**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.

Vytvorili sme novú tabuľku carts, v ktorej sú uložené nákupné košíky prihlásených používateľov, aby im zostal uložený obsah, aj keď sa prihlásia na novom zariadení. Do tejto tabuľky sa pri registrácii vytvorí nový záznam pre daného používateľa s prázdnym obsahom.

## 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.

Prístup do admin časti – prístup do admin časti stránky sme realizovali formou tlačidla v obrazovke profilu, aby bol oddelený od častí stránky, ktoré súvisia s nakupovaním. Tlačidlo sa zobrazí len používateľom, ktorí sú zároveň aj adminmi. To je zabezpečené pomocou ItemPolicy, ktorá kontroluje v databáze, či je prihlásený používateľ admin.

# PROGRAMOVACIE PROSTREDIE

Vo vypracovaní projektu sme použili odporúčané technológie – Laravel a databázu PostgreSQL. Ako editor sme používali Visual Studio Code, a PhpStorm.

# 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 pre neprihláseného používateľa.

Ak je používateľ prihlásený, používame na uloženie košíka priamo databázu, do ktorej sa zapíše obsah nákupného košíka – všetky produkty, a pre každý produkt jeho množstvo. Týmto sa zabezpečí, že používateľ si môže otvoriť svoj košík aj na druhom zariadení.

Ak je množstvo rovné nule, tak sa z košíka vymaže, rovnako ako keby bolo stlačené priamo tlačidlo na vymazanie.

Ak užívateľ klikne na pridať produkt do košíka, a tento produkt tam už má, tak sa len pridá jeden kus. Ak klikne na pridať viacero kusov (z prehľadu produktu), tak sa pripočíta tento počet k už pridaným produktom, a zapíše sa to do Session, alebo priamo do databázy (ak je prihlásený).

Pridanie produktu do košíka – ak používateľ nie je prihlásený, vytvorí sa nová inštancia triedy Cart, do ktorej sa zapíše tento nový produkt (plus existujúci obsah košíka) a uloží sa do Session. Ak používateľ je prihlásený, tak sa produkt pridá do jeho košíka, ktorý je načítaný z databázy.

Pre prihlasovanie a registráciu sme využili riešenie, ktoré už bolo predpripravené v rámci Laravel aplikácie, akurát sme zmenili šablónu, ktorú aplikácia používa na vlastnú, a upravili sme si modelz tak, aby vyhovovali našim potrebám.

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. Tu sa využíva filtrovanie produktov priamo z databázy, pomocou query builder-u, takže nepracujeme so zbytočne veľa produktami, ktoré sa nevyužijú.

Na stránkovanie sme použili Paginator priamo z Laravelu, 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 zobrazených na stránke je tiež riešený cez Paginator, 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 je možné 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. Filtre sú zo Session vymazané po navštívení domovskej stránky, aby neboli aplikované pri ďalšom prehliadaní produktov.

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).