Fakulta informatiky a informačných technológií STU v Bratislave

**Semestrálny projekt - elektronický obchod**

**Adam Pašteka, René Revay**

2025

**Obsah**

[1. Zadanie 4](#_Toc196905486)

[2. Diagram fyzického dátového modelu 5](#_Toc196905487)

[2.1. Starý 5](#_Toc196905488)

[2.2. Nový 6](#_Toc196905489)

[2.3. Odovodnenie a zmeny 6](#_Toc196905490)

[2.3.1. Tabuľka user\_items 6](#_Toc196905491)

[3. Návrhové rozhodnutia 7](#_Toc196905492)

[3.1. Riešenie rolí používateľov 7](#_Toc196905493)

[3.1.1. Guest 7](#_Toc196905494)

[3.1.2. User 7](#_Toc196905495)

[3.1.3. Admin 7](#_Toc196905496)

[3.2. .... 7](#_Toc196905497)

[4. Opis implementácie 8](#_Toc196905498)

[4.1. Prihlásenie user a admin 8](#_Toc196905499)

[4.2. Vyhľadávanie 8](#_Toc196905500)

[4.3. Pridanie produktu do košíka 8](#_Toc196905501)

[4.4. Stránkovanie (pagination) 8](#_Toc196905502)

[4.5. Základné filtrovanie 8](#_Toc196905503)

[4.6. Zmena množstva produktu 8](#_Toc196905504)

[4.7. Pridávanie produktov do košíka pre prihláseného používateľa a hosťa 8](#_Toc196905505)

[4.8. Pridávanie nových produktov cez AdminPage 9](#_Toc196905506)

[5. Screenshots 10](#_Toc196905507)

[5.1. HomePage 10](#_Toc196905508)

[5.2. Search-filter 10](#_Toc196905509)

[5.3. Products 11](#_Toc196905510)

[5.4. Login 11](#_Toc196905511)

[5.5. Register 12](#_Toc196905512)

[5.6. Shopping-cart 12](#_Toc196905513)

[5.7. Order 13](#_Toc196905514)

[5.8. Admin-page 13](#_Toc196905515)

[5.8.1. Add new product 14](#_Toc196905516)

[5.8.2. Delete product 14](#_Toc196905517)

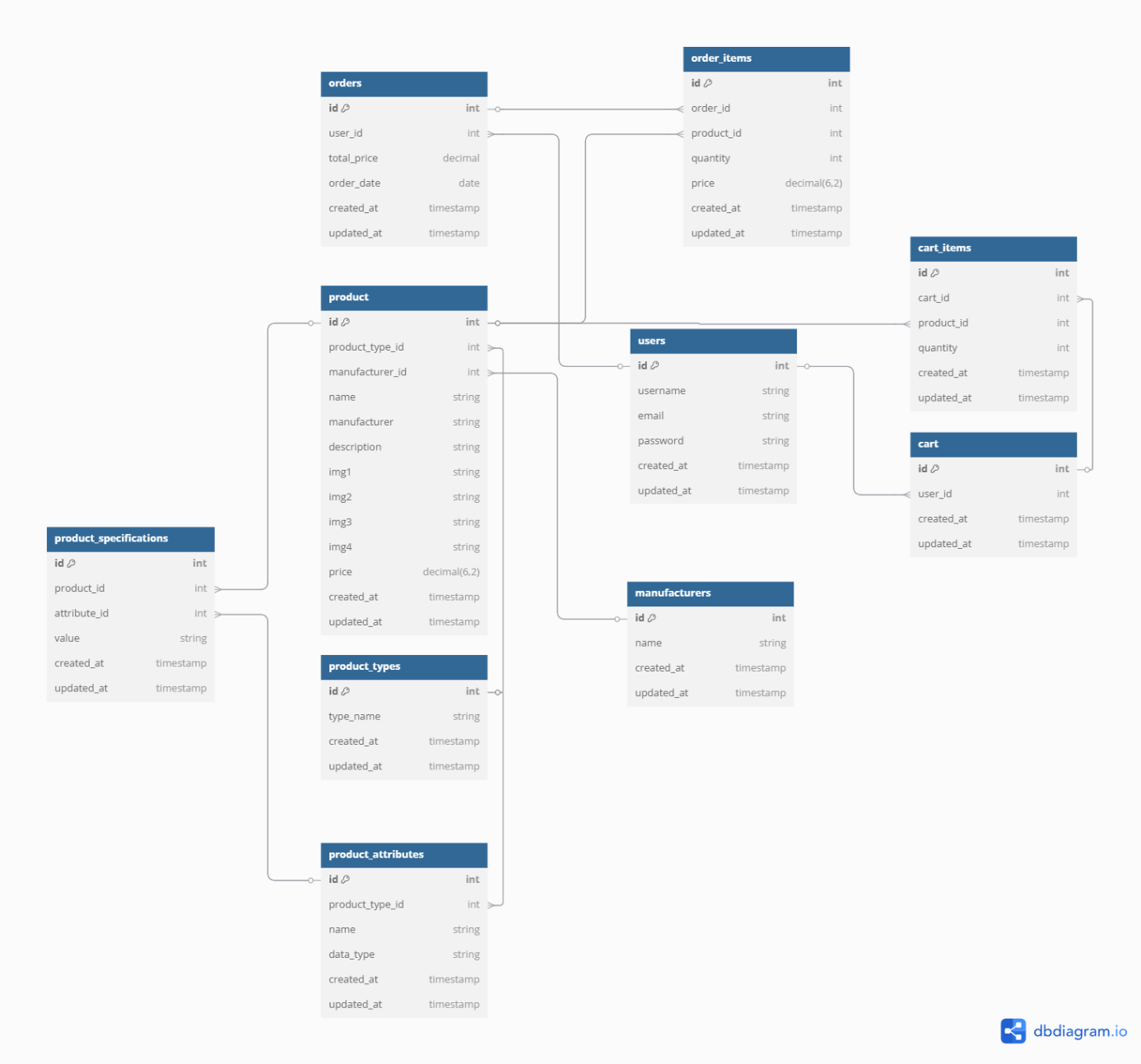
[5.8.3. Add product 15](#_Toc196905518)

[5.8.4. Edit product 16](#_Toc196905519)

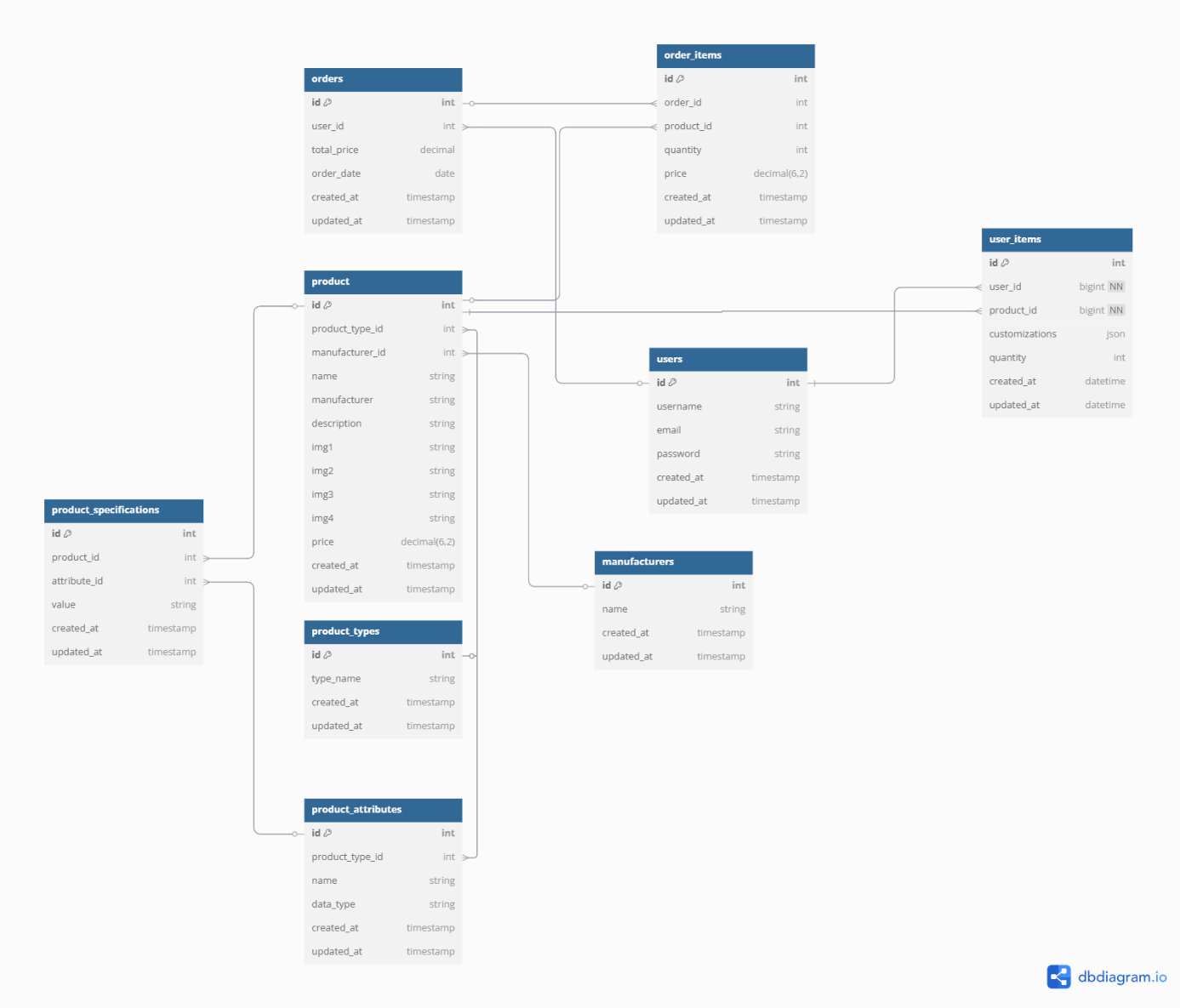
1. Zadanie

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.

1. Diagram fyzického dátového modelu
   1. Starý



* 1. Nový



* 1. Odovodnenie a zmeny

V novej verzii databázy boli tabuľky cart a cart\_items odstránené a nahradené jednou novou tabuľkou user\_items. Táto zmena prináša jednoduchší a flexibilnejší spôsob ukladania produktov priradených používateľom, napríklad pri ukladaní nakonfigurovaných produktov.

* + 1. Tabuľka user\_items
* priamo prepája používateľa s produktom (user\_id, product\_id),
* umožňuje ukladať rôzne prispôsobenia cez JSON stĺpec customizations,
* sleduje počet kusov (quantity),
* nahrádza potrebu zvlášť držať cart\_id.

1. Návrhové rozhodnutia
   1. Riešenie rolí používateľov
      1. Guest
   * Vybrané produkty sa dočasne uložia na strane klienta sessionStorage vo formáte JSON,
     1. User
   * Po prihlásení môže vkladať produkty do tabuľky user\_items
     1. Admin
   * Pridáva, upravuje a maže produkty v e-shope,
   1. ....
2. Opis implementácie
   1. Prihlásenie user a admin

Formulár POSTuje na routu /login, ktorú obsluhuje LoginController::login(). Tá overí prihlasovacie údaje cez Laravel Auth, pri úspechu presmeruje na predvolenú stránku a pri neúspechu vráti späť s chybovou flash správou.

* 1. Vyhľadávanie

GET na /search-filter/{type?} volá ProductController::searchFilter(). Získa voliteľný type param (ID kategórie) a query parameter name, zostaví Eloquent dotaz s ->where('type\_id', $type) a/nebo ->where('name', 'like', "%{$name}%"), výsledok stránkuje (napr. ->paginate(10)) a posiela do Blade view.

* 1. Pridanie produktu do košíka

V detaili produktu je tlačidlo „Pridať do košíka“, ktoré odosiela POST na /shopping-cart/add (môže byť aj inline JS). V CartController::add() sa položka vloží alebo navýši v session-ovom „cart“ poli a používateľ dostane flash správu „Produkt pridaný do košíka“.

* 1. Stránkovanie (pagination)

V oboch prípadoch (zoznam produktov aj výsledky vyhľadávania) sa používa Laravel paginator. V controllere sa volá ->paginate(n), v Blade šablóne sa potom vykreslia odkazy metodou {{ $products->links() }}.

* 1. Základné filtrovanie

Vedľa vyhľadávacieho poľa sú tlačidlá pre jednotlivé typy produktov (napr. Luky, Kuše, Praky…), ktoré odkazujú na /search-filter/{type}. Pri kliknutí sa do controllera pošle type a výsledky sa obmedzia pomocou ->where('type\_id', $type).

* 1. Zmena množstva produktu

Na stránke košíka sa pri zmene hodnoty <input type="number" name="amounts[...]"> spustí JS funkcia updateTotal(), ktorá aktualizuje cenu, a následne sendQuantityUpdate(productId, quantity) posiela AJAX POST na /shopping-cart/update-quantity. V CartController::updateQuantity() sa upraví položka v session-ovom „cart“ poli a vráti sa JSON s úspechom.

* 1. Pridávanie produktov do košíka pre prihláseného používateľa a hosťa

Keď používateľ, ktorý nie je prihlásený (guest), klikne na tlačidlo „Pridať do košíka“, aplikácia odošle POST požiadavku na /shopping-cart/add. Backend v CartController::add() rozpozná, že používateľ nie je autentifikovaný, a pridanú položku (ID produktu a jeho množstvo) uloží do session, konkrétne do poľa session('cart'), kde sa každé ID produktu mapuje na aktuálne množstvo. Po úspešnom pridaní sa session košík aktualizuje a používateľovi sa zobrazí flash správa „Produkt pridaný do košíka“. Tento hosťovský košík existuje iba počas trvania session prehliadača; jej vyprázdnením alebo zatvorením sa košík odstráni.

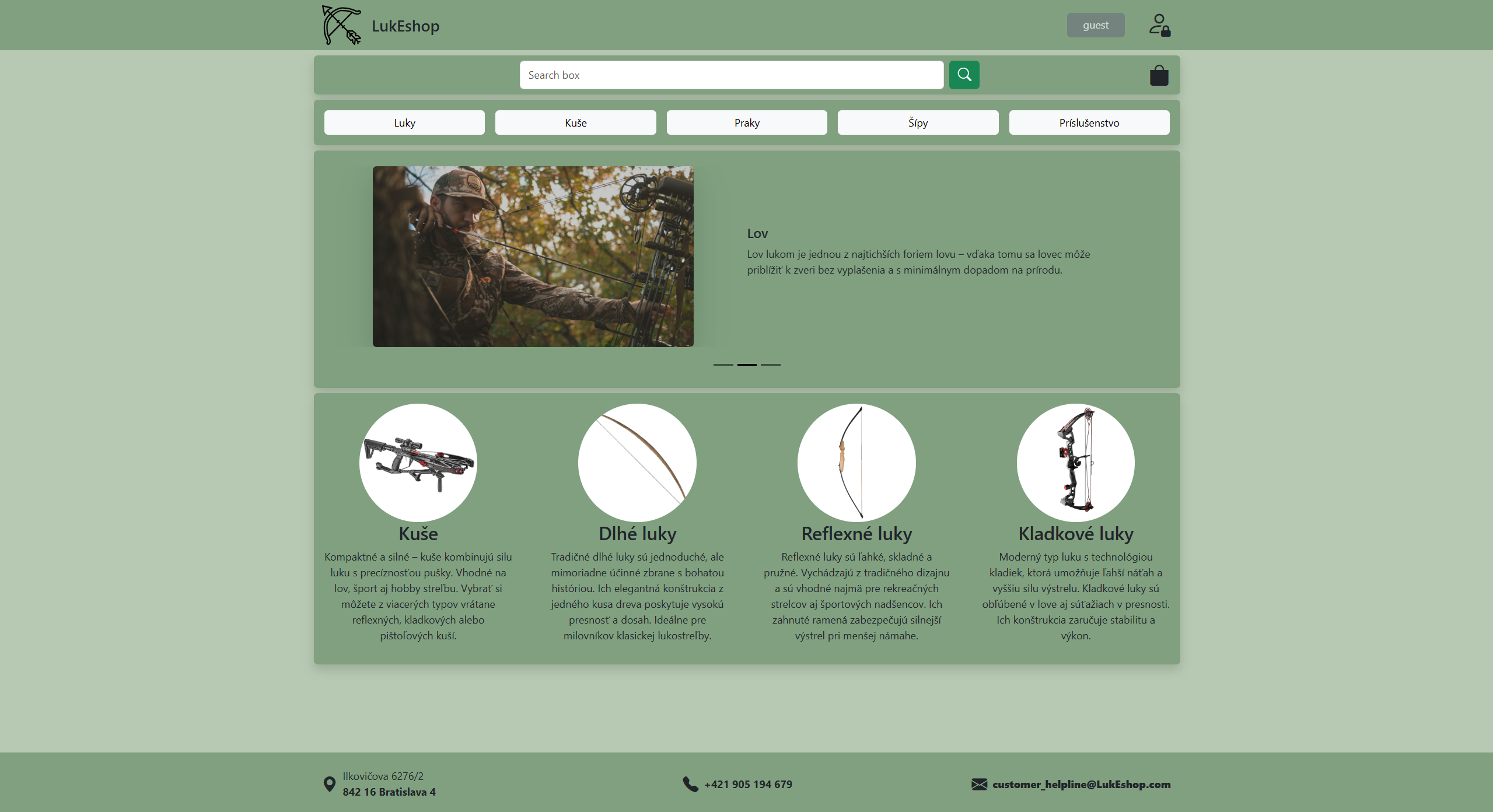
V prípade prihláseného používateľa prebieha pridanie veľmi podobne – klik na „Pridať do košíka“ spustí rovnakú POST požiadavku na /shopping-cart/add. V CartController::add() sa najskôr overí stav autentifikácie pomocou Auth::check(). Ak je používateľ prihlásený, položka sa namiesto session uloží do databázovej tabuľky cart\_items volaním metódy Cart::updateOrCreate(), ktorá buď vytvorí nový záznam pre daného používateľa a produkt, alebo zvýši množstvo existujúcej položky. Navyše, vždy keď sa používateľ prihlási a v session je ešte hosťovský košík, aplikácia v rámci CartService::mergeGuestCart() iteruje cez všetky položky zo session a presunie ich do databázy pre tohto používateľa, pričom množstvá sa pri zhodných ID produktov spočítajú. Po úspešnom zlúčení sa hosťovský košík v session vyprázdni. Vďaka tomu má prihlásený používateľ k dispozícii svoj košík uložený trvalo v databáze, ktorý sa zároveň plynulo doplní o produkty, ktoré si pred prihlásením vybral ako hosť.

* 1. Pridávanie nových produktov cez AdminPage

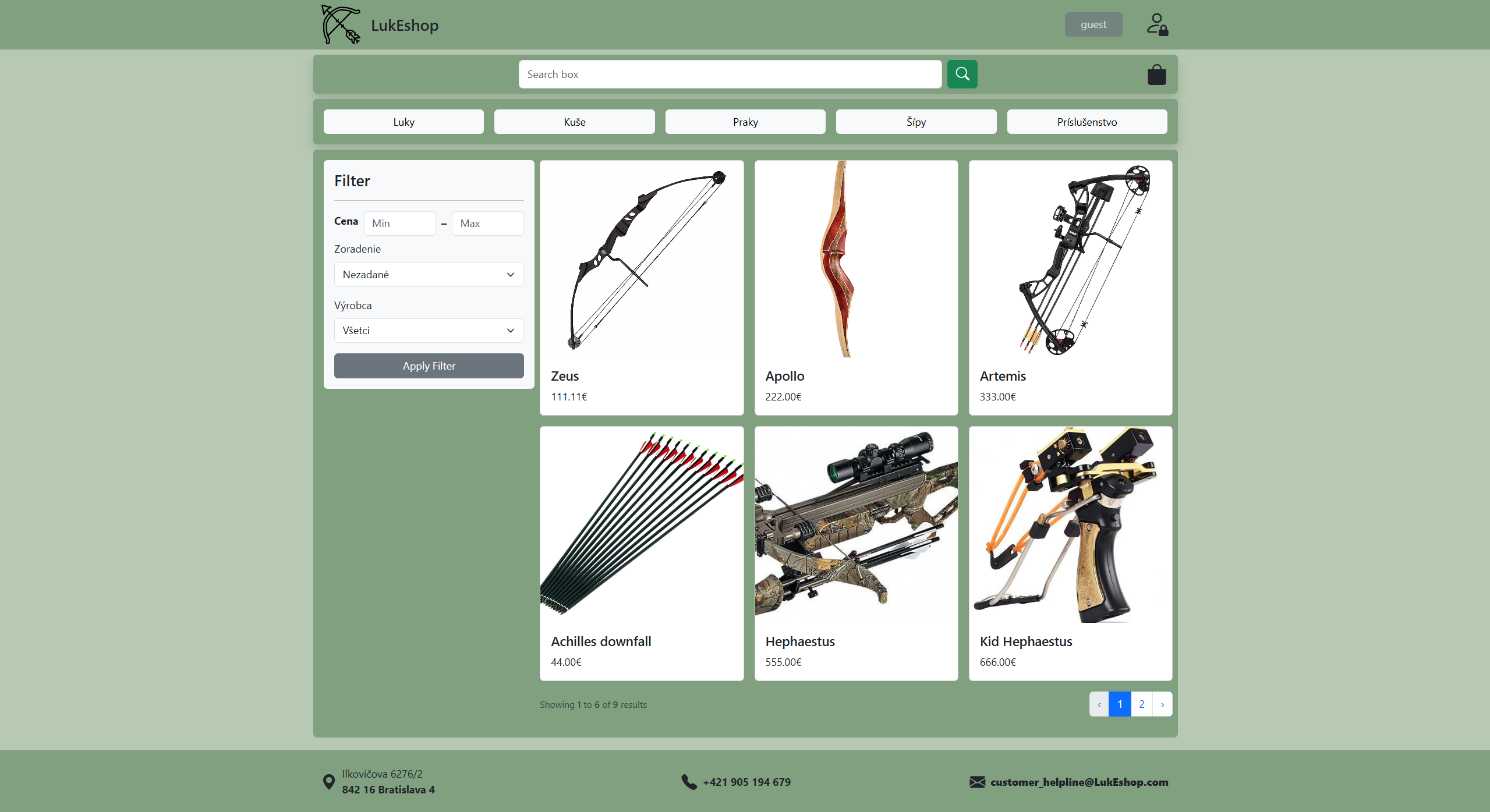
Keď je administrátor prihlásený, zobrazí sa mu špeciálne administrátorské rozhranie, v ktorom vidí už existujúce produkty a môže zvoliť možnosť „Add Item“. Po kliknutí na „Add Item“ sa najskôr zobrazí výber preddefinovaných typov – luk, kuša, prak a pod. – kde administrátor vyberie kategóriu nového tovaru, a až potom klikne na „Add Product“. Aplikácia mu následne zobrazí formulár, v ktorom vyplní názov, popis, cenu, dostupné množstvo a vybranú kategóriu produktu. Po odoslaní tento formulár pošle POST požiadavku na routu /admin/products/store, ktorú v ProductAdminController::store() spracuje metóda zodpovedná za validáciu vstupov (napríklad pomocou Laravel Request s pravidlami required, numeric a pod.). Ak validácia prebehne úspešne, pomocou Eloquent modelu Product sa vytvorí nový záznam v tabuľke products, vrátane automatického uloženia časových pečiatok created\_at a updated\_at.

Ak administrátor nahráva aj obrázky, každé súborové pole sa v rámci metódy store() uloží do priečinka storage/app/public/products, pričom cesta k súboru sa následne zapíše do tabuľky product\_images s cudzím kľúčom na príslušný produkt. Po uložení všetkých údajov kontrolér vyprázdni prípadné cache kľúče súvisiace so zoznamom produktov a obnoví vyhľadávací index, aby bol nový produkt okamžite dostupný vo frontende. Nakoniec aplikácia administrátora presmeruje späť na zoznam produktov s flash správou „Produkt úspešne pridaný“.

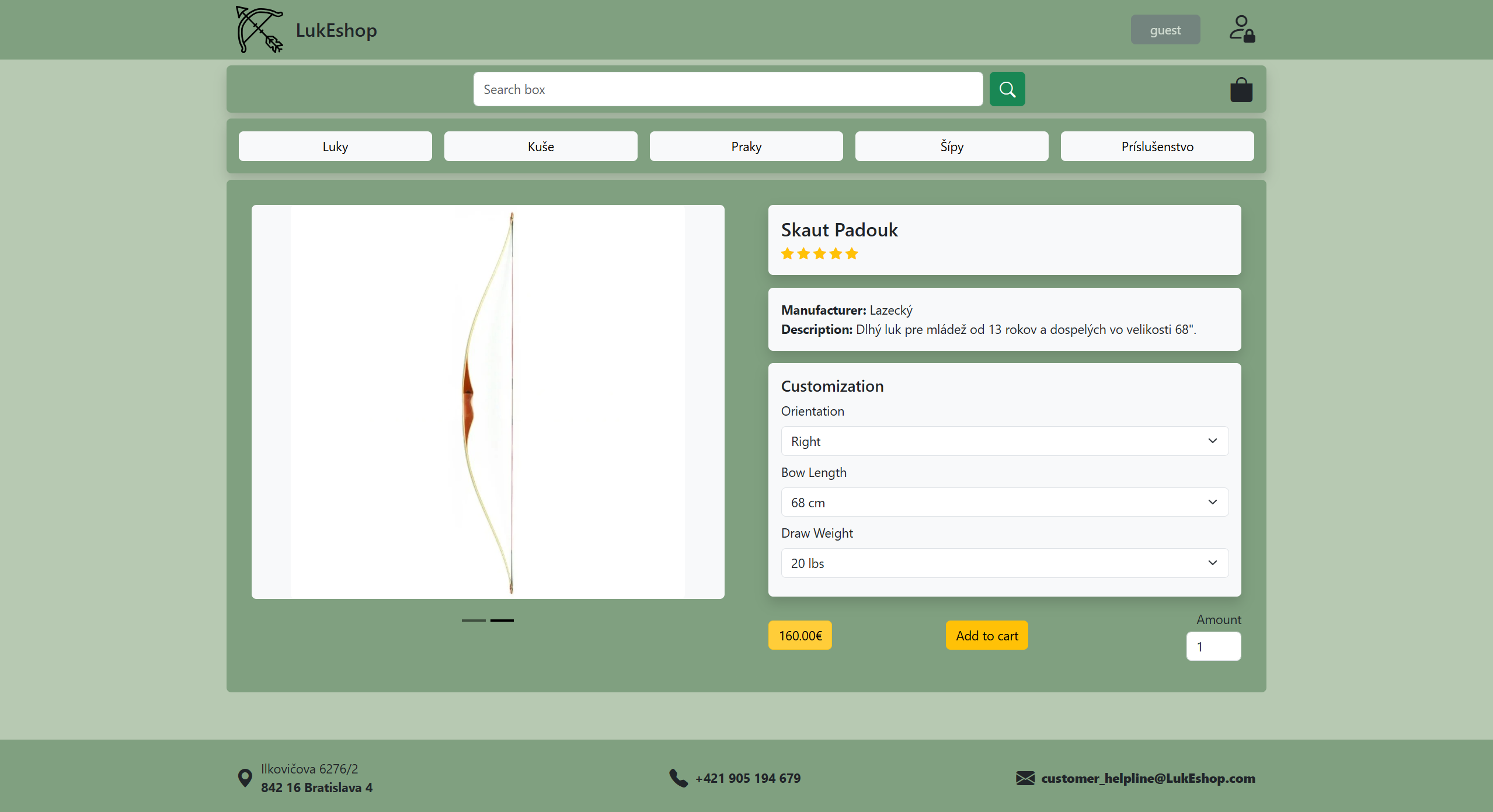
1. Screenshots
   1. HomePage



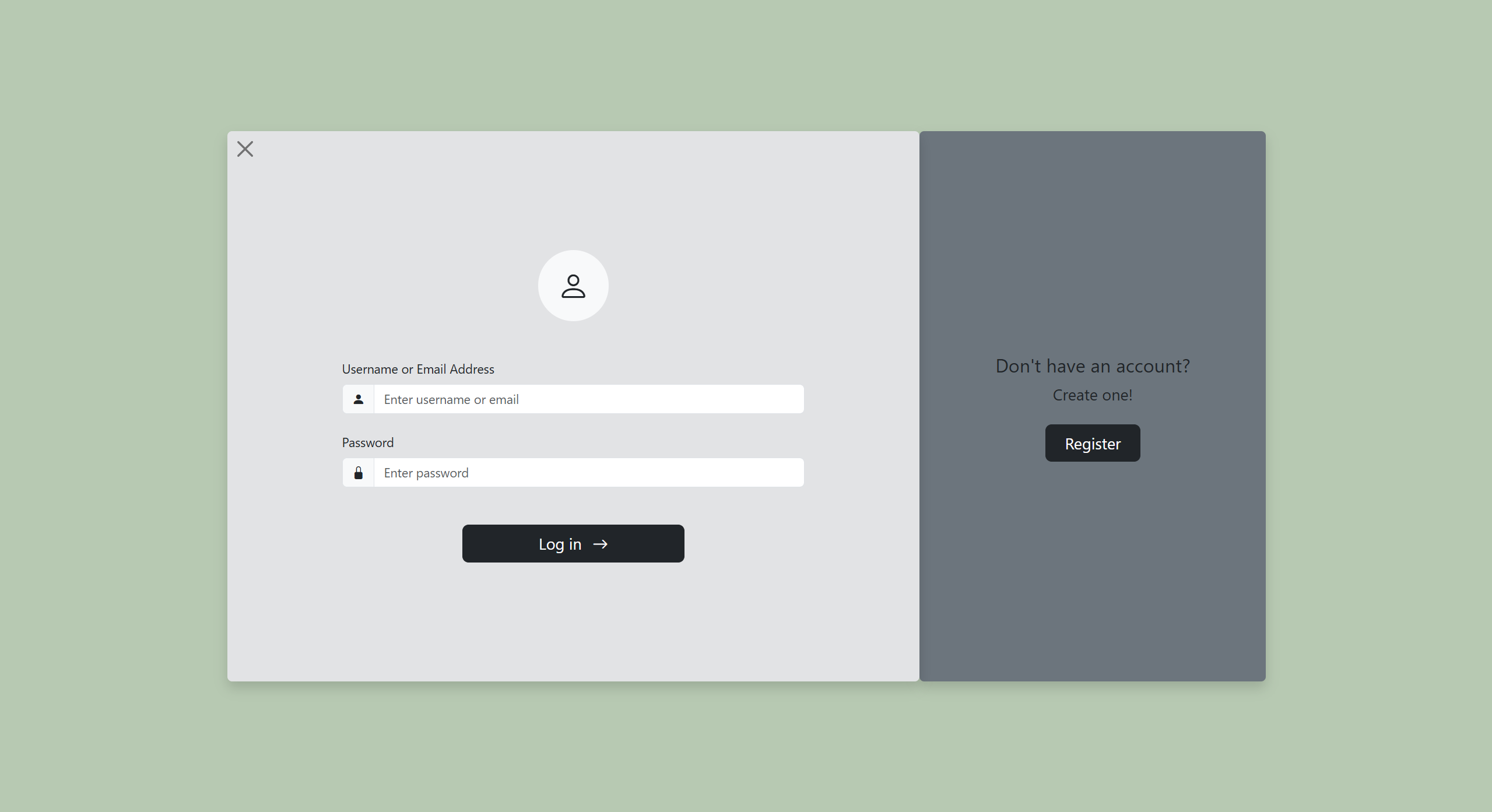
* 1. Search-filter



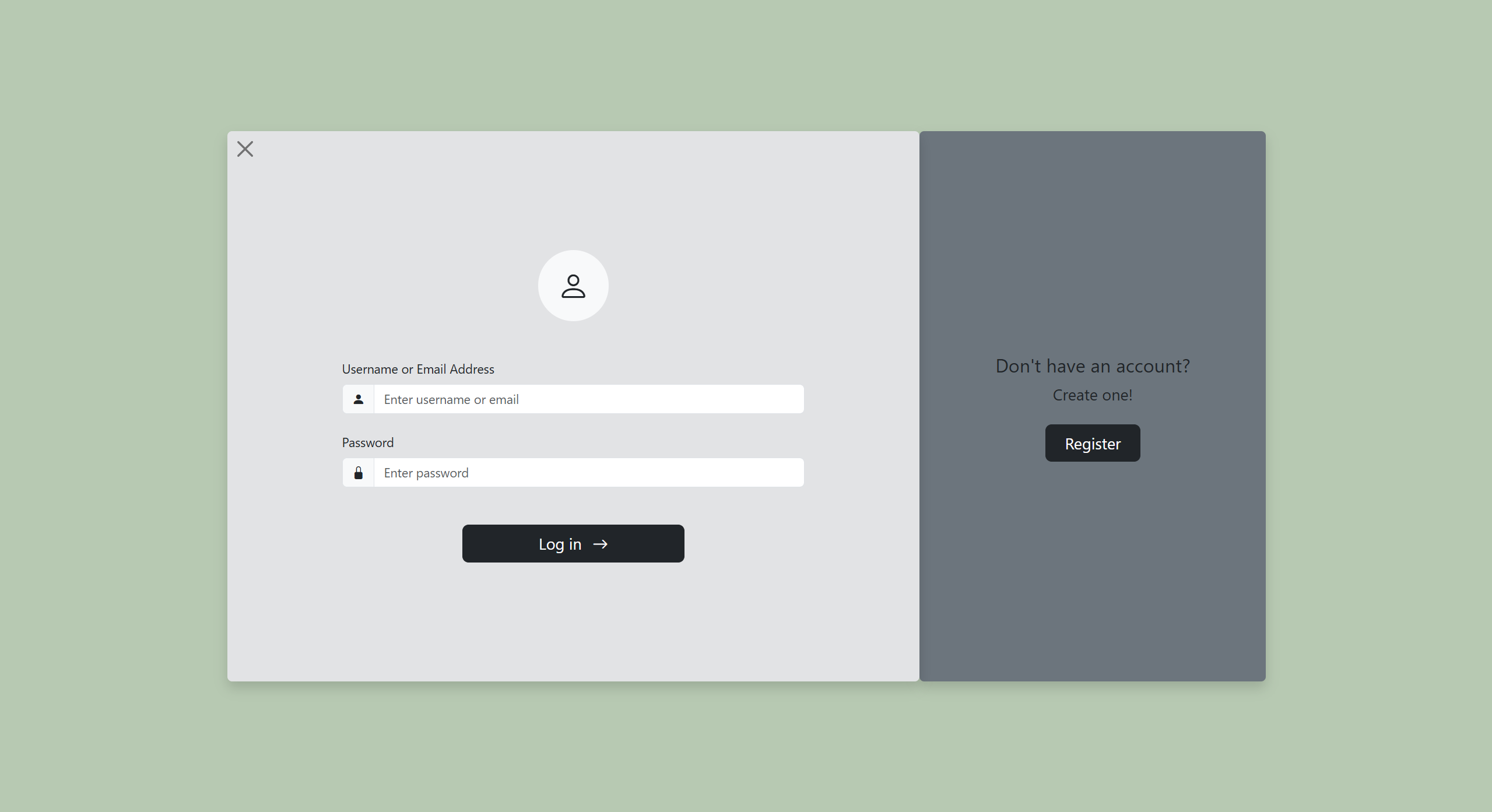
* 1. Products



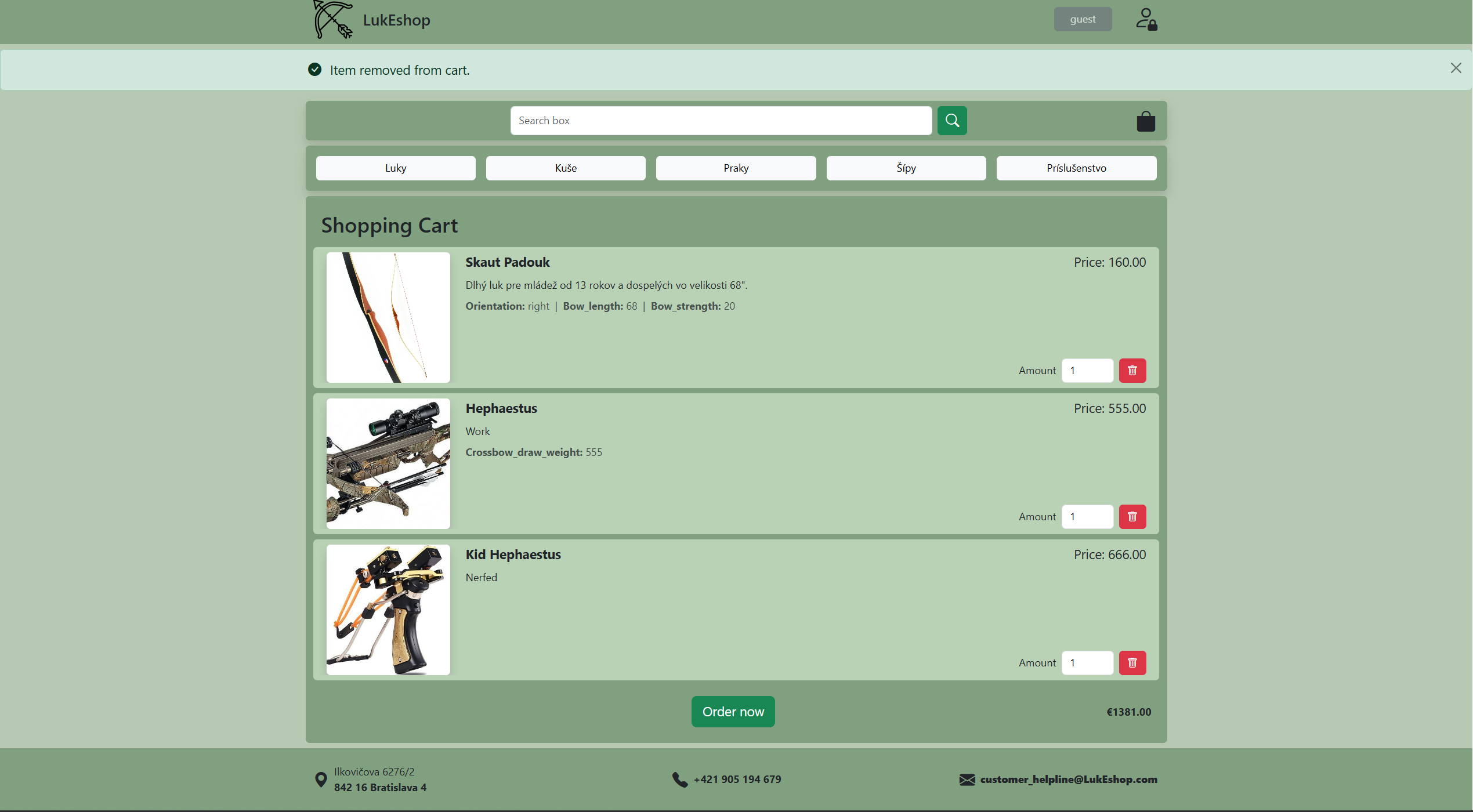
* 1. Login



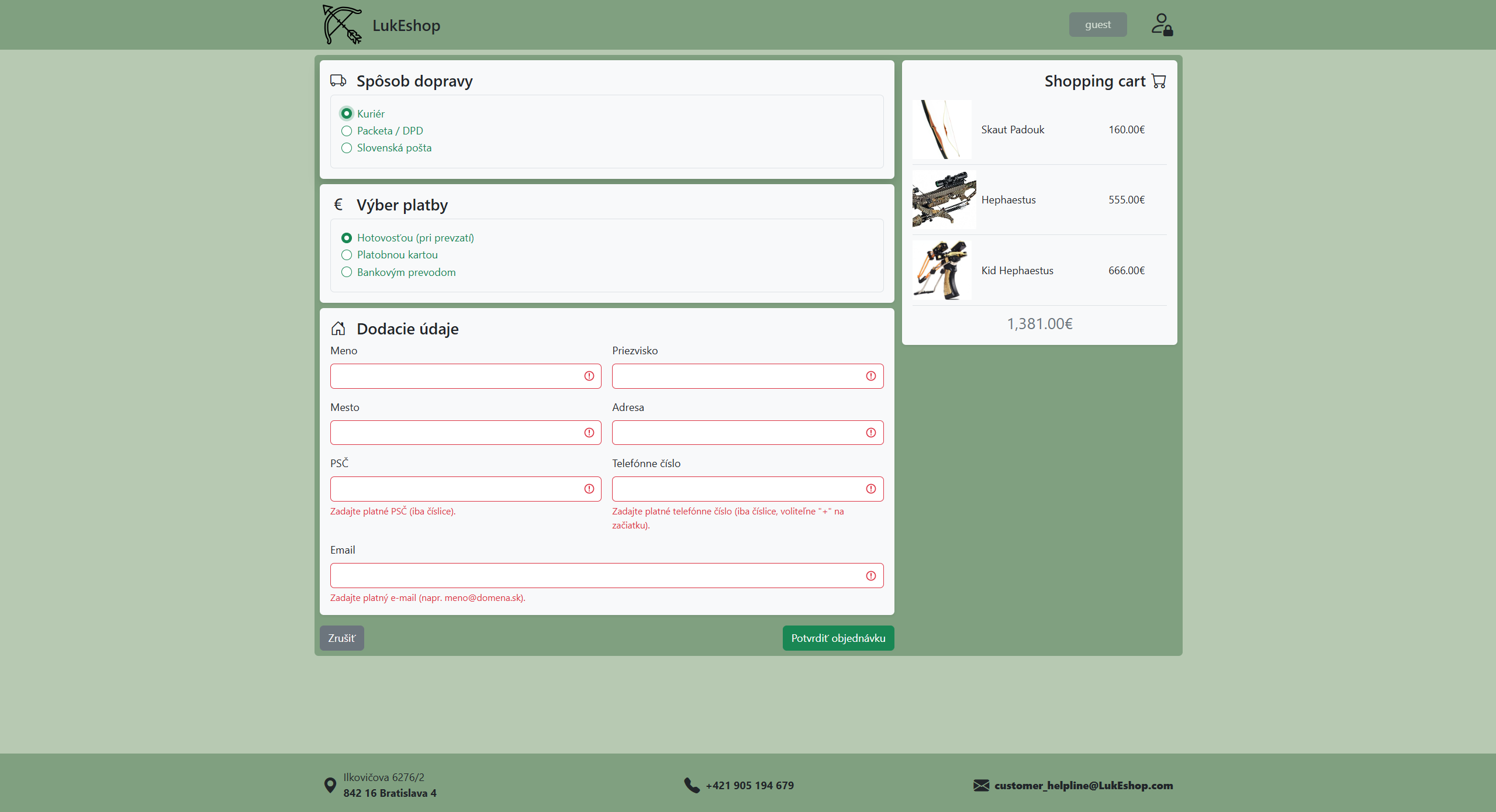
* 1. Register



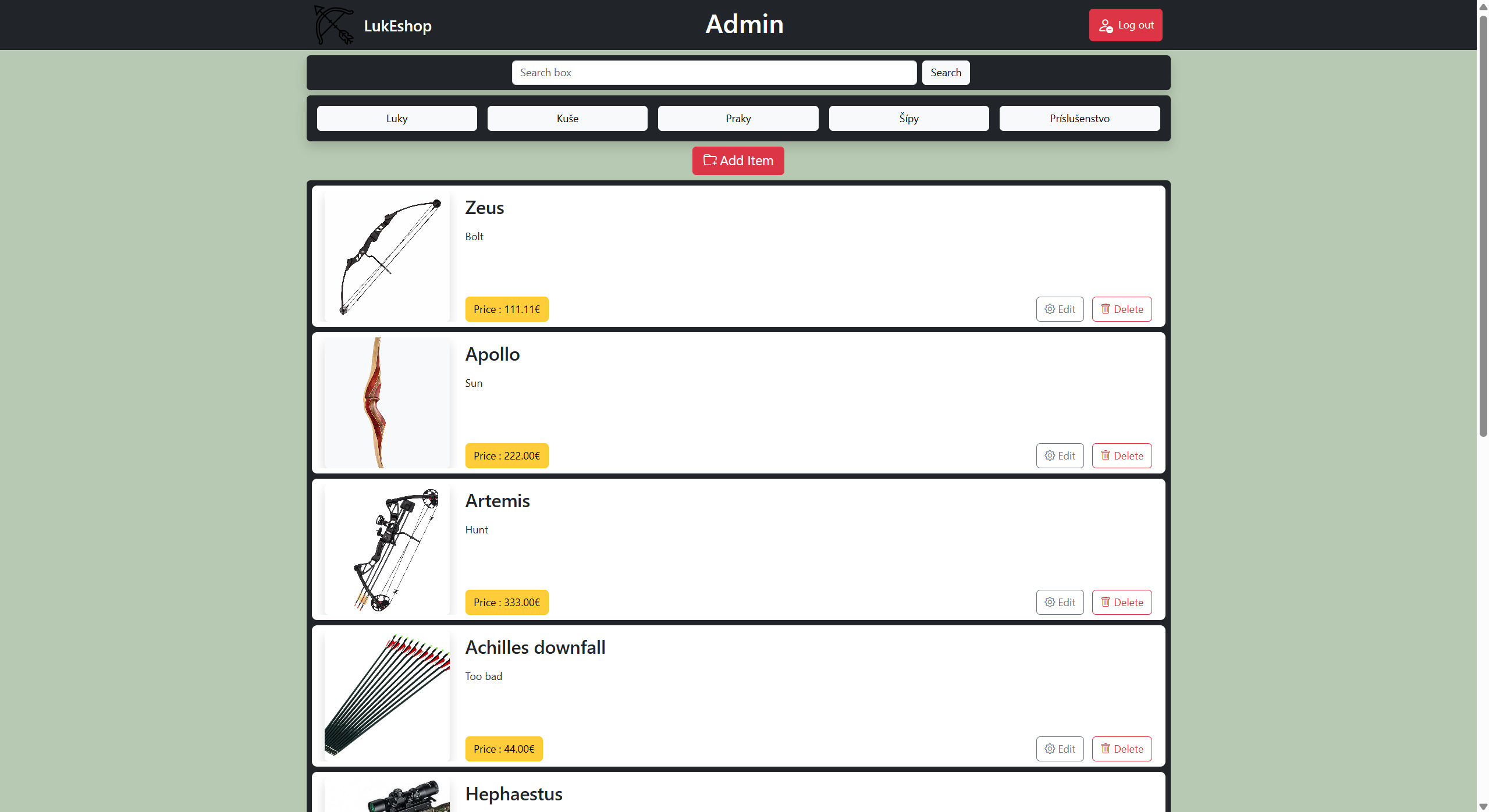
* 1. Shopping-cart



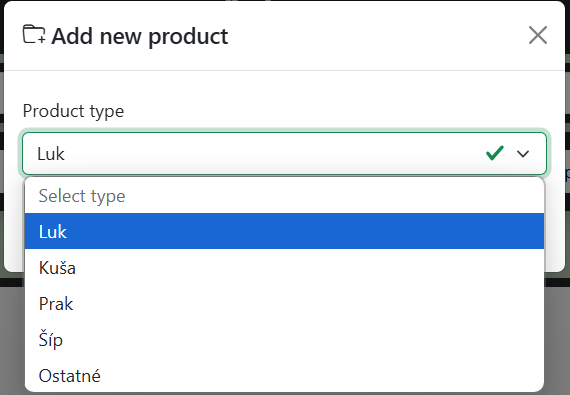
* 1. Order



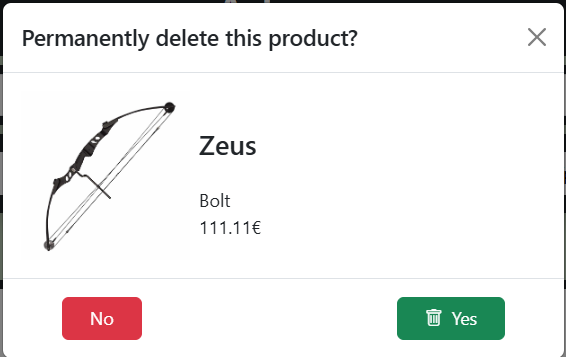
* 1. Admin-page



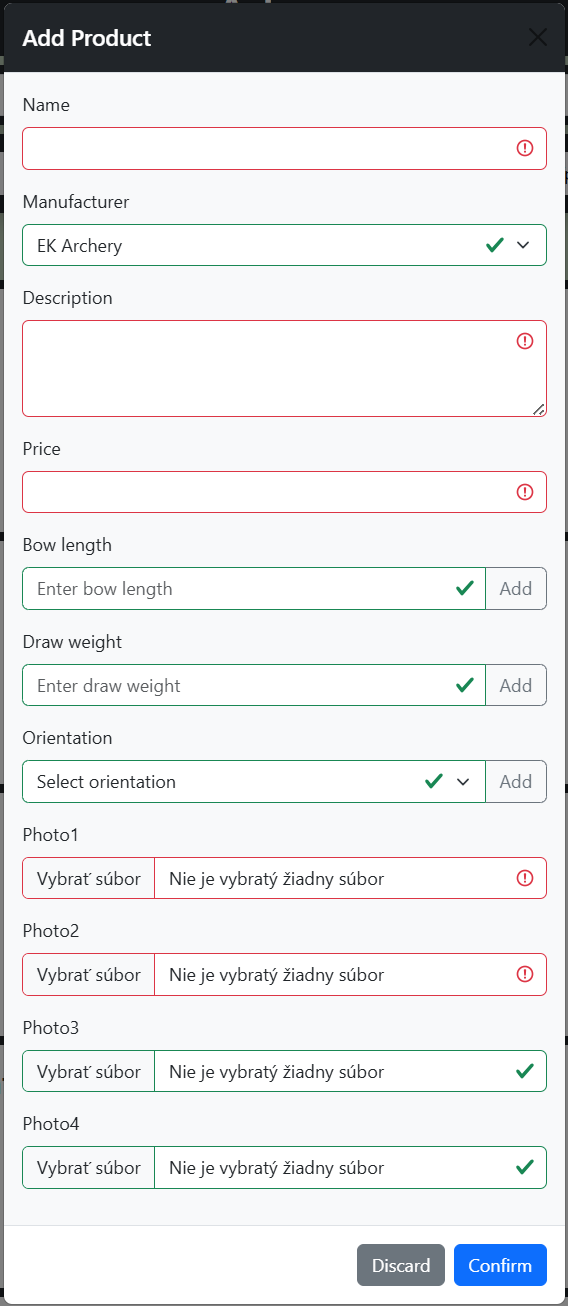
* + 1. Add new product



* + 1. Delete product



* + 1. Add product



* + 1. Edit product

