:

- Navigation:** 'Domov' and 'Produkty' buttons on the top left; 'Admin Používateľ' and 'Odhlásiť sa' buttons on the top right.
- Header:** 'KVP' logo at the top center.
- Back Button:** 'Späť' (Back) button above the form title.
- Form Title:** 'Nový produkt' (New Product).
- Fields:**
  - Názov produktu:** Text input with placeholder 'Zadajte názov'.
  - Cena:** Price input with a currency symbol '€' and a value of '0'.
  - Počet kusov:** Quantity input with a value of '0' and a unit 'ks'.
  - Kategória:** Dropdown menu with 'Kúrenie' selected.
  - Informácie o produkte:** Text input with placeholder 'Info'.
  - Vložte obrázky:** File upload section with a button 'Vybrať súbory' and a message 'Nie je vybratý žiadny súbor'.
- Action Button:** 'Vytvoriť nový produkt' (Create new product) button at the bottom.

Obrázok 15 - Pridanie nového produktu

## Vymazanie alebo upravenie produktu

Na nasledujúcom obrázku môžeme vidieť obrazovku vymazania a upravenia produktu. Na tejto obrazovke vie administrátor upraviť údaje o produkte, pridávať alebo odoberať jeho obrázky alebo produkt vymazať.

The screenshot displays the KVP (Kuchyňa, Voda, Plyn) product management interface. At the top, there are navigation links: "Domov" and "Produkty" on the left, and "Admin používateľ" and "Otvoriť sa" on the right. The main content area is titled "KVP" and contains a form for managing a product. The form includes fields for "Název" (Name), "Cena" (Price), "Počet" (Quantity), and "TO". Below these fields is a section for "Informácia o produkte" (Product Information), which contains a text area for "Súhrn" (Summary) and a section for "Obrázky" (Images). The "Obrázky" section shows a list of images with a "Vložiť a zmeniť" (Add and change) button and a "Vymazať produkt" (Delete product) button. Below the images, there is a section for "Existujúce obrázky" (Existing images) with a "Vložiť a zmeniť" button and a "Vymazať produkt" button. The interface is clean and modern, with a white background and blue accents.

Obrázok 16 - Vymazanie alebo upravenie produktu