Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

**PC KOMPONENTE**

Strokovno poročilo

Mentor: Darjan Toth, prof. Avtor: Sebastjan Vidergar, R 4. C

Ljubljana, april 2022

Povzetek

Strokovno poročilo vsebuje opise in postopke izdelave spletne trgovine in konfiguratorja ter pojasnjuje nekatere odločitve za način izdelave. V spletni trgovini se prodaja računalniške komponente, katere se lahko nato uporabi v svetovalcu za sestavo računalnika (konfigurator). Izdelek vsebuje podatkovno bazo MariaDB, spletno trgovino in konfigurator, ki sta napisana v jezikih PHP, Javascript in HTML.

**Ključne besede:** spletna trgovina, konfigurator, javascript, jQuery, MySQL, MariaDB, naročilo, košarica

Vsebina

[1. Uvod 4](#_Toc100853433)

[2. Struktura podatkovne baze 5](#_Toc100853434)

[2.1. Kategorije 5](#_Toc100853435)

[2.2. Izdelek 6](#_Toc100853436)

[2.3. Naročilo 7](#_Toc100853437)

[2.4. Uporabnik 8](#_Toc100853438)

[2.5. Konfigurator 9](#_Toc100853439)

[3. Struktura datotečnega sistema 10](#_Toc100853440)

[3.1. Ločitev map: 10](#_Toc100853441)

[4. Uporabnik 12](#_Toc100853442)

[4.1. Registracija 12](#_Toc100853443)

[4.2. Prijava 13](#_Toc100853444)

[4.3. Ponastavitev gesla 13](#_Toc100853445)

[4.4. Administrativni uporabnik 14](#_Toc100853446)

[5. Trgovina 15](#_Toc100853447)

[5.1. Kategorije 15](#_Toc100853448)

[5.2. Iskalnik 16](#_Toc100853449)

[5.3. Filtriranje 16](#_Toc100853450)

[5.4. Prikaz izdelkov 17](#_Toc100853451)

[6. Košarica 18](#_Toc100853452)

[6.1. Delovanje 18](#_Toc100853453)

[6.2. Prikaz 19](#_Toc100853454)

[7. Naročilo 20](#_Toc100853455)

[7.1. Naslov 20](#_Toc100853456)

[7.2. Plačilo 21](#_Toc100853457)

[7.3. Pošiljatelj 22](#_Toc100853458)

[7.4. Zaključek 22](#_Toc100853459)

[8. Generator podatkov 23](#_Toc100853460)

[8.1. Potek delovanja 23](#_Toc100853461)

[8.2. Slike 24](#_Toc100853462)

[9. Konfigurator 25](#_Toc100853463)

[9.1. Hiter pregled delovanja 26](#_Toc100853464)

[9.2. Prikaz komponent 26](#_Toc100853465)

[9.3. Drevo odločanja 27](#_Toc100853466)

[9.4. Stanje konfiguracije 27](#_Toc100853467)

[9.5. Seštevalnik 27](#_Toc100853468)

[9.6. Prikaz stanja 28](#_Toc100853469)

[9.7. Poraba 28](#_Toc100853470)

[9.8. Shranjevanje konfiguracije 28](#_Toc100853471)

[9.9. Izboljšave 28](#_Toc100853472)

[10. Zaključek 29](#_Toc100853473)

[11. Viri 30](#_Toc100853474)

1. Uvod

Za svoj maturitetni izdelek sem si izbral izdelavo spletne trgovine in konfiguratorja za računalnik. Za to sem se odločil, ker sem vedel, da mi bo znanje javascripta in PHP-ja zelo pripomoglo pri rednem pouku. Moj cilj je bil, da naredim delujočo spletno trgovino, kjer bo uporabnik lahko opravil naročilo, dodal izdelke v košarico, lahko iskal izdelke po kategorijah ali imenu, in da imam delujoč konfigurator. Slednji deluje nad pričakovanji, saj deluje za raznolike izdelke in je zelo hiter. Pri tem sem dodal še nadgradnjo originalni ideji, in sicer shranjevanje konfiguracije v bazo in dinamično spreminjanje presežka napajalnikove moči. Dodatek izdelku je bil tudi generator podatkov, ki lahko ustvari naključne izdelke. Slednji so uporabljeni za prikaz v tem izdelku.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 1: Logotip spletne trgovine

2. Struktura podatkovne baze

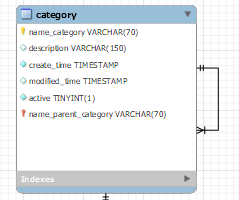
Za podatkovno bazo sem se odločal med PostgreSQL in MariaDB strežnikom. Na koncu sem se odločil za MariaDB, zato ker smo ga uporabljali pri pouku in bi tako lahko hitreje in bolj učinkovito delal na njem, kakor da bi se moral sproti učit tudi novo podatkovno bazo.

Vsa struktura podatkovne baze je narejena v programu MySQL Workbench, ki mi je omogočil lahko ustvarjanje baze in hitrih popravkov. Nato sem uporabil orodje, s katerim sem pretvoril vizualno bazo v SQL kodo, ki sem jo nato prenesel v MariaDB strežnik.

Podatkovna baza je sestavljena iz 24 tabel. Veliko tabel tudi hrani čas ustvarjenja »create\_time«, ki se nastavi ob vnosu podatka s CURRENT\_TIMESTAMP. Podobno se hrani tudi čas zadnje spremembe »modified\_time« s CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP. Hrani se tudi, če je podatek aktiven z atributom »active«, ki je tipa tinyint(1) (boolean). Vsi tuji ključi so nastavljeni, da se ob posodabljanju ali brisanju tudi oni posodobijo/zbrišejo, kar omogoča odstranjevanje neuporabljenih podatkov. Vsi števni primarni ključi uporabljajo možnost AUTO-GENERATE, ki poskrbi za to, da se ključi ne podvajajo in jih ni potrebno ročno/programsko vnašati.

2.1. Kategorije

Tabela kategorija je rekurzivna tabela, ki je narejena tako zato, da imamo lahko neomejeno podkategorij za določeno kategorijo. Realizirano je tako, da je tabela povezana sama nase in tako imamo sestavljeni primarni ključ, po katerem nato iščemo. Za glavne kategorije (niso vezane na nobeno starševsko kategorijo) namesto tujega ključa vstavimo znak »x«, kar predstavlja prazno kategorijo.

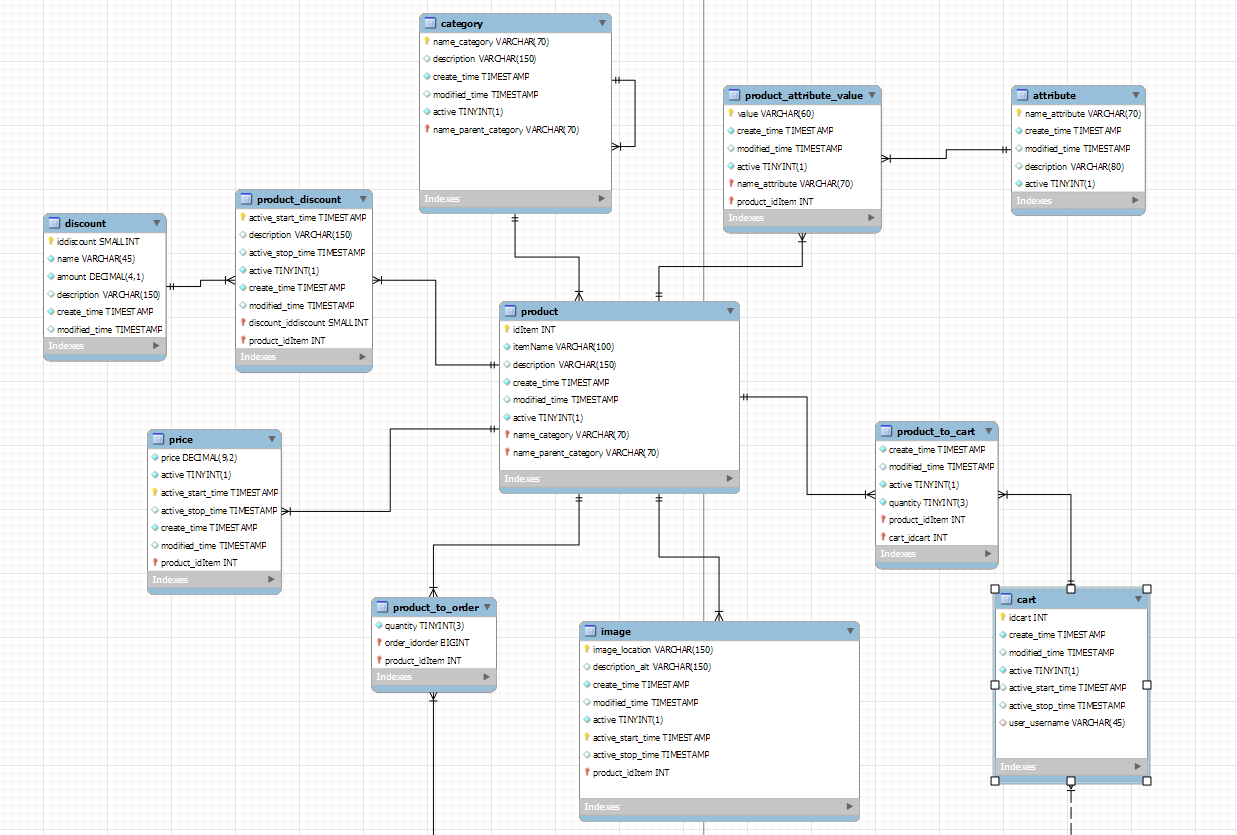


Slika 2: Tabela kategorija

2.2. Izdelek

Tabela izdelek je vezana z več drugimi tabelami, kakor lahko vidimo na spodnji sliki. Za identifikacijo izdelka se uporablja ključ, ki je sestavljen iz primarnega ključa ID izdelka in sestavljenega tujega ključa iz tabele kategorija. Tako lahko izdelek postavimo v pravo kategorijo in zagotovimo, da izdelki niso v napačnih kategorijah. Na to tabelo so še povezane tabele:

* popust, ki je v povezavi mnogo proti mnogo;
* cena, ki omogoča hranjenje vseh cen skozi zgodovino izdelka;
* slika, ki hrani lokacijo slik, vezanih na izdelek;
* atribut izdelka, ki je vezan mnogo proti mnogo in omogoča vezavo več različnih vrednosti na en atribut;
* dodatni tabeli »product\_to\_order« in »product\_to\_cart«, ki služita temu, da lahko dodamo več izdelkov v košarico ali v naročilo.

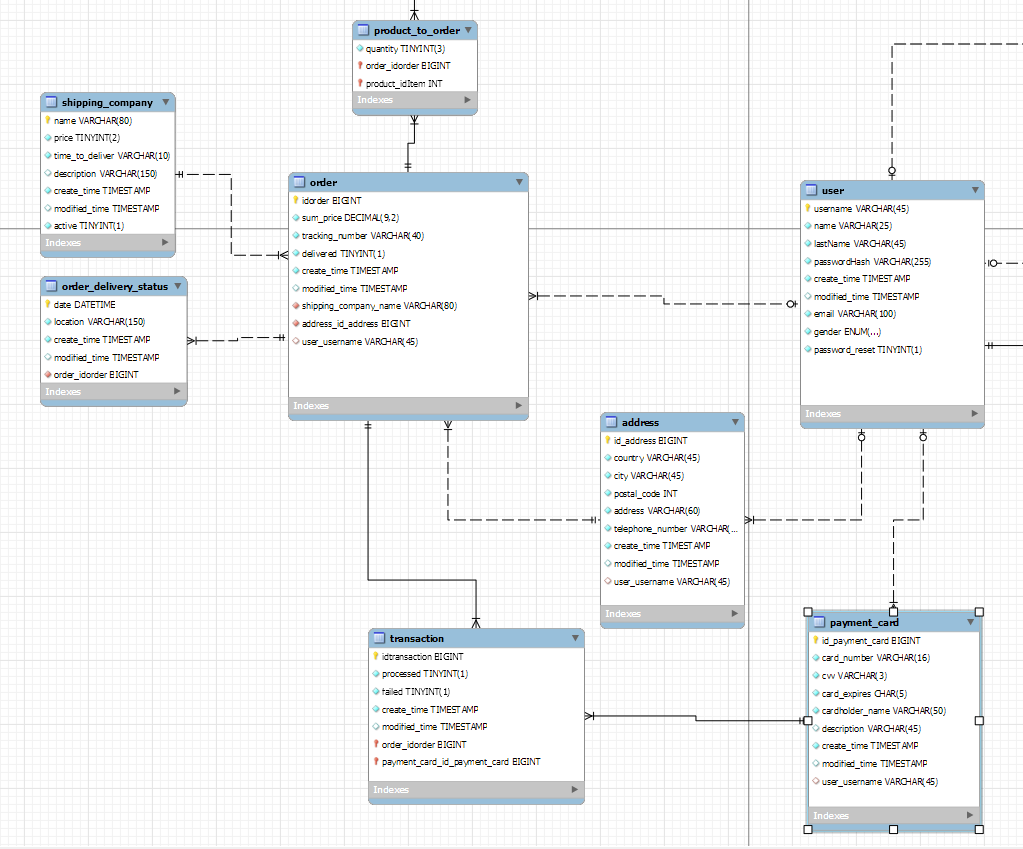


Slika 3: Tabela izdelek

2.3. Naročilo

Tabela naročilo je narejena tako, da vse deluje tudi če uporabnik ni prijavljen, zaradi tega imamo v tej tabeli tuji ključ za uporabnika, ki ga ni potrebno vnesti. Tukaj imamo še več dodatnih atributov kot so »sum\_price«, ki nam pove skupno ceno naročila, »tracking\_number«, ki ga priskrbi podjetje za dostavo in še »delivered«, kar pomeni da je naročilo bilo dostavljeno. Na tabelo naročilo so še vezane:

* Podjetje za dostavo, ki ima lahko več naročil;
* Status naročila, v katerem lahko hranimo več stanj dostave naročila;
* Uporabnik, ki se uporabi če je prijavljen;
* Naslov, v katerem se hrani naslov, ki je dodan trenutno ali če je oseba prijavljena se lahko uporabi shranjen naslov;
* Transakcija in plačilna kartica, to omogoča, da imamo lahko več poskusov plačila z isto kartico in v tem se lahko hrani podatke o tej transakciji.

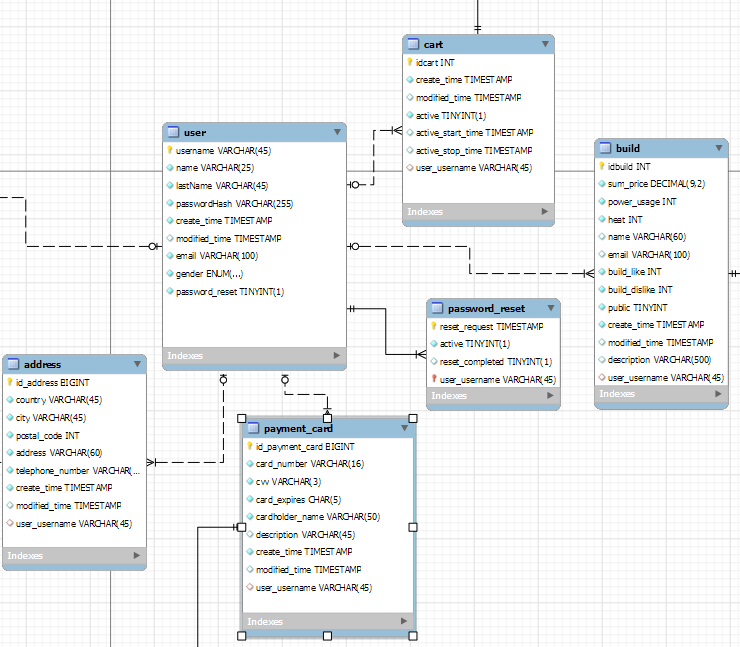


Slika 4: Tabela naročilo

2.4. Uporabnik

Tabela uporabnik vsebuje vse potrebne lastnosti uporabnika, da lahko zaključimo nakup in da lahko z njim komuniciramo. Imamo tudi polje »password\_reset«, ki nam pove če je uporabnik zahteval ponastavitev gesla. Povezana je tudi z tabelami:

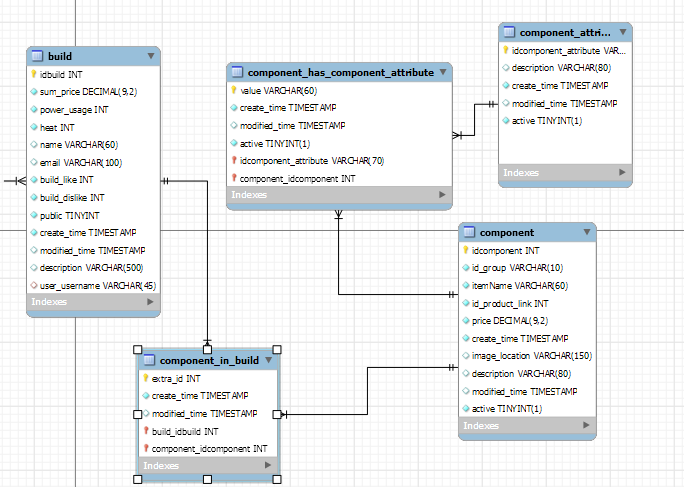
* Naslov, ki omogoča hranjenje naslovov;
* Plačilna kartica hrani vse uporabnikove kartice;
* Košarica, hrani uporabnikove košarice in tudi tisto, ki je uporabljena nazadnje, tako da lahko uporabnik do nje dostopa iz drugih mest;
* Tabela »password\_reset« v kateri hranimo vse poskuse zamenjave gesel;
* Konfiguracija, ki hrani vse uporabnikove konfiguracije.



Slika 5: Tabela uporabnik

2.5. Konfigurator

Glavna tabela je »component«, ki hrani vse lastnosti komponent (izdelek) brez dodatnih tabel. V tej tabeli tudi hranimo povezavo na izdelek »id\_product\_link«. Ta je vezana na atribute v mnogo proti mnogo vezavi isto kot pri izdelku. Vse te komponente pa se lahko poveže v konfiguracijo, ki vsebuje skupne lastnosti komponent. V vmesni tabeli je dodan atribut »extra\_id«, ki pa je potreben zato, da lahko dodamo več istih komponent.



Slika 6: Tabela komponenta

3. Struktura datotečnega sistema

Struktura se je razvijala skozi celotno izdelavo izdelka. Datoteke so med sabo ločene v mape za lažjo berljivost in organiziranost. V vsaki mapi je nato še naknadno ločeno na module pri katerih se uporablja.

3.1. Ločitev map:

»data\_generator«, v katerem so shranjene datoteke povezane z generatorjem izdelkov;

»database«, v katerem so shranjene vse ustvarjenje verzije podatkovne baze;

»db\_query«, tukaj so vse datoteke, ki obdelujejo podatke v bazi;

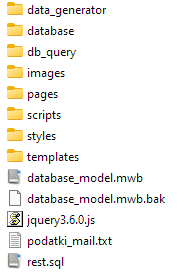
»images«, v tej mapi so shranjene sistemske slike in slike izdelkov;

»pages«, tukaj so dodane glavne strani do katerih uporabnik lahko dostopa;

»scripts«, vsebuje vse ukazne datoteke, ki so napisane v jeziku »javascript«;

»styles«, tukaj so dodani vsi stili spletne strani;

»templates«, vsebuje vse predhodno napisane predloge.



Slika 7: Datotečni sistem

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 8: Podmapa

4. Uporabnik

Za vstop na spletno stran, nakupovanje in ustvarjanje konfiguracij ni zahtevano, da je uporabnik registriran ali prijavljen. Registracija ponuja dodatni možnosti, ki so zelo priročne za vračajoče uporabnike in jim pohitri ter olajša izkušnjo na spletni strani. Ene izmed teh možnosti so: shranjevanje košarice, shranjevanje naslovov in kreditnih kartic ter shranjevanje konfiguracij.

4.1. Registracija

Potek registracije je preprost in intuitiven. Uporabnik mora vnesti unikatno uporabniško ime, svoj e-poštni naslov, geslo, ime, priimek in izbrati svoj spol. Vnos uporabniškega imena mora biti med 3 in 30 znaki. Implementirana je tudi preprost pregledovalnik za geslo, ki preveri, da vsebuje vsaj eno veliko črko in vsaj eno številko. Za dodatno varovalko uporabniku je postavljeno polje za ponovni vnos gesla v primeru, da se zmoti, saj je vsak vnos znaka v polje geslo pretvori v znak »•« in naredi geslo nevidno. Ob pritisku gumba »registracija« se na strani strežnika preverijo točnost podatkov in ob napaki podatka se na uporabniškem vmesniku prikaže ustrezna napaka. Ob uspešni registracije se uporabniku pošlje e-pošto in ga vrne na prijavno okno, kjer se lahko nato prijavi.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 9: Registracija

4.2. Prijava

V račun se prijavimo z uporabniškim imenom in geslom. Ob prijavi se preverijo podatki in se uporabniku prikažejo ustrezne napake. Če je prijava uspešna, se v PHP sejo shranita uporabniško ime in zakodirano geslo in nato se prikaže uporabniški menu.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 10: Prijava

4.3. Ponastavitev gesla

Uporabniku je podana možnost ponastavitve gesla. Geslo se ponastavi tako, da se izbere »ponastavitev gesla« na prijavnem oknu in se vnese e-poštni račun uporabnika. Sistem nato preveri, če uporabnik s tem računom obstaja in pošlje e-pošto s povezavo na stran, kjer lahko ponastavimo geslo. Medtem se v bazi nastavi, da je uporabnik zahteval ponastavitev gesla, to je storjeno v primeru, da bi nekdo poskušal spremeniti geslo brez odobritve uporabnika. Ko je novo geslo vneseno, in je preverjeno s strani strežnika, je uporabnik preusmerjen na prijavno okno, kjer se lahko prijavi z novim geslom.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 11: Ponastavitev gesla

4.4. Administrativni uporabnik

Administrativni uporabnik je bil dodan tako, da imamo PHP tabelo, v kateri so shranjena vsa uporabniška imena od administratorjev. Tako lahko ob vsaki prijavi preverimo, če se uporabniško ime uporabnika, ki je prijavljen nahaja v tej tabeli. Na nadzorni plošči ima administrator dostop do vseh tabel v bazi in lahko v njih vnaša podatke ter jih pregleduje. Poleg tega ima še možnost uporabe generatorja izdelkov.

5. Trgovina

Trgovina prikazuje izdelke, ki se jih lahko kupi. Ponuja pa tudi načine, kako lahko pridemo do teh izdelkov in jih filtriramo po najvišji in najnižji ceni in po abecedi. Dodan je tudi prikaz kategorij, ki prikazuje, v kateri kategoriji se nahajamo ali kam spada izdelek. Implementirana sta tudi iskalnik in omejen prikaz izdelkov na stran.

Vse deluje na podlagi URL-ja, s katerim aplikacija identificira, kaj uporabnik od nje zahteva in mu vrne ter prikaže ustrezne podatke. S takim pristopom je omogočeno tudi deljenje povezav, ki prikažejo iste podatke.

5.1. Kategorije

Menu je sestavljen iz naslova »KATEGORIJE« in iz samih glavnih kategorij, ki jih pridobi iz baze. Vse kategorije se naložijo na stran že ob prvem nalaganju, ampak ostanejo skrite v svojih podmenijih. Ob kliku na eno izmed kategorij se najprej spremeni barva trenutnega besedila iz bele v črno in odpre se podmeni (če obstaja). Ob tem se spremeni tudi URL naslov, v katerega se doda parametre za iskanje te kategorije. Če je bila izbrana, katera izmed drugih glavnih kategorij, se najprej izklopi aktivnost drugih kategorij in podmenija ter se ponovi postopek za izbor glavne kategorije. Ob izbrani podkategoriji, se najprej obarva besedilo, nato se preveri za podmenije in izklopi vse ostale kategorije na tej ravni. Postopek se ponavlja, saj je narejen rekurzivno za vse kategorije. Pri podkategoriji se v URL parametri dodajo in se ne zbriše kot pri glavnih kategorijah.



Slika 12: Aktivna podkategorija

5.2. Iskalnik

Iskalnik deluje tako, da imamo zato vnaprej določeno polje, kamor se vpiše iskalni niz in ob pritisku »lupe«, to je iskalno stikalo ali ob pritisku tipke »ENTER« na tipkovnici. Ob zagonu iskanja se spremeni URL na ta parameter in se pošlje zahtevek na strežnik za te izdelke. Iskalnik nato išče izdelke v bazi glede na ime in ni pomembna velikost črk.



Slika 13: Iskalnik

5.3. Filtriranje

Filtriranje deluje na strani strežnika in takrat, ko uporabnik pošlje zahtevo po prikazu izdelkov, se poleg iskalnih parametrov pošlje tudi filtracijske parametre. Nato se na strežniku, preden se podatki pošljejo, izvede filtriranje glede na parametre. Na voljo imamo sortiranje po abecedi, ki presortira izdelke glede na njihovo ime od »a« do »ž«. Sortiranje po najnižji ceni, ki sortira izdelke od najmanjše do najvišje cene in še sortiranje po najvišji ceni, ki sortira izdelke od najvišje do najmanjše cene. Ob izboru enega izmed filtrov se v URL doda parameter za sortiranje, ki je nato uporabljen za izbor sortiranja.



Slika 14: Filtrirni menu

5.4. Prikaz izdelkov

Ob uspešnem zahtevku za prikaz izdelkov se na uporabnikovi strani ustvarijo elementi, ki vsebujejo pomembne podatke o izdelku, kot so slika, cena, ime in popust. Ti elementi se nato združijo v mrežo, ki deluje tako, da se oblikuje glede na širino zaslona. To pomeni, če je zaslon širši, se lahko prikaže več izdelkov v širino. Prikazanih je lahko največ 30 izdelkov na stran. Po teh straneh se lahko premikamo z dvema stikaloma, ki omogočata premik naprej in nazaj po straneh. Ta proces je hiter in brez ponovnega osveževanja strani. V elementu za prikaz izdelka se nahaja tudi stikalo, ki lahko doda ta izdelek v košarico. Na pritisk kjerkoli drugje v tem elementu se odpre prikaz izdelka. Ta vsebuje meni s slikami in omogoča prehod med njimi. Vsebuje tudi ime in bolj podroben opis izdelka in njegovih atributov. Dodano je še stikalo in orodje za izbero koliko teh izdelkov bo dodanih v košarico. Maksimalno se lahko doda 10 izdelkov in najmanj enega. V prikazu kategorij se avtomatsko nastavijo kategorije, ki so povezane s tem izdelkom. Ustrezno se posodobi tudi URL, v katerega se shrani ime in id tega izdelka.

Slika, ki vsebuje besede besedilo, kompaktni disk, elektronika

Opis je samodejno ustvarjen

Slika 15: Prikaz izdelka

6. Košarica

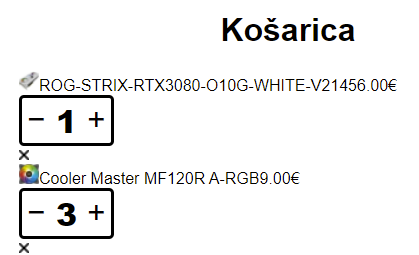
Košarica je sestavljena iz dveh elementov, ki prikazujeta vsebino. V teh dveh elementih lahko vidimo sliko, ime in ceno izdelka, ki ga imamo v košaric. Poleg tega lahko izdelke tudi odstranimo in spremenimo njihovo količino. Košarica lahko deluje tudi brez prijave uporabnika.

6.1. Delovanje

Košarica deluje tako, da ko se doda izdelek, aplikacija preveri, če je uporabnik prijavljen in tako ugotovi, kateri način delovanja mora izbrati. Prvi način delovanja deluje takrat, ko uporabnik ni prijavljen in shranjuje košarico v lokalno shrambo brskalnika, kjer se lahko nato ponovno uporabi. V lokalno shrambi je shranjen JSON objekt, ki je spremenjen v tekstovno obliko. V tem objektu so dodani id-ji izdelkov, ki so dodani kot ključi in njim se določi vrednost, ki pove, koliko teh izdelkov imamo v košarici. Drugi način začne delovati takrat, ko je uporabnik prijavljen. Tako se id izbranega izdelka in njegova količina pošljeta na strežnik, kjer se ga doda v bazo. Če uporabnik nima aktivne košarice, se mu ta avtomatsko ustvari in se izdelek poveže na to košarico.

6.2. Prikaz

Vsebino košarice lahko vidimo na točno določenem delu strani, kjer se lahko odstrani izdelke iz košarice in se jim doda ali zmanjša količino. Prikazana je tudi slika izdelka, cena in ime. Drugi način za ogled košarice je na voljo na glavnem meniju strani, ki se prikaže, če povlečemo kazalec od miške čez meni »košarica«. Nato se prikaže podmeni, ki pokaže tudi sliko, količino, ime in ceno izdelka ter skupno ceno. Vse podatke pridobi iz baze ali lokalne shrambe odvisno od načina delovanja košarice. Preden se izdelki dodajo na stran, se preveri njihovo integriteto, in ali obstajajo v bazi. V primeru napake se košarica ne prikaže in uporabniku vrne napako.



Slika 16: Prikaz košarice

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 17: Menijska košarica

7. Naročilo

Vnos podatkov za naročilo je narejen tako, da zahteva minimalno količino podatkov od uporabnika. Podatki se avtomatično shranjujejo med različnimi vnosnimi obrazci in je tako zagotovljeno, da se lahko uporabnik premika med obrazci brez izgube podatkov. Ko se podatki vnašajo v bazo, se nekatere spremeni v drugačno obliko.

7.1. Naslov

Na vnosnem obrazcu za naslov je potrebno izbrati državo prejemnika. Seznam držav je implementiran tako, da se države naložijo iz posebne JSON datoteke in za privzeto vrednost se nastavi »Slovenija«. Nato je potrebno vnesti ime mesta, poštno številko in naslov kraja dostave. Uporabnik vnese še svojo telefonsko številko, in če ni prijavljen na stran, mora vnesti še svojo e-pošto. Ko uporabnik vnese podatke in pritisne stikalo »naprej«, se podatki pošljejo na strežnik, kjer se preveri, če so podatki pravilni. Če je uporabnik prijavljen v sistem, mu ni potrebno vnašati e-pošte, saj je že vnesena v bazi. Dobi tudi možnost izbire svojih vnesenih naslovov, ki se prikažejo pod vnosnim obrazcem ali pa vnese svoj naslov, ki ga nato lahko shrani med svoje naslove. Ob primeru napak v podatkih se uporabniku prikaže, kje so napake.

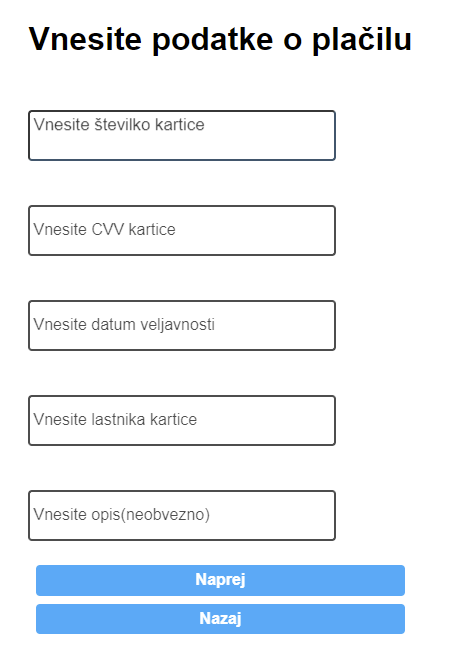
Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 18: Naslovni obrazec

7.2. Plačilo

Za možnost plačila je ponujena samo ena možnost, in to je plačilo s kartico. Uporabnik mora v vnosni obrazec vnesti številko kartice, ki je 16-mestna in aplikacija avtomatsko naredi presledke med razdelki številk za lažjo preglednost ter omeji dolžino številke. Nato je potrebno vnesti »CVV«, ki je 3-mestno število in datum veljavnosti kartice. Pri tem je že vnaprej nastavljena oblika zapisa in se preverja veljavnost datuma. Na koncu se vnese še lastnika kartice in neobvezen opis kartice, če si to uporabnik želi. Če je uporabnik prijavljen, mu je omogočeno, da si izbere kartico iz svojih kartic, lahko pa jo tudi doda. Ob pritisku stikala »naprej« se podatki pošljejo in preverijo.



Slika 19: Plačilni obrazec

7.3. Pošiljatelj

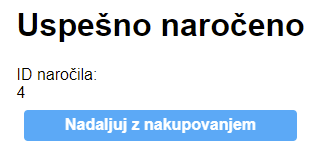
Uporabnik lahko izbira med različnimi kurirskimi službami, ki so mu na voljo. Poleg njihovega imena je prikazan tudi čas dostave ter cena, tako se lahko uporabnik lažje odloči.



Slika 20: Izbira pošiljatelja

7.4. Zaključek

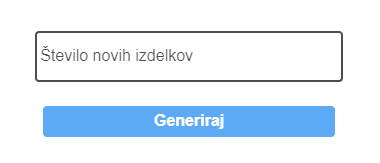
Na koncu uporabnik dobi pregled izdelkov in povzetek naročila. Če je uporabnik zadovoljen z izbranim, pritisne stikalo »Zaključi nakup«, ki nato uporabniku prikaže »ID« računa in mu pošlje e-pošto. Ob tem se podatki shranijo v bazo, in če uporabnik ni prijavljen, se košarica zbriše, drugače se košarica nastavi na neaktivno. Nato mu je omogočeno, da pritisne stikalo »Nadaljuj z nakupovanjem«, ki ga preusmeri nazaj na trgovino.



Slika 21: Končano naročilo

8. Generator podatkov

Ob začetku razvoja projekta sem se zavedal, da bom potreboval večjo količino podatkov za namene testiranja »konfiguratorja« in prikaza izdelkov. Zaradi tega sem se odločil, da bo hitreje, če naredim generator podatkov, ki bi avtomatsko ustvaril ustrezne podatke in jih dodal v bazo.



Slika 22: Vnosno polje za generator

8.1. Potek delovanja

Na začetku administrator določi, koliko izdelkov želi generirati. Ob pritisku na stikalo se pošlje število na strežnik, kjer se najprej iz baze pridobi vse kategorije in se jih vstavi v tabelo, ki je kasneje uporabljanja za preverjanje obstoječih kategorij. Nato se iz tabele izbere naključna kategorija, ki služi temu, da se lahko naprej določi znamka, ki je odvisna od določene kategorije. Nastavi se še ime, ki je odvisno od znamke in cena, ki je odvisna od izbrane kategorije. Izdelku se določijo atributi glede na določeno kategorijo. Vsi podatki se naključno izberejo iz že vnaprej določenih tabel, ki vsebujejo imena, najmanjše in največje vrednosti. Slike se naložijo z interneta in se shranijo v mapo na strežniku. Kodo za nalaganje slik sem naložil s spletne strani »PHP classes« in jo prilagodil svojim potrebam. Nato se dodajo kategorije za ta izdelek, če še ne obstajajo. Ko so uspešno dodane, se ustvari izdelek, ki je povezan na njih in tako pridobimo »ID« izdelka, na katerega nato povežemo vse ostale atribute, slike in ceno. Sočasno se dodajajo tudi komponente, ki so podobne izdelkom. Za razliko od izdelka, komponenta vsebuje samo eno sliko in samo eno ceno ter izbrane atribute. To pomeni, da je vmes dodan filter, ki dovoli samo določene atribute, ki so pomembni za »Konfigurator«. Ob vezanju izdelka in komponente na druge tabele se preverja za napake, in če jo najde, se zbrišejo do sedaj vneseni podatki za izdelek ali komponento in se ustavi generiranje podatkov. Ob tem se administratorju prikaže obvestilo za in vzrok za nastanek napake. Če je bilo dodajanje uspešno, se vnosno polje za generator obarva zeleno in se izpiše sporočilo »Dodajanje uspešno«.

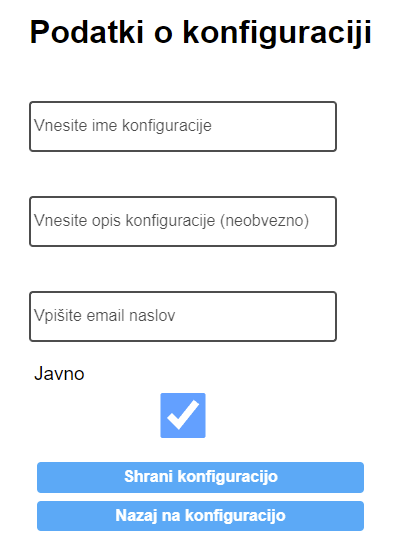
8.2. Slike

Pridobivanje slik deluje na PHP knjižnici »Client URL«. Postopek se začne tako, da se vzpostavi nova »cURL« seja, povezana na stran »Google images«, kateri je dodan parameter za iskanje po imenu. Nato se v datoteko shranijo povezave za nalaganje posamične slike in do tega kasneje dostopamo. Ko se postopek konča, se zažene nalaganje teh slik, ki deluje tako, da pridobljeno povezavo doda v spremenljivko »url«, kamor se še na začetek doda povezavo do strani od googla, ki je namenjena prenosu slik. To spremenljivko se nato uporabi v funkciji, »file\_get\_contents«, ki prenese sliko s strani, kasneje pa se uporabi funkcija »file\_put\_contents«, ki shrani sliko na strežnik. Slike se poimenujejo z imenom in dodatno številko, ki označuje vrstni red dodajanja. Takšen način je učinkovit, saj preprečuje podvajanje slik.

9. Konfigurator

Konfigurator je bil zelo zahteven za izdelavo in je vzel največ časa, saj nisem vedel, kako implementirati vse funkcionalnosti, ki sem jih želel, da jih ima. Na koncu sem se odločil, da bom uporabil drevo odločanja in svoj »seštevalnik«. Da se lahko prikaže vse konfiguratorjeve funkcionalnosti, je potrebno veliko raznolikih izdelkov z različnimi atributi, zaradi tega sem potreboval generator podatkov, kajti sam ne bi mogel vnesti toliko različnih komponent in izdelkov.

Celoten potek uporabe je zelo lahek, saj mora uporabnik samo izbrati komponente, ki so mu na voljo. Ko je je uporabnik zadovoljen s sestavljenim računalnikom, lahko shrani to konfiguracijo na stran. Prikaže se mu vnosni obrazec, kjer vnese ime in neobvezen opis konfiguracije. Če je uporabnik prijavljen, mu ni potrebno vnesti e-pošte, saj je že vnesena v bazo. Nato si lahko izbere ali je konfiguracija javna ali ne.



Slika 23: Podatki o konfiguraciji

9.1. Hiter pregled delovanja

Ko uporabnik izbere možnost dodajanja, se pošlje zahtevek na strežnik za vse komponente te skupine. Nato se vsako komponento pretvori v objekt in se uredi podatke, ki se jih pošlje v hiter pregled. Tam se preveri, če se komponento lahko doda v konfiguracijo, in če preverjanje potrdi pravilnost, se komponenta doda v seznam. Ko se vse komponente preverijo, se uporabniku prikaže seznam, iz katerega lahko izbira. Ob izbiri komponente se polno preveri ta komponenta in tudi tiste iz skupin, ki so odvisne od izbrane. Nato se seznam izbriše na vseh odvisnih skupinah in se ponovno naredi zahtevek, komponente se hitro preverijo, kar zagotavlja to, da se vse komponente na seznamu ujemajo s trenutno konfiguracijo. Ob tem se izdelek doda v glavno stanje in se posodobi cena, toplota in poraba ter se osveži prikaz na strani uporabnika. Osveži se tudi seštevalnik, ki ponovno preveri pravilnost konfiguracije in s tem zagotavlja kompatibilnost med komponentami. Nazadnje se shrani še glavno stanje v lokalno shrambo brskalnika in s tem omogoči ponovno nalaganje konfiguracije.

9.2. Prikaz komponent

Ob prvem prikazu strani se najprej naložijo prazna polja, ki predstavljajo vsako različno skupino komponent. To so: procesor, matična plošča, RAM, grafična kartica, shramba, procesorsko hlajenje, sistemsko hlajenje, napajalnik in ohišje. Če uporabnik želi dodati komponento v konfiguracijo, to stori s pritiskom na stikalo »plus«, ki prikaže seznam vseh komponent, ki so na voljo. Ob tem se ustvari stikalo »x«, s katerim lahko to komponento odstranimo iz konfiguracije. Ko uporabnik izbere željeno komponento, se prikažeta njena cena in slika. Če je ta skupina komponent »števna«, kar pomeni, da je možno imeti več istih komponent iz te skupine, se prikaže tudi možnost za dodajanje nove.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 24: Prikaz skupine komponent

9.3. Drevo odločanja

Drevo odločanja je implementirano v jeziku »javascript«, z objektom, v katerem so kot lastnost shranjena imena skupin in pod to lastnostjo je dodana funkcija. Tako se lahko preprosto kliče funkcije iz objekta. Vsaka izmed teh funkcij ima tri parametre, ki služijo temu, da se prejme primerjalne podatke, ali gre za hitro iskanje in na katerem indeksu bi ta komponenta bila vstavljena. V vsaki posamični funkciji so primerjalni stavki, ki primerjajo trenutno stanje konfiguracije z novimi podatki, ob tem za vsako primerjavo vrača svojo unikatno kodo, ki pove, kje se podatki niso ujemali. Če se podatki ujemajo s stanjem, funkcija vrne kodo »1«.

9.4. Stanje konfiguracije

Za delovanje konfiguratorja je potrebno hraniti njegovo stanje. Vsi podatki o komponentah se hranijo v objektu. Ta objekt se ustvari na začetku nalaganja strani iz predhodno napisane predloge. Za ohranjevanje in posodabljanje te predloge so narejene funkcije, s katerimi lahko vstavljamo komponente in jih varno brišemo. To pomeni, odstranijo samo vrednosti in se nastavijo na vrednost »null«. Funkcije poskrbijo tudi za ohranjanje možnosti tabel v posamičnih skupinah. To stanje se nato prenese v glavno stanje, kjer se poleg konfiguracije hranijo tudi poraba, cena in oddana toplota.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 25: Prikaz glavnega stanja

9.5. Seštevalnik

Seštevalnik je bil ustvarjen z namenom, da vse primerjave potekajo hitreje, tako da ni potrebno za vsako primerjavo preštevati vseh podatkov. Deluje tako, da se ob vsakem uspešnem dodajanju komponente v konfiguracijo shrani trenutno stanje in se izvede proces, ki odšteje vse povezane lastnostni v konfiguraciji in s tem preveri, če je pravilna.

9.6. Prikaz stanja

Ob vsakem uspešnem vnosu komponente v konfiguracijo se sproži tudi zahtevek za posodobitev prikaza komponent. Posodobi se prikaz tistih skupin, ki so odvisne od te, ki jim je bila dodana. Za skupine, ki so števne, se preveri tudi, če se lahko doda nove izdelke, zato da se nato odstrani ali doda možnost za dodajanje novih izdelkov te skupine. Ob tem se pod konfiguratorjem prikaže posodobljena poraba, cena in toplota.



Slika 26: Prikaz stanja o uporabniku

9.7. Poraba

Dodatna funkcionalnost tega konfiguratorja je možnost nastavitve presežka napajalnikove moči, kar je zelo priročno za uporabnike, ki želijo imeti možnost kasnejše nadgradnje računalnika. Začeten presežek moči je nastavljen na 10 %, kar pomeni, da se poraba konfiguracije poveča na 10 %. Največja dovoljena vrednost je 40 % in najmanjša 0. Povečana poraba se preverja v drevesu odločanja in ob vsakem spreminjanju vrednosti.

9.8. Shranjevanje konfiguracije

Ob vsakem spreminjanju glavne konfiguracije se stanje shrani v lokalno shrambo, kjer se nato lahko do nje dostopa. Shranijo se samo »ID-ji« komponent, cena, poraba in toplota. Ko stran zahteva nalaganje konfiguracije iz lokalne shrambe, se vrednosti prepišejo v trenutno glavno stanje in pošlje se zahteva na strežnik za podatke za vsak »ID« komponente. Nato se pregleda celotno konfiguracijo za napake.

9.9. Izboljšave

Ena izmed izboljšav, ki bi zelo pohitrila izbor in nalaganje komponent v seznam je to, da bi bil celoten konfigurator narejen na strani strežnika. Bilo bi hitrejše zaradi tega, ker ko bi uporabnik poslal zahtevo po tej skupini komponent, bi strežnik poslal samo tiste komponente, ki jih lahko prikaže.

10. Zaključek

Pri izdelavi maturitetnega izdelka mi je uspelo realizirati vse zastavljene cilje in jih tudi nadgraditi. Skozi potek izdelave se je moje razumevanje za programske jezike PHP, javascript in HTML zelo izboljšalo, kar je pripomoglo k boljšim ocenam pri strokovnih predmetih. Največja nadgradnja znanja je bila pri jeziku SQL, saj sem imel preteklo šolsko leto zelo malo znanja o tem. Ob tem sem se tudi naučil, da je pomembno ustvarjati redne varnostne kopije, zato da se lahko v primeru izbrisa podatkov naredi obnovo.

11. Viri

* <https://jquery.com/> (10. 4. 2022)
* [https://www.phpclasses.org/package/10014-PHP-Get-images-from-Google-search-that-match-a-keyword.html](https://www.phpclasses.org/package/10014-PHP-Get-images-from-Google-search-that-match-a-keyword.html%20%20)  (10. 4. 2022)
* [https://developer.mozilla.org/en-US/docs/Web/JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript%20) (10. 4. 2022)

Izjava o avtorstvu

Izjavljam, da je strokovno poročilo PC komponente v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

10. 4. 2022 Sebastjan Vidergar