SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií  
Ilkovičova 2, 842 16 Bratislava 4

**E-shop ReadMe**

Základy webových technológií

Dana Hrivnáková, Róbert Junas, Kateryna Lytvynenko

Cvičenie: Streda 12:00

Cvičiaci: Ing. Eduard Kuric, PhD.

7. 5. 2023

**Obsah**

[1. Zadanie 3](#_Toc134273025)

[2. Diagram fyzického dátového modelu 4](#_Toc134273026)

[3. Návrhové rozhodnutia 5](#_Toc134273027)

[4. Implementácia prípadov použitia 5](#_Toc134273028)

[5. Snímky obrazoviek 8](#_Toc134273029)

# 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 vybratej 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
    - pridanie produktu do košíka (ľubovolné 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

Obrázok, na ktorom je diagram, schematický

Automaticky generovaný popis

* Do tabuľky *photos* bol pridaný stĺpec *is\_cover*, na lepšie zobrazovanie obálky knihy na stránke a kvôli zlepšeniu formuláru, resp. zistenia, ktorá nahratá fotka je obálka.
* Pridanie tabuľky *carts*, pre lepšie dodržanie zásad pomenovaní v Laravel a kvôli aktualizácií množstva v košíku pre neprihláseného používateľa cez API endpoint
* Premenovanie tabuľky *cart\_items* na *book\_cart, order\_items* na *book\_order,* a *order\_details* na *orders*, pre dodržanie názvoslovia Laravel-u
* Zmena cudzieho klúča v *book\_cart user\_id* na *cart\_id*, kvôli novej tabuľke pre košík.
* Zmena názvu cudzieho klúča v *book\_order* z *order\_detail\_id* na *order\_id*
* Zmena názvu alebo pridanie atribútov *created\_at* a *updated\_at* vo väčšine tabuliek, kvôli konzistentnosti s Laravel modelmi

*Poznámka:* atribúty *language, type, role, payment, delivery* sú naďalej enumerácie, ale použítý nástroj ich ukazuje ako varchar(255)

# Návrhové rozhodnutia

* Použitie JavaScript knižnice *OwlCarousel* na vytvorenie carouselu na hlavnej stránke a na profile knihy, pretože carousel z knižnice Bootstrap nespĺňa naše požiadavky ovládania a zobrazovania položiek.
* Aj neprihlásený používateľ má vytvorený košík v tabuľke *carts* v databáze, takéto košíky nemajú pridelený *user\_id.* Takáto implementácia nám dovoľuje zozbierať analytické údaje, ako napr. koľko návštevníkov za určitý čas prišlo na našu stránku a vložilo produkty do košíka a neobjednalo . Zároveň pri registrácií sa takémuto košíku pridelí *user\_id* a nastaví sa hodnota *updated\_at*. Toto tiež má analytickú hodnotu v tom, že vieme, koľko používateľov sa po pridaní produktov do košíka registrovalo. Jedinou nevýhodou je nadbytočné ukladanie už nepoužívaných košíkov, čo môže byť riešené tým že po určitej dobe sa takéto košíky z databázy vymažú.
* Pomocou enumerácie máme definované 2 role: *admin, user.* Pričom user má prístup len stránku eshopu a *admin* má prístup aj do administrátorského rozhrania a môže vytvarať, editovať a vymazávať knihy. Autentifikácia admina je spravená cez middleware *auth a* náš middleware *AdminMiddleware* zabezpečuje to že prístup ku stránkam administrátorského rozhrania bude mat len používateľ s rolou “admin”, a ak nie je nikto prihlásený presmeruje na stránku */admin/login/.* Ak prihlásený používateľ má rolu *user,* tak bude presmerovaný na hlavnú stránku.

# Implementácia prípadov použitia

* **Zmena množstva pre daný produkt** – Zmena množstva môže nastať v 3 prípadoch:
  + *Pridanie z profilu* – používateľ zmení hodnotu množstva na profile a stlačí vložiť do košíka, vtedy sa vyvolá v *CartController* metóda *addProduct*, ktorá ak existuje záznam v košíku, pripočíta nové množstvo k predošlému
  + *Pridanie z košíku* – používateľ zmení hodnotu množstva na profile a stlačí vložiť do košíka, vtedy sa vyvolá v *CartController* metóda *addProduct*, ktorá ak existuje záznam v košíku, inkrementuje predošli počet o jedna.
  + *Zmena množstva v košíku –* v košíku si používateľ môže zvoliť presne aké množstvo chce. Aby sme predišli prekresľovaniu stránky pri každom kliknutí +/-, sme pomocou JavaScriptu a knižnice axios vytvorili asynchrónne volanie, ktoré v *CartController*volá funkciu *update,* ktorá do databázy nastaví novú hodnotu pre knihu.
* **pridanie produktu do košíka** – funguje podobne ako zmena množstva, s tým že ak používateľ nemá v *session* uložený košík, tak košík ak prihlásený používateľ má košík tak sa mu do *session* uloží id košík, na základe tohto id sa do košíka vloží položka, ak používateľ košík nemá, tak sa vytvorí nový košík a do *session* sa uloží toto id.
* **registrácia** - vykonáva sa pomocou *RegistrationController.*

Vo vnútri metódy register(Request $request): sa overujú poskytnuté údaje formulára pomocou metódy $request->validate(). Pravidlá validácie sú definované pre rôzne polia formulára ("meno", "priezvisko", "e-mail", "heslo", atď). Ak niektoré z validačných pravidiel zlyhá, používateľ je presmerovaný späť na registračnú stránku s príslušnými chybovými správami o validácii.

Ak validácia prebehne úspešne, nový používateľ sa vytvorí v databáze pomocou metódy User::create(). Na základe údajov z formulára sa priradí meno, priezvisko, e-mail a hashované heslo používateľa. Používateľ ihneď po registrácii bude prihlasený pomocou metódy Auth::login($user) a bude presmerovaný na domovskú stránku.

* **prihlásenie** – vykonáva sa pomocou *LoginController* pre obe roly.

Ak rola je "user", metóda check(LoginRequest $request) spracúva proces odoslania prihlasovacieho formulára a overenia. Pomocou metódy $request->validate() sa kontroluje, či e-mail existuje v tabuľke *users*. Ak e-mail neexistuje, vráti sa chybová správa o overení. Heslo sa overuje pomocou funkcie Auth::attempt(). Ak heslo je nesprávne, tiež sa vráti chybová správa.

V prípade že overenie prebehlo úspešne, ak v relácii existuje parameter 'cart\_id', načíta sa z databázy príslušný objekt Cart. V opačnom prípade vytvorí nový objekt Cart, uloží ho do databázy a jeho ID uloží do relácie. Nakoniec je používateľ presmerovaný na domovskú stránku.

Ak sa prihlasuje admin, proces overenia použivateľského mena a hesla prebieha rovnáko pomocou adminCheck(Request $request), len sa ešte dodátočne kontroluje či je rola overovaného používateľa "admin". Ak je, používateľ je presmerovaný na stránku '/admin/list'.

Ak rola overovaného používateľa nie je "admin", používateľ bude odhlasený pomocou funkcie Auth::logout(), vráti sa chybová správa a používateľ je potom presmerovaný späť na predchádzajúcu stránku.

* **vyhľadávanie** – Ak používateľ zadá určitý dotaz do searchbaru v navigácii stránky, zobrazia sa výsledky zodpovedajúce dotazu. Dotaz sa hľadá medzi titulom, autormi, popise a ISBN knihy. Ak používateľ zadá dotaz, v ktorom je medzera, tak sa všetky slová hľadajú jednotlivo, nie ako jeden celý reťazec a je medzi nimi vzťah AND.

Na používateľskom rozhraní ho vykonáva *CatalogController* vo funkcii *index*. Na admin rozhraní ho vykonáva *ProductController* vo funkcii *index*.

* **Stránkovanie** – je implementované pomocou funkcie paginate(N), kde N je číslo kníh, ktoré chceme zobraziť na stránke. Vykonáva sa vždy nakonci, ako získame všetky knihy, ktoré chceme podľa podmienok zobraziť.

$books = $books->paginate(6);

Na stránke sa následne zobrazuje pomocou funkcie links(), pričom dané poradie stránky sa vždy pridá na koniec.

{!! $booksQ->appends(['type' => $type, 'q' => $q, 'genre' => $genre, 'lang' => $lang, 'pages' => $pages, 'sort' => $sort])->links() !!}

* **základné filtrovanie** – Používateľ môže filtrovať podľa žánru, počtu strán, typu alebo jazyka knihy. *CatalogController* vo funkcii *index* vyberie z databázy všetky knihy, ktoré vyhovujú filtrom. Medzi žánrom, počtom strán, jazykom a typom je vzťah AND. Napr. sa vyberú všetky audioknihy v slovenskom jazyku. Avšak ak používateľ vyberie viac filtrov jedného typu, napr. viac žánrov, medzi nimi je vzťah OR. Čiže sa vyberú audioknihy v slovenskom jazyku žánra Beletria alebo Fantasy. Ak používateľ vyberie viac filtrov jedného typu, tak ich do URL zapisujeme pomocou čiarok. URL príkladu by vyzerala teda nasledovne:

<http://127.0.0.1:8000/katalog?type=audiokniha&genre=1%2C3&lang=slovensky>

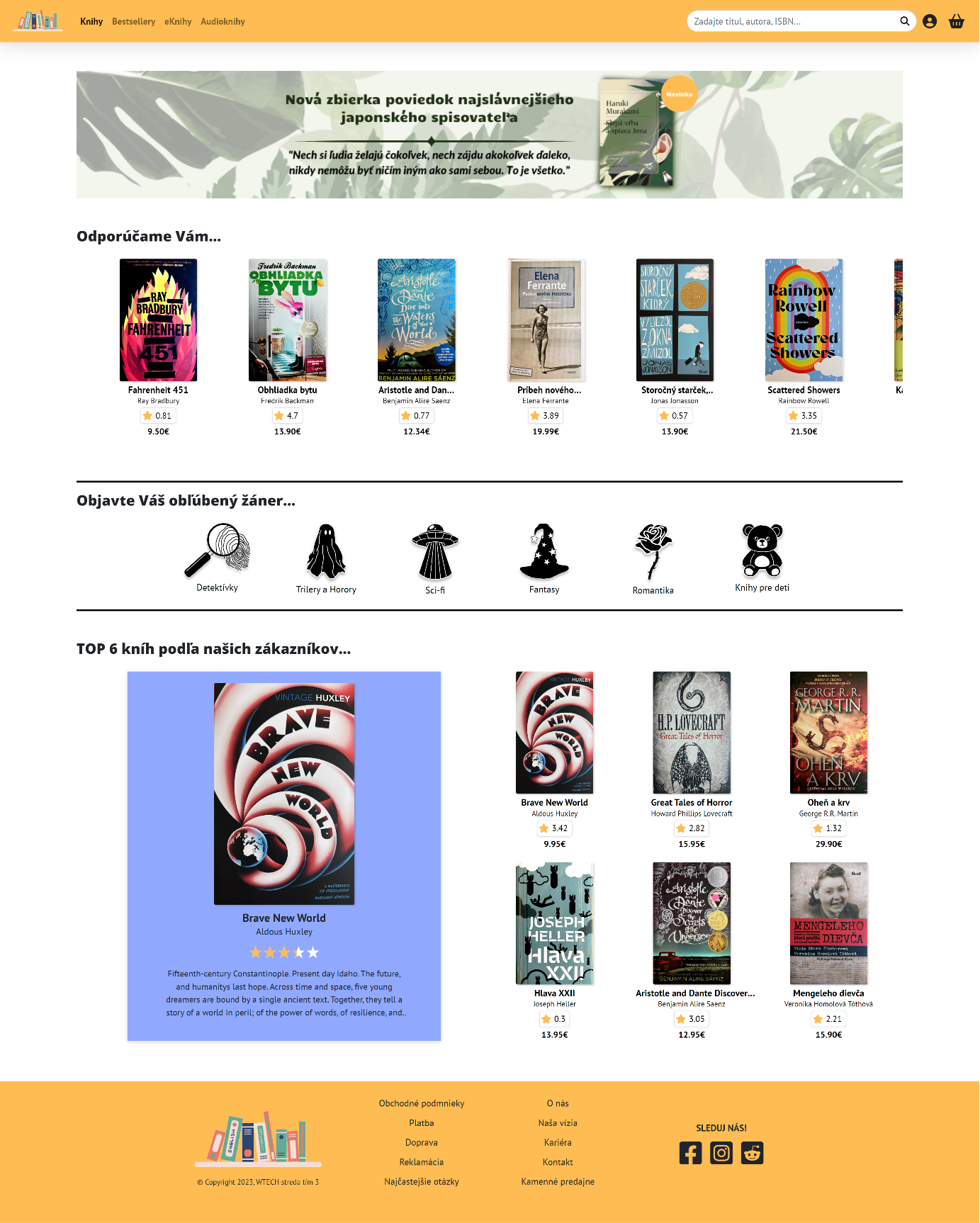
Pomocou atribútov URL sa aj aplikuje atribút *checked* na checkboxy, aby používateľ vedel, ktoré filtre sú aktívne.

* **Preusporiadanie produktov** –Používateľ môže produkty presuporiadať pomocou 4 kritérií:
  + **Novinky** – produkty sú zoradené podľa dátumu pridania do databázy od najnovšieho
  + **Bestsellery** - všetky naše produkty sú bestsellery, preto sú zoradené abecedne
  + **Najlacnejšie** - produkty sú zoradené podľa ceny vzostupne
  + **Najdrahšie** - produkty sú zoradené podľa ceny zostupne

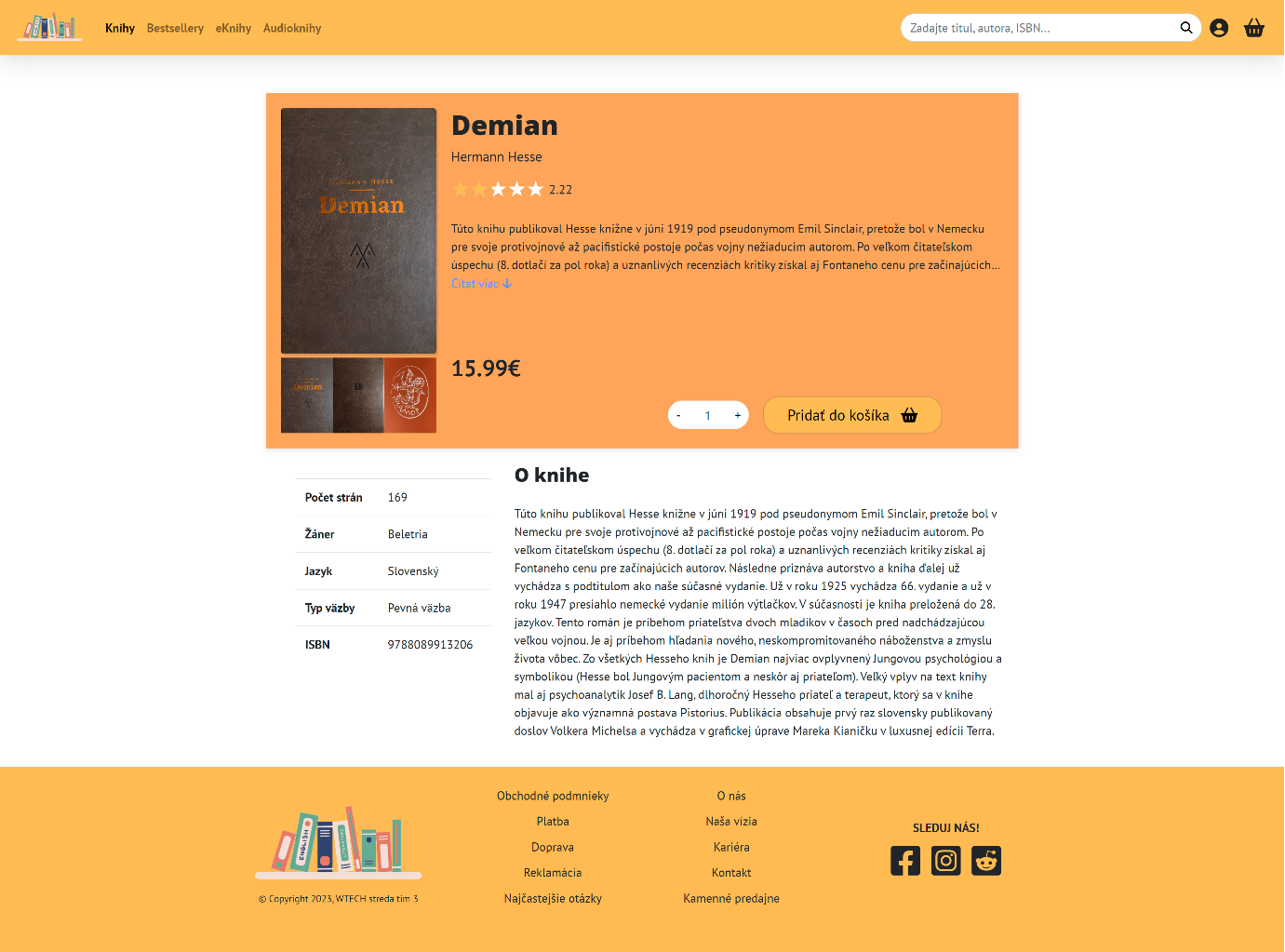
Preusporiadanie vykonáva *CatalogController* vo funkcii *index*.

# Snímky obrazoviek

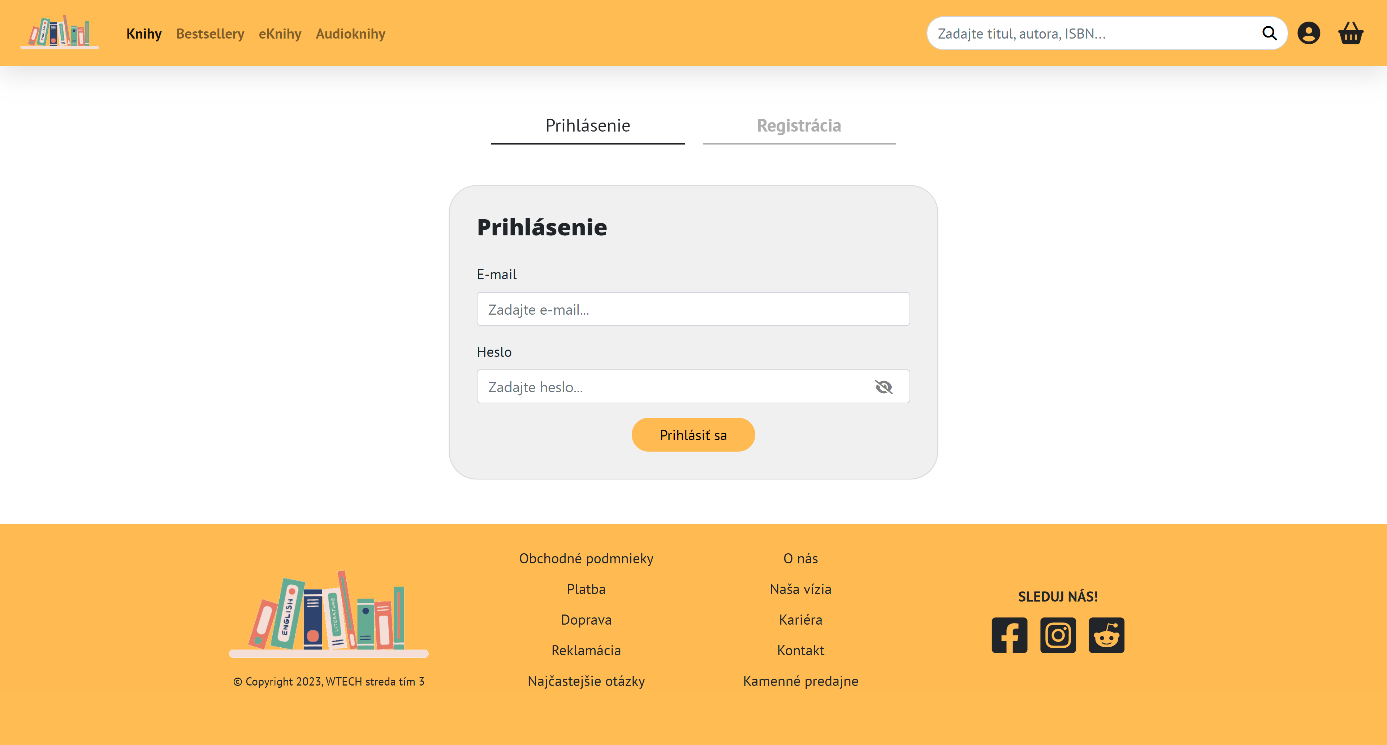
* Homepage



* Detail produktu



* Prihlásenie



* Nákupný košík

