Moderne tehnologije u izradi Internet trgovine

Štitić, Robert

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:703085

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK
Stručni studij

MODERNE TEHNOLOGIJE U IZRADI INTERNET TRGOVINE

Završni rad

Robert Štitić

Osijek, 2018.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. KORIŠTENE TEHNOLOGIJE 2

2.1. Visual Studio Code 3

2.2. Magento 3

2.3. Vue Storefront 4

3. IZRADA INTERNET TRGOVINE 6

3.1. Integracija Vue Storefront-a 6

3.2. Postavljanje e-mail notifikacija za malen broj proizvoda u skladištu 9

4. UPRAVLJANJE KATALOGOM 10

4.1. Kategorije 12

4.2. Proizvodi 13

4.3. Prikaz najnovijih i najprodavanijih proizvoda na naslovnoj stranici 15

5. ZAKLJUČAK 20

LITERATURA 21

SAŽETAK 22

ABSTRACT 23

ŽIVOTOPIS 24

PRILOZI (DVD) 25

1. UVOD

Razvojem Interneta, web aplikacija i weba općenito, raste interes poduzetnika za pravo web poslovanje, odnosno pokretanje Internet trgovine. Osnovna značajka Internet trgovine je da treba svom vlasniku omogućiti maksimalnu mogućnost samostalnog upravljanja sadržajem, a kupcu mogućnost da lagano pronađe proizvode koji ga zanimaju i naruči ih. Administracija trgovine treba bi omogućiti vlasniku veliki stupanj samostalnog upravljanja svojim asortimanom (kategorijama, podkategorijama..., pa sve do artikla). U ovom radu je detaljno objašnjeno kako napraviti Internet trgovinu koristeći Magento 2 za pozadinski sustav (eng. backend) i Vue Storefront za pristupni dio (eng. frontend). U drugom poglavlju opisane su teorijske osnove vezane za izradu Internet trgovine. Treće poglavlje detaljnije opisuje postupak izrade, dok se u četvrtom poglavlju opisuje upravljanje kategorijama i proizvodima. Peto poglavlje donosi glavne zaključke.

1.1. Zadatak završnog rada

Zadatak je napraviti aplikaciju namijenjenu za Internet trgovinu. Na naslovnoj stranici trebaju biti istaknuti najprodavaniji i najnoviji proizvodi. Potrebno je napraviti prijavu za dva tipa korisnika. Kupac treba moći puniti košaricu, platiti i pratiti narudžbe, a administrator unositi, uređivati i brisati kategorije i proizvode te pregledavati sve narudžbe. Također treba dobivati obavijest mailom kada na skladištu bude manje od 3 proizvoda.

2. KORIŠTENE TEHNOLOGIJE

Za izradu pozadinskog sustava Internet trgovine korišten je Magento 2 za koji je potrebno na operacijskom sustavu imati instaliran Apache 2 Web Server, MySQL 5 bazu podataka i PHP7 sa sljedećim ekstenzijama: bc-math, ctype, curl, dom, ImageMagick, intl, mbstring, mcrypt, hash, openssl, PDO, SimpleXML, soap, spl, libxml, xsl, zip, json i iconv. PHP je programski jezik koji se orijentira po C i Perl sintaksi, razvio ga je Rasmus Lerdorf, a namjena mu je razvoj dinamičkih web stranica. Ističe se širokom podrškom raznih internet protokola i baza podataka. Izvršava se na poslužitelju te ga možemo koristiti za slanje i primanje kolačića, prikupljanje podataka iz obrazaca, dinamičko generiranje sadržaja stranice itd. Za pisanje koda korišten je Microsoftov alat Visual Studio Code. Pristupni dio je napravljen u Vue Storefront-u kako bi trgovina bila progresivna web aplikacija. Vue Storefront koristi Redis i Elasticsearch za serviranje podataka. Prema [1] Redis je open source in-memory struktura podataka koja se koristi kao baza podataka, predmemorija i posrednik poruka. Implementira distribuiranu ključ-vrijednost pohranu sa opcionalnom izdržljivošću. Ime potječe od REmote DIctionary Server. Prema mjesečnim rangiranjima od DB-Engines, Redis je često rangiran najpopularnijom ključ-vrijednost bazom podataka. Redis je također rangiran kao #4 NoSQL baza podataka po zadovoljstvu korisnika i prisutnosti na tržistu. Podržava podatkovne strukture kao što su nizovi, hashovi, popisi, skupovi, sortirani skupovi s upitima raspona, bitmape itd. Redis ima ugrađenu replikaciju, Lua skriptiranje, LRU (Least Recently Used) iseljenje, transakcije i različite razine ustrajnosti na disku te pruža visoku dostupnost preko Redis Sentinela i automatsko particioniranje s Redis Clusterom. Zbog dizajna baze podataka, uobičajeni slučajevi korištenja su predmemorirane sesije, predmemorija za cijelu stranicu, aplikacije s redovima poruka, ploče s rezultatima, brojanje itd. Velike tvrtke poput Twittera koriste Redis, Amazon Web Services nudi Redis u svojem portfelju i Microsoft nudi Redis Cache u Azureu. Prema [2] Elasticsearch je distribuirani, opsežni alat za pretraživanje i analitiku. Može se koristiti za pretraživanje svih vrsta dokumenata. Pruža skalabilno pretraživanje i ima brzinu pretraživanja blizu realnom vremenu. Elasticsearch se distribuira, što znači da se indeksi mogu podijeliti na krhotine, a svaka krhotina može imati nula ili više replika, a svaki čvor ima jedan ili više krhotina i djeluje kao

koordinator da delegira operacije nad ispravnim krhotinama. Usmjeravanje se obavlja automatski, povezani podaci se često pohranjuju u istom indeksu, koji se sastoji od jednog ili više primarnih krhotina i nula ili više replika. Nakon što je indeks kreiran, broj primarnih krhotina ne može se mijenjati. Elasticsearch je razvijen u Java programskom jeziku. Službeni klijenti dostupni su u Java, .NET (C#), PHP, Python, Apache Groovy, Ruby i mnogim drugim jezicima. Prema rangiranju DB-Engines, Elasticsearch je najpopularnija tražilica za poduzeća.

2.1. Visual Studio Code

Prema [3] Visual Studio Code je uređivač izvornog koda kojeg su razvili Microsoft za Windows, Linux i MacOS. Uključuje podršku za ispravljanje pogrešaka, ugrađeno upravljanje Git-om, označavanje sintakse, inteligentno kompletiranje koda, isječke i refaktoriranje koda. Također je prilagodljiv, tako da korisnici mogu promijeniti temu, tipkovničke prečace i postavke. Besplatan je i open-source, iako je službeni download pod vlastitom licencom. Visual Studio Code temelji se na Electronu, razvojnom okruženju koji se koristi za implementaciju aplikacija Node.js za radnu površinu izvodeći se na Blink layout engine. U istraživanju Stack Overflow 2018 Developer Survey, Visual Studio Code je rangiran kao najpopularniji alat za razvojne programere, s 34,9% koji tvrde da ga koriste od 75,398 ispitanika. Podržava niz programskih jezika i niz značajki koje mogu biti dostupne za određeni jezik.

2.2. Magento

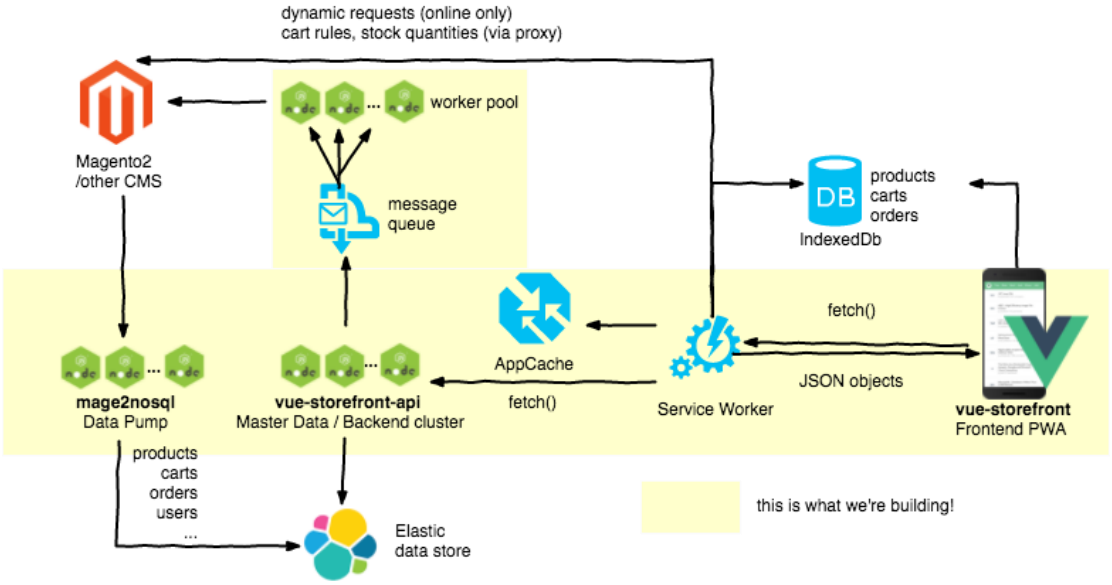
Prema [4] Magento je platforma za izradu Internet trgovina napisana u PHP programskom jeziku. Softver je izvorno razvio Varien, Inc, uz pomoć volontera. Varien je 31. ožujka 2008. objavio prvu opću verziju softvera. Dana 17. studenog 2015. Magento 2.0 je izdan, s ciljem pružanja novih načina za povećanje angažmana korisnika, lakše navigacije, poboljšanih stopa pretvorbe i generiranja prihoda vlasnicima trgovina. Ima alate potrebne za poslovanje koji poboljšavaju korisničko iskustvo softvera. Među poboljšanim značajkama verzije 2 nad verzijom 1 su sljedeće: smanjeni problemi zaključavanja tablica, poboljšano predmemoriranje stranica, skalabilnost poduzeća, ugrađeni bogati isječci za strukturirane podatke, nova struktura datoteka s lakšom prilagodbom, preprocesiranje CSS-a pomoću LESS & CSS URL

resolvera, poboljšane performanse i strukturiranija kodna baza. Magento koristi MySQL ili MariaDB sustav za upravljanje relacijskom bazom podataka, PHP programski jezik i elemente Zend okvira te primjenjuje konvencije objektno orijentiranog programiranja i Model-View-Controller arhitekture. Magento također koristi Entity-Attribute-Value model za pohranu podataka. Magento 2 je predstavio Model-View-ViewModel arhitekturu za pristupni dio pomoću JavaScript biblioteke Knockout.js.

2.3. Vue Storefront

Prema [5] Vue Storefront je headless PWA (Progressive Web Application) za Internet trgovine. Pored headless računala i softvera, najnoviji oblik headless tehnologije može se naći na web stranicama. Tradicionalne web stranice imaju svoj pozadinski sustav i pristupni dio. Svi dijelovi rade s istom kodnom bazom i komuniciraju izravno jedni s drugima. Međutim, u headless instalaciji, pristupni dio je samostalan program koji putem API-a (Application Programming Interface) komunicira sa pozadinskim sustavom. Oba dijela djeluju zasebno jedan od drugoga, a mogu se čak postaviti i na zasebne poslužitelje, čineći minimalnu verziju višeposlužiteljske arhitekture. Most između dva dijela je API klijent. Krajnji se dijelovi API-a mogu međusobno povezati. Progresivne web aplikacije su web aplikacije koje su regularne web stranice, no imaju mogućnost pojave korisniku kao tradicionalne aplikacije ili izvorne mobilne aplikacije. Progresivne web aplikacije poboljšavaju postojeću web tehnologiju, bez obzira na izbor preglednika jer su izgrađeni s progresivnim poboljšanjem kao temeljnim načelom. Kao takve, one ne zahtijevaju odvojeno povezivanje ili distribuciju – dopuštaju korisnicima da zadrže aplikacije koje korisnici smatraju najkorisnijima na početnom zaslonu bez upotrebe trgovina aplikacija za instalaciju. Mogu se identificirati kao „aplikacije“ zahvaljujući W3C (World Wide Web Consortium) manifestima i opsegu registracije service workera koji tražilicama omogućuju pronalaženje. Pored toga, sigurne su jer se poslužuje putem HTTPS-a (HyperText Transfer Protocol Secure) kako bi se spriječilo njuškanje (eng. sniffing). Objavljivanje progresivne web aplikacije jednako je kao i za bilo koju drugu web stranicu. Najveće prednosti ove tehnologije mogu se naći u optimizaciji performansi i fleksibilnosti softvera. Od travnja 2018. progresivne web aplikacije podržavaju preglednici Microsoft Edge, Google Chrome, Mozilla Firefox i

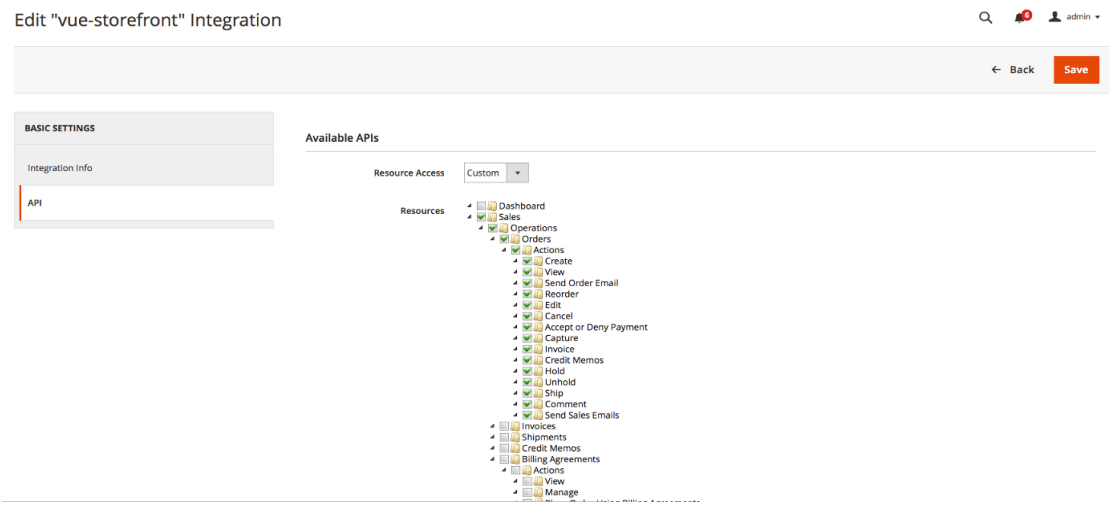
Safari. Za izdavača aplikacije najveća je prednost ušteda troškova u smislu razvoja i održavanja aplikacija. Prosječni troškovi progresivne web aplikacije mogu biti do deset puta manji u usporedbi s izvornim aplikacijama. Magento razvojni tim je pokrenuo razvoj PWA Studio upravo kako bi razvijanje PWA Internet trgovina bilo lakše. Iako Magento PWA Studio implementacija koristi ReactJS za pristupni dio, postoje i druge Magento PWA implementacije koje koriste različite tehnologije poput Vue Storefront-a kojega je moguće implementirati i u druge platforme osim Magento platforme poput Prestashop-a.



Sl. 2.1. Vue Storefront arhitektura [6]

3. IZRADA INTERNET TRGOVINE

Za instalaciju Magento platforme potrebno je napraviti praznu bazu podataka i pokrenuti setup.php datoteku. Zatim treba pratiti korake, odnosno odabrati jezik trgovine, vremensku zonu, valutu, informacije o bazi podataka, informacije o administracijskom računu itd. Nakon uspješne instalacije kako bi integrirali Magento sa Vue Storefront aplikacijom potrebno je logirati se u administracijsko sučelje i pod System -> Integrations kliknuti „Add new integration“ te ispuniti ime integracije i lozinku administratora. Osim toga, potrebno je unutar kartice API označiti Catalog, Sales, My Account i Carts.



Sl. 3.1. Magento integracija

Nakon aktiviranja Magento 2 daje OAuth pristupne tokene: consumer key, consumer secret, access token i access token secret.

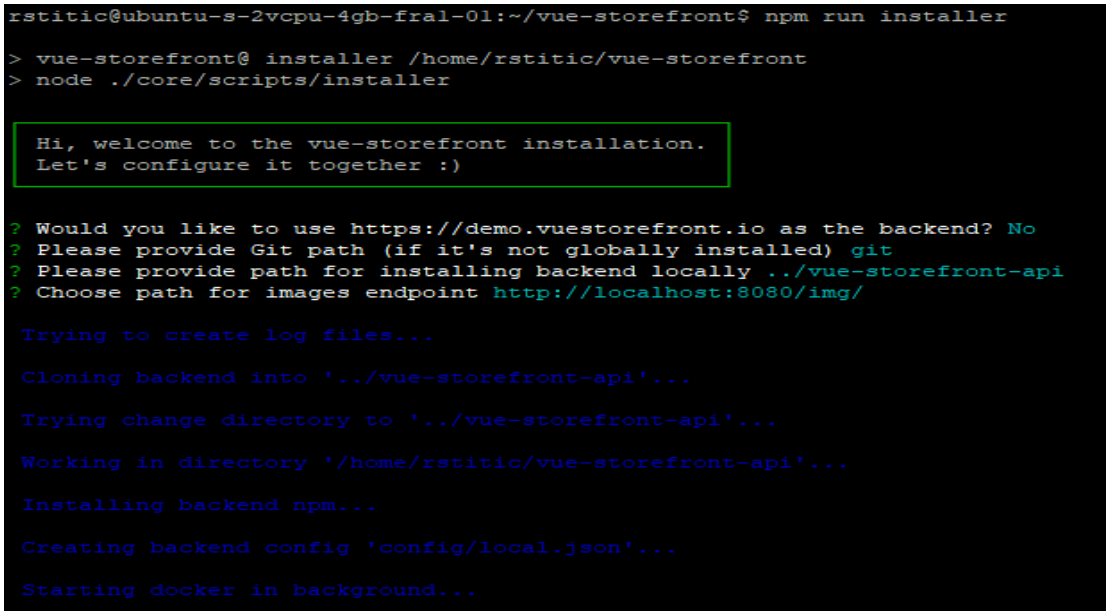
3.1. Integracija Vue Storefront-a

Da bismo počeli koristiti Vue Storefront, potrebno je znati arhitekturu sistema. Projekt podupiru tri zasebne Node.js aplikacije:

1. vue-storefront – glavni projekt gdje se nalazi kod pristupnog dijela, poput tema i stilova
2. vue-storefront-api – API sloj koji pruža podatke vue-storefront aplikaciji (sastoji se od Docker instanca za Redis i Elasticsearch)

3. mage2vuestorefront – podatkovni most zadužen za premještanje podataka iz Magento 2 u Vue Storefront

Node.js je JavaScript run-time okruženje koje izvršava JavaScript kod izvan preglednika. Omogućava pisanje alata za naredbeni redak i skripte koje poje se pokreću sa strane poslužitelja za izradu dinamičkog sadržaja web stranice. Za instalaciju Vue Storefront-a potrebna je Node.js verzija veća ili jednaka 8.0.0. Također je potrebno imati instaliran Docker i Docker Compose. Prema [7] Docker je računalni program koji obavlja virtualizaciju na razini operativnog sustava, što je poznato kao „kontejnerizacija“. Koristi se za pokretanje softverskih paketa koji se nazivaju kontejneri. U tipičnom primjeru jedan kontejner pokreće web poslužitelj i web aplikaciju, dok drugi kontejner pokreće poslužitelj baze podataka koji koristi web aplikacija. Kontejneri su međusobno izolirani i svaki kontejner ima svoje alate, biblioteke i konfiguracijske datoteke. Kontejneri mogu međusobno komunicirati kroz dobro definirane kanale, upravljaju jednim kernelom operacijskog sustava i stoga su laganiji od virtualnih strojeva. Docker Compose je alat za definiranje i pokretanje Docker aplikacija koje se sastoje od više kontejnera pomoću YAML (YAML Ain't Markup Language) datoteke. YAML je jezik za serializaciju podataka. U YAML datoteci se konfiguriraju usluge aplikacije. Vue Storefront ne koristi Magento bazu podataka direktno. Proizvodi, kategorije, pravila oporezivanja i atributi se spremaju u Elasticsearch bazu podataka koju koristi vue-storefront-api. Ovo je vrlo bitna značajka na koju treba pripaziti dok se razvija trgovina. Nakon kloniranja vue-storefront projekta i instaliranja potrebnih npm paketa potrebno je pokrenuti interaktivni program za instalaciju Vue Storefront aplikacije koji je prilično jednostavan za korištenje. Na prvom pitanju potrebno je potvrdno ili ne odgovoriti na pitanje da se želi koristiti demo verzija za pozadinski sustav, na drugom je potrebno postaviti git putanju (najčešće samo „git“ ako je instaliran u PATH). Zatim je potrebno postaviti image-proxy (image-proxy je dio vue-storefront-api-a i zaslužan je za promjenu veličine slika posluženih od Magento ili drugih pozadinskih sustava).



Sl. 3.2. Vue Storefront instalacija

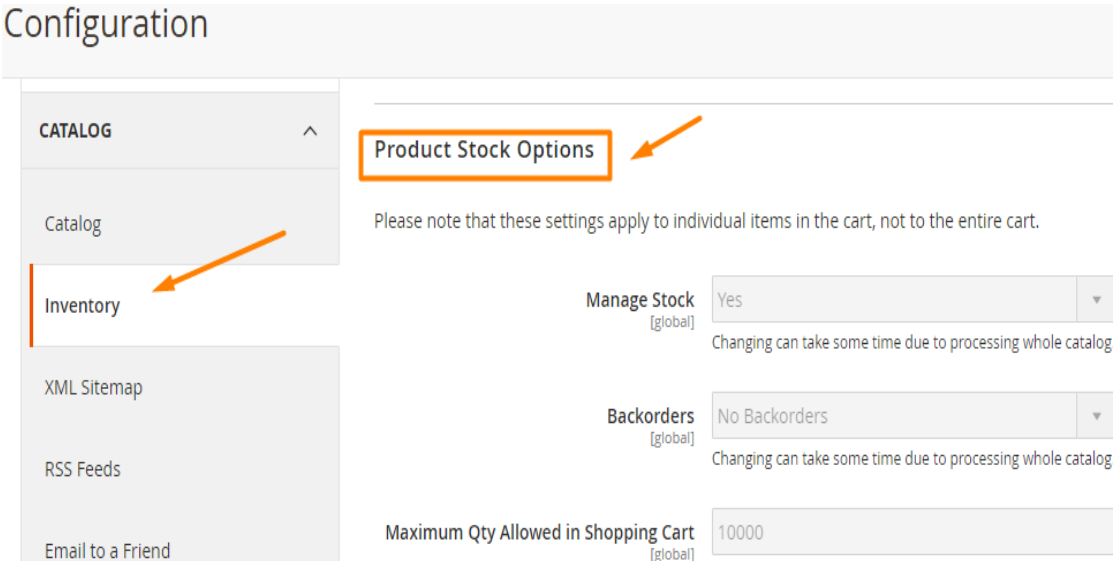
U tijeku instalacije u pozadini su pokrenute Elasticsearch i Redis instance te vue-storefront-api na portu 8080 i vue-storefront na portu 3000. Nakon toga potrebno je integrirati Magento sa lokalnom instancom, a zatim povezati Magento. OAuth tokene potrebno je upisati u mage2vuestorefront/src/config.js datoteku. Kako bi se svi proizvodi, kategorije i ostale bitne informacije unijele unutar Elasticsearch instance potrebno je izvršiti sljedeće naredbe u mage2vuestorefront direktoriju:

1. node cli.js taxrule
2. node cli.js attributes
3. node cli.js categories
4. node cli.js productcategories
5. node cli.js products

Pritom treba pripaziti da je redoslijed izvršavanja bitan, npr. naredba *node cli.js categories* popunjava Redis cache za daljnju uporabu *node cli products* itd. Sigurno je ponavljati ove naredbe ponavljaju tako da kreiraju ili ažuriraju postojeće zapise. Cli.js ima puno drugih načina za izvršavanje poput dinamičkih promjena, reda čekanja itd, no osnovni slijed za sinkronizaciju cijele Magento 2 baze podataka je prethodno prikazan. Vue-storefront-api je također odgovoran za sinkronizaciju narudžba i slika.

3.2. Postavljanje e-mail notifikacija za malen broj proizvoda u skladištu

Važno je zadovoljiti zahtjeve kupaca na vrijeme, stoga je korisno ispravno upravljati skladištem. U administracijskom sučelju pod Stores -> Settings -> Configurations -> Catalog -> Inventory treba postaviti Manage Stock „Yes“.



Sl. 3.3. Upravljanje skladištem preko Magento administracijskog sučelja

Slanje obavijesti izuzetno je učinkovito u upozoravanju administratora o stanju proizvoda. Obavještavanje o stanju može povećati prodaju te pruža vrijedne informacije o želji kupaca. Može pomoći i kod odabira koje vrste proizvoda ili cijena je potrebno ažurirati. Zatim je potrebno postaviti Notify for Quantity Below „3“. Ta opcija omogućava slanje obavijesti kada je na skladištu manje od 3 proizvoda.

4. UPRAVLJANJE KATALOGOM

Magento Catalog pruža funkcionalnost za stvaranje i održavanje proizvoda i kategorija. Dozvoljava administratoru upravljanje kategorijama i dodjeljivanje proizvoda kategorijama, no ne samo to. Sadrži različite aspekte, poput uređivanja opisa stavki, praćenje dostupnosti proizvoda, uploadanja slika proizvoda i još mnogo toga. Magento_Catalog modul se sastoji od:

- Api – direktorij koji sadrži ugovore o uslugama
    - Data – direktorij za sučelja podataka, svako sučelje mora imati model koji ga implementira (npr. sučelje za model proizvoda)
    - ProductRepositoryInterface.php – sučelja za spremišta (također moraju imati implementaciju)
- Block – direktorij za blokove korištene u izgledu za sučelje i pozadinu
    - Adminhtml – direktorij za blokove koji se koriste za pozadinski sustav
    - Category – direktorij za blokove na pristupnom dijelu
- Console – direktorij koja sadrži CLI naredbe
- Controller – direktorij koji sadrži kontrolere za pozadinski sustav i pristupni dio
    - Adminhtml – direktorij koji sadrži kontrolere za pozadinski sustav
    - Category – direktorij za kontrolere pristupnog dijela
- Cron – direktorij s kodom koji cron izvršava
- etc – direktorij koji sadrži XML konfiguracijske datoteke vezane za modul
    - frontend – direktorij koji sadrži XML konfiguracijske datoteke učitane na pristupnom dijelu
    - adminhtml – direktorij koji sadrži XML konfiguracijske datoteke učitane na pozadinskom sustavu
    - webapi_rest – direktorij koji sadrži XML konfiguracijske datoteke za REST API
    - webapi_soapt – direktorij koji sadrži XML konfiguracijske datoteke za SOAP API
    - acl.xml – Access Control List definicije
    - catalog_attributes.xml – zadani atributi za entitete kataloga
    - catalog_attributes.xsd – validacijska shema za catalog_attributes.xml

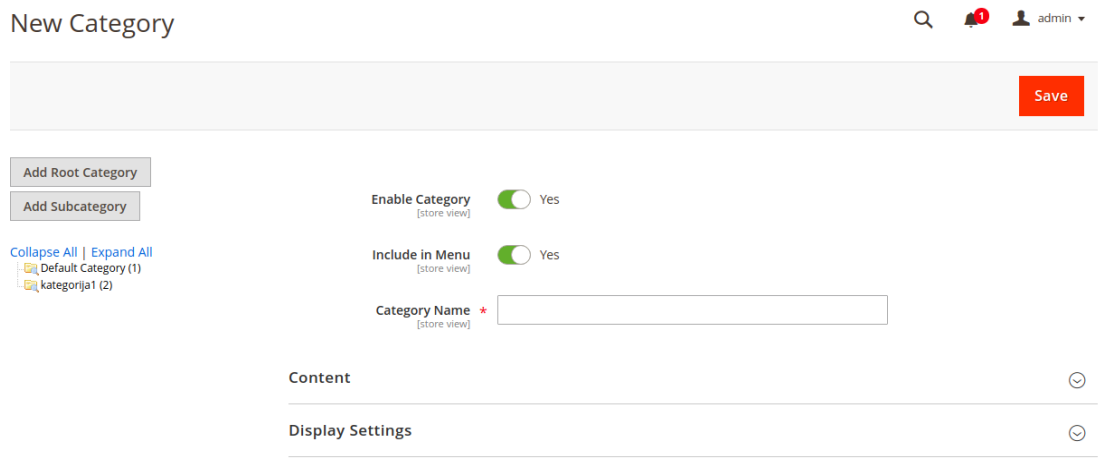
    - config.xml – zadane vrijednosti za opcije konfiguracije
    - crontab.xml – raspoređivanje cron zadataka
    - di.xml – ovisnosti vezane za dependency injection
    - events.xml – deklaracija promatrača za događaje
    - indexer.xml – postavke za različite indekse koje treba izvršiti prilikom promjene podataka
    - module.xml – datoteka za deklaraciju modula
    - product_* – postavke vezane za proizvod.
    - webapi.xml – web

  api deklaracijske putanje.
    - widget.xml – deklaracije widgeta.
- Helper – direktorij za razne helpere modula
- i18n – direktorij za datoteke sa prijevodom jezika
- Model – direktorij sa modelima za interakciju sa tablicama iz baze podataka
- Observer – direktorij sa promatračima događaja
- Plugin – around|before|after dodatci za razne javne metode
- Pricing – direktorij unutar kojeg su klase vezane uz cijene
- Setup – direktorij za instalaciju/ažuriranje sheme i podataka
- Test – direktorij sa unit testovima
- Ui – direktorij sa klasama vezanim uz UI komponente
- view – direktorij za preglede
    - adminhtml – direktorij za datoteke u administracijskom sučelju
        - layout – direktorij za XML layoute za adminhtml
        - templates – phtml predlošci za adminhtml
        - ui_compoenent – direktorij sa deklaracijama UI komponenti
        - web – direktorij sa stilovima (CSS, SCSS), JavaScript datotekama i slikama
        - requirejs-config.js – konfiguracija za Require.js biblioteku
    - base – direktorij za datoteke korištene na pristupnom dijelu i pozadinskom sustavu
    - frontend – direktorij za datoteke korištene na pristupnom dijelu

- composer.json – datoteka za konfiguraciju Composera (upravitelja paketima u PHP-u)
- registration.php – datoteka za registraciju modula

4.1. Kategorije

Kategorije su neophodne za izgradnju učinkovite kataloga i navigacije korisnika kako bi klijenti lako mogli pronaći proizvode koje žele. Oni su prilično osnovni koncept, no Magento ima puno funkcionalnosti ugrađenih u kategorije kako bi se u potpunosti iskoristila moć kategorija. Kategoriju je najlakše kreirati pomoću administracijskog sučelja. Potrebno je navigirati se u Catalog -> Categories i odabrati da li se želi kreirati nova kategorija ili podkategorija te unijeti podatke o kategoriji i kliknuti u gornjem desnom kutu gumb za spremanje nove kategorije.



Sl. 4.1. Dodavanje kategorija preko administracijskog sučelja

Kategorijama je moguće upravljati i preko API-a. Magento 2 REST API daje endpointe za brisanje, dohvaćanje, kreiranje i ažuriranje kategorija:

- DELETE /V1/categories/{categoryId}
- GET /V1/categories/{categoryId}
- POST /V1/categories
- PUT /V1/categories/{categoryId}

Kada se kreira kategorija preko administracijskog sučelja, pojavi se opcija za konfiguraciju postavki za optimizaciju prikaznih oglasa i tražilice. Da bi se konfiguriralo ove postavke pomoću POST V1/categories poziva, moguće je kao

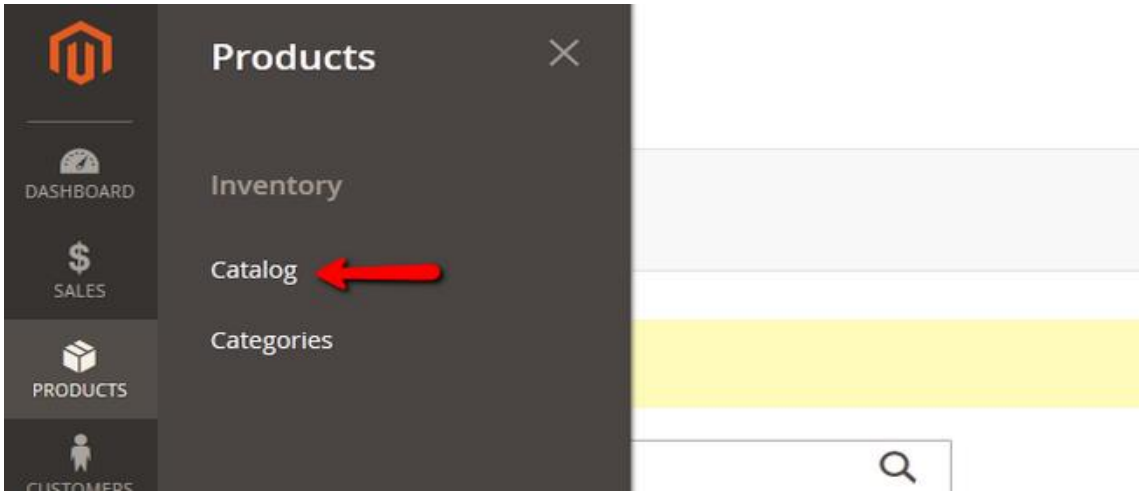
attribute_code vrijednosti odrediti sljedeće parametre: all_children, children, children_count, custom_apply_to_products, custom_design, custom_design_from, custom_design_to, custom_layout, update, custom_use_parent_settings, default_sort_by, description, display_mode, filter_price_range, image, is_anchor, landing_page, meta_description, meta_keywords, meta_title, page_layout, path, path_in_store, url_key i url_path. Moduli trećih strana (eng. third-party modules) mogu definirati druge prilagođene atribute.

4.2. Proizvodi

Magento podržava više tipova proizvoda:

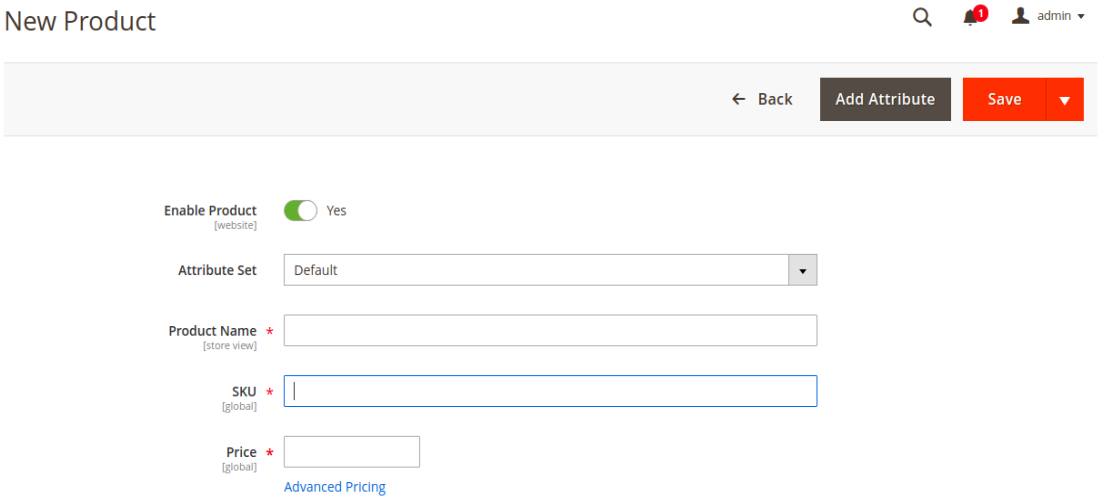
- Simple – je fizička stavka s jednim SKU. Jednostavni proizvodi imaju različite cijene i kontrole unosa koji omogućuju prodaju varijacija proizvoda.
- Grouped – predstavlja više samostalnih proizvoda kao skupinu. Možete ponuditi varijacije jednog proizvoda ili ih grupirati za promociju. Proizvodi se mogu kupiti odvojeno, ili kao grupa.
- Configurable – Podesivi proizvod izgleda kao jedan proizvod s popisom opcija za svaku varijaciju. Međutim, svaka opcija predstavlja zaseban, jednostavan proizvod s različitim SKU-om, što omogućuje praćenje oglasnog prostora za svaku varijaciju.
- Virtual – proizvodi koji nemaju fizičku prisutnost i obično se upotrebljavaju za usluge poput usluga, jamstva i pretplata. Virtualni proizvodi mogu se koristiti zajedno s Grouped i Bundled proizvodima.
- Bundled – omogućuju korisnicima da "izgrade svoje" iz asortimana opcija. Paket može biti poklon košara, računalo ili bilo što drugo što se može prilagoditi. Svaka stavka u paketu je zaseban, samostalan proizvod.
- Downloadable – proizvod koji se sastoji od jedne ili više datoteka koje se mogu preuzeti. Datoteke mogu biti na vlastitom poslužitelju ili biti navedene kao URL na bilo kojem drugom poslužitelju.

Proizvod je najlakše kreirati pomoću administracijskog sučelja. Potrebno je navigirati se u Products -> Catalog.



Sl. 4.2. Navigacija do kataloga proizvoda

Nakon navigiranja u katalog proizvoda pojavljuje se u gornjem desnom kutu opcija „Add Product“. Nakon toga pojavljuje se obrazac u kojem se unose atributi proizvoda poput imena, cijene itd. Na slici ispod je prikazan obrazac za unos novog proizvoda.



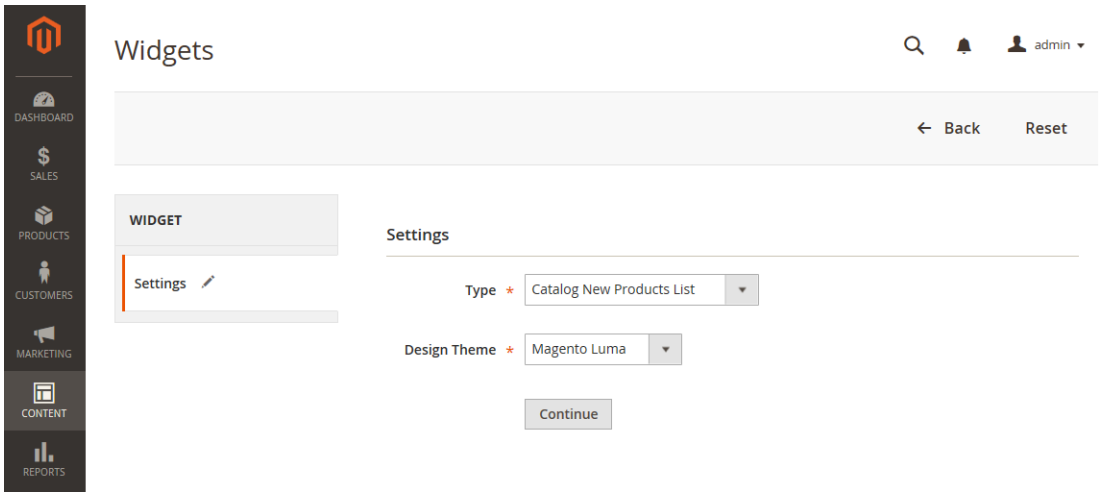
Sl. 4.3. Obrazac za unos novog proizvoda

Također je moguće i preko API-a upravljati proizvodima. Magento 2 REST API daje endpointe za brisanje, dohvaćanje, kreiranje i ažuriranje proizvodima:

- DELETE /V1/products/{sku}
- GET /V1/products/{sku}
- POST /V1/products
- PUT /V1/products/{sku}

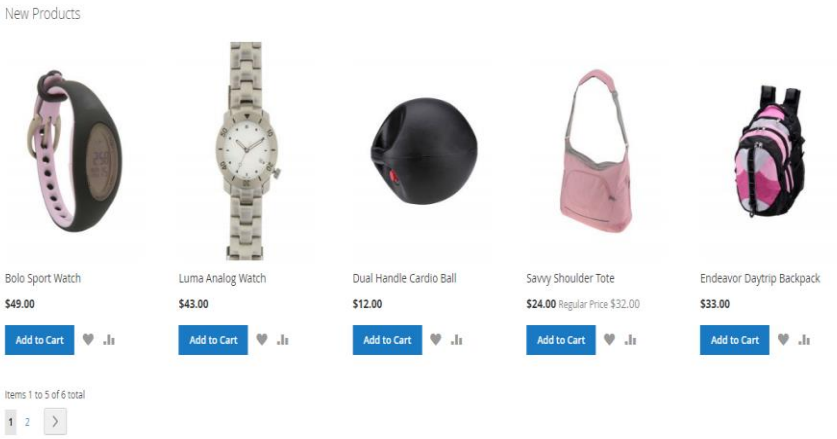
4.3. Prikaz najnovijih i najprodavanijih proizvoda na naslovnoj stranici

Kako bi se prikazali najnoviji proizvodi na naslovnoj stranici, potrebno je dodati widget. Widget je isječak koda koji omogućuje prikaz širokog raspona sadržaja i postavljanje na određene blokove trgovine. Magento platforma ima dostupan widget koji prikazuje najnovije proizvode te ga je samo potrebno postaviti na naslovnicu. Mnogi widgeti prikazuju stanje u realnom vremenu, dinamičke podatke i stvaraju prilike komunikaciju klijenata s trgovinom. Alat za widgete olakšava postavljanje postojećih sadržaja kao što su statički blokovi sa slikama i tekstom te interaktivnih elemenata. Kako bi dodali widget, potrebno je navigirati se u administracijskom sučelju u Content -> Elements -> Widgets te kliknuti „Add Widget“. U sekciji Settings potrebno je postaviti Type na „Catalog New Products List“ i pod „Design Theme“ odabrati temu na kojoj će widget biti prikazan.



Sl. 4.4. Widget Settings

Zatim je potrebno odabrati lokaciju na kojoj će se widget prikazati. U sekciji Layout Updates potrebno je postaviti Display On na „Specified Page.“, zatim Page na „CMS Home Page.“ i Block Reference na „Main Content Area.“ te spremiti postavke.



Sl. 4.5. Prikaz najnovijih proizvoda na naslovnoj stranici

Magento platforma sa modulima koji su dostupni nakon instalacije nema widget koji prikazuje najprodavanije proizvode te ga je potrebno kreirati. Magento platforma je sastoji se od velikog broja modula, te je preporučeno za izmjene sustava instalirati pojedinačne module, a ne mijenjati module iz Magento imenika. Moduli u Magento 2 kreiraju se unutar app/code/ direktorija. Za Stitic_Bestsellerwidget modul potrebno je napraviti datoteku app/code/Stitic/Bestsellerwidget/etc/module.xml.

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../../../../lib/internal/Magento/Framework/Module/etc/module.xsd">
 <module name="Stitic_Bestsellerwidget" setup_version="1.0.0."></module>
</config>
```

Prikaz koda 4.1. Datoteka module.xml za Stitic_Bestsellerwidget modul

Potrebno je napraviti i widget.xml datoteku unutar istog direktorija.

```
<?xml version="1.0" encoding="UTF-8"?>
<widgets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../../Magento/Widget/etc/widget.xsd">
 <widget id="stitic_bestsellerwidget"
class="Stitic\Bestsellerwidget\Block\Widget\Bestsellerdproduct">
  <label translate="true">Bestseller Product Widget</label>
  <description>Robert Stitic Bestseller Product Widget</description>
```

```
 <parameters>
  <parameter name="productcount" xsi:type="text"  visible="true" sort_order="1"
>
   <label translate="true">Product Count</label>
  </parameter>
  <parameter name="imagewidth" xsi:type="text"  visible="true" sort_order="2" >
   <label translate="true">Image Width</label>
  </parameter>
  <parameter name="imageheight" xsi:type="text"  visible="true" sort_order="3" >
   <label translate="true">Image Height</label>
  </parameter>
 </parameters>
</widget>
</widgets>
```

Prikaz koda 4.2. Datoteka widget.xml za Stitic_Bestsellerwidget modul

Zatim je potrebno kreirati app/code/Stitic/Bestsellerwidget/registration.php datoteku.

```
<?php \Magento\Framework\Component\ComponentRegistrar::register(
\Magento\Framework\Component\ComponentRegistrar::MODULE,
'Stitic_Bestsellerwidget',
__DIR__);
```

Prikaz koda 4.3. Datoteka registration.php za registraciju Stitic_Bestsellerwidget modula

Kako bi se najprodavaniji proizvodi prikazali na naslovnoj stranici potrebno je kreirati Magento blok i postaviti raspored (eng. layout) bloka na naslovnu stranicu ili koristiti neki postojeći blok sa naslovne stranice. Blok je klasa koja predstavlja skup metoda za manipuliranje Magento UI (User Interface). Nakon toga je u konstruktoru bloka potrebno učitati tvornicu (eng. factory) CollectionFactory. U objektno-orijentiranom programiranju tvornica je objekt kojemu je namjena kreiranje drugih objekata. Ispod je prikazan kod bloka koji služi za ispis najprodavanijih proizvoda.

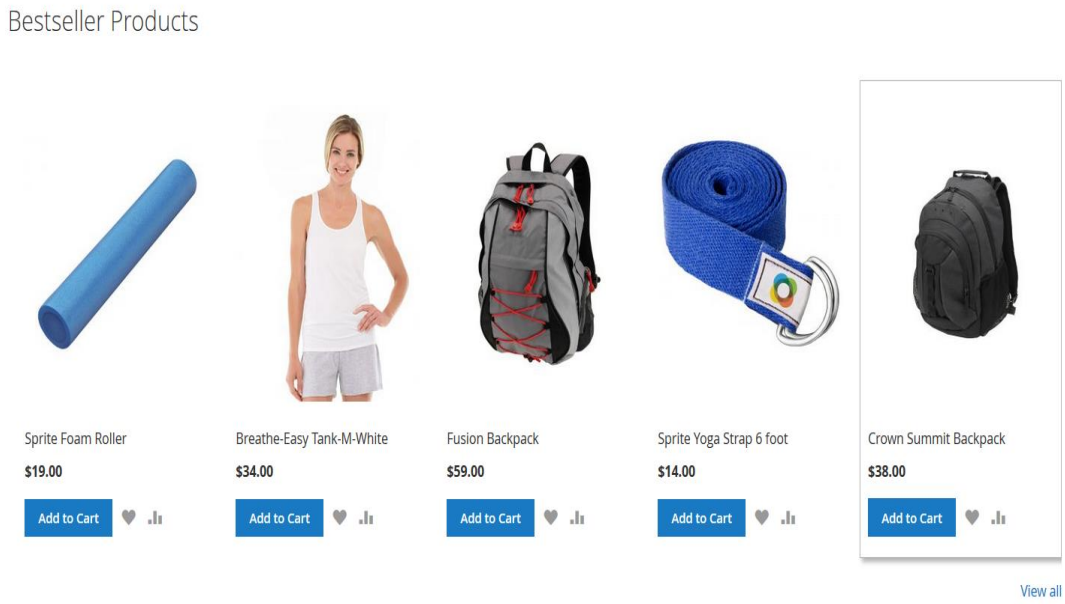
```php
<?php
namespace Stitic\Bestsellerwidget\Block;
class Bestsellerd extends \Magento\Framework\View\Element\Template
{
    protected $_coreRegistry = null;
    protected $_collectionFactory;
    public function __construct(
        \Magento\Backend\Block\Template\Context $context,
        \Magento\Framework\Registry $registry,
       \Magento\Sales\Model\ResourceModel\Report\Bestsellers\CollectionFactory
$collFactory,
        array $data = []
    ) {
        $this->_collectionFactory = $collFactory;
        $this->_coreRegistry = $registry;
        parent::__construct($context, $data);
    }

    public function _prepareLayout()
    {
        return parent::_prepareLayout();
    }

    public function getBestSellerData()
    {
        $collection = $this->_collectionFactory->create()->setModel(
            'Magento\Catalog\Model\Product'
        );
        return $collection;
    }
}
```

Prikaz koda 4.4. Klasa bloka za najprodavanije proizvode

Nakon toga možemo napraviti predložak koji će prikazati najprodavanije proizvode u Stitic/Bestsellerwidget/view/frontend/templates/widget/bestsellerdproduct.phtml datoteci. Na slici ispod prikazan je izgled bloka koji prikazuje najprodavanije proizvode.



Sl. 4.6. Prikaz najprodavanijih proizvoda

5. ZAKLJUČAK

Danas, kao i prije, Internet trgovine funkcioniraju na jednakom principu – dodavanje artikala u košaricu, ispunjavanje osobnih podataka na stranici za naplatu i naposljetku narudžbe koja obično završava slanjem e-mail poruke, kartičnim plaćanjem i/ili slanjem narudžbe i generiranjem uplatnice. Na tržištu postoje danas razne aplikacije namijenjene razvijanju Internet trgovina, no prije izbora bitno je razmisliti koliko vremena je upravitelj spreman uložiti u učenje korištenja aplikacije jer ponekad opcije koje aplikacije nude mogu zbuniti ili onemogućiti upravljanje. Neke od najpoznatijih platformi za razvijanje Internet trgovina su Magento, OpenCart, PrestaShop, WooCommerce, Sylius i sl. Osim vremena, potrebno je poznavati i temeljne funkcije koje trgovina treba ispunjavati za poslovanje – ne treba impresionirati korisnike mogućnostima već ostati na tragu onoga što je potrebno i bitno. Internet trgovina treba biti jednostavna i praktična za korisnika jer bi se inače mogli izgubiti u prevelikoj količini nepotrebnih podataka. Također je poželjno znati navike korisnika trgovine.

LITERATURA

[1] Internet : Redis
Dostupno na: https://redis.io/ [travanj 2018.]
[2] Internet : Elasticsearch: RESTful, Distributed Search & Analytics | Elastic
Dostupno na: https://www.elastic.co/products/elasticsearch/ [lipanj 2018.]
[3] Internet : Visual Studio Code
Dostupno na: https://code.visualstudio.com/docs/ [lipanj 2018.]
[4] Internet : Magento 2 Developer Documentation
Dostupno na: https://devdocs.magento.com/ [kolovoz 2018.]
[5] Internet : Vue Storefront
Dostupno na: https://www.vuestorefront.io/ [srpanj 2018.]
[6] Internet : Vue Storefront — how to install and integrate with Magento2
Dostupno na: https://medium.com/@piotrkarwatka/vue-storefront-how-to-install-and-integrate-with-magento2-227767dd65b2 [kolovoz 2018.]
[7] Internet : Docker
Dostupno na: https://en.wikipedia.org/wiki/Docker_(software) [kolovoz 2018.]

SAŽETAK

U ovom završnom radu govori se o modernim tehnologijama za izradu Internet trgovine. Spominjala se progresivna web aplikacija koja znatno poboljšava performanse trgovine u odnosu na izvorne aplikacije. Spomenuta je Magento platforma, detaljno je objašnjen princip izrade i razlog zašto je korišten za rješavanje ovoga zadatka. Objašnjeno je kako povezati Magento sa Vue Storefront progresivnom web aplikacijom. Objašnjeno je i kako upravljati Internet trgovinom i katalogom te slati notifikacije kada je na skladištu manje od 3 proizvoda. Na kraju je prikazano kako najprodavanije i najpopularnije proizvode istaknuti na naslovnoj stranici te su se prezentirali dobiveni eksperimentalni rezultati.

Ključne riječi: Katalog, Magento, progresivna web aplikacija, trgovina.

ABSTRACT

MODERN TECHNOLOGIES IN MAKING INTERNET COMMERCE

This paper is about modern technologies for creating Internet store. Progressive web application was mentioned which drastically improves shop's performance. Magento platform is mentioned, a detailed explanation of the shop creation and the reason why it is used to solve this task. It is explained how to integrate Magento with the Vue Storefront progressive web application. It is also explained how to manage the Internet store and its catalog and how to send notifications when there are less than 3 products in the inventory. At the end it is shown how to show most popular and bestseller products on homepage and presented the experimental results.

Keywords: Catalog, Magento, progressive web application, shop.

ŽIVOTOPIS

Robert Štitić je rođen 18.11.1996. u Sisku, nakon čega dolazi u Kutinu živjeti i školovati se. Sa 6 godina upisuje Osnovnu školu Mate Lovraka u Kutini. Po završetku kao što je i običaj upisuje srednju Tehničku školu u Kutini, koju također završava redovno i uspješno. Nakon toga, 2015. godine primljen je na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku gdje upisuje stručni studij informatike u trajanju od 3 godine. U međuvremenu radi poslove koji mu donose iskustvo i želju za radom.

Potpis: _____

PRILOZI (DVD)

Prilog 1. Pisana verzija završnog rada u .docx formatu

Prilog 2. Pisana verzija završnog rada u .pdf formatu

Prilog 3. Kompletan kod i baza podataka