

Slovenská Technická Univerzita, Bratislava
Fakulta Informatiky a Informačných Technológií

Elektronický obchod - semestrálny projekt

Základy webových technologíí

11.5.2025
Bratislava

Terézia Klieštiková
Alena Múková

Obsah

Zadanie	3
Databáza	3
E-R dátový model	3
Fyzický dátový model	4
Zmeny v dátovom modeli oproti fáze 1	4
Návrhové rozhodnutia	5
Opis implementácie vybraných prípadov použitia.....	7
Zmena množstva pre daný produkt.....	7
Prihlásenie.....	7
Vyhľadávanie	8
Pridanie produktu do košíka	8
Stránkovanie	9
Základné filtrovanie	10
Preusporiadanie produktov.....	10
Admin – pridanie produktu.....	10
Admin - odstránenie produktu.....	10
Admin - úprava produktu	11
Snímky obrazoviek	11
Používateľská príručka (Windows / MacOS)	17
Používateľská príručka na zakúpenie produktu.....	18
Záver	18

Zadanie

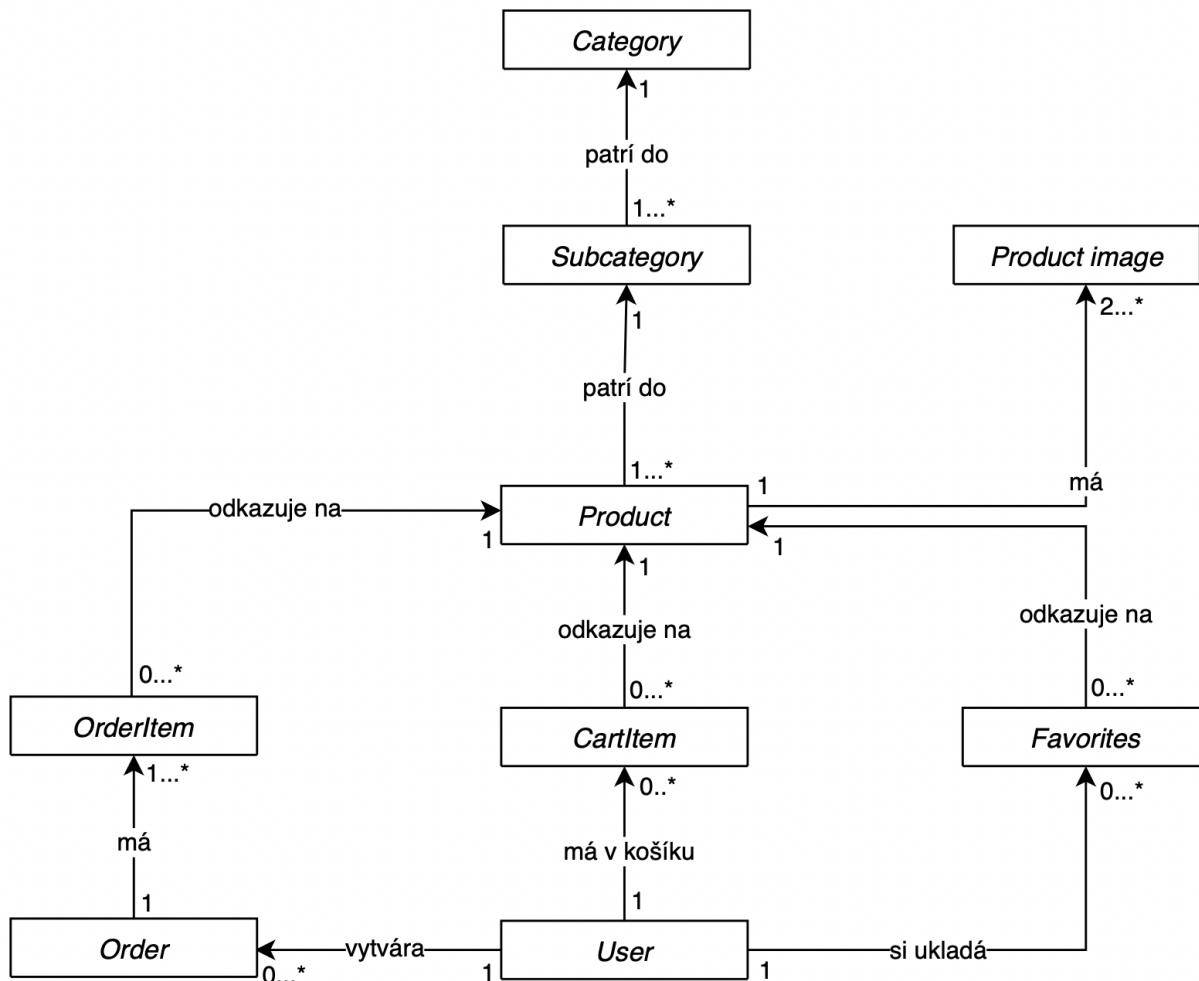
Našou úlohou bolo vytvoriť vlastnú webovú aplikáciu – e-shop, ktorá komplexne rieši typické prípady použitia elektronického obchodu. Aplikácia je vytvorená vo dvojici a pozostávať z klientskej aj administrátorskej časti. V rámci klientskej časti sme implementovali funkcie ako prehľad produktov s filtrovaním a stránkovaním, detail produktu, nákupný košík, registráciu, prihlásenie a možnosť uskutočniť objednávku. Administrátorská časť obsahuje správu produktov (vytváranie, úprava, mazanie).

Projekt bol vypracovávaný v 2 hlavných častiach. V prvej časti sme tvorili návrh používateľského rozhrania v podobe skíc v nástroji Figma a návrh dátového modelu. Z týchto skíc sme následne vytvorili HTML, CSS a JavaScript šablóny ktoré sú plne responzívne. V druhej fáze projektu sme aplikovali PHP framework Laravel, ktorý využíva server-side rendering. Vďaka tomu sa nám podarilo vytvoriť plne funkčný interaktívny e-shop.

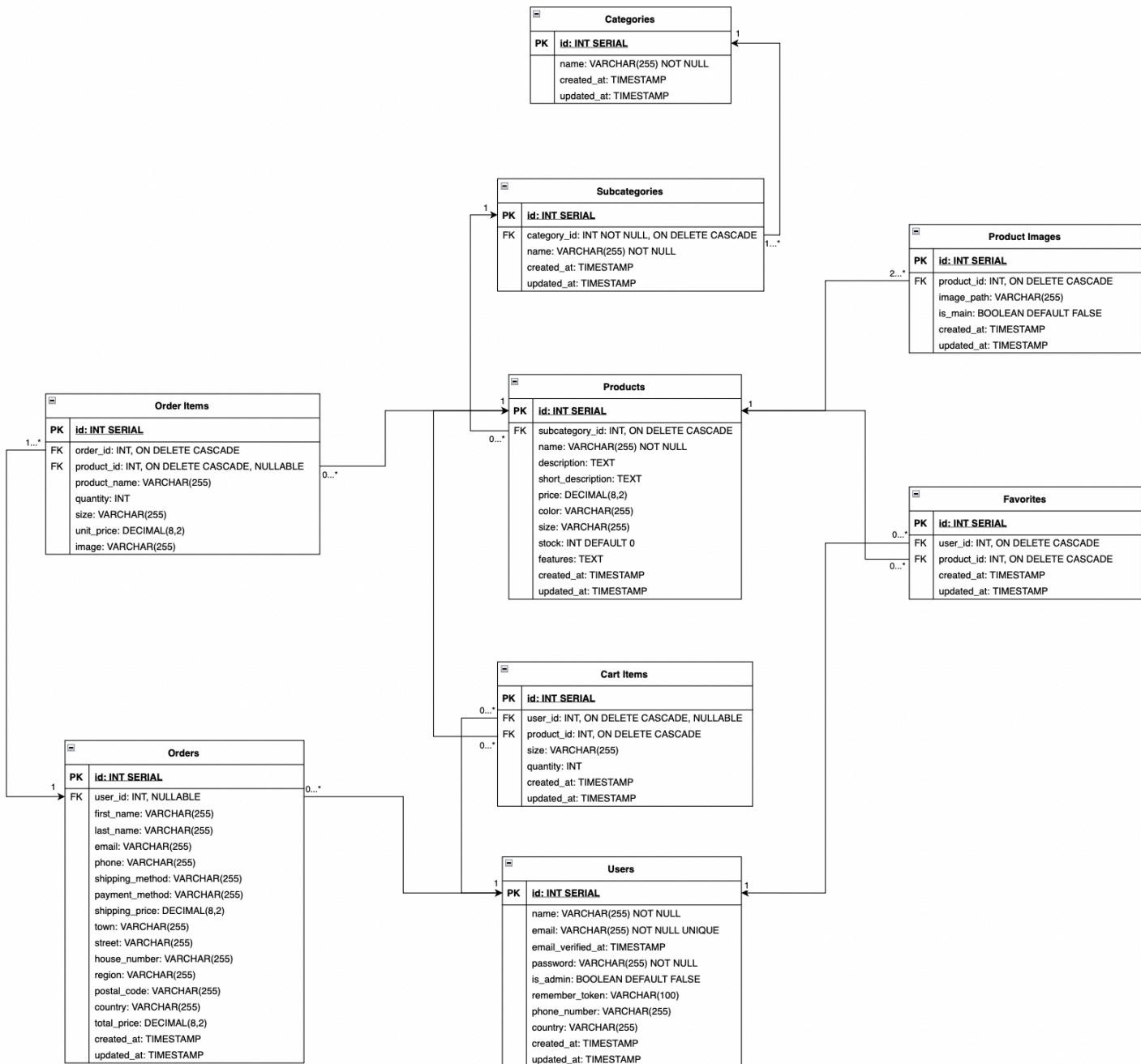
Databáza

Pracujeme s databázovým systémom PostgreSQL.

E-R dátový model



Fyzický dátový model



Zmeny v dátovom modeli oproti fáze 1

Pridanie tabuľiek **Category** a **Subcategory** umožnilo lepšiu organizáciu produktov pričom má používateľ možnosť filtrovať produkty aj podľa týchto entít. Tabuľky **Payment** a **Shiping** sme zahrnuli do tabuľky Orders, čím sme zabezpečili, že všetky dôležité informácie o objednávke sú v jednej entite a nemusíme zbytočne využívať JOIN (spájanie tabuľiek). Rozhodli sme sa odstrániť tabuľku **Search History**, pretože sme ukladanie histórie vyhľadávania neimplementovali – bolo by to zbytočné kvôli komplexite a nenašli sme tomu praktické využitie.

V tabuľkách pribudlo viac atribútov: Tabuľke **Products** pribudol textový atribút **short_description**, pretože sa v detaile produktu tento textový atribút zobrazuje a musí byť pridelený konkrétnemu produktu. Admin zároveň musí byť schopný zmeniť **short_description** nezávisle od základej **description**. Tabuľke **Users** pribudli atribúty **name**, **phone_number**, **country** na uchovanie základných informácií o používateľovi a kvôli automatickému vyplneniu dodacích údajov k objednávkam. Pribudli aj časový atribút **email_verified_at** a atribút **remember_token**. Tabuľka **OrderItem** pribudol atribút **product_name**, ktorý slúži najmä na zobrazenie objednávok v profile používateľa. Tabuľka **Product Image** má zmenený atribút **name** na **image_path** pretože meno obrázka je nepotrebné, potrebujeme ukladať cestu k nemu. Taktiež pribudol aj atribút **is_main**, ktorý určuje, či je tento obrázok hlavným obrázkom produktu (ten sa prioritne zobrazuje v liste produktov).

Tabuľkám **CartItem**, **Favorites**, **Product Image** pribudli časové atribúty **created_at**, **updated_at**.

Návrhové rozhodnutia

Pri vývoje webovej aplikácie sme si vybrali **PHP framework Laravel**, ktorý poskytuje robustný základ pre vývoj webových aplikácií a bol odporúčaný na prednáškach. Vďaka Laravelu sme zvládli vytvoriť backendovú logiku bez potreby pridávania externých knižníc.

Zvolili sme si použitie databázového systému **PostgreSQL**, keďže s týmto systémom máme skúsenosť z predmetu Databázové systémy a dokázal pokryť všetky naše požiadavky.

Pri návrhu používateľského rozhrania sme využili populárny **CSS framawork – Bootstrap**, ktorý nám poskytol preddefinované komponenty a respontívny grid systém vďaka čomu sme nemuseli definovať niektoré štýly samé od základov. Bootstrap je do projektu pridaný prostredníctvom CDN linku v hlavičke (head) layoutu.

Obrázky sú zobrazované v Blade šablónach pomocou HTML tagu `` a generujú sa dynamicky podľa src atribútu na základe cesty uloženej v databáze. Zvolili sme prístup, že v databáze máme pre každý obrázok uloženú iba cestu k súboru, nie binárny obsah súboru, čo zabezpečuje menšiu záťaž databázy. Samotné obrázky sa potom nachádzajú v priečinku projektu `/public/images`.

Kedže vytvárame fiktívny eshop pre účely predmetu WTECH a nevlastníme reálne produkty, čiže ani ich fotky, produkty sme čerpali z iných webových stránok alebo z open-source zdrojov z internetu. Chceli sme mať jednotný dizajn fotiek produktov. Nepodarilo sa nám nájsť dostatočne veľa open-source fotiek produktov do nášho e-shopu tak, aby bola medzi fotkami zachovaná konzistentnosť, a preto sme sa rozhodli použiť obrátky produktov z verejne dostupných stránok na internete. Po dôkladnom zvážení a rešpektovaní rozsahu použitia - nekomerčné a čisto študijné účely sme obrázky čerpali zo stránok pukassurfshop.com, mctavish.com, decathlon.sk, surflocos.com. Tieto obrázky sme zvolili pre ich vizuálnu kvalitu a estetiku. Sme si vedomé, že tieto obrázky podliehajú autorským

právam, a preto by boli v reálnom projekte nahradené vlastnými alebo licencovanými obrázkami.

Rozhodli sme sa neimplementovať komplexný **systém oprávnení a rolí**, keďže v našom eshope existuje iba zákazník a admin. Použitie plnohodnotného laravelovského systému pre roly a oprávnenia by zbytočne zvýšilo zložitosť riešenia. Usúdili sme, že v našom prípade je dostatočné rozlišovať admin používateľa na základe atribútu `is_admin = True` v databáze v tabuľke `users`. Bežný používateľ má prístup do eshopu, môže si prehliadať produkty a nakupovať. Admin používateľ nevie nakupovať produkty – vie si ich iba prezerať. Admin má však oprávnenie cez adminské rozhranie do ktorého sa vie prihlásiť upravovať, mazať alebo pridávať nové produkty.

Registrácia a prihlásenie používateľov využíva Laravel autentifikačnú vrstvu – rozhranie `Authenticatable` – ktoré zabezpečuje bezpečnú správu identity používateľov a šifrovanie hesiel. Rozlišovanie medzi zákazníkom a adminom je riešené cez vlastný boolean atribút `is_admin`.

Prístup do adminského rozhrania sme sa rozhodli realizovať oddeleným prihlasovacím rozhraním, ktoré nie je verejne viditeľné ani odkazované zo zákazníckej časti aplikácie. Prístup je možný iba cez URL `/login/admin`. Na tejto adrese je sprístupnený prihlasovací formulár, ktorý umožňuje prihlásenie iba administrátorom (používateľa, ktorí majú `is_admin = True`). Po úspešnom prihlásení je admin presmerovaný do administrátorského rozhrania, kde má prístup ku správe produktov. Toto riešenie minimalizuje viditeľnosť admin loginu (v budúcnosti by bolo vhodné ešte viac zatajiť URL na admin login) a zabezpečuje prístup iba adminom so správnymi prihlasovacími údajmi.

Pri **validácii údajov v nákupnom košíku** sme sa rozhodli pre riešenie na strane klienta pomocou JavaScriptu. Tento prístup nám umožnil okamžitú spätnú väzbu používateľovi bez potreby obnovovania stránky alebo volania servera. JavaScriptom sme kontrolovali povinné polia, správnosť emailovej adresy, číselné hodnoty pre telefón, PSČ a číslo domu, ako aj formát platobných údajov. V prípade platby kartou sme implementovali vlastnú validáciu dátumu expirácie, formátu čísla karty, mena na karte a CVC kódu.

Dôvodom pre túto voľbu bola jednoduchosť implementácie, zvýšenie používateľského komfortu a lepšia interaktivita formulárov. Vzhľadom na to, že ide o školský projekt bez reálnej integrácie platobných brán alebo externých služieb, nepovažovali sme za potrebné pridávať serverovú validáciu vo všetkých krokoch. V reálnom nasadení by sme samozrejme doplnili aj validáciu na strane servera pre zabezpečenie integrity a ochranu pred manipuláciou.

Aj keď má používateľ v našej aplikácii na výber viaceré spôsoby platby ako napríklad Apple Pay, PayPal, či GooglePay, tieto možnosti v projekte nie sú reálne implementované. Ich výber slúži len na demonštráciu používateľského rozhrania a možného budúceho rozšírenia.

V produkčnom nasadení by sme tieto platobné brány prepojili s príslušnými API a zabezpečili by sme celý proces ich integrácie a overovania.

Opis implementácie vybraných prípadov použitia

Zmena množstva pre daný produkt

Používateľ môže zmeniť množstvo konkrétneho produktu priamo v košíku prostredníctvom `<input type="number" name="quantity">`. Tento prvk je umiestnený vo formulári, ktorý sa automaticky odosielá pri zmene hodnoty `cez onchange="this.form.submit()"`. Používateľ si tiež môže vybrať množstvo daného produktu priamo v detaile produkta `product_detail.blade.php` kde sa pri zmene hodnoty aktualizuje formulár `cez on_change = "updateQuantity()"`. Metóda je definovaná v Javascripte `product_details.js`. Dôležitou vlastnosťou formulára je aj `@method = "PUT"`, pomocou ktorej aktualizujeme hodnoty existujúceho zdroja – teda nášho form.

Formulár smeruje na route definovanú v `routes/web.php`:

```
Route::put('/cart/update/{id}', [CartController::class, 'updateCartItem'])->name('updateCartItem');
```

Spracovanie zmeny sa vykonáva v metóde `updateCartItem()` v `CartController`.

- **Pre prihláseného používateľa** sa zmení záznam v databáze v tabuľke **CartItems** - nájde sa položka podľa `user_id` a `product_id`, následne sa aktualizuje veľkosť (`size`) a množstvo (`quantity`).
- **Pre neprihláseného používateľa** sa aktualizuje hodnota v session `session() ->get ('cart')` a nová verzia košíka sa uloží späť `session() ->put ('cart', $cart)`.

Následné zobrazenie položiek košíka sa vykoná v metóde `step1()` v rovnakom kontroléri.

Táto metóda načíta bud:

- dátá z databázy (ak je používateľ prihlásený), alebo
- zo session (ak nie je prihlásený),

a odošle ich do Blade šablóny `shopping_cart1.blade.php`.

Prihlásenie

Používateľ sa do svojho konta v eshope prihlásuje cez rozhranie vytvorené v blade šablóne `login.blade.php` (admin sa prihlasuje cez `admin_login.blade.php` a prihlásenie spracováva `AdminController` vo funkcií `login()`). Používateľ vidí dva formuláre:

Registračný formulár:

Formulár obsahuje polia pre meno, email, telefónne číslo, krajinu, heslo a potvrdenie hesla. Odosielá sa POST požiadavkou a je zabezpečený pomocou csrf. Keď používateľ vyplní všetky polia formulára, tak formulár sa odosielá buttonom Register. Samotná registrácia sa spracováva v `UserController` vo funkcií register, ktorá validuje vstupné údaje pomocou `validate()` kde sa kontroluje jedinečnosť emailu a správnosť a zhoda vytvoreného hesla.

Používateľ sa zapisuje do databázy do tabuľky **users** cez `User::create()` spolu s heslom, ktoré je zahashované funkciou `Hash::make()`. Nový registrovaný používateľ je ihneď prihlásený do svojho konta pomocou laravel funkcie `Auth::login()` a presmerovaný na hlavnú stránku. V prípade nesprávnych registrčných údajov je používateľ presmerovaný späť a zobrazia sa mu aj chyby, ktoré nastali pri validácii.

Prihlasovací formulár:

Login formulár obsahuje polia pre email a heslo. Ak pri validácii údajov nastanú chyby, tie sa následne taktiež zobrazia ako chybové hlášky. Formulár sa odosielá POST požiadavkou. A následne sa spracováva v `UserController` vo funkcií `login()`. Po odoslaní vykonám kontrolu, či sú polia vzplnené, a či sa snaží prihlásiť bežný používateľ (tento formulár nedovoluje prihlásenie adminovi). Následne sa vykoná kontrola správnosti prihlasovacích údajov pomocou laravel funkcie `Auth::attempt()` a v prípade správnych prihlasovacích údajov je používateľ prihlásený do svojho konta a presmerovaný na homepage. Ak je prihlásenie neúspešné, zobrazí sa chybová hláška s touto informáciou.

Vyhľadávanie

Používateľ dokáže vyhľadávať produkty podľa mena. Vyhľadávací formulár sa nachádzav blade šablónach `homepage.blade.php` a `products.blade.php`. (Ďalšie vyhľadávanie mám aplikované v adminovom rozhraní, kde admin dokáže vyhľadávať produkty, ktoré bude potrebuje na úpravu. Adminovské vyhľadávanie produktov funguje podobne ako vyhľadávanie bežného používateľa.)

Vyhľadávací formulár posielá GET požiadavku s parametrom ‘search’. Vyhľadávanie prebieha už v rámci aktuálnej URL adresy napr. `/products?search= ...` Vyhľadávanie spracováva `ProductController` v metódach `index()`, `byCategory()` alebo `bySubcategory()`. Vo všetkých metódach je vyhľadávanie implementované rovnako. Kontrolujem, či bol parameter `search` vyplnený. Ak áno, tak na query ktorá bude vyhľadávať vyhovujúce produkty v tabuľke **products** ktoré sa napokon používateľovi zobrazia aplikujem filter podľa `name` stĺpca s použitím `ILIKE`, ktoré je case-insensitive fulltext vyhľadávanie v PostgreSQL databáze.

Pridanie produktu do košíka

Používateľ môže pridať produkt do nákupného košíka priamo z prehľadu produktov alebo zo stránky detailu produktu. Táto funkcia využíva HTTP metódu POST, keďže ide o vytvorenie novej položky v košíku

Pridanie z katalógu produktov `products.blade.php` :

Zoznam produktov zobrazuje pre každý produkt ikonu na pridanie alebo odstránenie z košíka. Ak produkt ešte nie je v košíku, zobrazí sa formulár s POST požiadavkou. Ak je používateľom admin, tak sa mu ikona na pridanie a odstránenie produktu do košíka nezobrazí. Admin si teda produkt zakúpiť nemôže.

Pridanie z detailu produktu product_detail.blade.php :

Na stránke detailu si používateľ môže zvoliť veľkosť a množstvo. Formulár sa po každej zmene aktualizuje pomocou onchange="updateSize()" a onchange="updateQuantity()". Metódy sú definované v Javascripte product_details.js. Po stlačení <button type="submit" form="add-to-cart-form"> sa odošle formulár s POST požiadavkou. Ak je používateľom admin, tak sa mu button na pridanie produktu do košíka nezobrazí. Admin si teda produkt zakúpiť nemôže.

Formulár smeruje na route definovanú v routes/web.php :

```
Route::post('/cart/add/{id}', [CartController::class, 'addToCart'])->name('addToCart');
```

Metóda addToCart v kontroléri CartController :

Ak je používateľ prihlásený, hľadáme existujúci záznam o danom produkте v košíku používateľa priamo cez databázu. Ak daný záznam už existuje, zvýši sa počet tohto produktu o 1 a ak produkt neexistuje, tak sa tento produkt vloží do databázy do tabuľky **CartItems** s cudzím kľúčom na používateľa, cudzím kľúčom na daný produkt, s jeho vybranou veľkoťou a počtom kusov.

Ak používateľ prihlásený nie je, pracujeme so sessions. Pomocou session()->get('cart') vyberieme aktuálne položky v košíku. Ak sa tam daný produkt už nachádza, zvýšime jeho počet o 1, a ak sa tam ešte nenachádza, pridá sa nová položka do session s cudzím kľúčom na produkt a jeho vybranou veľkosťou a počtom kusov. Cez session()->put('cart', \$cart) sa vloží daný záznam do session.

Stránkovanie

Stránkovanie aplikujem pri prezeraní produktov v blade šablóne products.blade.php v časti <div class="strankovanie"> (v admin profile je stránkovanie aplikované taktiež). Stránkovanie sa spracováva v ProductControlleri v metódach index(), byCategory() alebo bySubcategory(). Vo všetkých metódach je stránkovanie implementované rovnako – pomocou Laravel metódy paginate(12) ktorá automaticky zobrazí produkty do stránok po 12 produktov. 12 sme zvolili kvôli príjemnosti prehliadania produktov a taktiež kvôli tomu, aby sa dalo stránkovanie aplikovať pre ukážku aj na menší počet produktov. V budúcnosti by sa s touto hodnotou dalo jednoducho hýbať. Vo view pracujem s laravel metodami currentPage(), lastPage(), hasMorePages(), onFirstPage(), vďaka ktorým dokážem efektívne pracovať s jednotlivými komponentami vo view a pridať im funkčnosť zmeny stránky tak ako potrebujeme. Pri presune na novú stránku zachovávam všetky aktuálne nastavené filtre, zoradenia a vyhľadávania v URL pomocou request()->fullUrlWithQuery(...).

Základné filtrovanie

Naša webová aplikácia umožňuje filtrovanie zobrazených produktov na základe ceny, subkategórie farby a veľkosti. Rozhranie pre filtrovanie sa nachádza vo view products.blade.php. Filtrací formulár odosielá filtrové dátá cez URL a GET požiadavku. Vo filtrovacom formulári sa dá zvoliť cenový rozsah pomocou 2 `<input type="range">`. Subkategória, farba a veľkosť sa volia pomocou `<input type="checkbox">`. Filtrovanie sa spúšťa tlačidlom Apply filter.

Filtrovanie sa spracováva v `ProductController` funkciou `applyFilter()`. Vstupom je `$request` (GET dátá) a `$query` na ktorú sa filtrovanie bude aplikovať. Vo funkcií sa postupne kontroluje subkategória, farba a veľkosť pomocou `whereIn` a `where`. Cena sa kontroluje a aplikuje pomocou `whereBetween`. Výsledná `$query` potom pokračuje ďalej do funkcie, ktorá zobrazuje stránku s produktami.

Vďaka javascriptu v `products.js` je filter interaktívny. Pri načítaní stránky sa v blade šablóne automaticky predvyplnia checkboxy/sliders hodnoty na základe `value = request()`.

Preusporiadanie produktov

Vo view `products.blade.php` má používateľ možnosť preusporiadať si zobrazenie produktov podľa ceny alebo abecedy. Túto funkcionality implementuje úsek kódodu v `<div class="dropdown">`. Rozbalovacie tlačidlo s textom Sort by otvára zoznam možností. Každá možnosť je odkaz, ktorý aktualizuje URL parametre. Toto sa spracováva v `ProductController` vo funkcií `applySorting()`, ktorá číta parameter 'sort' a na základe toho sa vykoná `orderBy()`.

Admin – pridanie produktu

Keď je admin prihlásený vo svojom profile, po kliknutí na Add product button vo view, ktoré je v `admin_profile.blade.php` sa mu zobrazí overlay s formulárom na pridanie nového produktu. Formulár obsahuje polia - niektoré sú povinné a bez nich formulár nepôjde odoslať. V `admin_profile.js` sa validujú polia ešte pred odoslaním formulára a poskytuje adminovu spätnú vazbu v reálnom čase. Ak vyplní všetky povinné polia, tak sa odblokuje button na odoslanie formuláru. Samotné pridanie produktu sa vykonáva v `AdminController` vo funkcií `add()`. Každý produkt je najskôr validovaný a následne uložený do databázy - do tabuľky **products** čím predchádzame chybám. Obrázky sa fyzicky ukladajú do `public/images/products/` a ich názov sa zapisuje do tabuľky **product_images**.

Admin - odstránenie produktu

V administrátorskom rozhraní (`admin_profile.blade.php`) je pri každom produkте tlačidlo Delete. Po kliknutí na tlačidlo sa zobrazí potvrzovacie okno (`deleteOverlay`), v ktorom musí admin tlačidlom Yes potvrdiť vymazanie produktu z eshopu. To odošle DELETE požiadavku na URL adresu, ktorá je dynamicky nastavená pomocou `admin_profile.js` podľa ID produktu, ktorý sa vymazáva. Server spracováva požiadavku v `AdminController` v metóde `delete()`, ktorá vymaže všetky obrázky produktu z databázy z tabuľky

product_images, ktoré patria tomuto produktu a potom aj samotný produkt z **products** tabuľky.

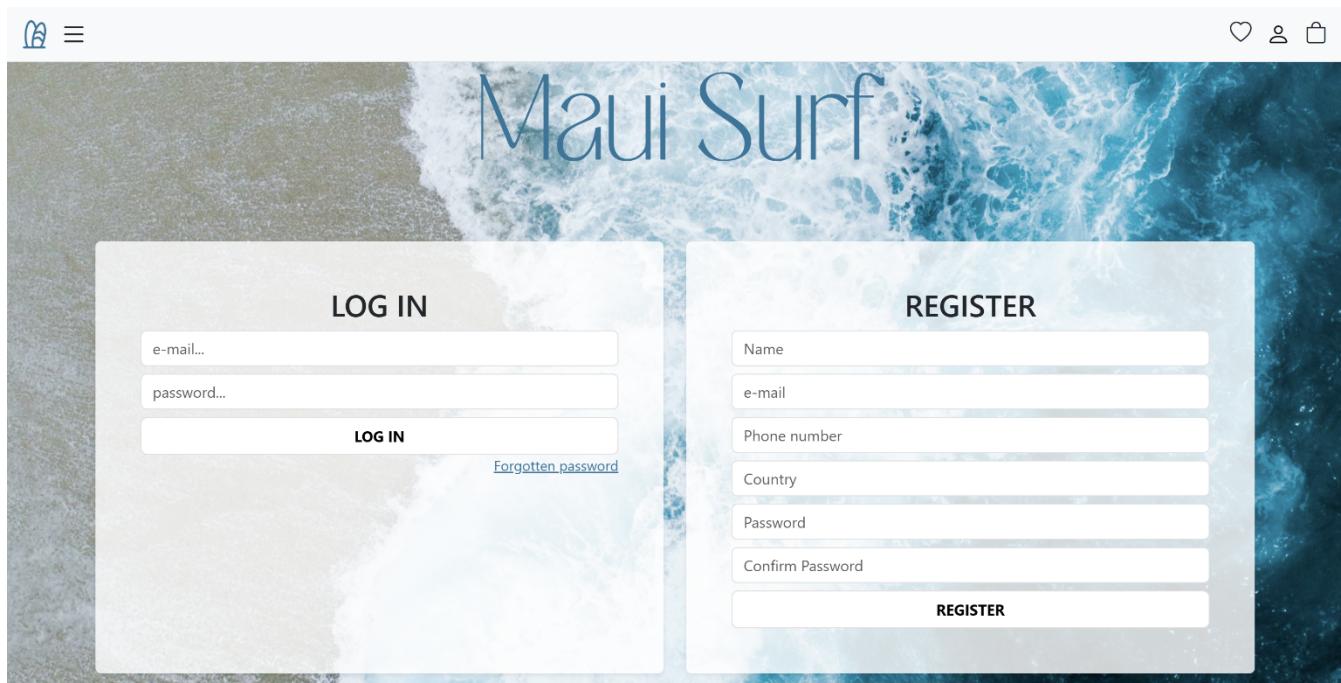
Rozhodli sme sa, že pri vymazávaní produktu, s čím súvisí vymazávanie príslušných obrázkov nebudeme vymazávať obrázok aj fyzicky z priečinka ale iba záznam z databázy. Toto rozhodnutie sme spravili preto, lebo máme nedostatok obrázkov a niektoré obrázky sú priradené viacerým produktom.

Admin - úprava produktu

V administrátorskom rozhraní (`admin_profile.blade.php`) je pri každom produkte tlačidlo Edit. Po kliknutí na tlačidlo sa vďaka `admin_profile.js` načíta formulár so všetkými predvyplnenými údajmi pre daný produkt. Administrátor následne môže zmeniť tieto údaje. Zabezpečujeme, že produkt musí mať vždy minimálne 2 fotografie. Po odoslaní PUT požiadavky sa formulár spracováva v `AdminController` vo funkcií `edit()`. Prebehne validácia údajov a zmeny v databáze sa vykonajú iba pre tie atribúty, ktoré sú nastavené správne. V opačnom prípade na týchto atribútoch úprava neprebehne.

Snímky obrazoviek

PRIHLÁSENIE ZÁKAZNÍKOV



Homepage

The waves are calling - ride them in style! Find the best surf gear here.

SURFBOARDS EQUIPMENT ACCESSORIES

More

Surfboards

Equipment

Accessories

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

CUSTOMER SERVICE

Share your experience with your friends on socials

Return & Delivery
Reviews
About us

Prehľad produktov

Maui Surf

Search...

SURFBOARDS EQUIPMENT ACCESSORIES

WAVE RIDER PRO \$499.99

WAVE HUNTER \$399.99

ABC RIDER \$199.99

STORMSURGE \$899.99

SUNDRIFTER \$349.99

REEF RIDER \$1,999.99

OCEAN PULSE \$1,199.99

BLUE ECHO \$783.00

SALTSPRINT \$1,299.99

SEA ZEN \$199.99

DRIFT NOVAN \$399.99

MAUI GO \$1,399.99

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

CUSTOMER SERVICE

Return & Delivery
Reviews
About us

Share your experience with your friends on socials

Facebook Instagram LinkedIn Twitter

DETAIL PRODUKTU

[Maui Surf](#)

Wave rider Pro
★★★★★ [Reviews](#)

Fast and responsive shortboard for your daily surfing sesh

Size: S | Amount: 1 pc
499.99\$

[Buy](#)

Features

- Carbon fiber rails
- Tri-fin setup

Description

High-quality surfboard for experienced surfers.

Delivery & Returns

Reviews

We think you will like

POUŽÍVATEĽSKÝ PROFIL

[Maui Surf](#)

Your orders

Product	Size	Amount	Prize
Wave rider Pro	S	1 pc	\$499.99
BlueEcho	L	1 pc	\$783.00
Maui vibes	M	1 pc	\$129.99
Maui fast Fin	S	3 pc	\$329.97

Alena

[LOG OUT](#)
[EDIT PROFILE](#)
[CHANGE PASSWORD](#)

Profile details

Email: a@a.com
Phone: 1234567890
Country: Slovensko

"You can't stop the waves but you can learn to surf"

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

CUSTOMER SERVICE

Return & Delivery
Reviews
About us

Share your experience with your friends on socials

[Facebook](#) [Instagram](#) [LinkedIn](#) [Twitter](#)

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

CUSTOMER SERVICE

Return & Delivery
Reviews
About us

Share your experience with your friends on socials

[Facebook](#) [Instagram](#) [LinkedIn](#) [Twitter](#)

KOŠÍK – 1. krok

Maui Surf

What's in my bag

①. step ----- ②. step ----- ③. step

	Wave rider Pro	<input type="button" value="remove"/>
S ▾	1	
Price: 499.99 \$		
	BlueEcho	<input type="button" value="remove"/>
L ▾	1	
Price: 783.00 \$		
	Maui vibes	<input type="button" value="remove"/>
M ▾	1	
Price: 129.99 \$		
	Maui fast Fin	<input type="button" value="remove"/>
S ▾	3	
Price: 109.99 \$		

Total: 1,742.95 \$

[Continue Shopping](#) [Checkout](#)

KOŠÍK – 2. krok

Maui Surf

Fill out your information

①. step ----- ②. step ----- ③. step

Customer Information	Billing Information
Alena	Town
Last Name	Street
a@a.com	House Number
1234567890	Region
Shipping Method	Postal Code
<input checked="" type="radio"/> Standard Shipping 3,99\$ <input type="radio"/> Express Shipping 6,99\$	Country
Payment Method	
<input checked="" type="radio"/> Card <input type="radio"/> Apple Pay <input type="radio"/> PayPal <input type="radio"/> Google Pay <input type="radio"/> Cash On Delivery	

Shipping: 3,99\$
Total: 1746.94\$

[Go Back](#) [Continue](#)

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

CUSTOMER SERVICE

Return & Delivery
Reviews
About us

Share your experience with your friends on socials

KOŠÍK – 3. krok

≡ Maui Surf ♥ ⌂ ⌂

The last step
①. step ----- ②. step ----- **③. step**

Supported payment methods

Card Information

Name on Card

Card Number

MM/YY CVC

Shipping: 3.99\$
Total: 1746.94\$

[Go Back](#) [Purchase](#)

CONTACT CUSTOMER SERVICE Share your experience with your friends on socials

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

Return & Delivery
Reviews
About us

≡ ♥ ⌂ ⌂

POTVRDENIE o objednaní

Thank you for your purchase!

You got a confirmation email
Your order is on its way to you!



[Continue Shopping](#)

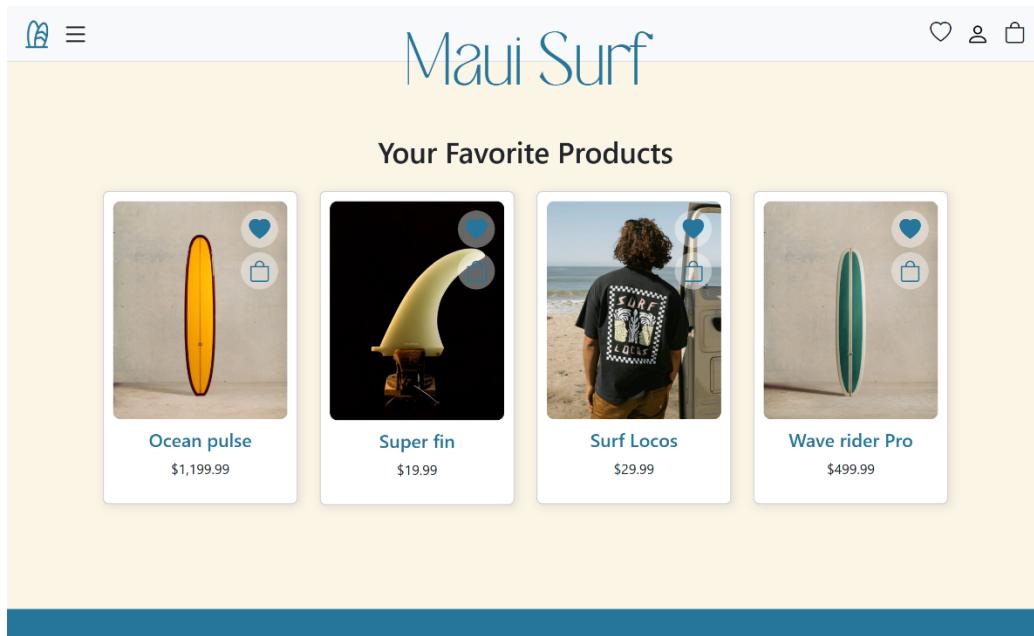
CONTACT CUSTOMER SERVICE Share your experience with your friends on socials

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

Return & Delivery
Reviews
About us

FAVORITES



The screenshot shows the 'Your Favorite Products' section of the Maui Surf mobile application. It displays four items with their names, prices, and small control icons (heart and bag).

- Ocean pulse: \$1,199.99
- Super fin: \$19.99
- Surf Locos: \$29.99
- Wave rider Pro: \$499.99

CONTACT

Email: example@email.com
Phone: +123 456 789
Address: 123 Street, City, Country

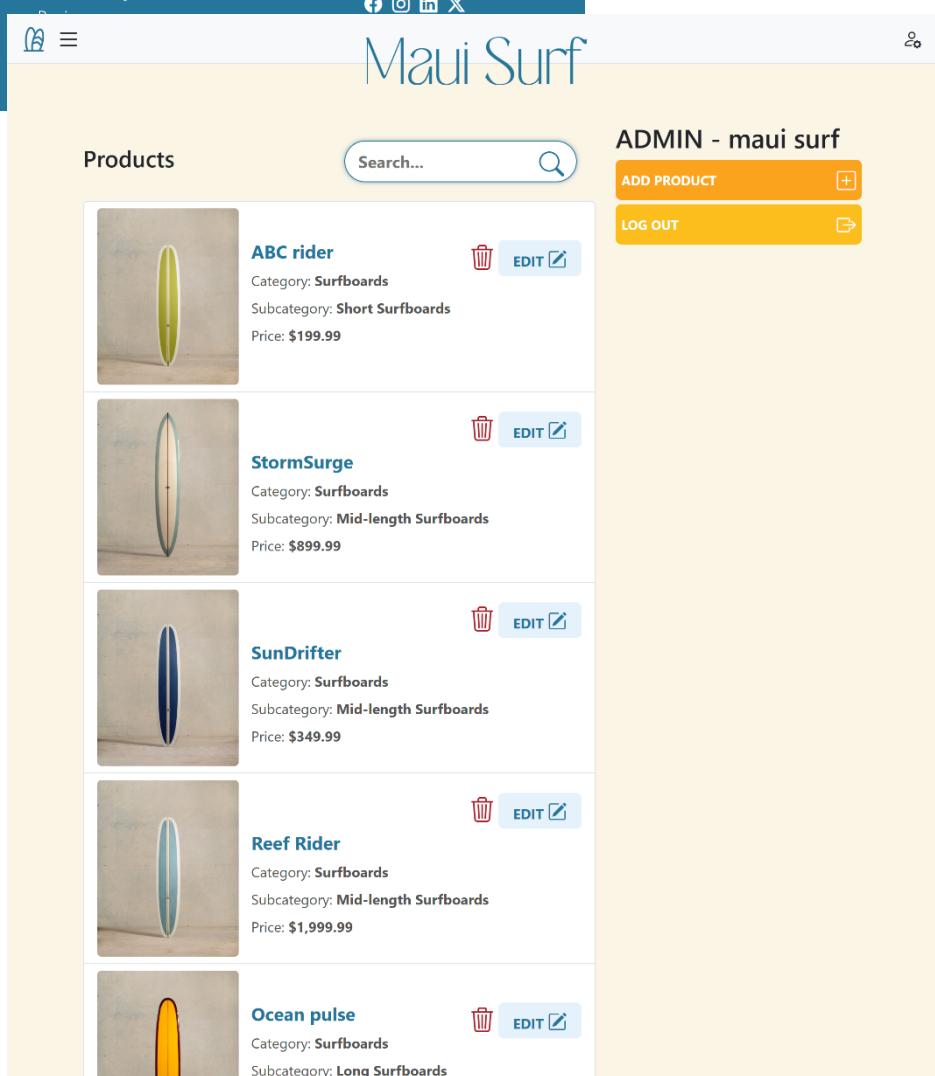
CUSTOMER SERVICE

Return & Delivery

Share your experience with your friends on socials



ADMIN rozhranie



The screenshot shows the 'Products' section of the Maui Surf administration interface. It lists five surfboards with their details and edit/delete buttons.

Product Name	Category	Subcategory	Price
ABC rider	Surfboards	Short Surfboards	\$199.99
StormSurge	Surfboards	Mid-length Surfboards	\$899.99
SunDrifter	Surfboards	Mid-length Surfboards	\$349.99
Reef Rider	Surfboards	Mid-length Surfboards	\$1,999.99
Ocean pulse	Surfboards	Long Surfboards	

ADMIN - maui surf

ADD PRODUCT

LOG OUT

Používateľská príručka (Windows / MacOS)

Táto príručka vás prevedie celým procesom úspešného spustenie nášho Laravel e-shop projektu. Nevyžaduje žiadne predchádzajúce nastavenie – každý krok je vysvetlený.

1. Predpoklady:

Je potrebné, aby ste pred spustením mali nainštalované tieto nástroje:

- PHP - minimálne verzia 8.1
- Composer
- PostgreSQL

2. Stiahnutie nášho projektu:

- Stiahnite si súbor nášho projektu z AIS – **wtech_mauisurf.zip**
- Súbor môžete stiahnuť aj z nášho githhubu:
<https://github.com/xmukova/WTECH-surf>
- Rozbalte zip súbor

3. Otvorte priečinok v termináli:

```
cd nazov_projektu
```

4. Nainštalujte si PHP závislosti:

```
composer install
```

5. Pripravte si súbor .env

- Skopírujte .env.example do .env

```
cp .env.example .env
```
- Upravte si ho podľa svojich nastavení

```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432 (alebo iný port v závislosti od vašej PostgreSQL konfigurácie)
DB_DATABASE=databaza (alebo iný názov)
DB_USERNAME=postgres
DB_PASSWORD=heslo (heslo do PostgreSQL)
```

6. Vygenerujte aplikáčny kľúč

```
php artisan key:generate
```

7. Spusťte migrácie a naplňte databázu dátami:

```
php artisan migrate
php artisan db:seed
```

8. Spusťte server

```
php artisan serve
```

Prihlásanie ako Admin

Ak sa chcete prihlásiť do rozhrania pre Admina, prihlásenie je možné iba cez url:

<http://127.0.0.1:8000/login/admin>

Prihlasovacie údaje pre admina sú:

e-mail: admin@mauisurf.com

heslo: heslokleslo

Prihlásanie ako zákazník

Ak sa chcete prihlásiť do zákazníckeho účtu, je potrebné sa najprv registrovať, čím si vytvoríte vlastný účet.

Používateľská príručka na zakúpenie produktu

V košíku sme implementovali viacero validácií, ktoré musia byť splnené v každom kroku v košíku. V **prvom kroku košíka** je potrebné mať aspoň jeden produkt na to, aby bolo možné presunutie na druhý krok. V **druhom kroku košíka** musia byť všetky polia vyplnené a to tak, že **meno, priezvisko, mesto, ulica a región** musia obsahovať aspoň jedno písmeno, **číslo domu, PSČ, tel.č.** musia obsahovať iba čísla a email musí mať formát mailovej adresy.

Po výbere metódy platby **Cash on Delivery** sa preskakuje tretí krok košíka (vyplnenie karty), pretože sa platí na dobierku. V opačnom prípade v **treťom kroku košíka** je potrebné zadať v poli Name on Card aj meno aj priezvisko, ktoré musia obsahovať iba písmená. V poli Card Number je potrebné vložiť presne 16 číslic. MM/YY reprezentuje dátum expirácie karty a túto validáciu sme tiež implementovali – vložené číslo musí byť validný dátum v budúcnosti. V poli CVC je potrebné vložiť 3 až 4 číslice.

Záver

V rámci semestrálneho projektu sme úspešne navrhli a implementovali vlastnú webovú aplikáciu - e-shop, ktorá spĺňa všetky požiadavky definované v zadaní. Projekt obsahuje plnohodnotnú klientsku časť pre bežného používateľa, ako aj administrátorskú zónu určenú na správu produktov. V aplikácii sme zabezpečili funkcionality ako prehľad produktov, filtrovanie, detail produktu, správa košíka, realizácia objednávky, registrácia a prihlásenie používateľa. Administrátor má možnosť produkty vytvárať, upravovať a mazať, vrátane nahrávania obrázkov.

Pri vývoji sme využili odporúčaný Laravel framework a PostgreSQL databázu, pričom sme dôsledne navrhli dátový model a zohľadnili princípy dobrej softvérovej architektúry. Projekt bol vypracovaný tímovo, pričom obe členky tímu sa aktívne podieľali na všetkých jeho častiach. Výsledná aplikácia plne zodpovedá požiadavkám zadania a je pripravená na rozšírenie alebo nasadenie v produkčnom prostredí. Našu priebežnú prácu sme počas implementácie projektu pridávali na [GitHub repozitár](#).