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.

26. 3. 2023

Obsah

[1. Zadanie 3](#_Toc134200557)

[2. Diagram fyzického dátového modelu 3](#_Toc134200558)

[3. Návrhové rozhodnutia 4](#_Toc134200559)

[4. Implementácia prípadov použitia 4](#_Toc134200560)

[5. Snímky obrazoviek 6](#_Toc134200561)

# 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

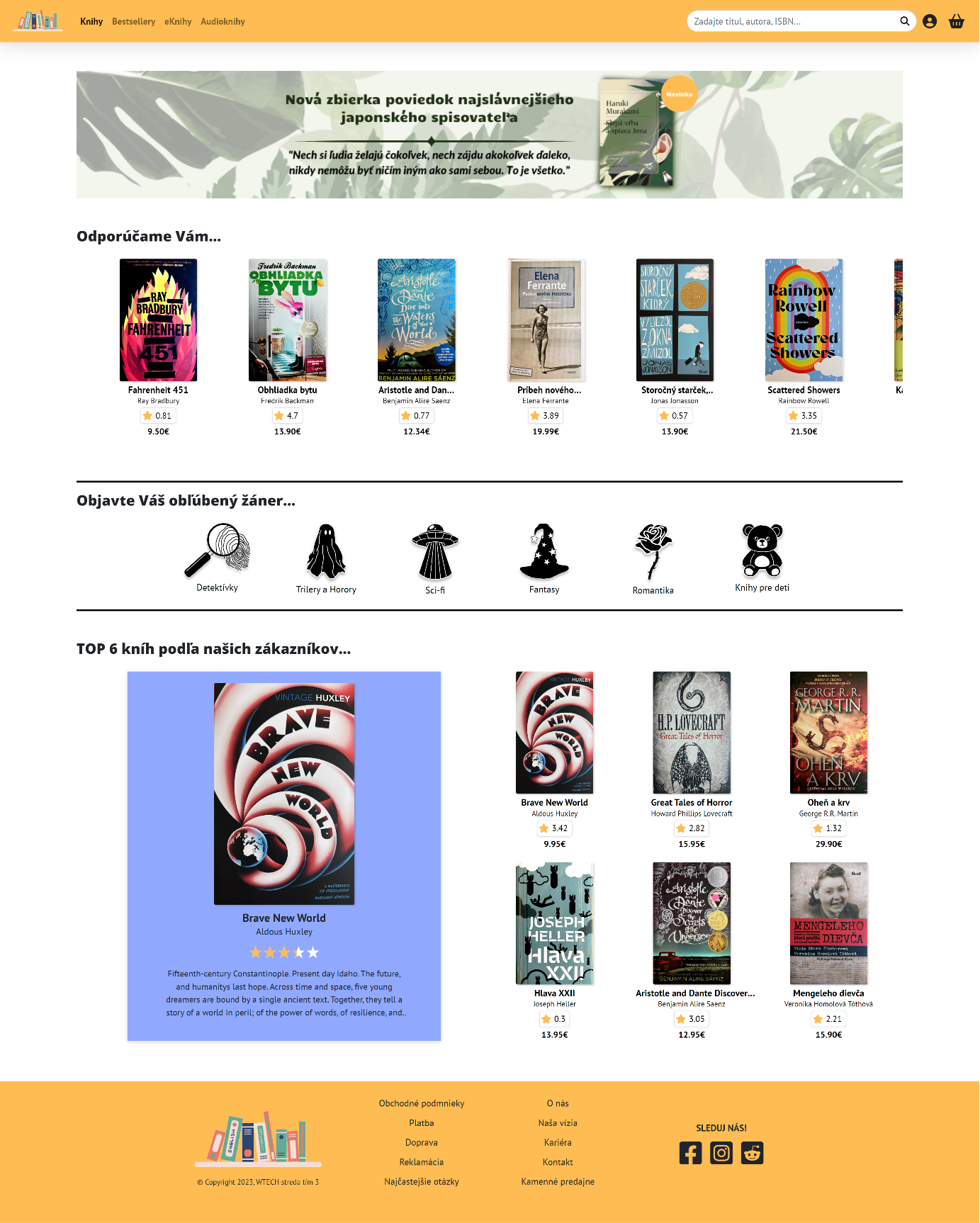
* Použitie JavaScript knižnice *OwlCarousel* na vytvorenie carouselu na hlavnej stránke a na profile knihy, pretože carousel z knižnice Bootstrap nespĺňa naše požiadavky ovládania a zobrazovania položiek.
* Aj neprihlásený používateľ má vytvorený košík v tabuľke *carts* v databáze, takéto košíky nemajú 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.
* ...

# Implementácia prípadov použitia

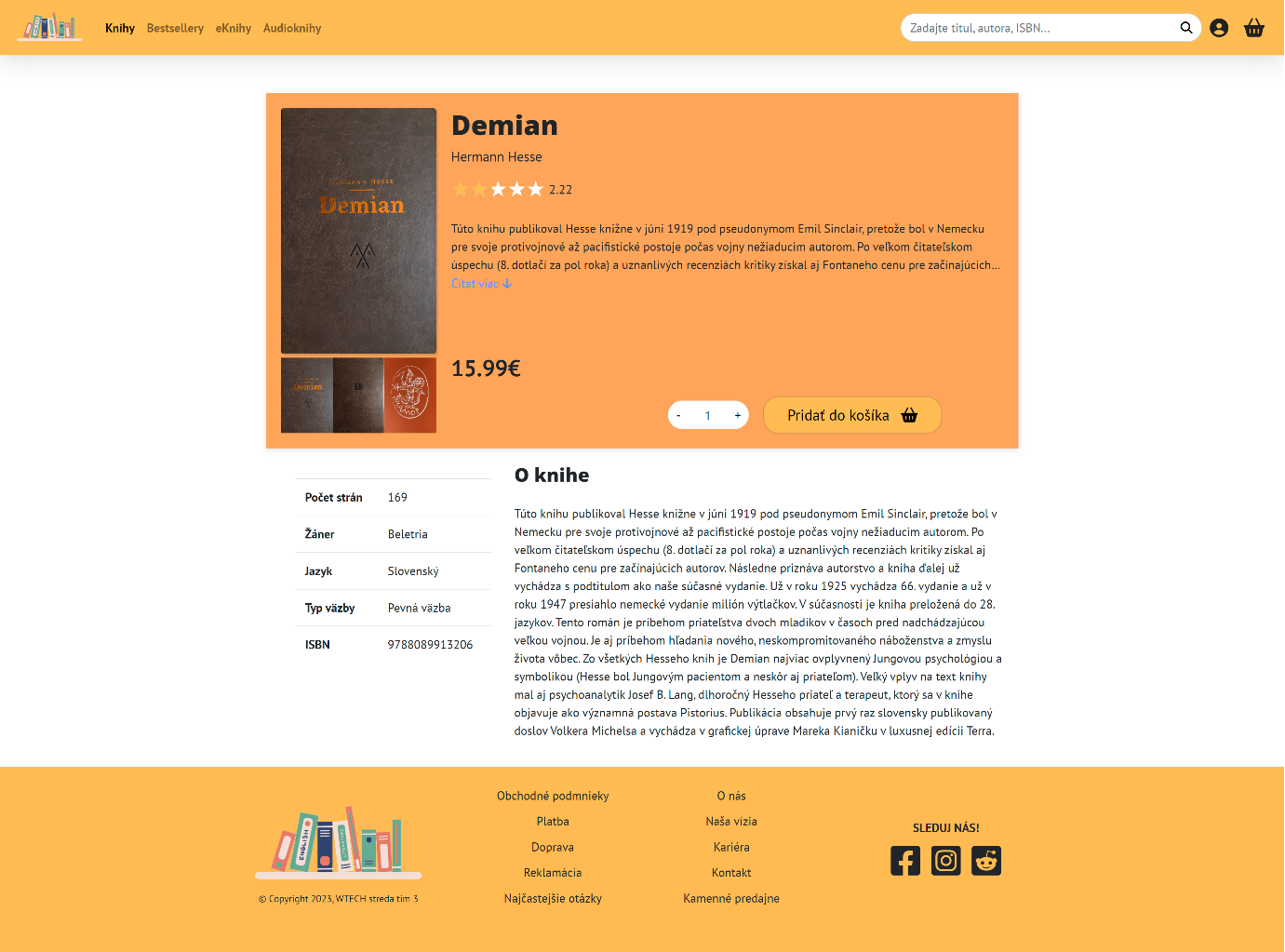
* **Zmena množstva pre daný produkt** – Zmena množstva môže nastať v 3 prípadoch:
  1. *Pridanie z profilu* – používateľ zmení hodnotu množstva na profile a stlačí vložiť do košíka, vtedy sa vyvolá v *CartController* metóda *addProduct*, ktorá ak existuje záznam v košíku, pripočíta nové množstvo k predošlému
  2. *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.
  3. *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.
* prihlásenie,
* vyhľadávanie,
* **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.
* stránkovanie,
* základné filtrovanie

# Snímky obrazoviek

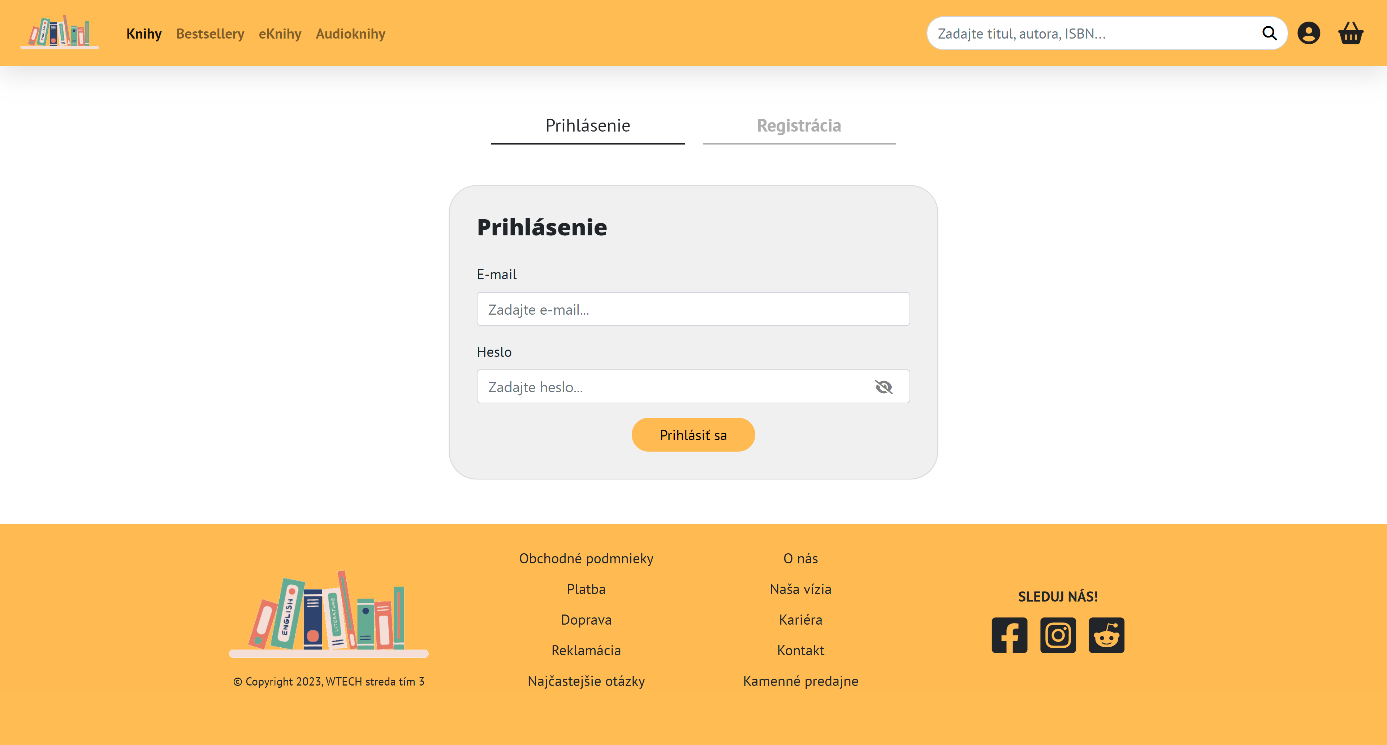
* Homepage



* Detail produktu



* Prihlásenie



* Nákupný košík

