**SLOVENSKÁ TECHINCKÁ UNIVERZITA V BRATISLAVE**

**FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ**

SEMSTRÁLNY PROJEKT – ELEKTRONICKÝ OBCHOD

**Dokumentácia**

ZÁKLADY WEBOVÝCH TECHNOLÓGIÍ

Matúš Baran

ID: 102852

Samuel Bubán

ID: 102879

4. 12. 2021

# ZADANIE

Vytvorte webovú aplikáciu - eshop, ktorá komplexne rieši nižšie definované prípady použitia vo vami zvolenej doméne (autosúčiastky a náhradné diely). Presný rozsah a konkretizáciu prípadov použitia si dohodnete s Vašim cvičiacim na cvičení.

# DÁTOVÝ MODEL

Diagram

Description automatically generated

Zmeny oproti pôvodnému návrhu:

V tabuľke items sme namiesto používania price a sale zaviedli nový systém, new\_price a old\_price. Vďaka tejto zmene sme nemuseli počítať novú cenu po zľave na viacerých miestach, ale použijeme priamo new\_price (po zľave).

V tejto tabuľke sme ďalej vymazali aj riadky s odkazmi na obrázky produktov. Namiesto toho sme zaviedli nový systém, kde sú obrázky rozdelené podľa rozlíšenia do príslušných priečinkov, a sú uložené s prislúchajúcim ID produktu. Vďaka tomuto už nepotrebujeme mať záznam o tom, kde sa daný obrázok nachádza.

V tabuľke categories sme vymazali odkaz na obrázok, a namiesto toho sme nechali obrázky len hlavným kategóriám, a ostatné majú len názov. Ďalej sme vymazali riadok filter\_json, ktorý mal udržiavať filtre prislúchajúce danej kategórii, a namiesto toho sme použili globálne filtre.

Vytvorili sme novú tabuľku filters, ktorá obsahuje záznamy o všetkých filtroch, ktoré môžu byť aplikované na predmety. V stĺpci name sú názvy filtrov a v stĺpci values (text) sú ich hodnoty, oddelené bodkočiarkou.

## Možné vylepšenia:

Namiesto použitia shipping\_type a payment\_type typu varchar by bolo lepšie vytvoriť novú tabuľku, ktorá by obsahovala všetky možné typy platieb a druhú, ktorá by obsahovala typy dopravy. Ďalej by tu boli zapísané ich názvy a tiež cena, a v tabuľke orders by sme namiesto textu použili odkaz na tieto dve nové tabuľky.

# NÁVRHOVÉ ROZHODNUTIA

Využili sme viacero externých knižníc:

doctrine/dbal – potrebovali sme premenovať stĺpce v databáze

guzzlehttp/guzzle – TODO

intervention/image – aby sme mohli zmeniť rozlíšenie obrázku, ktorý používateľ zadá pri vytváraní alebo úprave produktu

Vytvorenie novej tabuľky admins na riešenie rolí používateľov – keďže väčšina užívateľov bude len klasický zákazník, prišlo nám zbytočné plytvať miestom, a pridávať každému informáciu o jeho roli. Preto sme vytvorili novú tabuľku admins, kde sú len ID-čka používateľov, ktorý majú admin právomoci.

Žiadny odkaz na obrázky v databáze – nakoľko každý produkt má len jeden obrázok (v 3 rôznych rozlíšeniach), rozhodli sme sa im napevno priradiť cestu, pre každé rozlíšenie jednu. Názov obrázku je teda len ID daného produktu, a podľa toho v ktorom priečinku sa nachádza, také má rozlíšenie.

TU EŠTE MOŽNO DAČO

# PROGRAMOVACIE PROSTREDIE

IDK

# IMPLEMENTÁCIA PRÍPADOV POUŽITIA

Zmena množstva pre daný produkt – nákupný košík je riešený pomocou novej triedy Cart, ktorá je uložená v rámci Session. Keď sa stlačí tlačidlo na zmenu množstva (plus alebo mínus), zavolá sa funkcia z tejto triedy košíka, ktorá zmenu vykoná, a uloží svoj nový obsah do Session. Uložením košíka do session sa zabezpečila prenositeľnosť košíka aj pre neprihláseného používateľa.

Pre prihlasovanie a registráciu sme využili riešenie, ktoré už bolo predpripravené v rámci Laravel aplikácie. Len sme si upravili šablónu, ktorú aplikácia používa na vlastnú, a aj modely sme si upravili podľa potreby{upravovali sme modely kvôli prihlasovaniu?}.

Vyhľadávanie sme si implementovali vlastné – vyhľadávacia fráza sa zadá ako parameter, a následne používame full-text prehľadávanie nad názvom produktu, jeho popisom, a parametrami. Ak sa fráza nachádza v niektorom z nich, produkty sú vrátené ako výsledok.

Stránkovanie sme použili priamo z laravelu už predpripravené, len sme si ho adaptovali. Zoradenie na stránke podľa ceny sme implementovali cez query builder, kde nám vráti priamo zoradené produkty už podľa potreby. Počet produktov na stránku je tiež riešený laravel postupom, len použijeme vlastné hodnoty na presné zadanie.

Filtrovanie sme implementovali procedurálne. To znamená, že máme vlastnú tabuľku pre filtre (kde sú aj príslušné hodnoty každého filtra), a do tejto tabuľky vieme pridávať nové filtre s novými hodnotami, a automaticky sa pridajú aj do prehľadu kategórii. Keď si používateľ zvolí príslušné filtre, pošlú sa cez dotazník naspäť na server, kde sa spracujú všetky hodnoty, a uložia sa do Session (aby sa zachovali). Prehľadávanie prebieha priamo v query builder-i nad hodnotou info\_json v každom produkte.

V tomto smere by bolo možné našu aplikáciu ešte zlepšiť tak, že by sa zobrazili iba filtre, ktoré sú vhodné pre dané produkty (ak si zvolíme brzdovú kvapalinu, tak nebude ponúkať filter nad materiálmi ako sú hliník alebo meď).