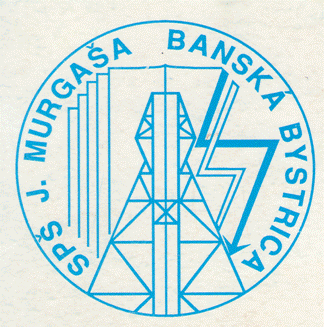
**STREDNÁ PRIEMYSELNÁ ŠKOLA JOZEFA MURGAŠA, HURBANOVA 6, BANSKÁ BYSTRICA**

****

**WEBOVÁ STRÁNKA E-SHOP**

MATURITNÁ PRÁCA PČOZ forma b)

SPRIEVODNÁ DOKUMENTÁCIA

**PATRIK GREXA, IV. B**

**BANSKÁ BYSTRICA 2018/2019**

**STREDNÁ PRIEMYSELNÁ ŠKOLA JOZEFA MURGAŠA, HURBANOVA 6, BANSKÁ BYSTRICA**

Meno a priezvisko: Patrik Grexa

Trieda: IV. B

Školský rok: 2018/2019

Odbor: 2694 M informačné a sieťové technológie

Názov práce: Webová aplikácia E-shop

Zadanie práce: Zostrojiť funkčný internetový obchod

Vytvoriť administrátorské rozhranie

Možnosť vytvoriť si vlastný profil

Vytvoriť technickú dokumentáciu

Dátum zadania: 24. 10. 2018

Konzultant: Ing. Ivan Koštial

Podpis konzultanta: ................................

**ČESTNÉ VYHLÁSENIE**

Čestne vyhlasujem, že maturitnú prácu som vypracoval samostatne s použitím uvedenej odbornej literatúry.

 Banská Bystrica 11. 2. 2019 ............................................

vlastnoručný podpis autora

**Obsah**

[Zoznam Obrázkov 7](#_Toc1575890)

[Úvod 9](#_Toc1575891)

[1 Ciele práce 10](#_Toc1575892)

[2 Teoretické východisko 12](#_Toc1575893)

[2.1 Flask 12](#_Toc1575894)

[2.1.1 Werkzeug 12](#_Toc1575895)

[2.1.2 Jinja 12](#_Toc1575896)

[2.2 Backbone.js 12](#_Toc1575897)

[2.2.1 Backbone.Model 13](#_Toc1575898)

[2.2.2 Backbone.Collection 13](#_Toc1575899)

[2.2.3 Backbone.Router 14](#_Toc1575900)

[2.2.4 Backbone.View 14](#_Toc1575901)

[2.3 jQuery 15](#_Toc1575902)

[2.3.1 Ajax 15](#_Toc1575903)

[2.4 Python 15](#_Toc1575904)

[2.5 JavaScript 15](#_Toc1575905)

[2.6 Materialize 16](#_Toc1575906)

[3 Metodika práce 17](#_Toc1575907)

[3.1 Main.py 17](#_Toc1575908)

[3.2 Global\_model.js 20](#_Toc1575909)

[3.3 Kosik\_model.js 21](#_Toc1575910)

[3.4 Obrazky\_model.js 21](#_Toc1575911)

[3.5 Prehlad\_model.js 21](#_Toc1575912)

[3.6 Produkt\_model.js 22](#_Toc1575913)

[3.7 Kosik.html 22](#_Toc1575914)

[3.8 Prihlasit.html 23](#_Toc1575915)

[3.9 Produkt\_editacia.html 23](#_Toc1575916)

[3.10 Profil.html 24](#_Toc1575917)

[3.11 Registrovat.html 24](#_Toc1575918)

[4 Výsledky práce 25](#_Toc1575919)

[4.1 Hlavná stránka – prehľad 25](#_Toc1575920)

[4.2 Detail produktu 27](#_Toc1575921)

[4.3 Nákupný košík 28](#_Toc1575922)

[4.4 Nákupný košík 29](#_Toc1575923)

[4.5 Formulár 30](#_Toc1575924)

[4.6 Prihlásiť sa 31](#_Toc1575925)

[4.7 Registrácia používateľa 31](#_Toc1575926)

[4.8 Profil používateľa 32](#_Toc1575927)

[4.9 Objednávky 33](#_Toc1575928)

[4.10 Detail objednávky 34](#_Toc1575929)

[4.11 Zoznam produktov pre editáciu 35](#_Toc1575930)

[4.12 Produkt editácia 36](#_Toc1575931)

[5 Záver 37](#_Toc1575932)

[5.1 Stručné zhodnotenie 37](#_Toc1575933)

[5.2 Význam práce 37](#_Toc1575934)

[5.3 Možnosti využitia v praxi 37](#_Toc1575935)

[5.4 Získané poznatky, zlepšenie stránky 38](#_Toc1575936)

[Zoznam použitej literatúry 39](#_Toc1575937)

[Zoznam príloh 40](#_Toc1575938)

[Prílohy 41](#_Toc1575939)

# Zoznam Obrázkov

[**Obrázok 1** Prehľad 25](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575364)

[**Obrázok 2** Hlavná stránka 25](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575365)

[**Obrázok 3** Filter 25](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575366)

[**Obrázok 4** Navigácia 26](#_Toc1575367)

[**Obrázok 5** Pätička 26](#_Toc1575368)

[**Obrázok 6** Balíček produktov 28](#_Toc1575369)

[**Obrázok 7** Reklama 28](#_Toc1575370)

[**Obrázok 8** Nákupný košík 29](#_Toc1575371)

[**Obrázok 9** Odoslanie objednávky 30](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575372)

[**Obrázok 10** Prihlásiť tlačidlo 31](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575373)

[**Obrázok 11** Prihlásenie 31](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575374)

[**Obrázok 12** Registrácia 31](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575375)

[**Obrázok 13** Kontaktné údaje zmena 32](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575376)

[**Obrázok 14** Heslo zmena 33](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575377)

[**Obrázok 15** História objednávok 33](#_Toc1575378)

[**Obrázok 16** Objednávky 34](#_Toc1575379)

[**Obrázok 17** Údaje produktu 34](#_Toc1575380)

[**Obrázok 18** Údaje zákazníka 35](#_Toc1575381)

[**Obrázok 19** Pridať produkt 35](file:///C:\Users\grexa\Desktop\dokumentacia%20najvsia%20Patrik%20grexa.docx#_Toc1575382)

[**Obrázok 20** Zoznam produktov admin 36](#_Toc1575383)

[**Obrázok 21** Produkt editácia 36](#_Toc1575384)

[**Obrázok 22** Pridávanie obrázku 36](#_Toc1575385)

# Úvod

Pre elektronický obchod som sa rozhodol z niekoľkých dôvodov. Prvým bolo získať nové vedomosti a vyskúšať si tvorbu komplexnej webovej stránky. Stránku som považoval za najlepšiu možnosť ako demonštrovať vedomosti, ktoré mám. Taktiež si myslím, že to bola pre mňa najvhodnejšia téma.

Cieľov som si zvolil viacero. Chcel som vytvoriť obchod na prezeranie oblečenia, filtrovanie produktov, objednávanie nákupu. Taktiež som si zvolil za cieľ vytvoriť funkcionalitu registrácie a následného prihlasovania sa na stránku, ktorá bude mať pre prihlásených užívateľov samozrejme výhody. Tieto výhody pozostávajú z automatického vypísania kontaktných údajov pri kupovaní si tovaru. Uľahčenie nákupu, možnosť vidieť históriu svojich nákupov na stránke a taktiež možnosť zmeny osobných údajov zákazníka slúži na to profil zákazníka. Taktiež som sa zameral na to, aby mal administrátor rozšírené možnosti oproti zákazníkom a aby mohol spracovať objednávky a pridávať produkty zo svojho vlastného profilu. Taktiež má prístupný detail každej objednávky s údajmi, ktoré zákazník vyplnil.

Problematika vzniká v momente, keď sa pokúšate o prepojenie pomerne veľkého množstva databáz nadväzujúcich navzájom na seba. Potrebujete taktiež zabezpečiť komunikáciu so serverom. Je potrebné ovládať technológie, potrebné na fungovanie vašej technológie. Dôležité je tiež zvoliť si vhodné balíčky, vybrať si jazyk pre správnu funkčnosť. V mojom prípade som si vybral jazyk python, ktorý je podľa môjho názoru veľmi vhodný na webové aplikácie.

Pre webovú aplikáciu je tiež dôležitá responsibilita, to znamená, že je nutné, aby bola stránka vhodná pre obrazovky rôznych veľkostí. Ďalšou nutnosťou je rýchlosť načítania stránky, stránka by mala fungovať plynulo, bez sekania a načítavanie by nemalo trvať pridlho. Vhodná je aj prehľadnosť stránky, aby bola intuitívna a zákazník sa v nej dokázal jednoducho orientovať.

# Ciele práce

Mojím hlavným zámerom bolo vytvoriť funkčný internetový obchod, určený na nákup tovaru ľubovoľného druhu. Taktiež mi šlo o príjemné administrátorské prostredie, kde administrátor môže upravovať objednávky a pridávať produkty do svojho obchodu. Mojou úlohou teda bolo vytvoriť funkčnú databázu a prepojiť ju s python serverom. Ďalej zabezpečiť komunikáciu stránky a servera. Vytvoriť funkčnú konštrukciu a systém, ktorý bude medzi sebou komunikovať a plniť moje požiadavky.

V začiatkoch bolo potrebné vytvoriť modely a návrhy stránok, ktoré sa budú môcť následne používať. Vytvoriť funkčnosti ako je preposielanie dát pomocou formulárov. Uloženie cookies na stránku alebo nastavenie ich životnosti. Ďalej bolo potrebné vytvoriť prehľad produktov a detail produktov na prezeranie tovaru. Po úspešnom vytvorení podstránky bolo potrebné vytvoriť košík – miesto, kam sa vybrané produkty uložia a kde bude mať zákazník možnosť záväzne si objednať tovar. V košíku bolo potrebné ešte vytvoriť formulár s kontaktnými údajmi. Formulár by mal taktiež obsahovať „Spôsob dopravy“ a „Spôsob platby“.

Stránka by mala mať ešte dva ďalšie dôležité celky a to administrátorské prostredie, ktoré som už spomínal a užívateľské prostredie, ktoré zanedlho popíšem.

Administrátorské prostredie by sa malo skladať zo správy produktov a z objednávok. Správca môže jednoducho upravovať produkty, pridávať im obrázky, upravovať popis a podobne taktiež má prehľad o všetkých produktoch na stránke. Bude môcť vidieť aj detailný popis produktov. Druhá záložka, do ktorej sa dokáže dostať iba administrátor sú objednávky. Je to vlastne zoznam všetkých objednávok v obchode. V tabuľke by sa mali nachádzať Stav objednávky, cena a dátum kedy sa objednávka vykonala tieto informácie sú najdôležitejšie. Po kliknutí na ID objednávky bude administrátor presmerovaný na detail objednávky, kde sú podrobné údaje o objednávke.

Ako zákazník, tak aj správca budú mať prístup ku svojmu profilu. Budú sa v ňom nachádzať kontaktné údaje používateľov. Taktiež sa tu bude nachádzať možnosť zmeny hesla . Každému používateľovi, ktorý už v minulosti vykonal nejakú objednávku sa naskytne možnosť vidieť históriu svojich nákupov, stav objednávky a podobne po kliknutí na „Detail objednávky“ sa mu zobrazia podrobné údaje o objednávke.

Mojím zámerom je vytvoriť komplexný funkčný internetový obchod, ktorý bude spĺňať základné požiadavky ako používateľa tak aj administrátora.

# Teoretické východiská

## Flask

Flask je mikro web framework písaný pomocou Pythonu. Ako mikroframework je klasifikovaný pretože nevyžaduje nijaké konkrétne nástroje alebo knižnice. Neobsahuje v sebe žiadnu databázu, validátor a ani žiadne ďalšie komponenty tretích strán, komunity. No podporuje rozšírenia, ktoré pridávajú aplikácii vlastnosti správajúce sa ako keby boli implementované vo Flasku samotnom.

Mikroframework Flask je založený na projektoch z Pocoo a to z Werkzeug a Jinja2.

(1)

### Werkzeug

Werkzeug začal ako jednoduchá kolekcia rôznych nástrojov pre WSGI aplikácie a stala sa jednou z najpokročilejších modulov WSGI. Zahŕňa výkonný debugger, plne vybavené request a response objekty, HTTP nástroje na spracovanie tagov, hlavičky riadenia cache, http dátumy, spracovanie súborov cookie, odosielanie súborov, výkonný URL smerovací systém a množstvo doplnkových modulov. (2)

### Jinja

Jinja je šablónový prostriedok pre Python a je licencovaný pod BSD License a taktiež od Roacher Similiar pre Django. Django je podobný web framework s tým rozdielom, že využíva knižnice. Používa sa pri väčších projektoch. Na svoju prácu využíva šablóny, ktoré sú neoddeliteľnou súčasťou Flasku. (3)

## Backbone.js

Backbone.js dodáva štruktúru webovým aplikáciám poskytovaním modelov s väzbou na kľúčovú hodnotu a vlastných eventov, kolekcií s bohatým rozhraním pre programovanie aplikácií (API) s počítateľnými funkciami, pohľadmi určujúcimi spracovanie jednotlivých udalostí a spájaním tohto všetkého do vášho API cez JSON rozhranie. (4)

;

### Backbone.Model

Modely sú základom každej JavaScriptovej aplikácie, obsahujú ako interaktívne dáta tak aj veľkú časť logiky, ktorá obklopuje aplikáciu. Napríklad konvencie, validácie a výpočtové vlastnosti.

**Základné používané príkazy**

**extend** – na vytvorenie vlastnej triedy - Modelu musíte rozšíriť Backbone.Model a zadať mu vlastnosti inštancie.

**constructor** – Keď vytvárate inštanciu modelu, môžete odovzdať počiatočné hodnoty atribútov, ktoré budú vsadené do modelu. Ak definujete počiatočnú funkciu, vyvolá sa pri vytvorení modelu.

**get** - Získa aktuálnu hodnotu atribútu z modelu.

**set** – Do atribútu v danom modeli nastaví požadovanú hodnotu. Prípadne zmení hodnotu požadovaného atribútu.

**has –** vráti true ak má atribút nenulovú hodnotu.

**fetch** – Zlučuje stav modelu s atribútmi odobranými zo servera prenesením do Backbone.sync. Umožňuje komunikáciu medzi Backbone a serverom.

### Backbone.Collection

Kolekcie sú usporiadané sady modelov. Každý event ktorý sa spustí v modeli bude tak isto spustený priamo v kolekcii.

**Základné používané príkazy**

**extend** - Pri vytvorení vlastnej triedy kolekcie, rozšírite Backbone. Collection, ktorá obsahuje vlastnosti inštancie.

**model –** Prepísaním tejto vlastnosti, určíte triedu modelu, ktorú kolekcia obsahuje. Ak je definovaná, môžete preniesť objekty a polia raw atribútov, ktoré chcete pridať.

### Backbone.Router

Webové aplikácie často poskytujú prepojiteľné URL adresy pre dôležité lokácie v aplikácii. Až donedávna boli na prepojenie týchto URL používané fragmenty hash, ale s príchodom histórie API je teraz možné použiť štandardné URL.

**Základné používané príkazy**

**route –** Manuálne vytvorená trasa pre smerovač. Na zadanie cesty môžeme použiť string alebo regulárny výraz. Každé zhodné zachytenie z trasy bude odovzdané ako spätné volanie.

### Backbone.View

Hlavnou myšlienkou je usporiadať rozhranie do logických pohľadov (Views), podporovaných modelmi, z ktorých každý môže byť nezávisle aktualizovaný, keď sa model zmení, bez toho, aby ste stránku museli znova načítať.

**Základné používané príkazy**

**el –** Všetkypohľady majú vždy DOM (Dokument objektových modelov) element, či už boli vložené do stránky alebo nie. V tomto smere môžu byť pohľady renderované a vložené do DOM naraz, v záujme vysoko výkonného vykresľovania používateľského rozhrania s čo najmenej opravami.

**$** - ak je jQuery nalinkované, každý view má $ funkciu, ktorá spúšťa volanie v rámci view elementu.

**$el –** Objekt jQuery zachytený vo view elemente. Užitočný odkaz namiesto toho aby sa prvok v DOM opakoval.

**render** – základná implementácia renderovania v Backbone je, že nerobí vôbec nič, alebo tak málo že v podstatne neurobí nič. Prepísanie tejto funkcie vaším kódom, ktorý renderuje šablónu viewu z dátového modelu a aktualizuje ju s novým HTML. Dobrým zvykom je použiť return this na koniec renderovania.

## jQuery

Je to malá rýchla a na funkcie bohatá knižnica JS. Pomocou nej dokážeme manipulovať s HTML dokumentami v JS, naplánovať udalosti, animácie a taktiež Ajax oveľa jednoduchšie s používateľsky prívetivou API, ktorá funguje pre veľkú škálu prehliadačov. (5)

### Ajax

Umožňuje aktualizovať webovú stránku bez nutnosti znovu načítania. Požiada si o dáta zo servera, po tom čo bola stránka už načítaná. Obdrží dáta zo servera, po tom čo bola stránka už načítaná. Posiela dáta serveru z pozadia.

## Python

Python je interpretačný vysoko úrovňový jazyk pre všeobecné programovanie. Python si berie na dôraz čitateľnosť kódu a to hlavne za použitia výrazných medzier. Poskytuje konštrukcie, ktoré umožňujú jasné programovanie v malej aj vo veľkej škále. (6)

Python obsahuje dynamický typ systému a automatickú správu pamäte.

## JavaScript

Je vysoko-úrovňový jazyk a interpretačný programovací jazyk. Je to jazyk, ktorý je tiež charakterizovaný ako dynamický, prototypový a multi-paradigmatický.

JS umožňuje interaktivitu vo webových stránkach a je teda neoddeliteľnou súčasťou webových aplikácií. (7)

## Materialize

Vytvorený a navrhnutý spoločnosťou Google, Material Design je dizajnový jazyk, ktorý kombinuje klasické princípy úspešného dizajnu spolu s inováciami a technológiou. Cieľom spoločnosti Google je vyvinúť systém dizajnu, ktorý umožňuje jednotnú používateľskú skúsenosť vo všetkých svojich produktoch na akejkoľvek platforme. (8)

# Metodika práce

V prvom rade bolo potrebné vytvoriť spojenie s databázou, toto spojenie som realizoval pomocou súboru Main.py. V tomto súbore sa nachádzajú všetky mnou napísané funkcie, potrebné na fungovanie stránky. Ďalej som vytvoril modely v jazyku JavaScript, ktoré vytvoria zoznamy produktov alebo tabuľky s dátami z databázy. Následne bolo potrebné vytvoriť .html súbory a dotvoriť vzhľad pomocou materialize.css.

## Main.py

Pre správne fungovanie stránky bol potrebný takmer tisíc riadkový kód vyžadovaný na správnu komunikáciu so serverom. Využívam ho na vytváranie podstránok, prácu s databázou, vytváranie cookies. Bez súboru main.py by som nebol schopný komunikovať s databázou a ani so serverom.

Funkcia get\_db()

Pomocou tejto funkcie získavam spojenie na databázu SQLite, spojenie sa ukladá do globálnej premennej.

Funkcia close\_connection(*exception*)

Uzavriem pripojenie na databázu SQLite pri ukončení aplikácie.

Funkcia query\_db(*query*, *args*=(), *one*=False, *commit*=False)

Táto funkcia vykonáva SQL príkazy. Napríklad select, update, delete podľa parametrov, ktoré do nej vložím.

Funkcia prihlasenie\_nutne(*f*)

Pomocou funkcie prihlasenie\_nutne(f) zisťujem, či je používateľ prihlásený pomocou premennej login. Pokiaľ nie, zákazníka presmerujem na podstránku s prihlásením. Túto informáciu získavam z cookies kam som ju vložil v inej funkcii.

Funkcia zisti\_prihlasenie(*f*)

Funkciou *zisti*\_prihlasenie overujem dva parametre.

Za prvé, či je používateľ prihlásený. Podobným spôsobom ako funkcia prihlasenie\_nutne. Pokiaľ áno, do premennej \_prihlaseny zapíšem hodnotu true a do premennej \_login zapíšem hodnotu z request.cookies['login'].

Za druhé overím, rolu používateľa, či sa prihlasuje ako zákazník, alebo ako administrátor. Táto informácia je uložená v request.cookies['role']. Pokiaľ je v request.cookies['role'] hodnota 'ADMIN' do premennej \_je\_admin zapíše hodnotu true.

Funkcia vytvor\_session(*request*)

Session je podobné, no na rozdiel od cookies dáta sú uložené na serveri. Je to časový interval, počas ktorého môže byť klient prihlásený na serveri po vypršaní tohto intervalu je klient odhlásený. Dáta, ktoré sú potrebné aj po vypršaní danej session, som dočasne uložil v priečinku na serveri.

Funkcia vloz\_do\_kosika(*produkt\_id*, \**args*, \*\**kwargs*)

Funkciu vloz\_do\_kosika zabezpečím vloženie produktu do nákupného košíka. Ak je košík prázdny vytvorí objednávku a zapíšu sa do nej dáta konkrétneho produktu. Pokiaľ sa v objednávke už nachádza.

Pokiaľ existuje request.cookies['session\_id']

prihlasit\_put(\**args*, \*\**kwargs*)

Funkcia prihlasit\_put je put metóda. Má za úlohu vložiť nejakú informáciu, či už do databázy alebo do samotnej stránky. Do cookies na stránke vložím dve premenné a to login a role, ale iba za predpokladu, že po zavolaní tejto metódy sú vyplnené správne políčka vo formulári. Pokiaľ nie funkciou odošlem správu „neprihlásený“. Nastaví taktiež životnosť cookies.

odhlasit\_get()

Jednoducho iba nastavím životnosť cookies na nulu a tým ich vymažem. Po vymazaní nás presmeruje na hlavnú stránku – prehlad.html.

zobraz\_kosik2( \**args*, \*\**kwargs*)

Funkciou preberám viacero hodnôt, či už z JavaScriptu pomocou JSONu alebo z cookies. Vyberám hodnoty produktov (cena, názov a podobne) z databázy. Pracujem s údajmi zákazníka. Jeho úlohou je aktualizovať dáta o objednávke pridať spôsob dopravy a spôsob platby, zmeniť stav objednávky na obj – objednané.

Funkcia vrat\_produkty\_json()

Vytvoril som ju na princípe, že vyberie všetky produkty z databázy, údaje vloží do poľa a pretransformuje ich do univerzálneho jazyka JSON.

Funkcia zobraz\_detail\_objednavky(*objednavka\_id* = None, \**args*, \*\**kwargs*)

Pokiaľ potrebuje admin nájsť kontaktné údaje zákazníka, ktorý si objednal konkrétny produkt urobí tak pomocou tento metódy, ktorá vyberá dôležité údaje o zákazníkovi.

Funkcia put\_obrazok\_nastav\_hlavny(*obr\_id* = None, \**args*, \*\**kwargs*)

V editácii produktov mám možnosť pomocou jedného kliknutia nastaviť obrázok ako hlavný. Zabezpečujem to pomocou tejto funkcie, ktorá po každom kliknutí nahrá označený obrázok na hlavný a ostatné nahrá na vedľajšie.

Funkcia get\_produkt\_editacia(*produkt\_id* = None, \**args*, \*\**kwargs*)

Za predpokladu, že existuje aspoň jeden produkt v databáze pomocou funkcie vyberiem z databázy údaje pridelené jednému konkrétnemu produktu. Následne funkcia vykreslí stránku produkt\_editacia.html aj s id produktu, ktorý meníme a údajmi, ktoré produkt má ako napríklad cena, názov, popis a tak ďalej.

put\_registracia(\**args*, \*\**kwargs*)

Z JSONu najprv získam údaje, ktoré zákazník uviedol vo formulári. Potom overím, či sa používateľské meno už nepoužíva. Pokiaľ nie, vytvorím nový záznam s príslušným meno, heslom, id a rolou zákazník. Po registrácii prihlásim zákazníka. Pokiaľ sa neodhlási jeho prihlásenie bude aktívne 2 hodiny a potom ho odhlási samo, vymažú sa jeho cookies.

Funkcia get\_profil(\**args*, \*\**kwargs*)

Pomocou funkcie get\_profil zisťujem údaje používateľa, podľa jeho mena. Následne tieto údaje posielam do kwargs, aby mohli byť použité vo formulári.

Funkcia put\_kontakt\_zmena(\**args*, \*\**kwargs*)

Vyberiem si údaje používateľa, ktoré sa nachádzajú v JSONe prihlásený používateľ ich pravdepodobne zmenil. Vložím tieto pozmenené údaje do databázy a odošleme odpoveď o úspešnej zmene údajov.

Funkcia put\_sprava()

Odosielam správy z pätičky na databázu. Zapíšem do databázy email a správu, ktorú používateľ odošle pomocou pätičky. Nie je to skutočné posielanie emailu. Správa sa zapíše do databázy.

## Global\_model.js

V tomto súbore vytváram všetky modely. Pridávam im premenné, ktoré môžu následne používať.

Backbone model- ProduktModel

Vytvorí predpis s názvami premenných, ktoré sa následne musia používať, dá sa chápať ako trieda v jave.

Backbone collection KategoriaZoznam

Je to už kolekcia dát, ktorá používa ako svoj predpis model KategoriaModel, odkaz na tieto dáta je /kategorie.

globalModel

V globalModel vytvárame, už konkrétne objekty podľa toho, akú kolekciu potrebujeme. Tento objekt môžeme už používať a vložiť do neho konkrétne hodnoty.

## Kosik\_model.js

V súbore kosik\_model generujem produkty, ktoré zákazník zložil do košíka.

Funkcia onPlusMinusClick

Zjednodušene vlastne iba odpočítava alebo pripočítava počet produktov bez toho, aby bolo po každej zmene potrebné aktualizovať stránku, aby sa zmeny prejavili. Pokiaľ sa počet konkrétneho produktu dostane na nulu, jednoducho ho odstránime z nákupného košíka.

Backbone view- KosikZoznamView

Tento view je určený na to, aby sme mohli generovať ľubovoľný počet riadkov tabuľky v nákupnom košíku, konkrétne sa jedná o produkty, ktoré si zákazník vybral a vložil do nákupného košíka. Zobrazuje informácie o konkrétnom produkte, ktoré sa nachádzajú v databáze.

## Obrazky\_model.js

V súbore obrazky\_model.js generujem obrázky, ktoré sa nachádzajú v priečinku obrázkov.

Backbone view- ObrazkyZoznamView

Zobrazuje štyri náhodné obrázky produktov. Funkcia je určená na zaujatie zákazníka. Využíva funkciu random. Vytvorí odkaz s obrázkom, ktorý vás presmeruje na prehľad konkrétneho produktu.

## Prehlad\_model.js

V súbore prehlad\_model generujem filtrovaný zoznam, ktoré sa nachádzajú v databáze.

Backbone view- KategoriaZoznamView

Tento view je potrebný na filtrovanie produktov podľa kategórie. Vyberie všetky kategórie, ktoré sa nachádzajú v databáze a vygeneruje ich na stránku. Podľa id filtruje produkty v ponuke.

Backbone view- ProduktZoznamView

Toto je zobrazenie vyfiltrovaných produktov. Funkcia vygeneruje iba produkty a ku nim príslušné atribúty iba ak sa ich Id kategórie zhoduje s id, ktorú si vybral zákazník.

## Produkt\_model.js

V súbore prehlad\_model generujem detail produktov, vytvoril som v ňom informácie o cene produktu, obrázku, popise a podobne.

Funkcia vlozDoKosika

Táto funkcia požaduje hodnotu produkt\_id. Pomocou jQuery zistí obr\_id zabalí toto id do JSONu a odošle v balíčku dát ako samostatnú premennú. Po úspešnom vykonaní zákazníka presmeruje na domovskú stránku.

Backbone view- ProduktDetailView

Po kliknutí na produkt v prehľade zabezpečí generáciu detailu o produkte pomocou dopredu definovanej šablóny. Keďže informácie z databázy vyberáme podľa id v url stránky v prípade id, ktoré v databáze nie je, vypíšeme chybovú hlášku.

Backbone view- GaleriaObrazkyView

V detaile produktov vygeneruje všetky obrázky, ktoré zodpovedajú danému produktu. Zobrazí ten istý produkt, zle z iných pohľadov, prípadne rozdielne farby produktu.

## Kosik.html

Kosik.html som využil na vytvorenie podstránky, kde sa nachádza formulár a tabuľka s produktami.

Funkcia odosliObjednavku

Po niekoľkých kontrolách odošle hodnoty, ktoré sú vložené vo formulári v tvare JSONu. Po úspešnom vykonaní tejto operácie, zabije cookie, ktoré bolo zodpovedné za objednávku. Pri ďalšom nákupe dostane zákazník nové cookie s novou hodnotou.

Funkcia zobrazFormular

Štandardne je formulár vypnutý, až po tom čo sa zákazník rozhodne ukončiť nákup a odoslať objednávku, po tom čo klikne na tlačidlo objednať sa nastaví element formulára na „visible“ a zobrazí sa celý formulár.

Funkcia onPlusMinusClick

Zaznamenáva id položky a zmenu, ktorú zákazník vykonal, či sa rozhodol pridať tovar alebo odobrať tovar. Po úspešnom vykonaní tejto funkcie sa stránka nenápadne znovu načíta. Je to put metóda, čiže predáva hodnoty pol\_id a zmena ďalej.

## Prihlasit.html

Je vytvorená na prihlasovanie sa zákazníka do svojho profilu.

Funkcia akcia\_prihlasit

Z formulára vyťahuje pomocou jQuery prihlasovacie meno a heslo. Následne ich prekóduje na tvar JSONu a put metódou odošle ďalej. Po úspešnom vykonaní sa presmeruje na domovskú stránku.

## Produkt\_editacia.html

Vytvoril som ju pre potrebu editovania produktov.

Funkcia nastav\_hlavny

Je to jednoduchá funkcia, ktorá nastavuje hlavný obrázok. Hlavný obrázok je ten, ktorého id sa odosiela v tejto funkcii.

$("#form-obrazok-upload").submit(*function*(*event*)

Pomocou tejto funkcie nahráme obrázky na server, konkrétne do priečinku img/velke a zapisujeme názov obrázku na obrázok samotný a aj do databázy.

$("#form-produkt").submit(*function*(*event*)

Preberá hodnoty ako názov, popis, kategória, cena a posiela ich pomocou JSONu na server.

## Profil.html

Vytvoril som ju na spravovanie profilu používateľa.

Funkcia akcia\_kontaktne\_udaje\_zmena

Kontroluje všetky osobné kontaktné údaje pre prípad nezrovnalostí. Kontrola funguje na báze minimálneho počtu znakov. Pokiaľ zistí, že je počet znakov menší než je povolené vypíše chybovú hlášku a núti užívateľa zadať napríklad dlhšiu prezývku. Po tom, čo sú údaje správne odošle ich na server pomocou JSONu.

Funkcia akcia\_profil\_zmena

Taktiež kontroluje dĺžku údajov, pokiaľ je kratšia vypíše chybovú hlášku. Umožňuje meniť heslo používateľa aj s nutnosťou zopakovať heslo. Heslá sa následne porovnajú a uložia iba vtedy pokiaľ sa zhodujú. Pokiaľ nie, užívateľ je vyzvaný heslo napísať znovu.

## Registrovat.html

Vytvoril som ju na registráciu zákazníka do internetového obchodu.

Funkcia validacia\_povinny\_udaj

Je určený na kontrolu dĺžky údajov. Do tejto funkcie sa zadáva premenná, ktorá má byť skontrolovaná, chybová hláška a minimálny počet znakov.

Funkcia akcia\_registracia

Funkcia slúži na registráciu nových zákazníkov. Je vyžadované prihlasovacie meno, heslo a potvrdenie hesla. Po úspešnom vyplnení formulára a následnom overení dĺžky hesiel sa na server odošle správa v tvare JSON so zadanými údajmi. O úspešnom zaslaní nás informuje hláška „Registrácia bola úspešná!“

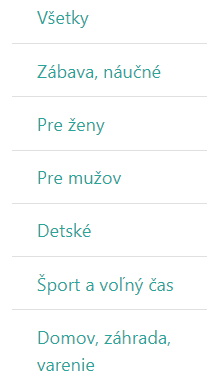
# Výsledky práce

## Hlavná stránka – prehľad

**Obrázok 1** Prehľad

**Obrázok 2** Hlavná stránka

Na prvotnej hlavnej stránke môžete vidieť prehľad všetkých produktov, ktoré máme k dispozícii. Každý takýto produkt sa skladá z obrázku produktu, jeho názvu a ceny. Tieto údaje sú volané z databázy. Podľa počtu produktov v databáze vytvorí rovnaký počet produktov na stránke. Podmienkou je, aby bol produkt v stave ‘A’ aktívny. Obrázok v každého produktu sa vyberá podľa hlav\_obr, ak je jeho hodnota 1 je to hlavný obrázok. Po prejdení myškou cez produkt (obrázok, nadpis, cena) sa vám naskytne možnosť odkazu do detailu produktu. V prípade potreby ďalších informácií o produkte stačí kliknúť na tri zvislé bodky. Po kliknutí na zvislé bodky sa vám zobrazí popis produktu, bez toho aby bolo potrebné prejsť na detail produktu.



**Obrázok 3** Filter

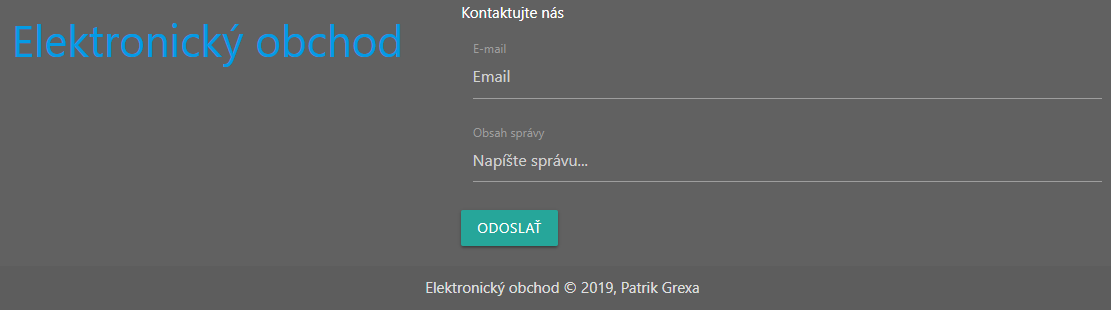
Po ľavej strane stránky je umiestnené filtrovanie produktov. Kategórie sa skladajú zo všetkých produktov a jednotlivých druhov magazínov podľa obsahu. Na filtrovanie sa využíva databáza. Každý produkt je prepojený s tabuľkou tab\_kat pomocou jedinečného identifikátora. Ten určuje kategóriu. v ktorej sa bude produkt nachádzať. Stránka sa snaží byť intuitívna a prístupná.



**Obrázok 4** Navigácia

Na vrchu stránky je taktiež niekoľko odkazov, ktoré smerujú na ďalšie podstránky, nachádzajú sa v každej podstránke a slúžia ako navigácia. V tejto časti stránky je taktiež názov obchodu. Stránka je responzívna a po zmenšení obrazovky sa navigácia premení na mobilnú verziu. Celé menu sa bude nachádzať na ľavej časti. Bude fungovať ako posuvník,

Na spodnej časti sa nachádza pätička obsahuje odkaz na domovskú stránku – prehľad, názov obchodu. Má v sebe zahrnutú funkciu odoslať správu do databázy, či už v prípade nespokojnosti, dobrej referencie alebo návrhu na zlepšenie fungovania internetového obchodu. Táto pätička sa nachádza v prehľade, detaile produktu a taktiež v nákupnom košíku. Odošle správu na server a následne do databázy.



**Obrázok 5** Pätička

## Detail produktu



V detaile produktu sa nachádza ľavý bok. Neobsahuje, ale už filtrovanie kategórií. Tiež obsahuje hlavičku s tlačidlami a pätičku na kontaktovanie - posielanie správ.

Detail je rozdelený na dve časti pravú a ľavú. Ľavá strana sa skladá z obrázku produktu.

Pravá časť obsahuje názov produktu, popis produktu, cenu produktu a tlačidlo na pridanie produktu do košíka. Túto funkčnosť som zvolil pre väčší komfort zákazníka a taktiež ako možnosť pre rozšírenie množstva produktov. Pod obrázkami sa nachádza popis produktu, ten je taktiež vytiahnutý z databázy. Pod textom v pravej časti sa nachádza už len tlačidlo “Pridaj do košíka”. Po kliknutí na toto tlačidlo sa do košíku pridá záznam o danom produkte a zákazníka odkáže na prehľad – hlavnú stránku pre prístup ku ďalšiemu tovaru. Záznam sa skladá z miniatúry obrázku, názvu produktu jeho ceny a počtu.



**Obrázok 6** Balíček produktov

V balíčku produktov sa môže nachádzať viacero produktov. Prezrieť si ich môžete kliknutím na obrázok. Po kliknutí sa vám zobrazí zväčšenina obrázku.



**Obrázok 7** Reklama

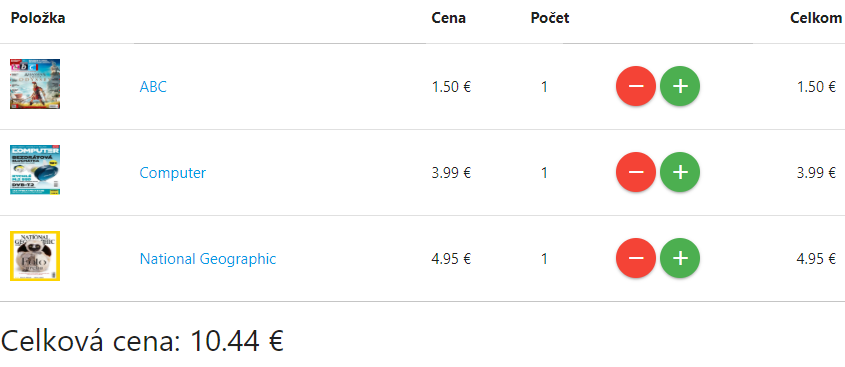
Pod informáciami o produkte je vytvorený zoznam štyroch náhodných produktov ako zaujímavé položky. Po každom aktualizovaní stránky sa vyberú štyri iné produkty. Fungujú ako odkazy na produkty na obrázku. Pod takýmto zoznamom je ešte pätička.

## Nákupný košík

Podstránka s nákupným košíkom je tvorená z vopred vytvoreného návrhu, v ktorom sa nachádzajú tlačidlá navigácie ako hlavička a pätička pre odosielanie správ s kontaktným údajom - email. Pod nákupným košíkom sa bude nachádzať vopred vytvorený návrh so zoznamom štyroch náhodných produktov. Pre rýchly prístup ku ďalšiemu tovaru.

## Nákupný košík

V prípade, že je nákupný košík prázdny, vytvorí sa oznámenie o tom, že si zákazník do košíka ešte nič nevložil.

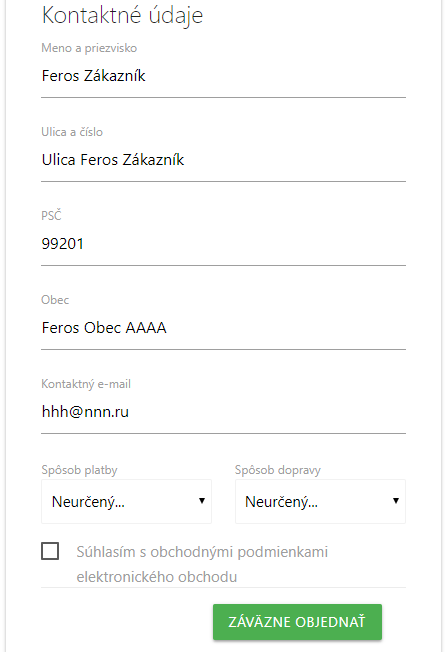


**Obrázok 8** Nákupný košík

Nákupný košík je tvorený z tabuľky obsahujúcej stĺpce obrázka, názvu produktu, ceny za jeden kus, počtu produktu, pripočítania alebo odpočítania počtu, a z celkovej ceny. Cena za všetky produkty je následne zobrazená ako posledný riadok v tabuľke pod názvom celková cena. V tom istom riadku sa nachádza ešte tlačidlo objednať, ktoré je umiestnené na pravej strane riadku a po kliknutí odkryje formulár na objednanie produktov.

## Formulár

Formulár obsahuje informácie potrebné pre zaslanie objednávky a to z :

Mena a priezviska kupujúceho

Adresy jeho bydliska

PSČ

Obce v ktorej má trvalý pobyt

Spôsob platby, a to Neurčený, Dobierka, Prevodom na účet, Zrýchlená platba.

Spôsobu dopravy, spôsoby sú nasledovné: Neurčený, Osobný odber v hodnote 0 €, Slovenskou poštou v hodnote 3 € a Kuriérom v hodnote 5 €.

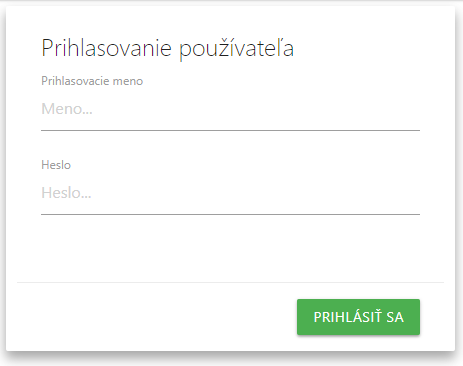
Na konci formulára sa bude nachádzať súhlas s obchodnými podmienkami nášho obchodu a pod tlačidlom Záväzne objednať. Toto tlačidlo odošle INSERT do databázy s vyplnenými údajmi.

**Obrázok 9** Odoslanie objednávky

## Prihlásiť sa

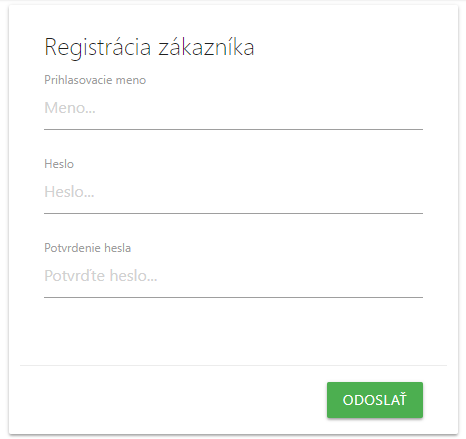
**Obrázok 10** Prihlásiť tlačidlo

Podstránka prihlásiť sa je tvorená z vopred vytvoreného návrhu, v ktorom sa nachádzajú tlačidlá navigácie a pätička pre odosielanie správ s kontaktným údajom - email.



Časť pre prihlasovanie tvorí nadpis „Prihlásenie používateľa“ pod ním sa nachádzajú dve políčka prvé je pre prihlasovacie meno druhé je pre heslo. Pod políčkami je tlačidlo Prihlásiť sa pre prihlásenie užívateľa. V databáze sú všetky prihlasovacie údaje. Pri zhode mena a hesla prebehne úspešné prihlásenie overuje sa taktiež rola používateľského účtu. Účet môže mať rolu zákazníka, alebo administrátora. Ak sa prihlasuje administrátor po prihlásení a presmerovaní na prehľad sa mu v navigácii zobrazia ďalšie funkcionality, ktoré sú pre obyčajného zákazníka neprístupné. Tlačidlo prihlásiť sa, sa zmení na tlačidlo Odhlásiť sa, po kliknutí na toto tlačidlo sa prihlásený užívateľ odhlási. Po odhlásení sa v stránke vymažú cookies, ktoré držali informácie o prihlásení a zablokuje prístup na administrátorské podstránky, na profil používateľa a aj na editáciu produktov.

**Obrázok 11** Prihlásenie



**Obrázok 12** Registrácia

## Registrácia používateľa

Registrácia zákazníka je veľmi jednoduchá. Stačí vyplniť prihlasovacie meno, zadať heslo a potvrdiť ho v ďalšom stĺpčeku. Týmito niekoľkými krokmi sa vykoná registrácia. Pre úplné vyplnenie profilu je potrebné ešte vypísať kontaktné údaje v profile zákazníka.

## Profil používateľa

**Obrázok 13** Kontaktné údaje zmena



**Obrázok 14** Heslo zmena

Profil používateľa slúži na zobrazenie osobných údajov zákazníka, ale aj administrátora. Umožňuje meniť ako heslo, tak aj kontaktné údaje zákazníka pre prípad, že by sa napríklad presťahoval, ale stále by chcel využívať služby nášho obchodu. Údaje sú rozdelené do dvoch formulárov. Prvý obsahuje login, heslo a políčko pre potvrdenie hesla. Druhý obsahuje osobné kontaktné údaje. V prípade, že zákazník už vykonal na svojom účte objednávku pod formulármi je zobrazená tabuľka s históriou jeho nákupov. Po kliknutí na „Detail objednávky“ sa mu zobrazí podrobný výpis produktov, ktoré si kúpil. Taktiež si môže nájsť informácie, na ktorú adresu sa nákup zaslal. Prípadne v akom stave je objednávka. Pred nákupom je potrebné vyplniť kontaktné údaje v profile.



**Obrázok 15** História objednávok

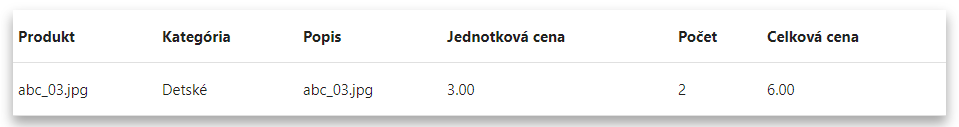
## Objednávky

Administratívne rozhranie umožňuje pozorovať všetky objednávky v databáze. Stav konkrétnych objednávok, celkovú sumu objednávky, jej id a dátum objednania. Nachádza sa tu jednoduchá tabuľka s atribútmi Id, Stav objednávky, Cena a Dátum a čas. Pre detailné zobrazenie objednávky je potrebné kliknúť na jej id, to vás zavedie na detail objednávky. Tabuľka sa snaží byť prehľadná a ľahko pochopiteľná. 

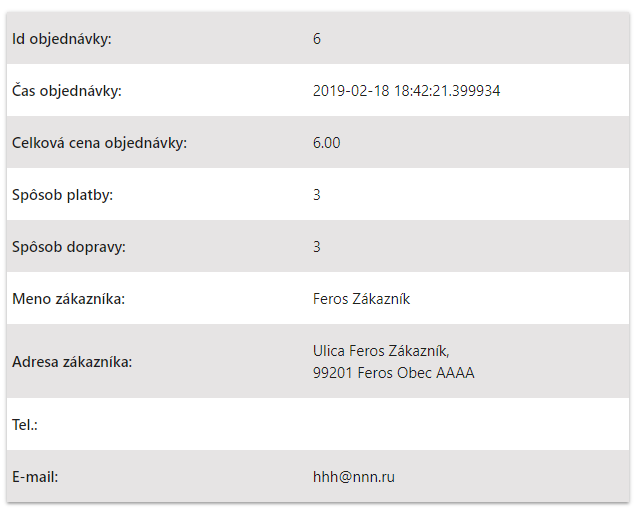
**Obrázok 16** Objednávky

## Detail objednávky

V detaile sa nachádzajú dve tabuľky. V jednej sú detailné údaje o objednávke a to Id objednávky, čas objednania, Celková cena, Spôsob platby, Spôsob dopravy, Meno a priezvisko zákazníka, Adresa zákazníka, Telefónne číslo (dobrovoľné) a Emailová adresa. V druhej tabuľke sa nachádzajú informácie o jednotlivých produktoch danej objednávky. Tabuľka sa skladá z názvu produktu, kategórie produktu, popisu produktu, ceny za kus, počtu a z celkovej ceny produktu.



**Obrázok 17** Údaje produktu

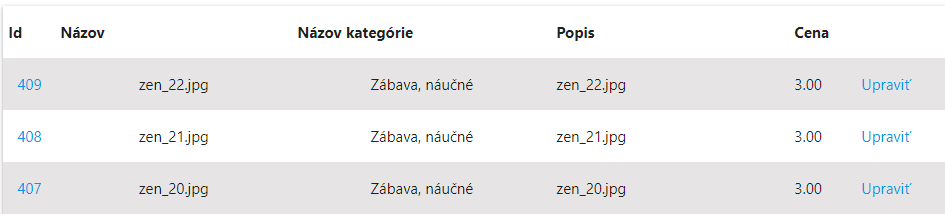


**Obrázok 18** Údaje zákazníka

## Zoznam produktov pre editáciu

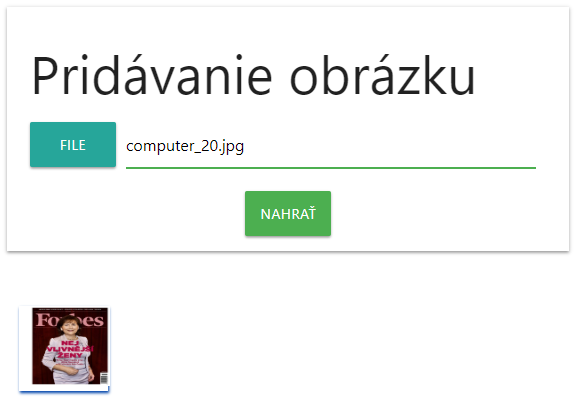
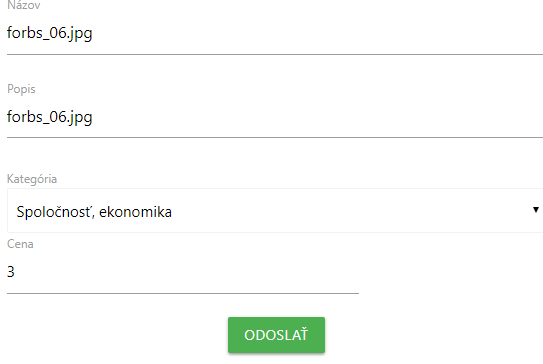
Prihlásenie sa pod účtom administrátora ponúka ešte možnosť pridávať nové produkty a editovať už existujúce. Na tejto podstránke je zobrazený zoznam všetkých produktov. Nachádza sa tu ID produktu, Názov produktu, Kategória produktu, Popis produktu a cena produktu. Po kliknutí na Id produktu nás presmeruje na detail produktu. Každý riadok má možnosť upraviť produkt. Po kliknutí na odkaz sme presmerovaný na podstránku Editácia/pridávanie produktu, kde máme možnosť vykonať zmeny.

**Obrázok 19** Pridať produkt



**Obrázok 20** Zoznam produktov admin

## Produkt editácia

Na tejto podstránke na nachádza formulár s údajmi o produkte, ktorý sme si v predchádzajúcom kroku vybrali. Máme možnosť meniť názov produktu, popis, jeho kategóriu a cenu. Sú to základné veci, ktoré sú potrebné. Ďalej máme možnosť nahrať na server obrázok, ktorý bude následne priradený konkrétnemu produktu. Obrázkov môže byť ľubovoľný počet. Vždy uvidíme aj celý názov obrázku, ktorý chceme vložiť. Obrázky pridelené ku produktu sú zobrazené pod „pridávaním obrázkov“ . Jeden z obrázkov je vždy výnimočný pretože je označený ako hlavný obrázok, tento obrázok sa zobrazuje v prehľade a je základne zobrazený aj v detaile produktu. Meniť hlavný obrázok je jednoduché stačí kliknúť na ľubovoľný obrázok pomocou myšky a obrázok sa zmení na hlavný. O tejto zmene vieme tak, že hlavný obrázok bude podfarbený modrou farbou***.***

**Obrázok 21** Produkt editácia

**Obrázok 22** Pridávanie obrázku

# Záver

## Stručné zhodnotenie

Vytvoril som responzívny web určený na predaj ľubovoľného tovaru zákazníkom priamo domov. Ponúkol som možnosť prehľadu všetkých produktov a následnému detailu jednotlivých produktov. Vytvoril som košík s vybranými produktami a formulár s kontaktnými údajmi zákazníka. Objednávku som odoslal na server a údaje putovali následne do databázy, kde sa spracovali a uložili do tabuliek. Aktívne som pracoval s databázou a uľahčil prácu s údajmi, ako pre zákazníka, tak aj pre administrátora. Vytvoril som praktické užívateľské prostredie na prehľad profilu na spracovanie objednávok a na úpravu produktov. Administrátor si informácie nemusí dohľadať v databáze, ale stačí mu upraviť dáta na stránke.

## Význam práce

Internetové obchody sú v súčasnosti neoddeliteľnou súčasťou internetu. Sú to podnikateľské možnosti aj pre obyčajných ľudí, ktorý chcú skúsiť niečo nové. Veľkou výhodou takéhoto obchodu je aj to, že nepotrebujete veľký kapitál. Stačí znalosť anglického jazyka a všetky výrobky, ktoré ponúkate vám môžu vyrobiť napríklad aj v Číne. Potreba internetových obchodov stále rastie a tento trend bude v najbližších rokoch určite pokračovať. Je preto potrebné učiť sa a vytvárať nové dizajny a nové koncepty internetových obchodov. Od tých najjednoduchších po zložitejšie, ktoré jeden človek nedokáže spravovať a je na to určený celý tím ľudí.

## Možnosti využitia v praxi

Stránka je síce plne funkčná, no určite nie je dokonalá. Na uvedenie stránky do ostrej prevádzky by bolo určite potrebné lepšie zabezpečenie. Lepšie šifrovanie údajov, aj odolnosť voči hackerským útokom. Je iste množstvo vecí, ktoré by bolo potrebné vylepšiť. Teším sa na možnosť pokračovať na svojom projekte vo voľnom čase. Získal som vďaka nemu množstvo skúseností, ktoré ako verím pretavím aj v budúcnosti do skutočne funkčnej stránky, ktorá bude spĺňať všetky požiadavky na profesionálnu webovú aplikáciu odolnú voči útokom.

## Získané poznatky, zlepšenie stránky

Ako som už načrtol, na stránke by som chcel vo voľnom čase pokračovať. Doplniť ju o platbu kartou dostatočnú bezpečnosť. Rád by som do nej pridal aj ďalšie možnosti pre administrátora ako je zmena stavu objednávky v prípade, že je objednávka už vybavená, prípadne počet kusov na sklade. Možností internetového obchodu je neúrekom a záleží iba na nás do akej miery ho chceme zdokonaľovať.

# Zoznam použitej literatúry

1. **Flask (web framework). *Wikipédia.* [Online] [Citace: 20. 2 2019.] https://en.wikipedia.org/wiki/Flask\_(web\_framework).**

**2. Werkzeug. *Pocoo.* [Online] [Citace: 20. 2 2019.] http://werkzeug.pocoo.org/.**

**3. Jinja. *Pocoo.* [Online] [Citace: 20. 2 2019.] http://jinja.pocoo.org/.**

**4. Backbone.js. *Backkbone.js.* [Online] [Citace: 20. 2 2019.] https://backbonejs.org/.**

**5. jQuery. *jQuery.* [Online] [Citace: 20. 2 2019.] https://api.jquery.com/.**

**6. Python (programming language). *Wikipédia.* [Online] [Citace: 20. 2 2019.] https://en.wikipedia.org/wiki/Python\_(programming\_language).**

**7. JavaScript. *Wikipédia.* [Online] [Citace: 20. 2 2019.] https://en.wikipedia.org/wiki/JavaScript.**

**8. Materializecss. *Materialize.* [Online] [Citace: 20. 2 2019.] https://materializecss.com/about.html.**

# Zoznam príloh

[Príloha A Vzorový zdrojový kód Python 41](#_Toc1575227)

[Príloha B Vzorový zdrojový kód JavaScript 41](#_Toc1575228)

[Príloha C Vzorový kód Prehlad\_model.js 42](#_Toc1575229)

[Príloha D Vzorový kód prehľad.html 43](#_Toc1575230)

[Príloha E Vzorový kód Filter.html 43](#_Toc1575231)

[Príloha F Vzorový kód SQL 43](#_Toc1575232)

[Príloha G Diagram SQL hierarchia 44](#_Toc1575233)

# Prílohy

Príloha A Vzorový zdrojový kód Python

@app.route("/kategorie", *methods*=['GET'] )

*def* vrat\_kategorie\_json():

SQL = '''

select kat\_kod, kat\_nazov from cis\_kategorie

'''

return jsonify(*collection*=[mapuj\_stlpce(('id','kategoria\_nazov'),i) for i in query\_db(SQL)])

pass

Príloha B Vzorový zdrojový kód JavaScript

*var* KategoriaModel = Backbone.Model.extend({

defaults: {

id: '',

kategoria\_nazov: '' /\* <-- názov kategórie \*/

},

url: "/kategorie"

});

*var* KategoriaZoznam = Backbone.Collection.extend({

model: KategoriaModel,

url: "/kategorie",

parse: parseCollection

});

*var* globalModel = {

//---------------------------------------

// VYTVORIME OBJEKTY V NAMESPACE

//---------------------------------------

kategoriaZoznam : new KategoriaZoznam([]),

getURLParameter : *function* (*sParam*) {

*var* sPageURL = window.location.pathname;

*var* sURLVariables = sPageURL.split('/');

return sURLVariables[sURLVariables.length - 1];

}

}

*var* jeNacitanyGlobalModel = false;

$(document).ready(*function*() {

jeNacitanyGlobalModel = true;

jeNacitanyGlobalModel = true;

globalModel.kategoriaZoznam = new KategoriaZoznam([]);

});

Príloha C Vzorový kód Prehlad\_model.js

*var* KategoriaZoznamView = Backbone.View.extend({

el: '#kategoria-filter',

initialize: *function* () {

this.render();

},

render: *function* () {

this.$el.html('');

this.$el.append('<a id="-1" href="#" class="collection-item kategoria-item">Všetky</a>');

globalModel.kategoriaZoznam.each(*function* (*model*) {

this.$el.append(

'<a id="'

+ model.get('id') + '" href="#" class="collection-item kategoria-item '

+ ((kategoriaFilter == parseInt(model.get('id'))) ? 'active' : '')

+ ' ">' + model.get('kategoria\_nazov')

+ '</a>');

}.bind(this));

return this;

}

});

Príloha D Vzorový kód prehľad.html

<main class="row">

{% include "filter.html" %}

<div class="col s12 m8 l10">

<div id="zoznam-produktov" class="row">

<!-- tu je miesto pre zoznam produktov, ktory vyuziva kategorie-->

</div>

</div>

</main>

Príloha E Vzorový kód Filter.html

<div id="kategoria-filter" class="collection col s12 m4 l2">

</div>

Príloha F Vzorový kód SQL

INSERT INTO tab\_obrazky (obr\_id,obr\_subor,pro\_id,hlav\_obr) VALUES (388, 'zen\_01.jpg', 388,1);

INSERT INTO tab\_obrazky (obr\_id,obr\_subor,pro\_id,hlav\_obr) VALUES (389, 'zen\_02.jpg', 389,1);

INSERT INTO tab\_obrazky (obr\_id,obr\_subor,pro\_id,hlav\_obr) VALUES (390, 'zen\_03.jpg', 390,1);

Príloha G Diagram SQL hierarchia

