

Semestrální projekt - elektronický obchod

Zdenko Kanoš & Tomáš Majerčík

WTECH

11. mája 2025

1. Zadanie a fyzický dátový model

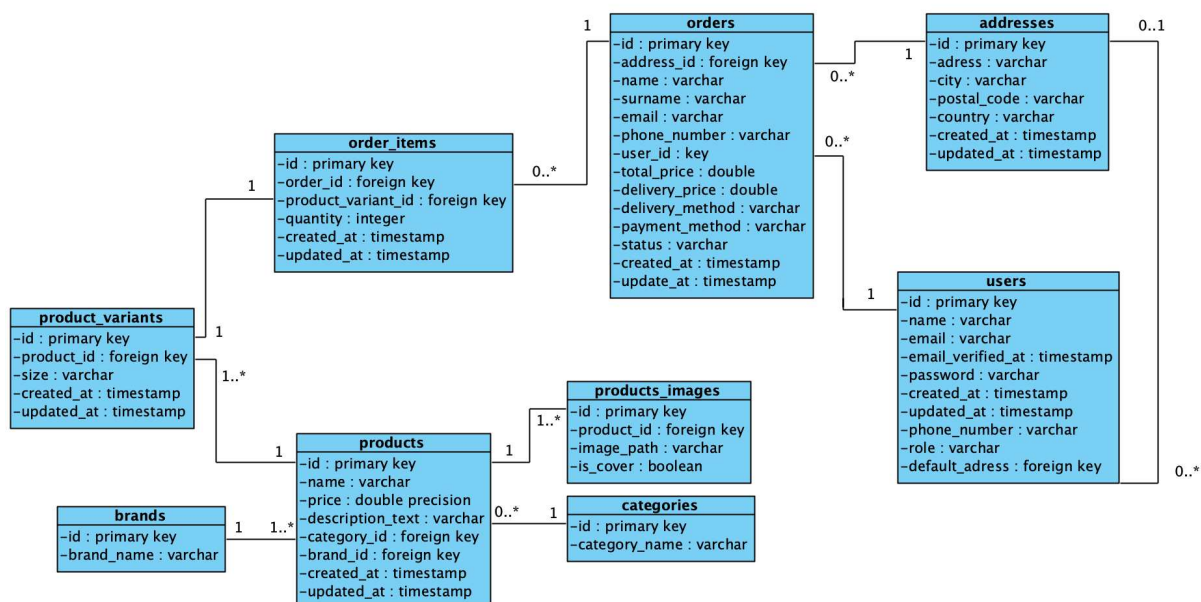
Zadanie

Cieľom semestrálneho projektu bolo vytvoriť plne funkčnú webovú stránku - **eshop**, ktorá umožňuje používateľom prehliadať, vyhľadávať a filtrovať produkty, spravovať nákupný košík, zadávať objednávky a vykonávať registráciu a prihlásenie.

Zároveň systém poskytuje **administrátorské rozhranie**, kde možno spravovať produkty vrátane obrázkov, upravovať ich a mazať. Projekt bol realizovaný ako tímová práca dvojice študentov, pričom každý musel významne prispieť k vývoju aplikácie.

Webová stránka bola vytvorená pomocou PHP frameworku **Laravel** s využitím server-side renderingu. Pri jej vývoji sme kladli dôraz na splnenie všetkých požiadaviek zadania, vrátane funkčnej klientskej a administrátorskej časti, práce s dátami, bezpečnosti používateľov a vypracovania dokumentácie.

Fyzický dátový model



Obr. 1: Fyzický dátový model

Zmeny oproti návrhu:

- Vymazaná tabuľka `user_adresses` - táto tabuľka bola vymazaná, pretože sme sa rozhodli, že v jednom čase môže používateľ mať uloženú len jednu adresu ako default. Z tohoto dôvodu bol odkaz na adresu preložený do stĺpca v tabuľke `users`.
- Presunutie ceny z `product_variants` do `products` - dôvodom je fakt, že v našom eshope sa cena pre rôzne veľkosti nerozlišuje
- Vymazaná tabuľka `payments` - táto tabuľka nebola potrebná, pretože informácia o spôsobe platby je jednoducho uložená v stĺpci `payment_method` v tabuľke `payments`.
- Ďalšie úpravy, zahŕňali pridanie timestamps pre `created_at`, `updated_at`, ktoré boli vyžadované od frameworku.
- V aktuálnom návrhu boli zmenené typy string na varchar aby reflektovali nastavenie v databáze

2. Návrhové rozhodnutia

Použité knižnice a ich dôvody

V projekte využívame externú knižnicu Sortable JS, ktorá umožňuje interaktívne presúvanie prvkov v rámci kontajnera (drag & drop). V našom prípade sme ju použili pre umožnenie zmeny poradia nahratých obrázkov produktu. Vďaka tomuto usporiadaniu je možné nastaviť hlavný obrázok pre produkt nakoľko ten je vždy prvý.

Roly a oprávnenia

Používateľské roly sme v projekte riešili pomocou stĺpca `role` v tabuľke `users`, kde sa rozlišuje medzi dvomi typmi používateľov: **admin** a **zákazník**.

Vzhľadom na jednoduchosť rolového systému sme nepoužili Laravel mechanizmy ako **Gates** alebo **Policies**. Namiesto toho sme roly kontrolovali priamo v Blade šablónach a v routách podľa hodnoty stĺpca `role`.

Admin používatelia:

- Majú prístup do administrátorského rozhrania,
- môžu pridávať, upravovať a vymazávať produkty,
- môžu spravovať produktové obrázky,
- majú prístup k štatistikám a prehľadu metrík objednávok.

Zákazníci:

- Majú prístup len ku klientskému rozhraniu e-shopu,
- môžu prehliadať produkty, filtrovať, vyhľadávať, zadávať objednávky,
- majú možnosť registrácie, prihlásenia a úpravy svojich údajov,
- v profile môžu prehliadať ich vytvorené objednávky a ich stav.

Zároveň sme zabezpečili, aby sa admin nemohol správať ako bežný zákazník – t.j. **admin používateľ nemôže pridávať produkty do košíka ani vytvárať objednávky**. Tieto funkcionality sú v rozhraní pre admina zámerne zablokované.

3. Programové prostredie

Pri vývoji aplikácie sme využívali odporúčané technológie. Backend sme implementovali v PHP frameworku **Laravel** s použitím šablónovacieho systému **Blade**. Ako databázový systém sme použili **PostgreSQL**.

Frontend sme realizovali pomocou **HTML5** a **čistého CSS**, bez použitia externých frameworkov ako Bootstrap alebo Tailwind, čo nám umožnilo mať plnú kontrolu nad štýlmi a responzivitou rozhrania.

4. Stručný opis implementácie prípadov použitia

Filtrovanie, vyhľadávanie a triedenie produktov:

Tieto funkcionality sme implementovali s využitím vlastného JavaScriptu, aby sme si udržali plnú kontrolu nad používateľským rozhraním a zároveň mohli pracovať s pôvodnými parametrami URL (napr. ponechať kategóriu počas filtrovania podľa ceny a značky). Nechceli sme, aby každá GET požiadavka úplne prepísala predchádzajúce parametre, a preto sme použili objekt `URLSearchParams`, ktorý umožňuje manipuláciu s URL bez straty predchádzajúcich hodnôt.

Filter a sort fungujú ako modálne okná, ktoré sa otvárajú cez JavaScript. Po potvrdení výberu (značka, cenový rozsah, veľkosti) sa vytvorí nová URL s parametrami a stránka sa načíta s novými výsledkami. Podobne funguje aj triedenie produktov podľa ceny a názvu. Celotextové vyhľadávanie sa aktivuje stlačením klávesy **Enter** a automaticky aktualizuje parameter `search` v URL.

- Kód pre filtrovanie: `applyFilter()`, triedenie: `applySort()`, vyhľadávanie: `handleSearchInput()`.
- Použité modálne komponenty: `filter-modal.blade.php`, `sort-modal.blade.php`.
- Využitý objekt: `URLSearchParams` na tvorbu a aktualizáciu query stringov.

Formuláre sú vo väčšine prípadov v projekte riešené pomocou AJAX (Asynchronous JavaScript and XML). Hlavným dôvodom, prečo používame AJAX, je zlepšenie používateľského zážitku – po odoslaní formulára nie je potrebný refresh, ale údaje sa odosielaajú na server na pozadí. Odpoveď zo servera sa následne spracuje priamo na stránke. Prináša výhody ako:

- Okamžité zobrazenie validačných chýb bez nutnosti obnovy stránky.
- Plynulejšia navigácia – presmerovanie prebieha len v prípade úspechu.
- Lepšia kontrola nad zobrazením chýb a aktualizáciou UI.

Pridanie produktu do košíka:

Proces pridania produktu do košíka prebieha v niekoľkých krokoch:

1. Zobrazenie detailu produktu

Používateľ si otvorí stránku `product_detail.blade.php`, kde sa zobrazí detail produktu. V rámci detailu produktu si môže zvoliť variant (veľkosť) a počet kusov.

2. Odoslanie požiadavky

Po stlačení tlačidla „Pridať do košíka“ sa formulár odošle pomocou metódy `POST` na endpoint `/cart/add`.

3. Spracovanie na strane servera

Backendová logika (v `CartController`) zabezpečí:

- pridanie položky do session pod kľúčom `cart`.
- v prípade prihláseného používateľa sa zapíše položka do databázy

4. Uloženie v session alebo databáze

Každá položka v košíku obsahuje:

- ID produktu a variantu,
- názov, obrázok, cena za kus,
- počet kusov,
- výslednú cenu ($\text{cena} \times \text{množstvo}$).

5. Spätná väzba používateľovi

Po úspešnom pridaní sa používateľovi zobrazí modal s možnosťou rýchleho prechodu do košíka, kde používateľ môže dokončiť jeho objednávku.

6. Obmedzenia a kontrola rolí

Admin používateľa nemajú povolené pridávať produkty do košíka. Tento scenár je ošetrený vypnutím tlačidiel pridávania do košíka. Pri opakovanom pridaní rovnakého produktu sa počet v košíku sčíta.

7. Prenositeľnosť košíka

Pokiaľ používateľ nebol prihlásený a mal v košíku produkty, po prihlásení alebo registrácii sa mu tieto produkty automaticky prenesú do jeho používateľského košíka. Vďaka ich uloženiu do databázy budú dostupné aj pri prístupe z iného zariadenia.

8. Zmena množstva

V košíku je možné upravovať množstvo produktov pomocou tlačidiel `+` a `-`. Používateľ môže pridať ľubovoľné množstvo produktov, no nie je možné znížiť počet pod hodnotu jeden. Pokiaľ chce používateľ produkt z košíka úplne odstrániť, je potrebné stlačiť tlačidlo `Remove`.

Stránkovanie:

Stránkovanie je implementované priamo v šablóne `product_page.blade.php` pomocou systému Laravel Paginator. Zoznam produktov je rozdelený do strán pomocou metódy `paginate()` na strane kontroléra a následne sa v šablóne vykresľuje vlastný HTML komponent.

Stránkovací panel zahŕňa:

- Tlačidlo `Previous` (predošlá stránka),

- číselné odkazy na aktuálne, predchádzajúce a nasledujúce stránky,
- skok na prvú a poslednú stránku,
- Tlačidlo **Next** (ďalšia stránka).

Pri generovaní jednotlivých odkazov sa zachovávajú všetky aktuálne GET parametre (napr. **search**, **brand**, **sizes**, **sort**), vďaka metóde `appends(request()->query())`. To zabezpečuje, že filtrovanie a vyhľadávanie funguje konzistentne aj pri prepínaní medzi stránkami.

V rámci UX bolo zabezpečené aj:

- zvýraznenie aktuálnej stránky (**active** trieda),
- zobrazenie bodiek (...) pre skracovanie zoznamu strán pri veľkom počte strán,
- deaktivovanie tlačidiel **Previous** a **Next**, keď nie sú dostupné.

Vďaka tomu si používateľ pri prechádzaní stránok zachová nastavené filtre a vyhľadávanie, čo zaručuje plynulý a konzistentný zážitok pri prehliadaní produktov.

Prihlásenie:

Funkcionalita prihlásenia bola implementovaná pomocou balíka **Laravel Breeze**, ktorý poskytuje jednoduchú a bezpečnú autentifikáciu postavenú na session.

Na stránke `auth.blade.php` sa nachádza prihlasovací formulár, v ktorom používateľ zadáva svoj e-mail a heslo. Po odoslaní formulára prostredníctvom POST požiadavky na endpoint `/login` sa údaje overujú na strane servera.

Laravel následne:

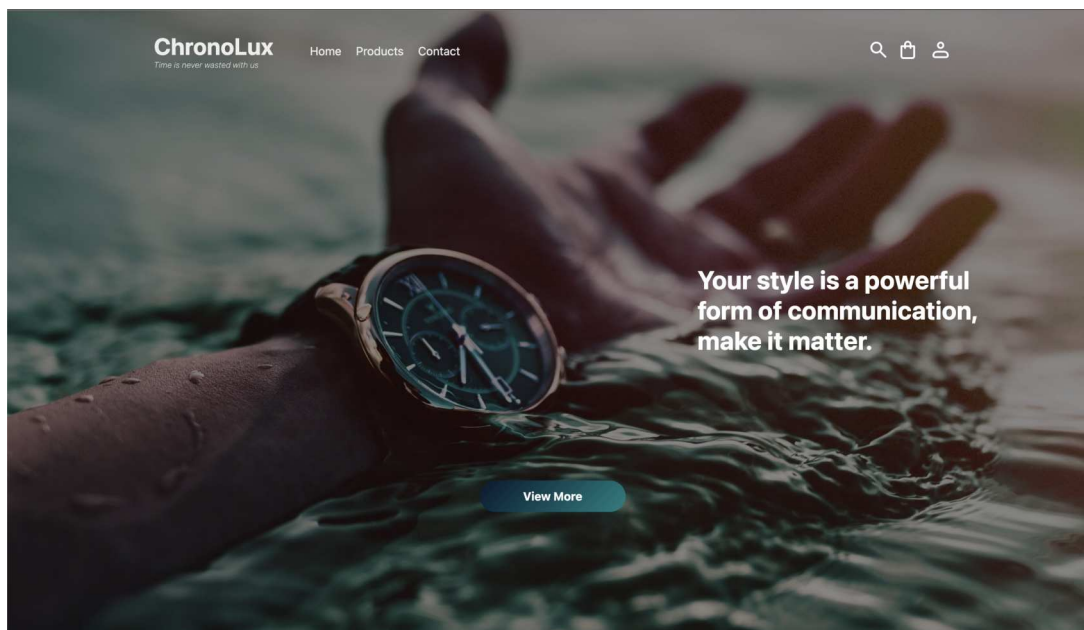
- overí zhodu e-mailu a hesla (hashovanie pomocou **bcrypt**),
- vytvorí autentifikačnú session,
- presmeruje používateľa do príslušnej časti aplikácie.

V prípade zlyhania (neplatné údaje) sa používateľovi zobrazí chybové hlásenie bez opustenia stránky. Po úspešnom prihlásení sa zákazník dostáva na stránku profilu.

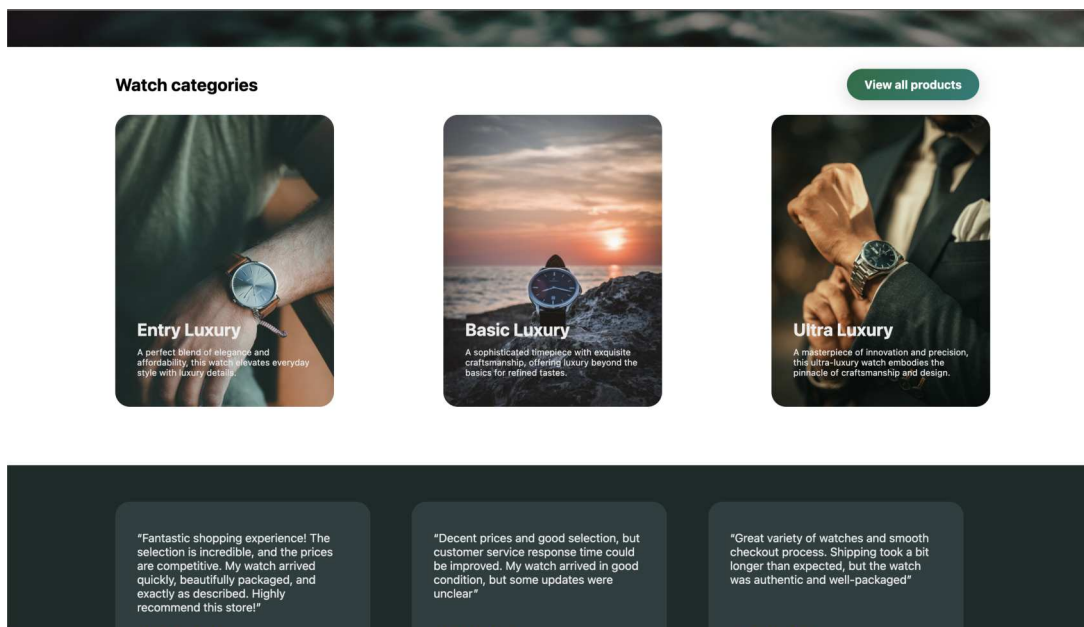
Riešenie podľa roly:

- Bežný používateľ je presmerovaný na klientské rozhranie.
- Používateľ s rolou **admin** je automaticky presmerovaný do administrátorského panela.
- Administrátor nemá prístup do používateľského profilu
- Bežný používateľ nemá prístup ku administrátorskému rozhraniu

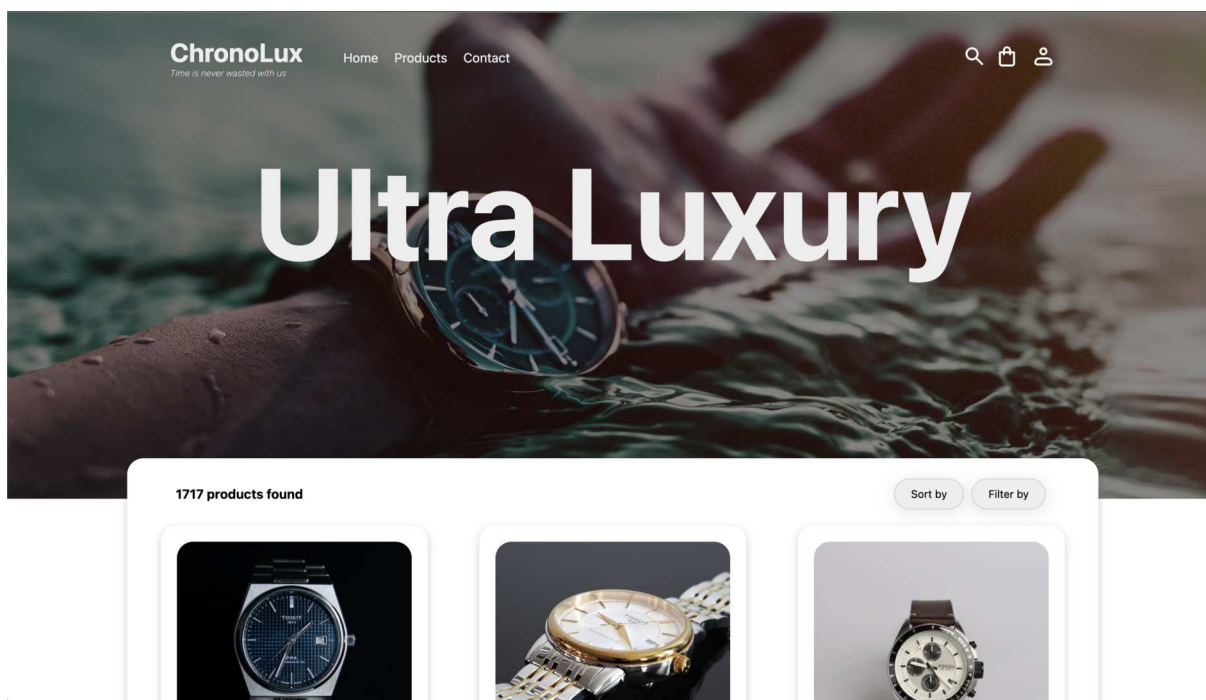
5. Snímky obrazoviek



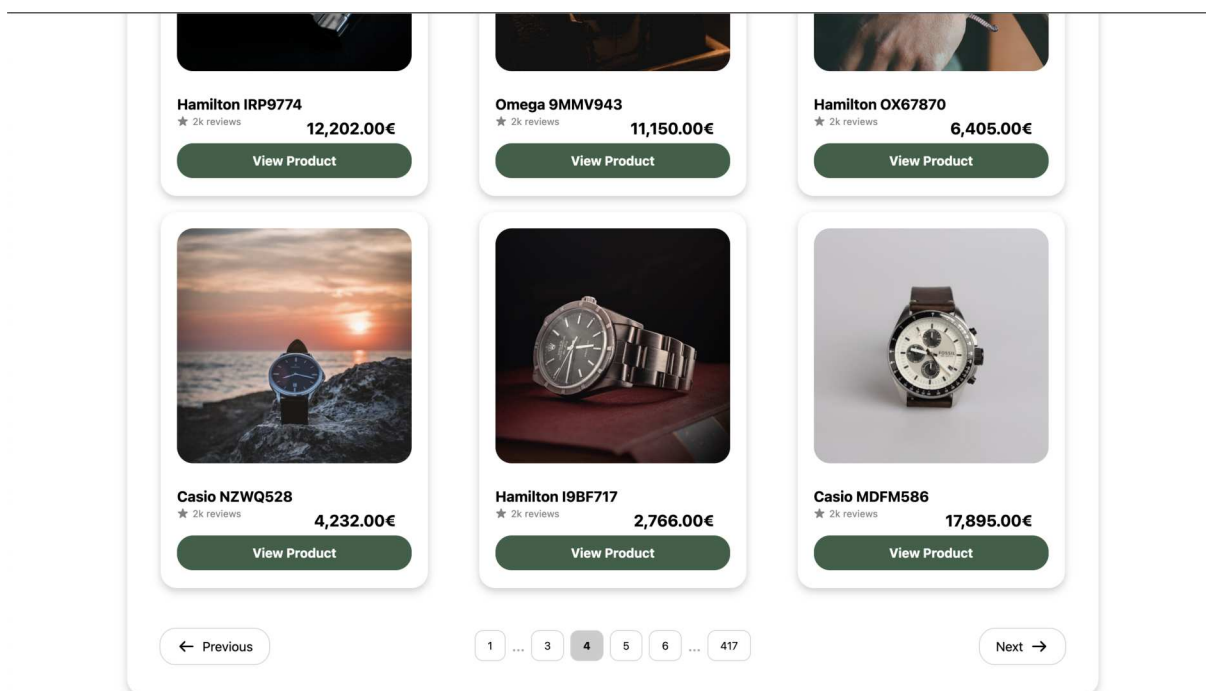
Obr. 2: Domovská obrazovka



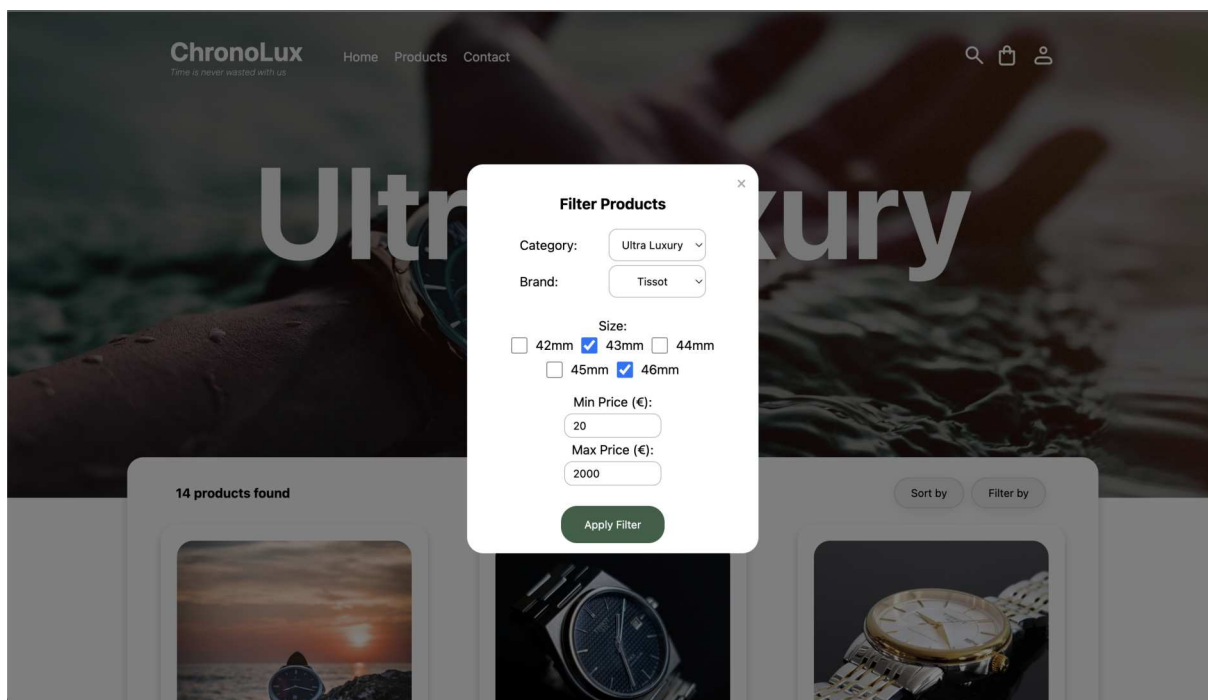
Obr. 3: Výber kategórie



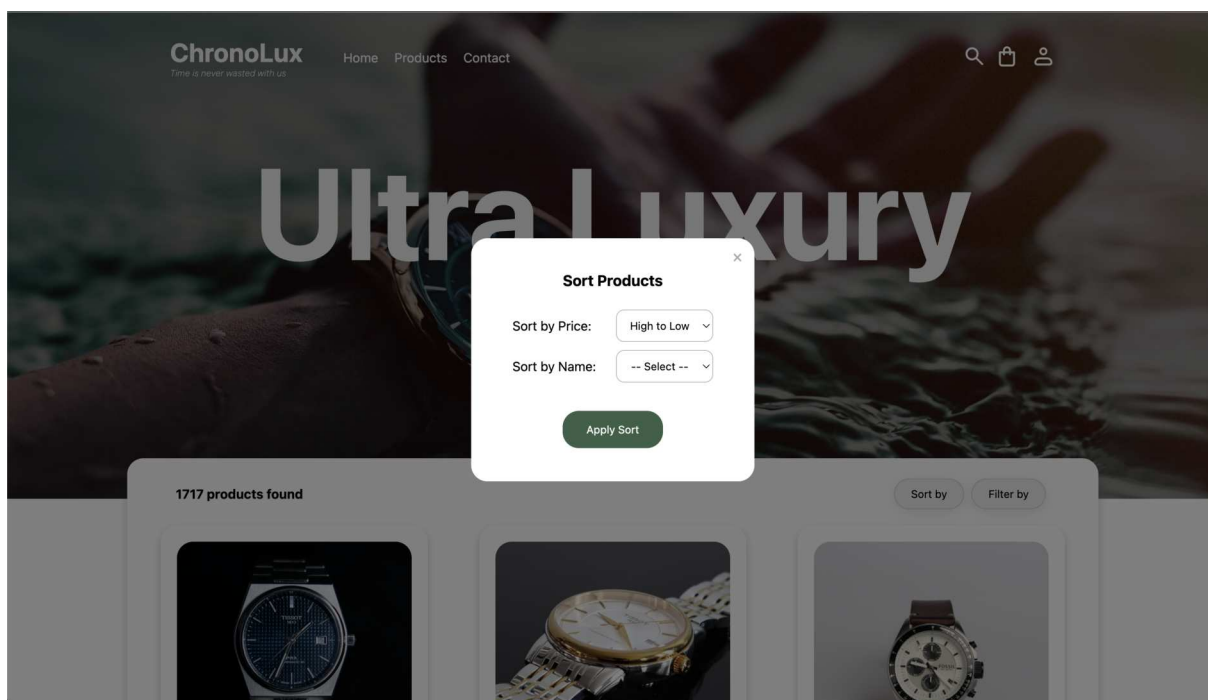
Obr. 4: Produkty v danej kategórii



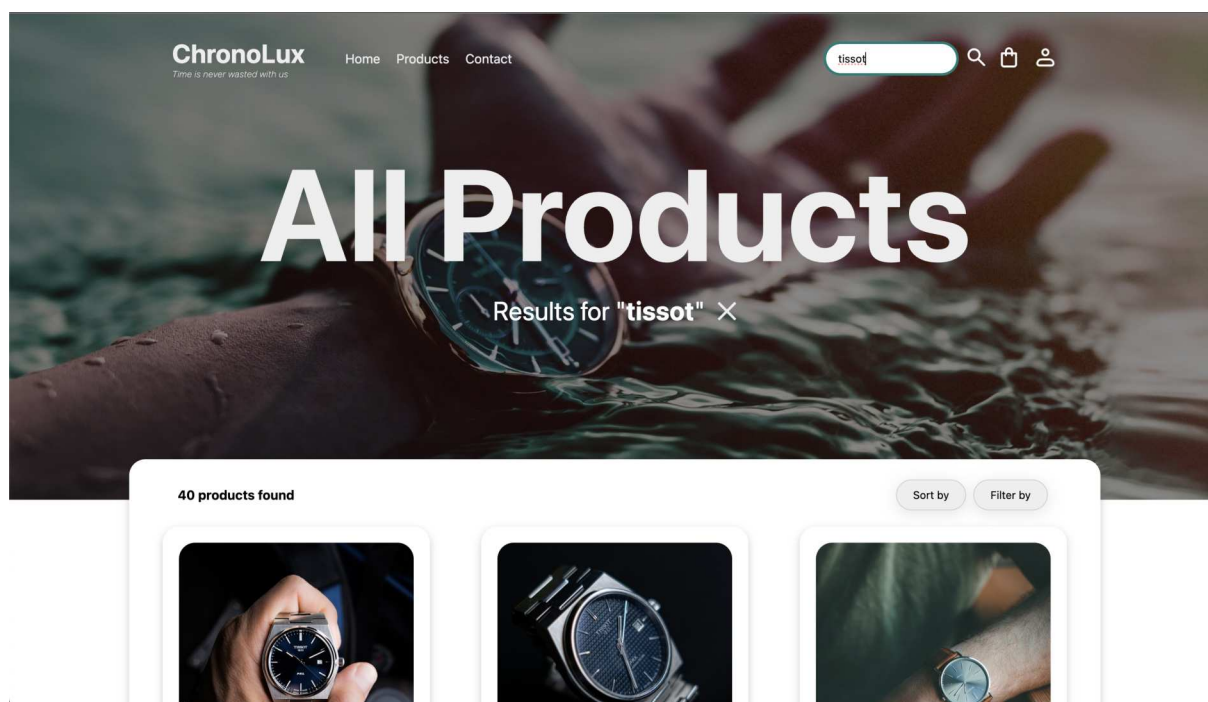
Obr. 5: Stránkovanie produktov



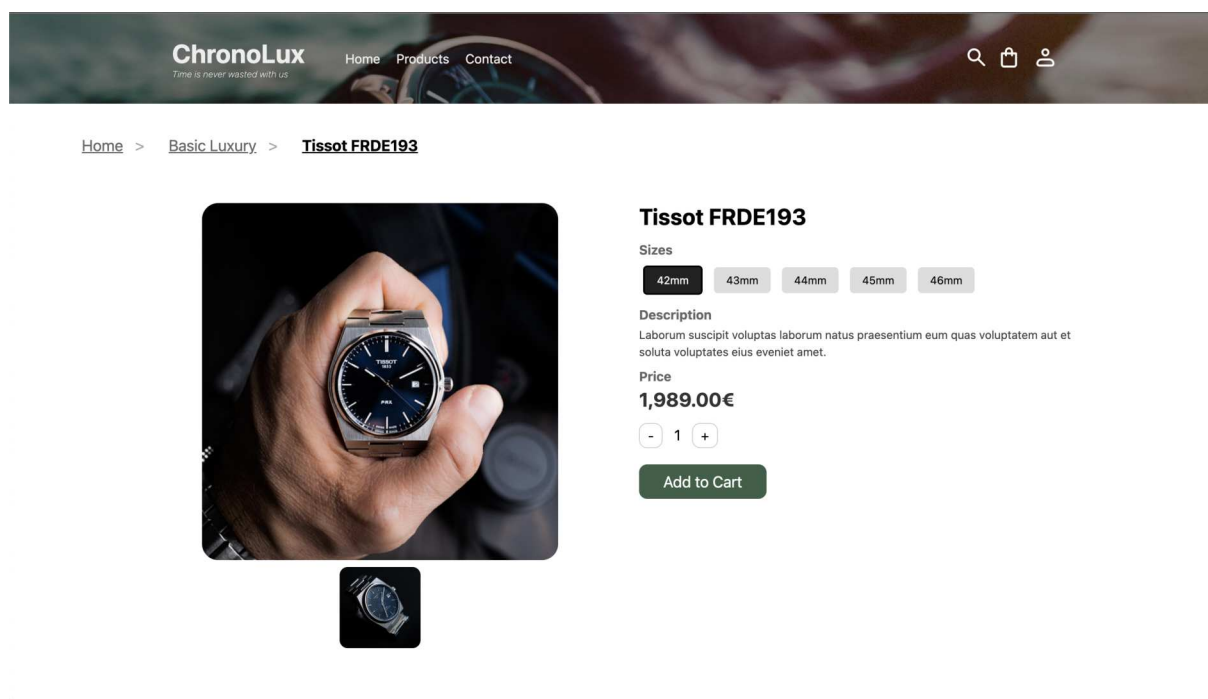
Obr. 6: Filter modal



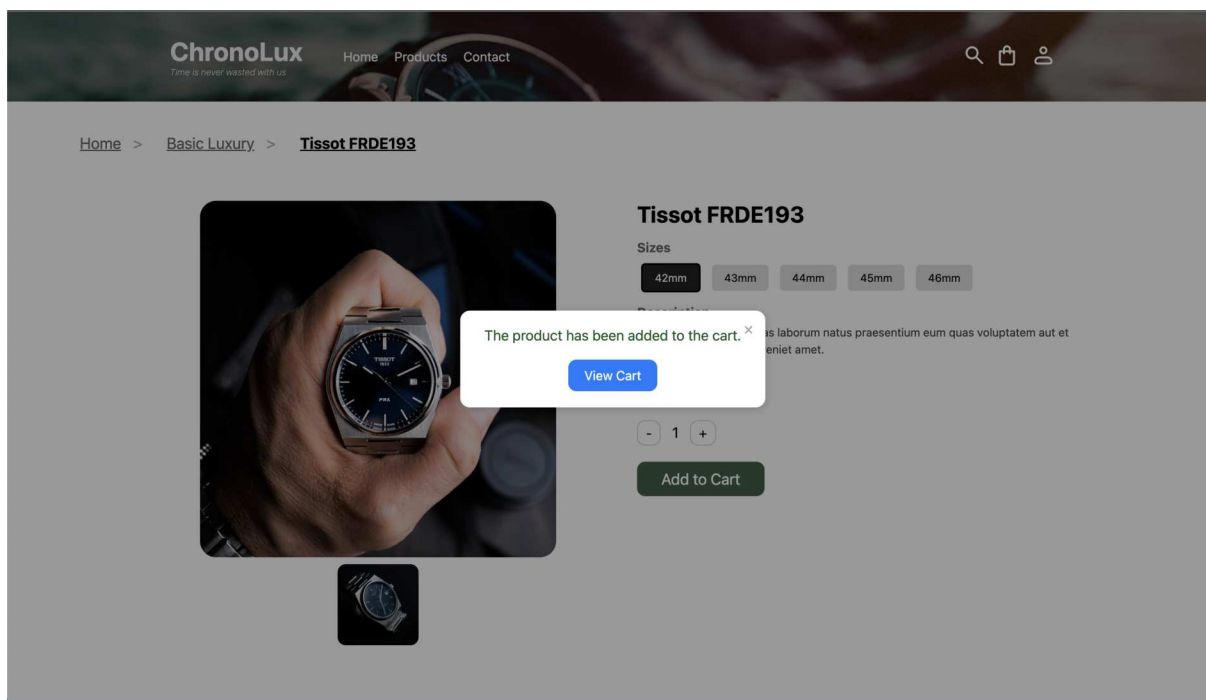
Obr. 7: Sort modal



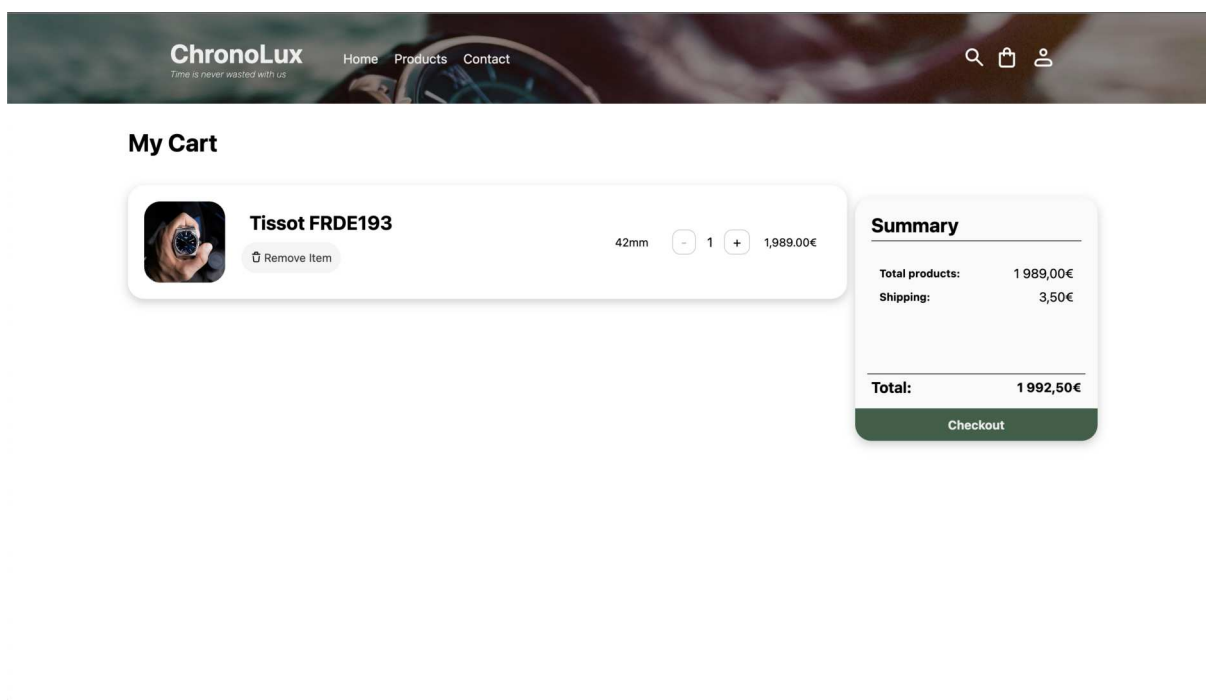
Obr. 8: Search



Obr. 9: Detail produktu



Obr. 10: Úspešné pridanie do košíka



Obr. 11: Košík

ChronoLux

Time is never wasted with us

Home

Products

Contact

My Cart

 >

Shipping

Shipping Information

☐ Use information from your account

Email

example@email.com

Name

John

Surname

Carrot

Address

Orange 123/45

Phone

0912 345 679

Postal Code

010 10

City

Carrot City

Country

Carroty

Delivery Method

☒ Packeta

☐ DPD

☐ Slovenská pošta

Summary

Total products:

1 989,00€

Shipping:

3,50€

Total:

1 992,50€

Payment

Obr. 12: Dopravné informácie

ChronoLux

Time is never wasted with us

Home

Products

Contact

My Cart

 >

Shipping

 >

Payment

Choose payment

Apple Pay

Google Pay

VISA

Mastercard

Summary

Total products:

1 989,00€

Shipping:

3,50€

Total:


1 992,50€

Pay Now

Obr. 13: Metóda platby



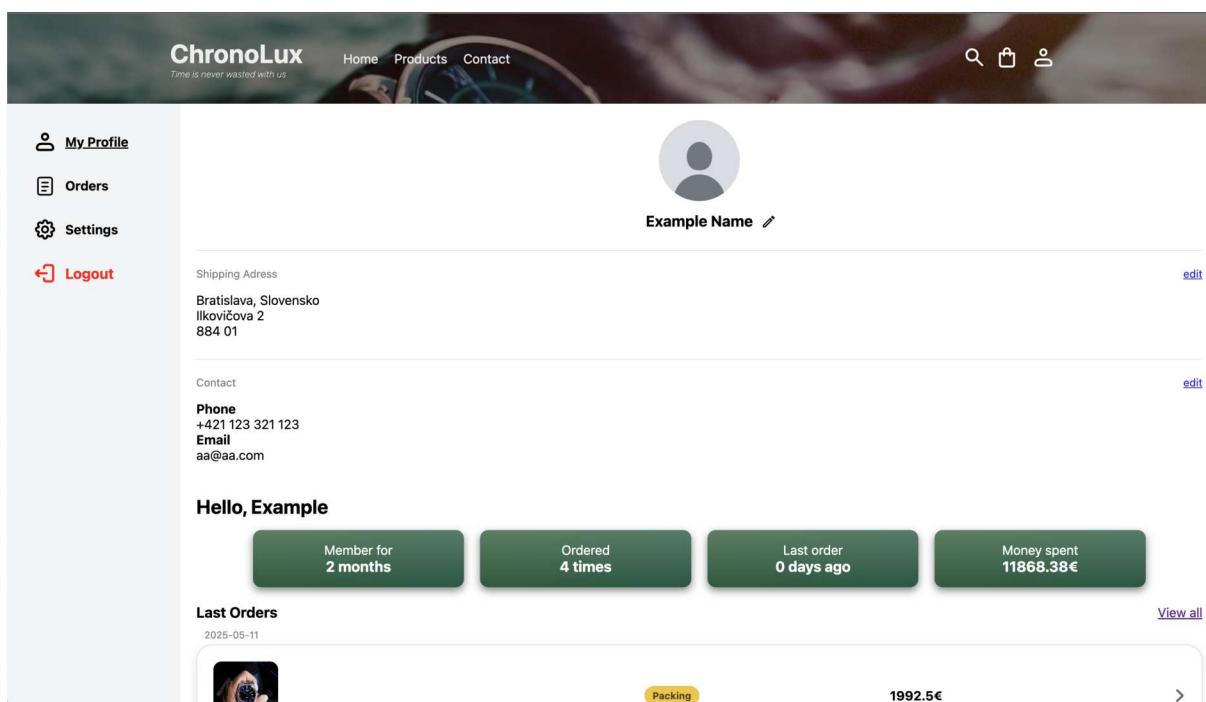
Thank you.
Your order was completed successfully.

 An email receipt including all necessary details has been sent to your email address. Please keep it for your records.

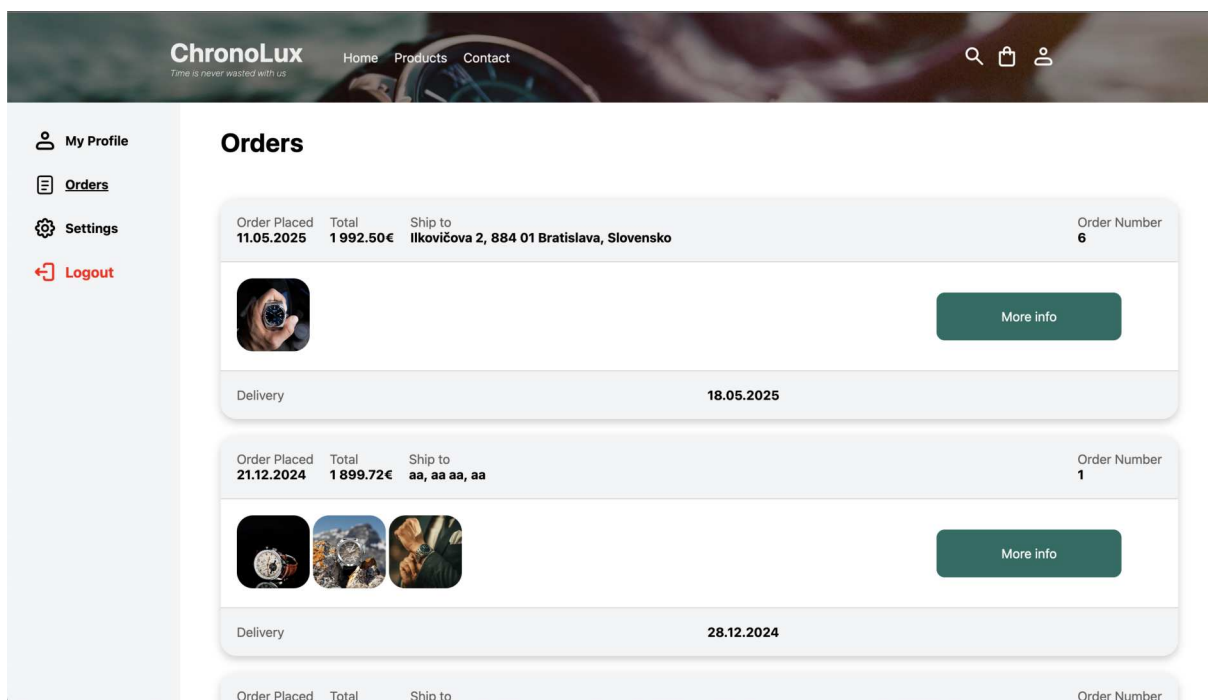
You can now check your order in your profile. Go to your profile and select 'orders', or press [here](#).



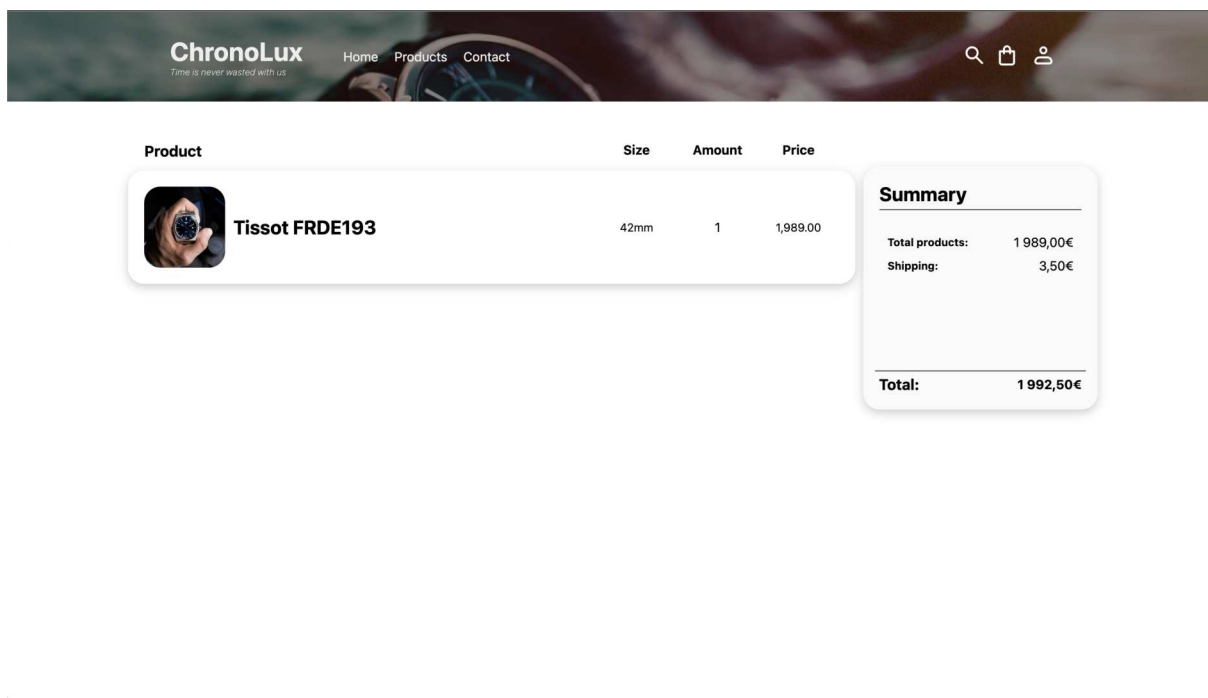
Obr. 14: Úspešné odoslanie objednávky



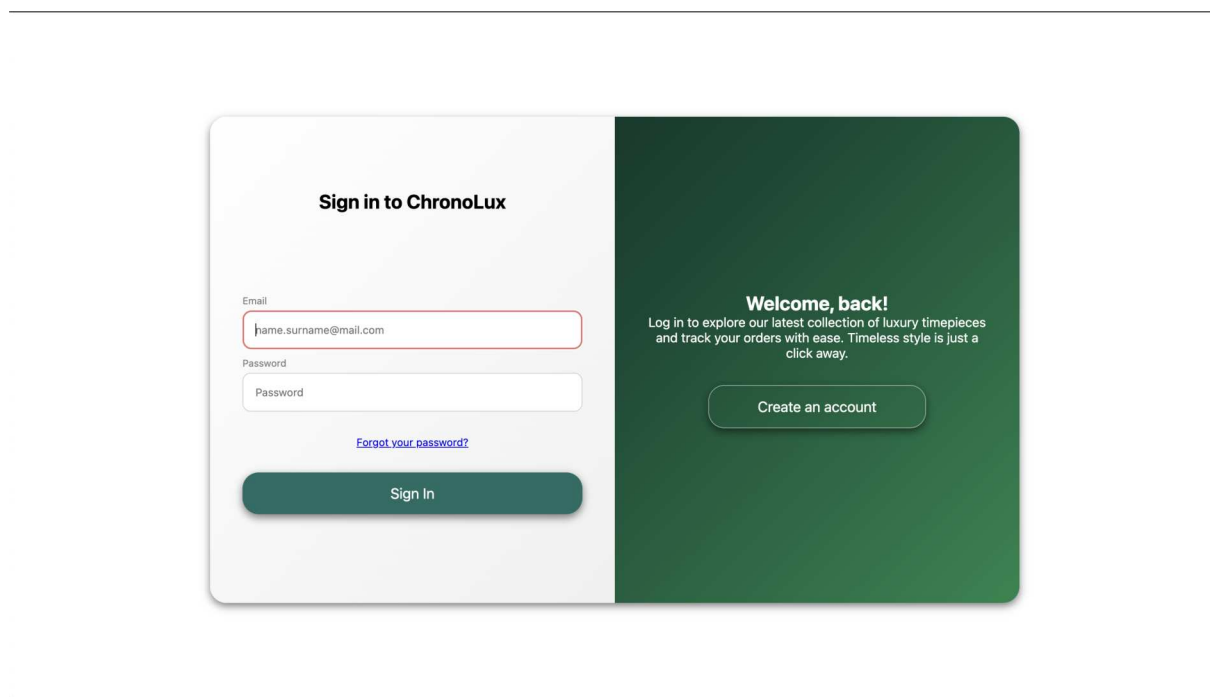
Obr. 15: Profil zákazníka



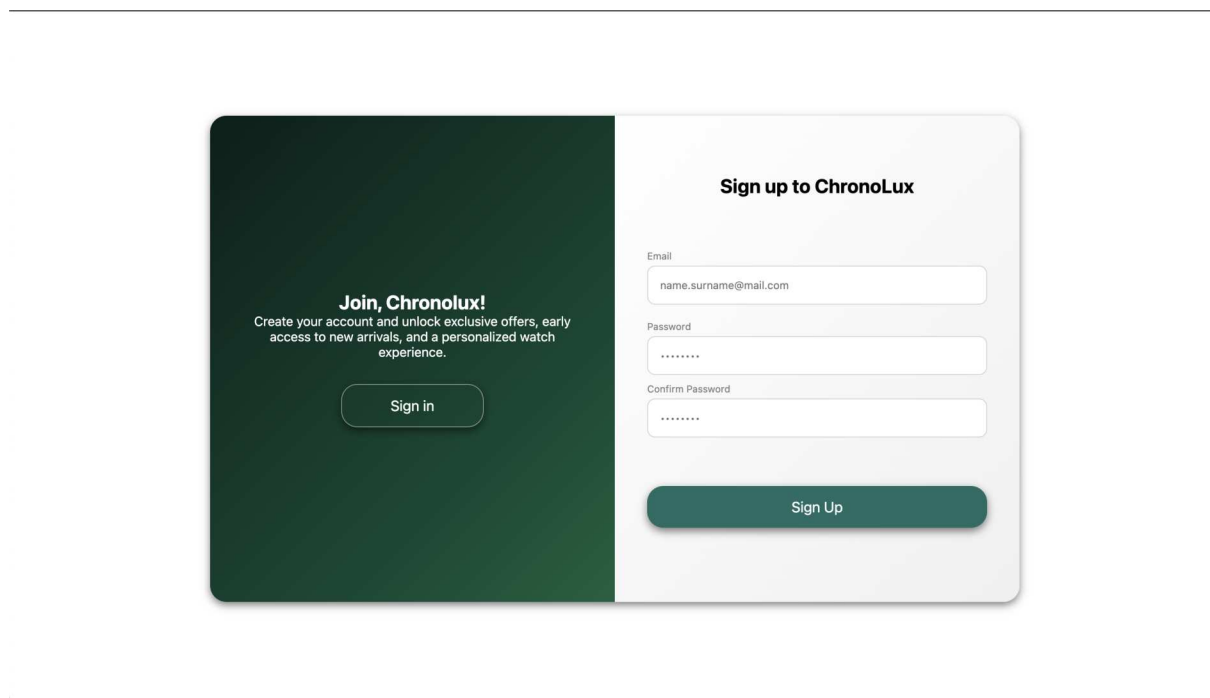
Obr. 16: Objednávky zákazníka



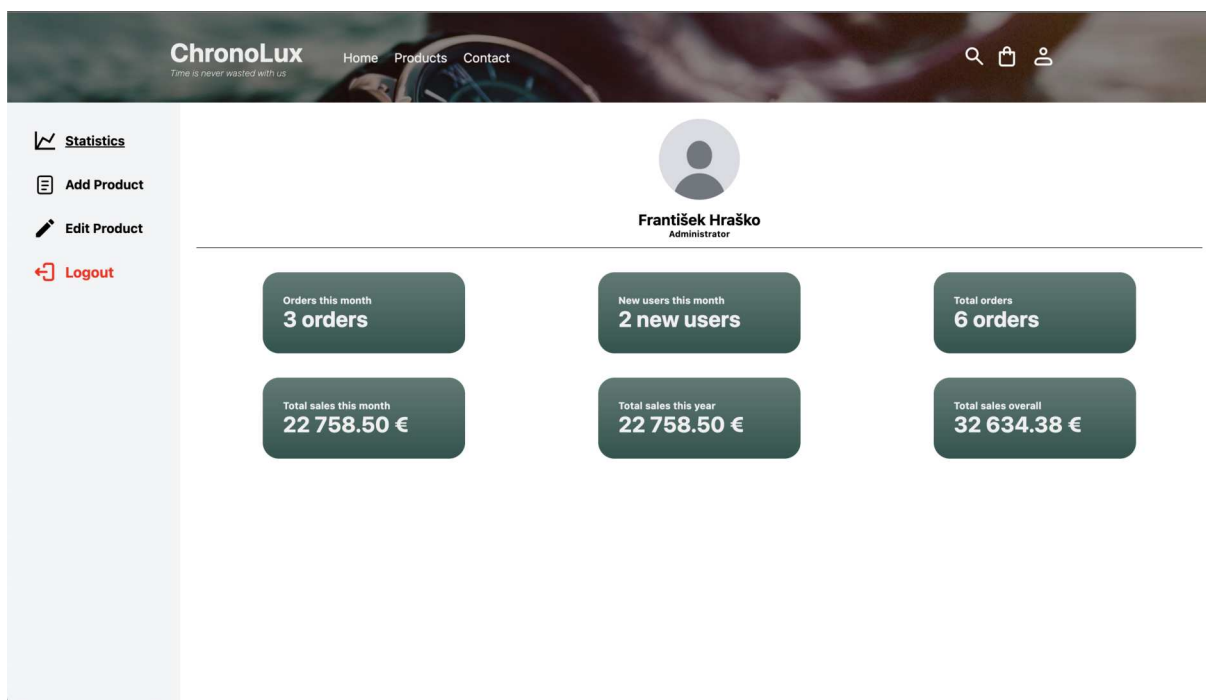
Obr. 17: Zhrnutie objednávky



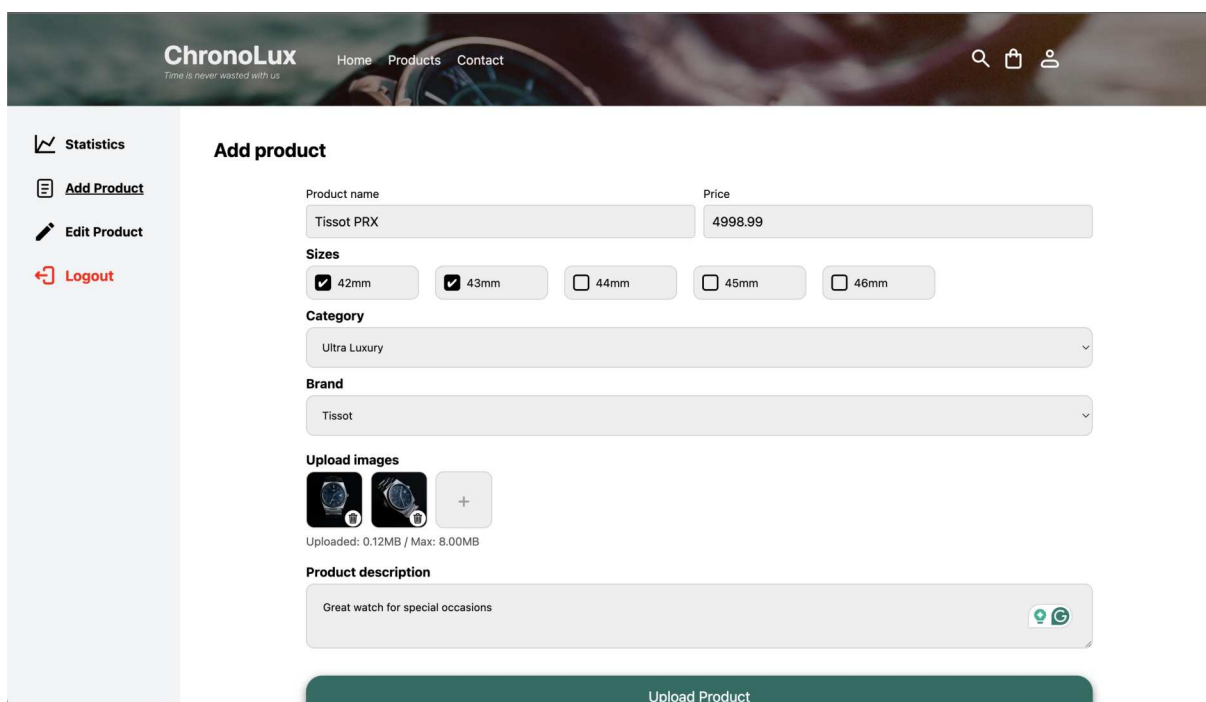
Obr. 18: Prihlásenie používateľa



Obr. 19: Registrácia používateľa



Obr. 20: Admin - Prehľad stavu eshopu



Obr. 21: Admin - Pridávanie produktov

ChronoLux

Time is never wasted with us

[Home](#)
[Products](#)
[Contact](#)

Statistics

Add Product

Edit Product

Logout

František Hraško

Administrator

Products

ID	Product Name	Category	Sizes	Price	Images	Description	Actions
1	Hamilton YRE0541	Basic Luxury	44mm	6,720.00€	1	Voluptas officia et deser...	
2	Rolex 0QRE184	Basic Luxury	45mm, 46mm, 42mm	9,067.00€	2	Earum sed ratione iure a...	
3	Citizen INB8797	Entry Luxury	45mm, 42mm, 46mm, 44mm	9,982.00€	2	Autem nemo quia vel ali...	
4	Mauron Musy KQ2E574	Entry Luxury	44mm	15,087.00€	3	Sit veritatis mollitia velit ...	
5	Breitling T2NU594	Basic Luxury	43mm, 45mm, 46mm, 42mm	7,368.00€	1	Soluta similique nemo a...	
6	Rolex SRFT163	Ultra Luxury	43mm, 46mm, 42mm, 45mm, 44mm	9,732.00€	1	Sequi velit perferendis la...	
7	Casio PTJM396	Basic Luxury	46mm, 44mm, 43mm, 42mm, 45mm	6,826.00€	2	Est neque qui ullam quis ...	
8	Fossil IESN696	Ultra Luxury	43mm, 42mm, 44mm, 46mm	17,526.00€	3	Officia omnis nesciunt m...	
9	Seiko 4WAE202	Entry Luxury	46mm, 45mm	15,683.00€	3	Omnis tempora sit ut ea ...	

Obr. 22: Admin - Prehľad všetkých produktov, s možnosťou náhľadu, editovania a vymazania produktu

ChronoLux

Time is never wasted with us

[Home](#)
[Products](#)
[Contact](#)

Statistics

Add Product

Edit Product

Logout

Edit product

Product name

Rolex BHYU942

Price

8585

Sizes

☐ 42mm
 ☐ 43mm
 ☒ 44mm
 ☒ 45mm
 ☐ 46mm

Category

Ultra Luxury

Brand

Rolex

Upload images

Uploaded: 0.13MB / Max: 8.00MB

Product description

Quis nisi fuga doloribus laborum qui vero laboriosam voluptas.

Confirm edit

Obr. 23: Admin - Editovanie produktu

Spustenie Laravel projektu

Táto sekcia popisuje kroky potrebné na spustenie Laravel projektu, ktorý sa nachádza v podadresári `/laravel/chronoLux/` po naklonovaní z GitHubu.

1. Klonovanie repozitára

```
git clone https://github.com/tomasmajercik/FIIT_WTECH2025.git
cd FIIT_WTECH2025/laravel/chronoLux/
```

2. Inštalácia PHP závislostí

```
composer install
```

Poznámka: Adresár `vendor/` je uvedený v `.gitignore`, preto je potrebné tieto závislosti nainštalovať lokálne.

3. Inštalácia frontendových závislostí

```
npm install
```

Poznámka: Inštalujú sa závislosti zo súboru `package.json`. Adresár `node_modules/` je takisto ignorovaný Gitom.

4. Konfigurácia prostredia

```
cp .env.example .env
```

5. Generovanie aplikačného kľúča

```
php artisan key:generate
```

Laravel používa tento kľúč na šifrovanie údajov (napr. session, cookies). Bez neho aplikácia nemusí fungovať.

6. Migrácia a naplnenie databázy dátami

```
php artisan migrate --seed
```

Týmto sa vytvoria tabuľky a predvolené dáta v databáze.

7. Vytvorenie symbolického odkazu na úložisko

```
php artisan storage:link
```

Týmto sa vytvorí odkaz `public/storage`, ktorý sprístupní uploadnuté súbory.

8. Spustenie aplikácie

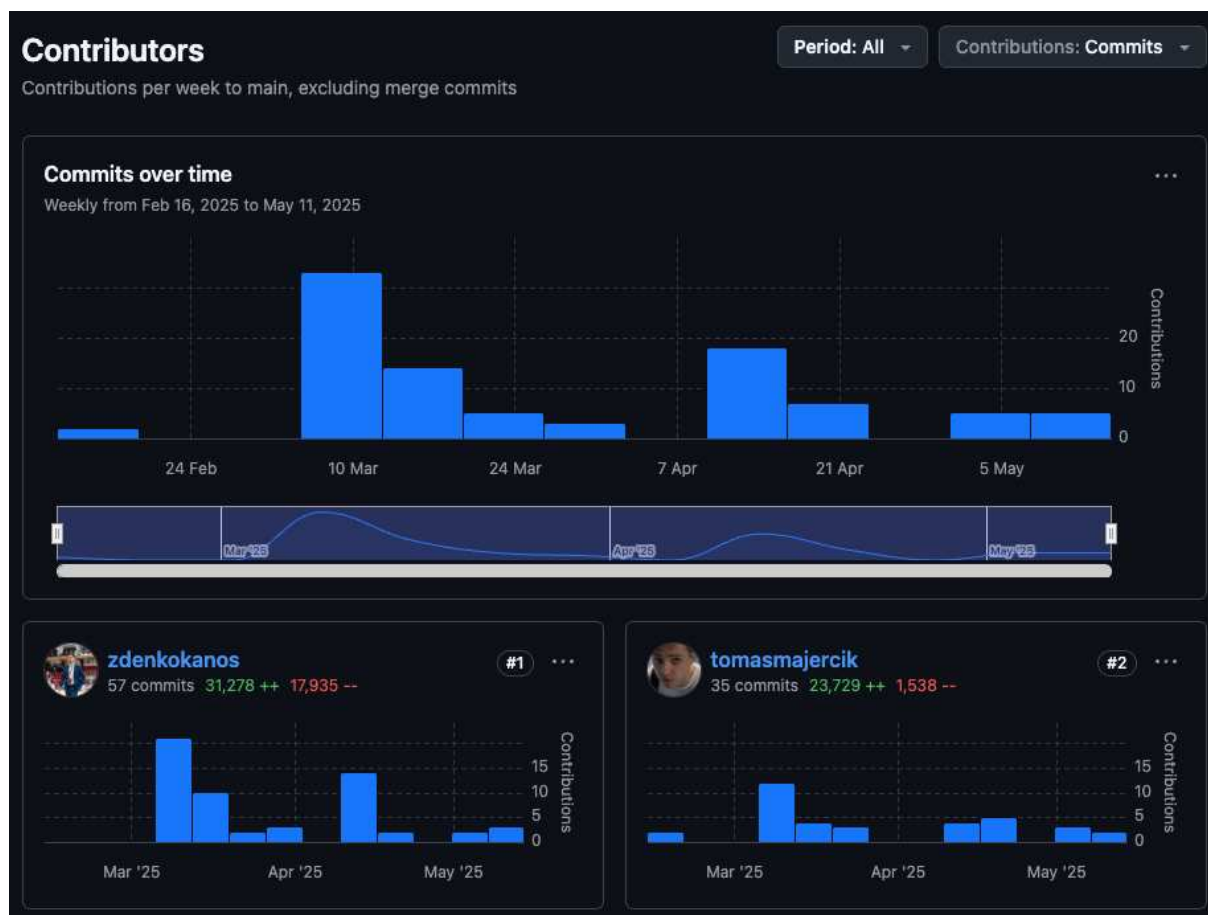
```
php artisan serve
```

Aplikácia bude dostupná na `http://127.0.0.1:8000`.

9. Prihlasovacie údaje

Admin: admin@admin.com Heslo: 1234567890

6. GitHub



Obr. 24: GitHub Štatistiky

Štatistiky nevystihujú reálny objem odvedenej práce, keďže každý z nás má odlišný spôsob práce s hlavnou vetvou – niekto projekt zakladal, niekto odstraňoval prebytočné alebo zabudnuté časti z predchádzajúcich commitov. Posledný hash commit bez dokumentácie: 8471a28

Referencie

Niektoré časti textu, boli preformulované alebo jazykovo upravené s využitím jazykového modelu spoločnosti OpenAI, ChatGPT.