

Semestrálny projekt - elektronický obchod

Rubchev Dmytro, Riabichenko Maksym

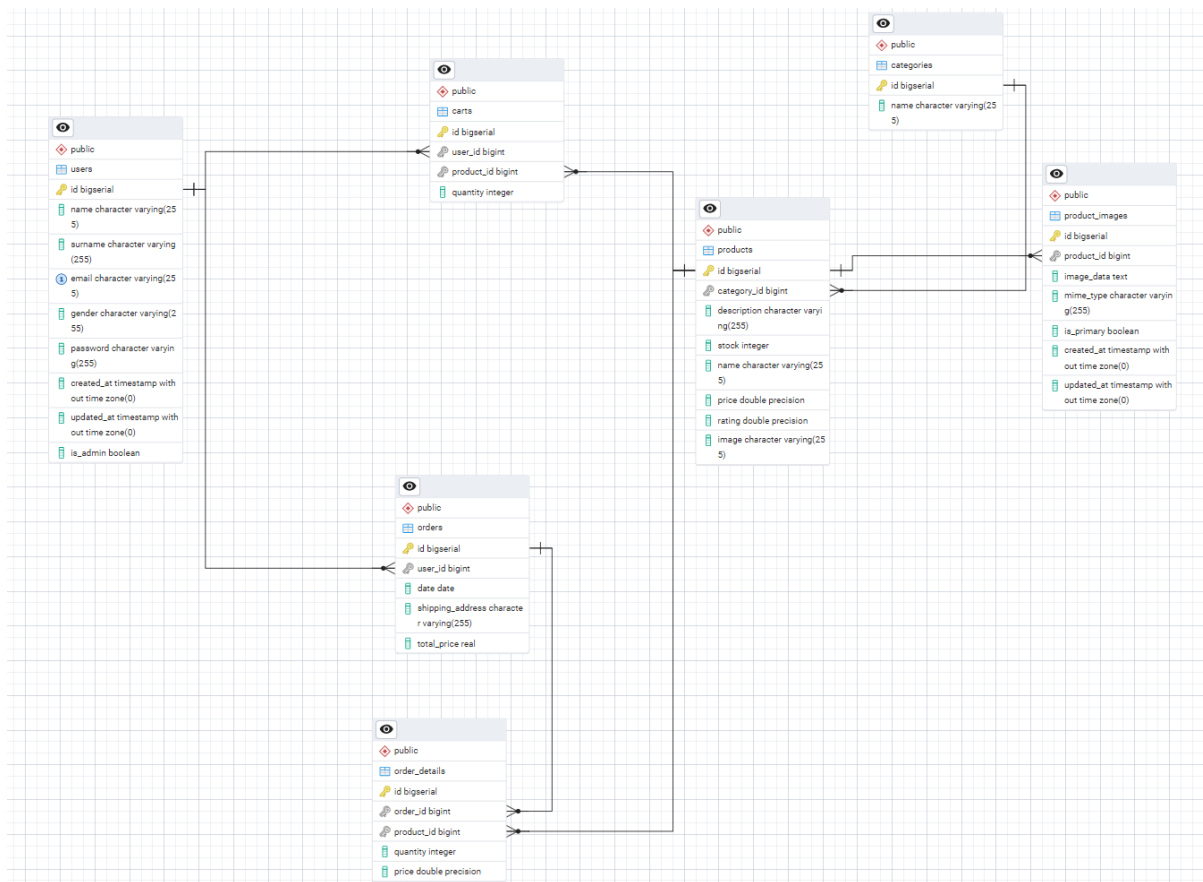
Zadaním tohto projektu bolo vytvoriť elektronický obchod. Tento projekt pozostáva z dvoch častí – klientská časť a admin časť. V klientskej časti sme mali spraviť:

- zobrazovanie zoznamu produktov s filtrovaním (podľa ceny/kategórií), preusporiadaním podľa ceny/hodnotenia, stránkovaním;
- zobrazovanie detailov produktu s funkciami pridania/odobratia z košíka;
- plnotextové vyhľadávanie nad katalógom produktov;
- košík, v ktorom by sme mohli vidieť zoznam produktov v košíku, meniť množstvo a odoberať produkty;
- zadanie dopravných a platobných údajov s validáciou;
- prenositeľnosť košíka v rámci účtu;
- registrácia, prihlásenie a odhlásenie zákazníka.

Admin časť pozostáva z:

- prihlásenia/odhlásenia admina;
- zoznamu produktov pre admina;
- možnosti pridania/odobratie produktu zo skladu;
- možnosti úpravy produktu;

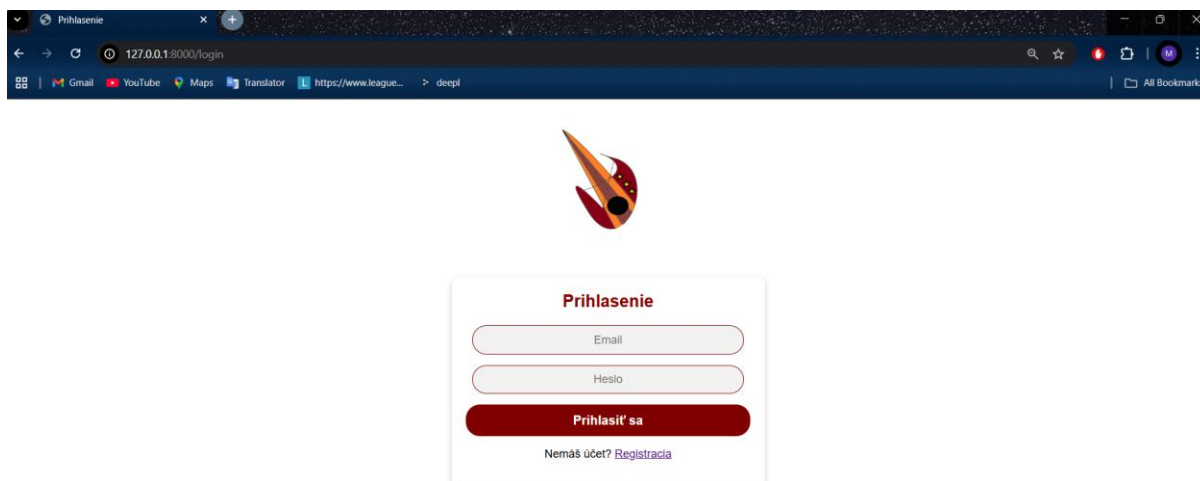
Fyzicky datovy model:



Zmeny:

Pridali sme novu tabuľku `product_images`, ktorá obsahuje obrázky pre produkty. Keďže každý produkt môže mať viacero obrázkov, ale jeden obrázok môže byť priradený najviac jednému produktu, je tam vzťah 1:N.

Prihlasenie



Prihlasenie

Email

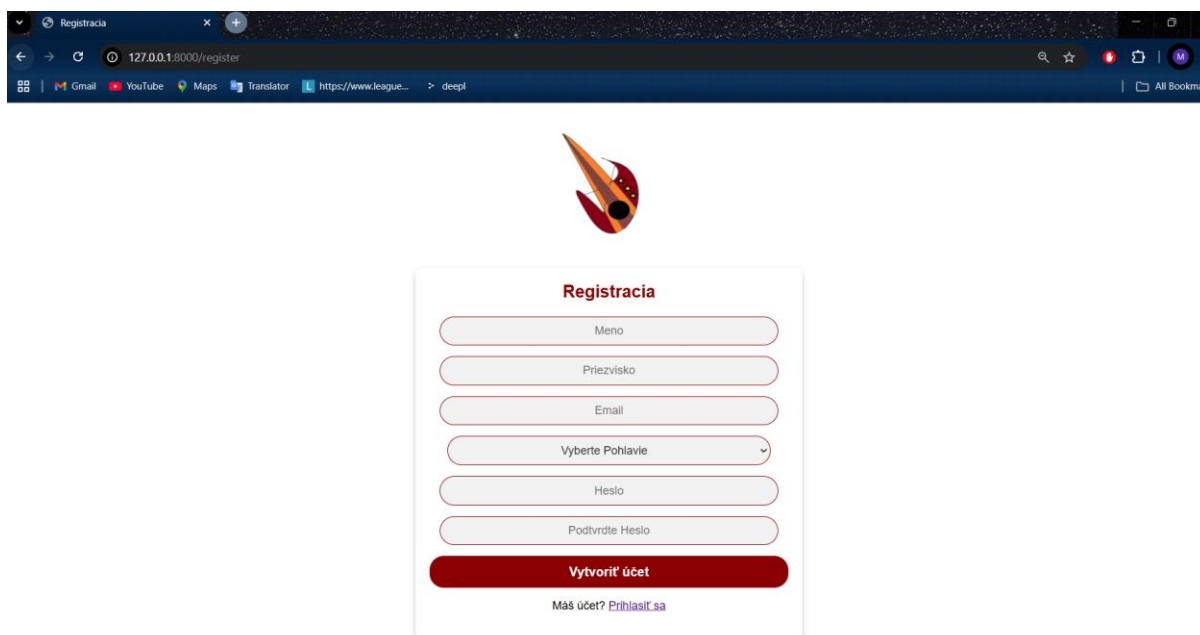
Heslo

Prihlasiť sa

Nemáš účet? [Registrácia](#)

Prihlasovací formulár v HTML šablóne odosiela e-mail a heslo na routu login s CSRF ochranou a zobrazuje validačné chyby. V AuthController metóda login validuje vstupy a používa Auth::attempt na overenie prihlasovacích údajov. Pri úspechu regeneruje session a presmeruje admina na /admin/admin_menu alebo bežného používateľa na /. Pri neúspechu vracia chybu Invalid credentials.

Registracia



Registracia

Meno

Priezvisko

Email

Vyberte Pohlavie

Heslo

Potvrďte Heslo

Vytvoriť účet

Máš účet? [Prihlasiť sa](#)

Registračný proces začína formulárom (v šablóne auth.register), ktorý odosiela meno, priezvisko, e-mail, pohlavie a heslo na routu register. V AuthController

metóda register validuje vstupy (napr. unikátny e-mail, heslo min. 8 znakov). Pri úspešnej validácii vytvorí nového používateľa s hashovaným heslom (Hash::make), prihlási ho (Auth::login) a presmeruje na /login s hláškou o úspechu. Pri chybe vracia validačné hlásky. Proces je zabezpečený CSRF ochranou a Laravel validáciou.

Katalog produktov

The screenshot displays the 'Obchod s hudobnými nástrojmi' e-shop catalog page. The page features a red header with a search bar, a category dropdown menu, and a price filter. The main content area shows a grid of musical instruments for sale, including Jazz Guitar, 7-String Guitar, Lefty Guitar, Flame Top Guitar, Double Pedal, Drumsticks, Practice Pads, and Classical Guitar. Each product card includes a star rating, price, and a 'Pridať do košíka' button. The page also includes a pagination bar at the bottom.

Product Name	Price (€)	Rating (Stars)
Jazz Guitar	499.99	5
7-String Guitar	399.99	5
Lefty Guitar	339.99	5
Flame Top Guitar	459.99	5
Double Pedal	159.99	5
Drumsticks	14.99	5
Practice Pads	49.99	5
Classical Guitar	149.99	5

Tato stránka je hlavnou stránkou nášho e-shopu, vytvorená v Laravele, rozširujúca šablónu layouts.app(ktora obsahuje header a footer). V bočnom

paneli (aside) zobrazuje zoznam kategórií produktov s odkazmi na filtrované zobrazenie podľa kategórie. Hlavná časť (section.products) obsahuje filter na zoradenie produktov (cena, hodnotenie) a cenový rozsah s posuvníkom. Produkty sú zobrazené v mriežke s obrázkom, názvom, hodnotením, cenou a tlačidlom na pridanie do košíka. Pri neprítomnosti produktov sa zobrazí hláška. Stránka podporuje pagináciu a zobrazuje notifikácie o úspechu alebo chybe (napr. po pridaní do košíka)

Filtrovanie:

Filtrovanie je implementované v metóde index kontroléra ProductController, kde sa na základe požiadavky (Request) filtrujú produkty podľa kategórie, cenového rozsahu (min/max cena s konverziou z čiarky na bodku) a zoradenia (cena vzostupne/zostupne, hodnotenie).

ProductController.php

```
13 class ProductController extends Controller
18 public function index(Request $request)
28
29 $minPrice = Product::min('price') ?? 0;
30 $maxPrice = Product::max('price') ?? 1000;
31 $products = Product::query();
32
33 if ($request->has('category')) {
34     $categoryId = $request->query('category');
35     $products->where('category_id', $categoryId);
36 }
37
38 // Обрабатываем min_price и max_price с запятой
39 $minPriceInput = $request->has('min_price') ? str_replace(' ', '', $request->min_price) : $minPrice;
40 $maxPriceInput = $request->has('max_price') ? str_replace(' ', '', $request->max_price) : $maxPrice;
41
42 $minPriceInput = (float) $minPriceInput;
43 $maxPriceInput = (float) $maxPriceInput;
44
45 $products->where('price', 'operator: >=', $minPriceInput);
46 $products->where('price', 'operator: <=', $maxPriceInput);
47
48 // Логируем значения для отладки
49 \Log::info('Price filter applied', [
50     'min_price' => $minPriceInput,
51     'max_price' => $maxPriceInput,
52 ]);
```

Vyhľadávanie:

Vyhľadávanie kombinuje backend (index a autocomplete v ProductController), kde sa vstup čistí a vyhľadáva v názvoch/popisoch produktov, a frontend (main_page.js), kde JavaScript volá /autocomplete cez fetch, zobrazuje návrhy (max. 5) a umožňuje navigáciu klávesnicou (šípky, Enter) alebo kliknutím s automatickým odoslaním formulára.

ProductController.php

```
54 if ($request->has('search')) {
55     $search = trim($request->query('search'));
56     $search = str_replace(['%', '_', '\\'], ['\\%', '\\_', '\\\\'], $search);
57     if ($search) {
58         $products->where(function($q) use ($search) {
59             $q->where('name', 'ILIKE', "%{$search}%")
60             ->orWhere('description', 'ILIKE', "%{$search}%");
61         });
62     }
63 }
```

Stránkovanie:

Stránkovanie: Stránkovanie je spracované v ProductController (metóda index) cez Laravelovu metódu paginate (12 produktov/stránku) s vlastnou šablónou (vendor.pagination.custom).

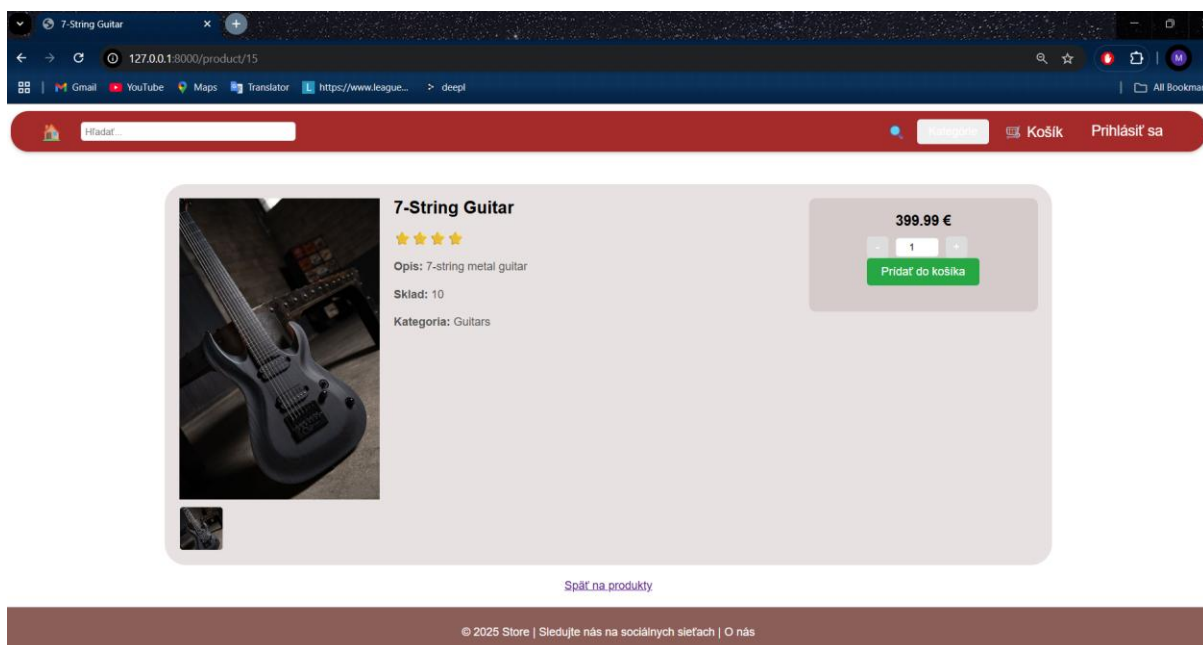
main_page.blade.php

```
82 <div class="pagination" id="pagination">
83     {{ $products->appends(request()->query()->links('vendor.pagination.custom')) }}
84 </div>
```

ProductController.php

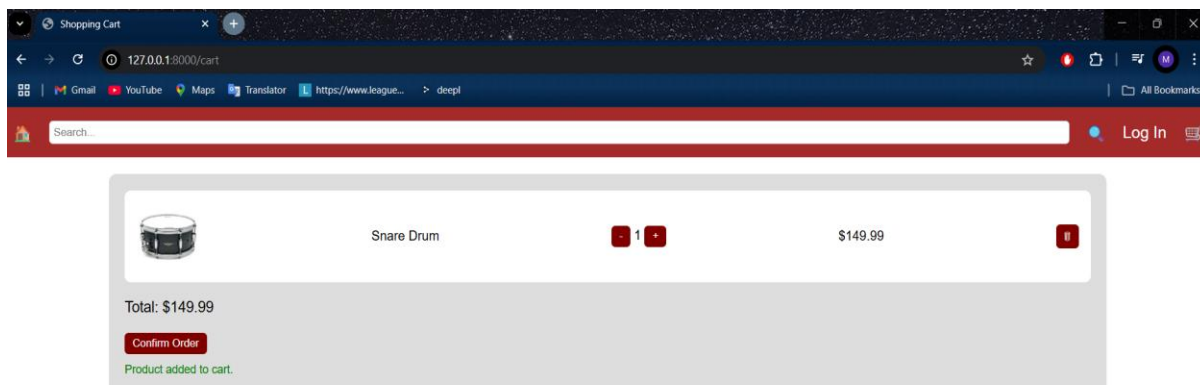
```
75 $perPage = $request->query( key: 'per_page', default: 12);
76 $products = $products->paginate($perPage);
77 $products->appends($request->query());
78
79 return view( view: 'main_page', compact( var_name: 'categories', ...var_names: 'products', 'minPrice', 'maxPrice'));
```

Stranka produktu



Táto stránka zobrazuje detail produktu. Obsahuje hlavný obrázok produktu s náhľadmi, informácie ako názov, hodnotenie, popis, sklad a kategóriu, a akcie na pridanie do košíka s ovládaním množstva. Notifikácie o úspechu alebo chybe sa zobrazujú po akciách (napr. pridanie do košíka). Odkaz „Späť na produkty“ vracia na hlavnú stránku.

Košík



Pridanie do košíka

Pridanie produktu do košíka je implementované cez formulár na stránke produktu alebo zoznamu produktov, ktorý odosiela POST požiadavku na routu `cart.add` s ID produktu a množstvom; v `CartController` sa položka ukladá do databázy pre prihláseného používateľa, čím je košík prenositeľný medzi uctami.

Odobratie z košíka

Odobratie produktu z košíka sa vykonáva cez tlačidlo v košíku, ktoré odosiela DELETE požiadavku na routu `cart.remove` s ID položky; v `CartController` sa položka odstráni z databázy a aktualizuje sa obsah košíka.

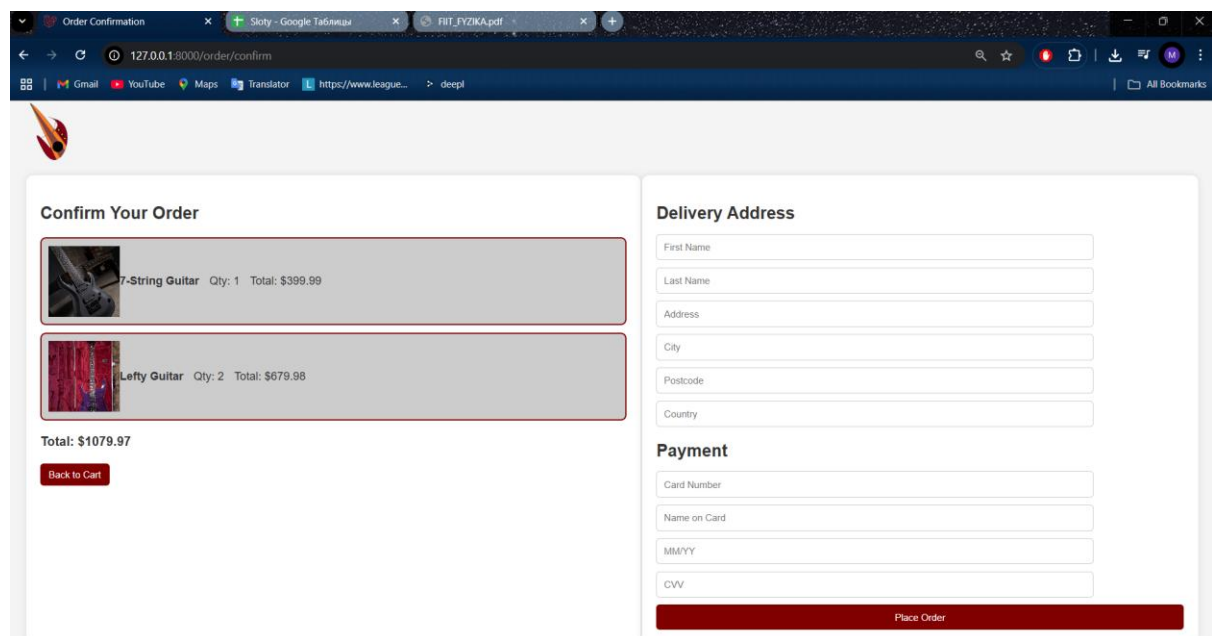
Košik bez prihlásenia

Ak používateľ nebol prihlásený, tak obsah jeho košíka sa uklada do session.

CartController.php



```
12 public function index(Request $request)
13 {
14     if (Auth::check()) {
15         $cartItems = Cart::where('user_id', Auth::id())->with('product')->get();
16     } else {
17         // Guest user - get items from session
18         $sessionCart = session()->get('cart', []);
19         $cartItems = collect($sessionCart)->map(function ($item) {
20             $product = Product::find($item['product_id']);
21             return (object)[
22                 'product' => $product,
23                 'quantity' => $item['quantity'],
24                 'id' => $item['product_id'],
25             ];
26         });
27     }
28
29     $totalPrice = $cartItems->sum(fn($item) => $item->quantity * $item->product->price);
30
31     return view('cart', compact('cartItems', 'totalPrice'));
32 }
```

Podtvrdenie objednávky



The screenshot shows a web browser window with the URL `127.0.0.1:8000/order/confirm`. The page has a dark blue header with a logo on the left. The main content area is white and contains two columns.

Confirm Your Order

- Item 1:  7-String Guitar Qty: 1 Total: \$399.99
- Item 2:  Lefty Guitar Qty: 2 Total: \$679.98
- Total: \$1079.97**
- [Back to Cart](#)

Delivery Address

First Name

Last Name

Address

City

Postcode

Country

Payment

Card Number

Name on Card

MM/YY

CVV

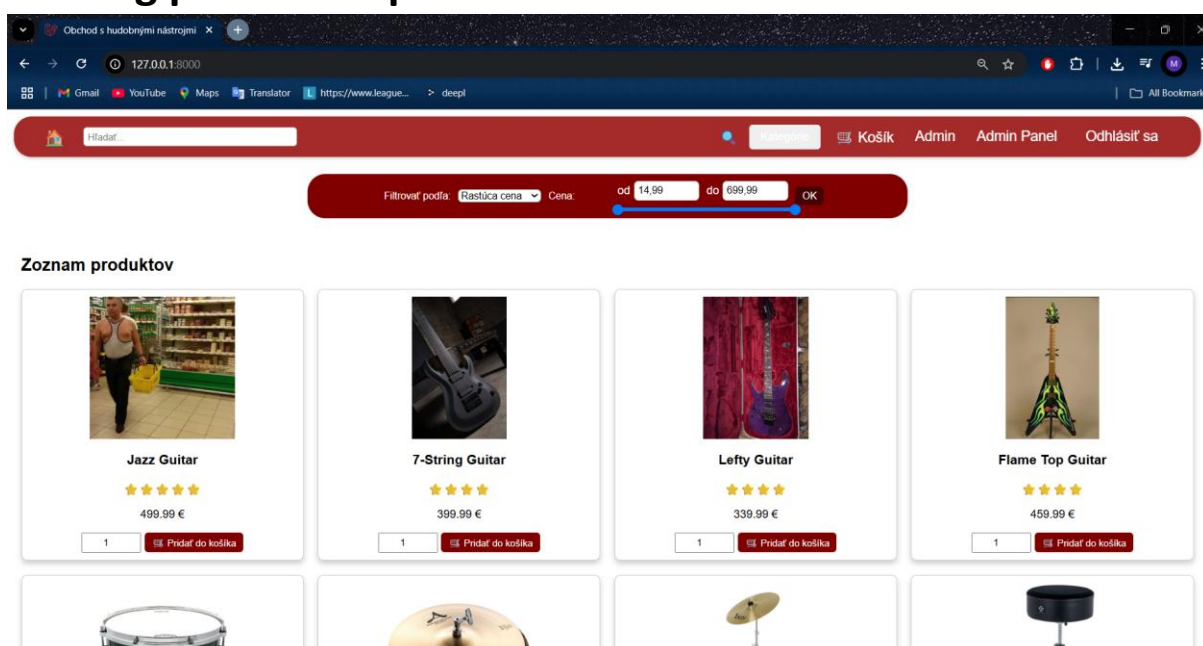
[Place Order](#)

Táto stránka slúži na potvrdenie objednávky, zobrazujúc obsah košíka a formulár na zadanie doručovacích a platobných údajov. Ak je košík prázdny, zobrazí sa hláška; inak sa vypíšu produkty s obrázkami, názvami, množstvom a celkovou cenou. Formulár zhromažďuje údaje ako meno, adresa, číslo karty a CVV, s validáciou na strane klienta aj servera (CSRF ochrana). JavaScript dynamicky aktualizuje celkovú cenu a kontroluje vyplnenie povinných polí pred odoslaním. Po odoslaní formulára sa objednávka spracuje cez routu `order.create`.

Administrátorke rozhranie

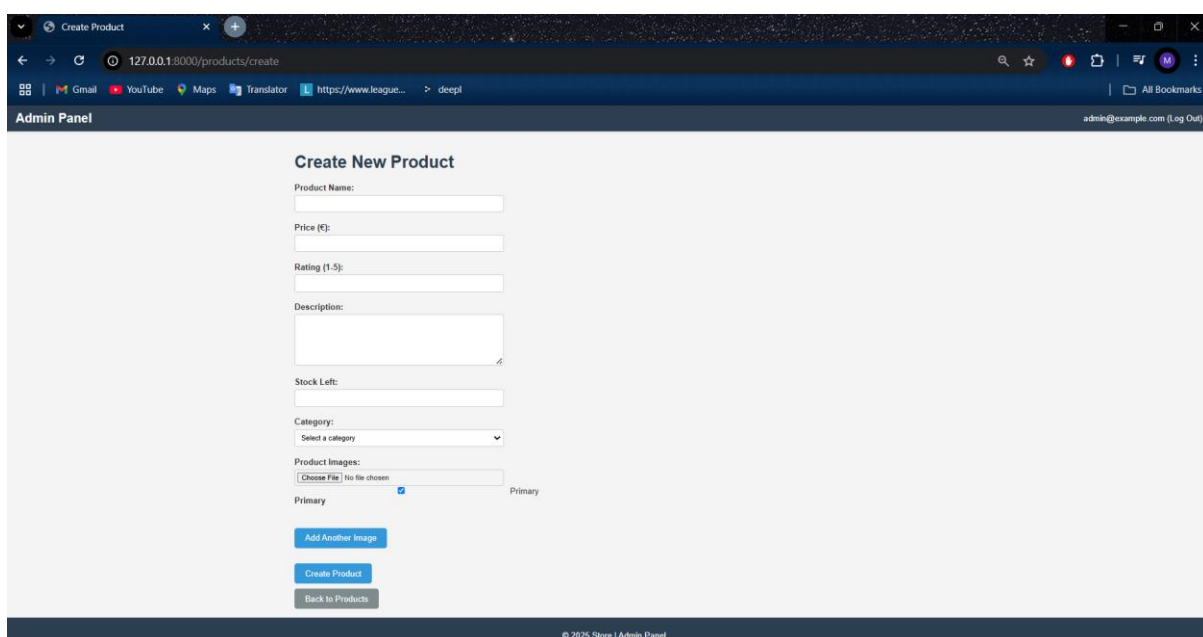


Katalog produktov pre admina

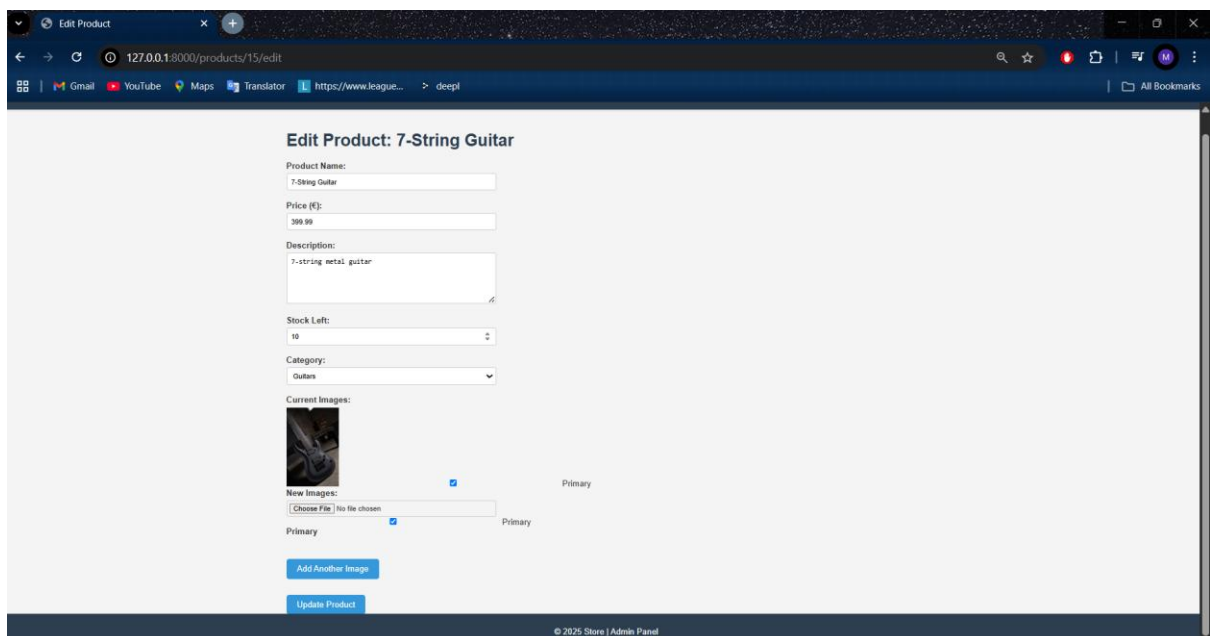
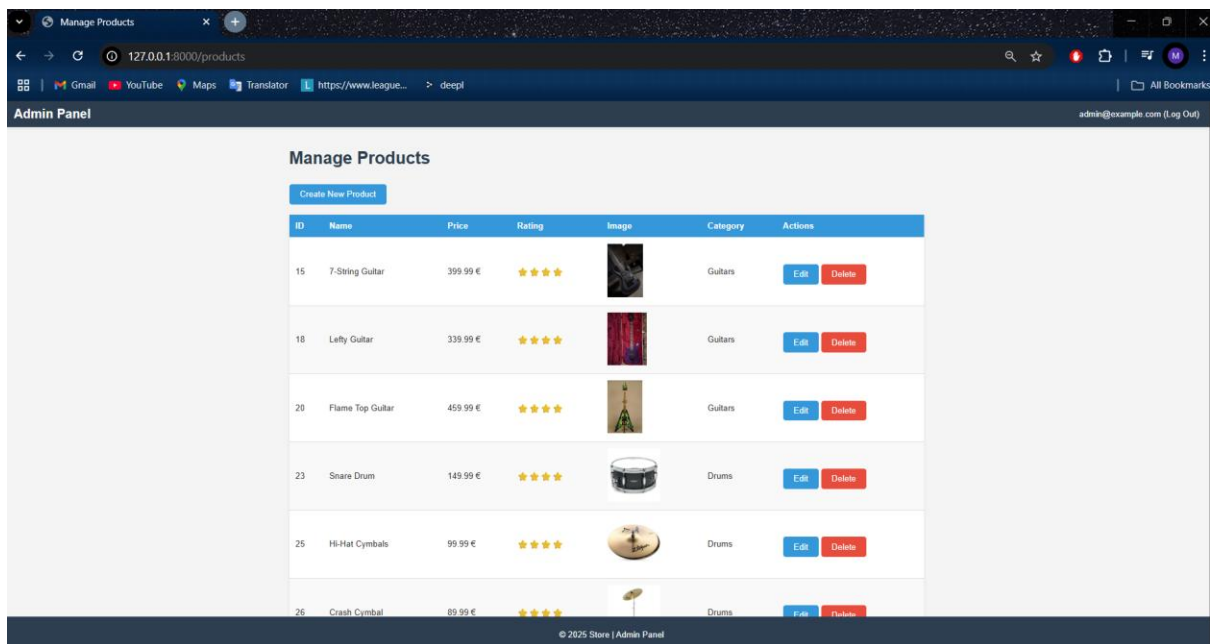


Admin takýmto spôsobom vie pozerať katalóg produktov a vie sa vrátiť na admin panel.

Pridávanie produktov



Úprava produktov



Zaver

Projekt úspešne implementuje elektronický obchod s plne funkčnou klientskou časťou, ktorá zahŕňa zoznam produktov s filtráciou, vyhľadávaním, stránkovaním, košíkom a registráciou, spolu s intuitívnym rozhraním pre správu objednávok. Admin časť poskytuje efektívne nástroje na správu produktov vrátane pridávania, úpravy a odstraňovania, s bezpečným prihlasovaním. Celkovo projekt spĺňa zadané požiadavky a ponúka robustné riešenie pre online obchod s hudobnými nástrojmi.