

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Urban Kovač

**Primerjava najpogosteje uporabljenih  
odprtokodnih sistemov za e-trgovanje**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Denis Trček

Ljubljana, 2018



COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge primerjajte najpogosteje uporabljene platforme za izdelavo spletne trgovine. Najprej raziščite, kateri sistemi se uporabljajo in v kakšne namene. Naredite poglobljeno tehnično in poslovno analizo vsakega sistema. Za vsak sistem izdelajte tudi testno spletno trgovino, ki bo primerna za praktično uporabo.



*Rad bi se zahvalil svojemu mentorju prof. dr. Denisu Trčku za strokovno svetovanje in pomoč pri izdelavi diplomske naloge. Zahvaljujem se tudi Nini Zupančič za lektoriranje.*





# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Pregled področja . . . . .	2
<b>2</b>	<b>Shopify</b>	<b>3</b>
2.1	Kaj je Shopify? . . . . .	3
2.2	Namestitev . . . . .	3
2.3	Predstavitev strukture . . . . .	4
2.4	Predloga . . . . .	4
2.5	Aplikacije . . . . .	5
2.6	API . . . . .	6
2.7	Osnovne funkcije platforme Shopify . . . . .	6
<b>3</b>	<b>WooCommerce</b>	<b>11</b>
3.1	Kaj je WooCommerce? . . . . .	11
3.2	Predstavitev strukture . . . . .	11
3.3	Namestitev . . . . .	12
3.4	Predloga . . . . .	12
3.5	Vtičniki . . . . .	13
3.6	Zanke . . . . .	13
3.7	Shortcodes . . . . .	14

3.8	Osnovne funkcije platforme WooCommerce . . . . .	14
<b>4</b>	<b>Magento</b>	<b>19</b>
4.1	Kaj je Magento? . . . . .	19
4.2	Zgodovina . . . . .	20
4.3	Predstavitev strukture . . . . .	21
4.4	Arhitektura sistema . . . . .	22
4.5	Namestitev . . . . .	24
4.6	Predloga . . . . .	26
4.7	Moduli . . . . .	28
4.8	Sistemske načini delovanja . . . . .	30
4.9	Osnovne funkcije platforme Magento . . . . .	31
4.10	Magento 1 vs Magento 2 . . . . .	36
<b>5</b>	<b>Primerjava najpopularnejših platform za spletno trgovanje</b>	<b>41</b>
5.1	Splošna primerjava . . . . .	41
5.2	Tržni delež . . . . .	44
5.3	Optimizacija spletnih strani (SEO) . . . . .	45
5.4	Cena in stroški . . . . .	46
5.5	Varnost . . . . .	46
5.6	Prijaznost do uporabnika . . . . .	46
5.7	Predloge . . . . .	47
5.8	Pomoč strankam . . . . .	47
5.9	Zaprto kodni ali odprto kodni programi za izdelavo spletne trgovine . . . . .	47
5.10	Zakaj izbrati Shopify? . . . . .	49
5.11	Zakaj izbrati WooCommerce? . . . . .	49
5.12	Zakaj izbrati Magento? . . . . .	50
<b>6</b>	<b>Sklep</b>	<b>51</b>
	<b>Literatura</b>	<b>52</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>CMS</b>	content management system	sistem za upravljanje z vsebinami
<b>PHP</b>	PHP Hypertext Preprocessor	strežniški skriptni programski jezik
<b>SQL</b>	Structured Query Language	strukturiran povpraševalni jezik za delo s podatkovnimi bazami
<b>HTML</b>	hypertext markup language	jezik za označevanje nadbesedila
<b>CSS</b>	cascading style sheet	prekrivni slogi
<b>MVC</b>	model - view - controller	model - pogled - krmilnik
<b>MVVM</b>	model - view - viewModel	model - pogled - modelPogled
<b>SEO</b>	search engine optimization	optimizacija strani za iskalnike
<b>XML</b>	extensible markup language	razširljivi označevalni jezik



# Povzetek

V tej diplomski nalogi je predstavljena analiza več sistemov za spletne trgovine, ki so na prvi pogled zelo podobne, a delujejo na popolnoma drugačen način.

Ko se sistemski upravitelj spletne trgovine prijavi v administracijo, ima na levi strani orodno vrstico, ki mu omogoča upravljanje s celotno spletno trgovino. Vsi sistemi imajo podobne nadzorne plošče in večino osnovnih orodij, vsi sistemi pa omogočajo tudi namestitev modulov in vtičnikov.

V tej diplomski nalogi proučimo, kako posamezni sistemi delujejo in kaj so glavne prednosti in slabosti posameznega sistema. Diplomaska naloga je namenjena vsem, ki upravljajo ali imajo v lasti spletno trgovino, hkrati pa tudi tistim, ki potrebujejo pomoč pri izbiri najbolj primerne rešitve za spletno trgovanje.

**Ključne besede:** spletne trgovine, Magento, WooCommerce, Shopify.



# Abstract

For my thesis, I decided to do a comparison of several systems for online shops that are at first glance very similar, but operate in a completely different way. When the administrator of an online store logs in, they can manage the entire online store. All systems are similar as well as the surveillance boards and most basic tools. All systems allow the installation of modules and plug-ins. In this thesis we study how individual systems work and what are the main advantages and disadvantages each of them. This thesis is for everyone who manages or owns an online store, as well as those who need help in choosing the most suitable online trading solution.

**Keywords:** e-commerce, Magento, WooCommerce, Shopify.





# Poglavje 1

## Uvod

V današnji hitro razvijajoči se tehnološki dobi ljudje vse bolj nakupujejo preko spleta, zato je za lastnike trgovin zelo pomembna kvalitetno izdelana spletna trgovina. Programerji za izdelavo nove spletne trgovine vedno izberejo neko platformo, ki je pravzaprav že popolnoma delujoča spletna trgovina, ki potrebuje še veliko prilagoditev. Te platforme pogosto nudijo le osnovne funkcionalnosti, ki pa jih je mogoče razširiti z vtičniki in moduli. Dve najbolj pogosteje uporabljene platformi sta Magento in WooCommerce, ki temeljita na programskem jeziku PHP, za prikaz vsebine pa uporabljata tehnologije kot so HTML, CSS in JS. Obe platformi sta brezplačni in omogočata namestitvev tem in modulov, ki pa so pogosto plačljivi.

Vse bolj so popularne tudi spletne trgovine kot storitev. Ti sistemi so plačljivi in zaprtokodni. Izvorna koda ni dostopna in je ni mogoče spreminjati, omogoča pa razvoj lastnih aplikacij in izdelavo nove ali urejanje obstoječe teme. Te platforme so primerne za manjše in enostavnejše spletne trgovine. Prednosti teh sistemov sta enostavnost in nizki stroški vzdrževanja. Od teh sistemov je najbolj razširjen Shopify, ki je primerjan z Magentom in WooCommerce.

V diplomski nalogi bomo torej podrobneje predstavili platforme Magento, WooCommerce in Shopify. Zanima nas, katera platforma omogoča več funkcionalnosti in je prijaznejša za razvijalce, lastnike in uporabnike.

## 1.1 Pregled področja

Na spletu obstaja že veliko primerjav sistemov za spletno trgovanje, ki so pogosto pomanjkljivo napisane ali pa nas želijo prepričati v izbiro sistema, ki ni najboljši za nas. Najboljši sistemi se še vedno razvijajo zelo hitro in so deležni večje posodobitve vsaj dvakrat letno. Zato vse primerjave, ki so starejše od treh let niso več primerne za ocenjevanje posameznih sistemov.

## Poglavje 2

# Shopify

### 2.1 Kaj je Shopify?

Shopify je zaprtokoden program za izdelavo spletne trgovine, ki deluje na tujem strežniku. To pomeni, da s celotnim delovanjem in gostovanjem spletne trgovine upravlja Shopify, vendar pa omogoča spreminjanje teme, ki jo lahko kupimo ali izberemo brezplačno. Jedro spletne trgovine Shopify je napisano v programskem jeziku Ruby.

Danes je Shopify z več kot 325.000 zgrajenimi spletnimi trgovinami eden od najbolj razširjenih sistemov za spletno trgovanje.

### 2.2 Namestitev

Osnovna namestitev poteka samo v treh korakih. V prvem koraku vnesemo elektronski naslov, geslo, ki ga bomo uporabljali za vstop v administracijo in ime trgovine. V drugem koraku vnesemo osnovne podatke o stanju naše trgovine. V zadnjem koraku so potrebni še naši osebni podatki in naslov domene, ki jo želimo uporabljati. S temi enostavnimi koraki smo postavili spletno trgovino do te točke, da lahko začnemo z vnosom izdelkov. Nameščena je tudi tema, ki že prikazuje vse osnovne strani s privzetim besedilom in prednastavljenimi slikami.

## 2.3 Predstavitev strukture

Celotna struktura Shopifyja ni vidna, ker je to zaprtokoden sistem, zato je spodaj predstavljena datotečna struktura predloge. Vse morajo vsebovati spodnjo strukturo datotek. Struktura predloge Simple je sledeča: Layout, Templates, Sections, Snippets, Assets, Config, Locales.

## 2.4 Predloga

Nova namestitev spletne trgovine vsebuje predlogo Debut, ki vsebuje le zahtevane datoteke za pravilno delovanje sistema. Predloge je enostavno menjati in dodajati. Na voljo imamo 10 brezplačnih in 49 plačljivih predlog, na spletu pa lahko najdemo še vsaj 500 plačljivih predlog, ki so združljive s sistemom Shopify.

Za lažje in hitrejše urejanje predloge se uporablja orodje Liquid. To je brezplačno programsko okolje za izdelovanje predlog, napisano v programskem jeziku Ruby. Predloga je sestavljena iz zgoraj navedenih map. Vsaka od njih vsebuje točno določene datoteke, nekatere od njih pa so obvezne za delovanje sistema.

V mapi Layout morata biti datoteki `password.liquid` in `theme.liquid`. Prva datoteka je namenjena preprečevanju dostopa do strani, ko je ta še v izdelavi, druga datoteka pa predstavlja osnovno strukturo strani in ne vsebine.

V mapi Templates so zahtevane datoteke: `index.liquid`, `404.liquid`, `page.liquid`, `password.liquid`, `product.liquid`, `search.liquid`, `article.liquid`, `blog.liquid`, `cart.liquid`, `collection.liquid`, `customers/account.liquid`, `customers/activate.liquid`, `customers/addresses.liquid`, `customers/login.liquid`, `customers/order.liquid`, `customers/register.liquid`, `customers/reset_password.liquid`, `gift_cards.liquid`, `list_collections.liquid`.

Vsaka od teh datotek predstavlja svojo stran oziroma množico tematsko enakih strani z različno vsebino. V tej mapi lahko ustvarimo novo predlogo strani ali urejamo obstoječe strani.

V mapah Section in Snippets se nahajajo datoteke s končno vsebino strani.

Tukaj ni zahtevanih datotek, vendar so zelo pomembne za pravilno strukturirano stran.

V mapi Assets so uvrščene datoteke za končni izgled. Tu so datoteke s programsko kodo javascript, scss in css. Ta mapa je namenjena tudi medijskim datotekam, kot so slike, ikone in video posnetki, ki so namenjene za izboljšanje oblikovanja spletne trgovine.

V mapi Config je zahtevana datoteka settings\_schema.json, ki vsebuje osnovne nastavitve predloge.

V zadnji mapi Locales so datoteke tipa JSON, ki so namenjene prevajanju spletne trgovine in administracije.

Končna vsebina strani se dodaja v datoteke, ki so v mapah Templates, Sections in Snippets. Mape z vsebino so ločene po hierarhiji od največje do najmanjše. V mapi Templates so datoteke, ki predstavljajo celotno stran, npr. blog, članek, zbirka, izdelek, košarica itn. V mapi Sections so datoteke, ki sestavljajo večji del strani, npr. glava, noga, stranska vrstica, vsebina članka, vsebina izdelka, paginacija bloga in izdelkov. V mapi Snippets pa so najmanjši bloki vsebine, ki jih najdemo na strani, npr. opis izdelka, komentarji, oznake, prijava na e-novice, socialna omrežja.

Taka hierarhija strani omogoča, da se html koda ne ponavlja in jo je lažje urejati.

## 2.5 Aplikacije

Shopify omogoča enostavno dodajanje funkcionalnosti s pomočjo aplikacij. Na njihovi spletni trgovini za aplikacije lahko najdete več kot 1000 aplikacij. Te so razvrščene v 11 skupin: pridobivanje izdelkov, trženje, prodaja, socialna omrežja, dostava, upravljanje zalog, storitve za stranke, računovodstvo, orodja, poročanje in prodajni kanal. Vse aplikacije delujejo preko API klicev.

## 2.6 API

Shopify je razvil dva ločena API-ja, Admin API in Storefront API. Prvi se uporablja samo znotraj administracije oziroma samo za delovanje vaše spletne trgovine. Ta API se največ uporablja za delovanje aplikacij in za oddaljeno urejanje predloge. Admin API omogoča prikazovnje podatkov v spletni trgovini z oddaljenega strežnika, mesečno plačevanje aplikacij, urejanje naročil in dostave, urejanje predloge, upravljanje strank, dodajanje in urejanje javascript kode in dodajanje dogodkov v spletno trgovino.

Drugi API je namenjen izboljšavi nakupovalne izkušnje za kupce. Storefront API omogoča ustvarjanje in urejanje strank, izdelavo edinstvenih košaric in blagajn, pridobitev izdelkov, ki se bodo prikazali na katerem koli spletnem mestu ali napravi in omogoča edinstveno izbiro oblike izdelka.

## 2.7 Osnovne funkcije platforme Shopify

Za boljšo primerjavo med platformami bomo predstavili osnovne funkcije, ki jih ponuja vsaka platforma. V spletno trgovino bomo vnesli izdelke in kategorije, uredili načine plačila in dostave, ter nastavili davek, kupone in darilne bone.

### 2.7.1 Kategorije

Shopify omogoča delitev izdelkov v skupine na več načinov. Skupine lahko ustvarimo glede na tip izdelka, proizvajalca, zbirko in oznako. Izdelava hierarhije kategorij je precej bolj zahtevna, saj noben izmed zgornjih načinov ne podpira relacije oče-sin. Enostavno hierarhijo lahko ustvarimo s skupno uporabo zbirk in tipov izdelka. Najprej moramo vsakemu izdelku določiti tip, ki je najnižji nivo v hierarhiji. Potem dodamo vse zbirke, ki se nahajajo v hierarhiji. Za pravilno delovanje hierarhije moramo višjim nivojem dodati vse tipe, ki jih vsebujejo nižji nivoji. Zbirke poleg kreiranja skupin omogočajo še veliko dodatnih funkcij. Vsaki zbirki določimo naslov, dodamo pa ji lahko še

sliko in opis. Izdelke lahko v zbirko dodamo ročno ali pa izberemo pogoje s katerimi določimo skupino s podobnimi atributi. Izdelke lahko ločimo glede na naslov, tip, proizvajalca, ceno, oznako, težo, zalogo in naslov variante. Vse te pogoje lahko združujemo in primerjamo na osem različnih načinov. Za boljše rangiranje strani v spletnih iskalnikih pa lahko zbirki nastavimo še naslov strani, meta opis in URL naslov.

### **2.7.2 Izdelki**

Vnašanje izdelkov v spletno trgovino Shopify je zelo enostavno in omejeno. Novemu izdelku moramo vnesti ime in ceno, ki sta obvezna podatka. Izdelku lahko dodamo še opis, slike, tip produkta, proizvajalca, zbirko, oznake in meta opis za boljše rangiranje. Shopify nam omogoča izdelavo variacij izdelka z največ tremi možnostmi, pri čemer lahko vsako variacijo ločeno urejamo.

### **2.7.3 Zaloga**

Shopify omogoča avtomatsko beleženje zaloge. Kupcu lahko preprečimo nakup, če izdelka ni na zalogi. Zalogo lahko povečamo večjemu številu izdelkov hkrati, število izdelkov pa lahko ločeno dodamo število izdelkov v prihod. Beleženje zaloge lahko izključimo v celotni spletni trgovini ali samo pri nekaterih izdelkih, vendar to ni priporočljivo, saj bomo tako hitro izgubili nadzor nad fizično zalogo.

### **2.7.4 Dostava**

Platforma omogoča dodajanje različnih vrst dostav glede na državo, težo in skupno ceno nakupa. Vse vrste dostav morajo imeti izbrano območje in ceno.

### **2.7.5 Kuponi**

Kuponi omogočajo, da kupcu ponudimo popust na izbrane izdelke ali na celoten nakup. Ob ustvarjanju novega kupona, moramo najprej vnesti kodo, ki

jo mora kupec vnesti pred zaključkom nakupa. Izbiramo lahko med tremi tipi kuponov: brezplačna poštnina, fiksni znesek ali odstotkovni znesek. Kuponu lahko nastavimo minimalni znesek, nad katerim ga bo mogoče uveljavljati. Kupon je lahko veljaven samo za določeno skupino kupcev ali pa za vse nakupe. Kupone lahko omejimo časovno, po številu uporabljenih kuponov, prav tako pa lahko preprečimo enemu kupcu večkratno uporabo.

### **2.7.6 Darilni boni**

Darilni bon lahko naročijo kupci, ki želijo podariti bon drugim uporabnikom. Darilni bon lahko časovno omejimo. Ustvarimo lahko le en bon in mu dodamo več variacij z različnimi zneski. Vsak bon vsebuje znakovno in QR kodo s katero bon vnovčimo.

### **2.7.7 Plačila**

Shopify priporoča uporabo plačilnih sredstev kot sta PayPal in kreditna kartica. Nakup lahko plačamo tudi s klasičnimi metodami, kot so plačilo ob dostavi, bančni depozit in denarno nakazilo. Poleg osnovnih plačilnih sredstev Shopify omogoča še alternativna plačila, kot so kriptovalute.

### **2.7.8 Davek**

Za pravilno delovanje spletne trgovine je potrebno pravilno nastaviti davke. Davke lahko vključimo v ceno izdelka ali pa davek prištejemo ceni izdelka. Višino davkov lahko geografsko ločimo po kontinentih, državah in regijah. Za digitalno blago lahko izdelamo ločeno zbirko, ki ji določimo drugačen davek.

### **2.7.9 Naročila**

Naročilo je sestavljeno iz osebnih podatkov kupca, načina pošiljanja, metode plačila in seznama izdelkov. Vsak nakup poteka v treh korakih. V prvem



koraku stranka vnese osebne podatke, v drugem koraku vnese način pošiljanja, ki je omejen na lokacijo kupca, v tretjem koraku pa ostane še metoda plačila.

Ko stranka zaključi z nakupom, se naročilo pojavi v administraciji. Lastnik trgovine vidi seznam vseh naročil z vsemi podatki. Vidne so tudi ostale aktivnosti kupca v spletni trgovini, kot so ostali nakupi, število ogledov spletne trgovine, naslov IP in GPS lokacija pridobljena iz naslova IP.



## Poglavje 3

# WooCommerce

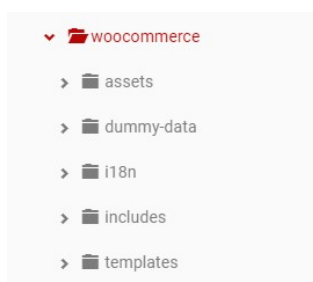
### 3.1 Kaj je WooCommerce?

WooCommerce je program za izdelavo spletne trgovine, ki deluje kot vtičnik platforme WordPress. Velika večina WordPress tem ima možnost brezplačne nadgradnje v spletno trgovino, kar je razlog za drugo najbolj razširjeno obliko spletnih trgovin na spletu. Program za izdelavo spletne trgovine WooCommerce je primeren za manjša podjetja, saj je sistem uporabniku prijazen in cenejši, kar pa ima za posledico izjemno slabšo kvaliteto in stabilnost spletne trgovine. [12]

### 3.2 Predstavitev strukture

Wordpress nima arhitekturnega modela, kot je MVC, zato je struktura Wordpresa zelo enostavna. deli se pa na tri glavne mape. Prva in najpomembnejša je wp-includes, ki vsebuje jedro Wordpresa in vse nujno potrebne funkcionalnosti. Druga je wp-admin, ki predstavlja administracijo. Ta del sistema omogoča prijavo uporabnika in njegov nadzor nad spletno trgovino. Tretja mapa je wp-content, ki vsebuje predloge, vtičnike in statične datoteke. Ta mapa je edina, ki se ob posodobitvi ne spremeni, zato so tukaj narejene vse programerske nadgradnje.

Datotečna struktura WooCommerca se nahaja znotraj mape plugins. V mapi assets se nahajajo slike, pisave, css in javascript koda. Mapa dummy-data nam omogoča uvoz testnih izdelkov v spletno trgovino. Mapa i18n je namenjena prevodom in nastavitvam specifičnih podatkov za posamezno državo. V mapi includes je sprogramirana vsa logika spletne trgovine. Sprednja stran (angl. Front End) datoteke se nahaja v mapi templates.



Slika 3.1: Datotečna struktura WooCommerca.

### 3.3 Namestitev

Za namestitev WooComerca je potrebno imeti nameščen WordPress. WooCommerca se naloži znotraj WordPress administracije. Namestitev poteka v petih korakih. Potrebno je vnesti lokacijo trgovine, valuto, prikaz cene, davek, enoto za težo in dolžino, ter metode plačila. Ko namestimo WooCommerca lahko začnemo z vnašanjem produktov ali pa izdelke uvozimo s CSV datoteko.

### 3.4 Predloga

Vse predloge se nahajajo v mapi themes. V uporabi imamo lahko samo eno temo. Če želimo predlogo urediti, je zelo priporočljiva uporaba podrejene predloge (angl. Child Theme). Ta tema se mora imenovati enako kot nadrejena predloga s končnico »-child«. Podrejena predloga mora vsebovati

datoteki `functions.php` in `style.css`. Taka predloga pohitri razvoj spletne trgovine in prepreči napake pri posodabljanju predloge in vtičnikov. Predlogi lahko dodajamo JavaScript datoteke s funkcijo `wp_enqueue_script()` in css datoteke s funkcijo `wp_enqueue_style()`.

### 3.5 Vtičniki

Vtičniki so paketi kode, ki razširjajo osnovno funkcionalnost WordPressa, ne da bi pri tem spreminjali jedro WordPressa. Vtičniki WordPress so sestavljeni iz kode PHP in drugih sredstev, kot so slike, CSS in JavaScript. Večina WordPress-ovih vtičnikov je sestavljena iz več datotek, toda vtič dejansko potrebuje samo eno glavno datoteko s posebej formatiranim DocBlock-om v glavi. Vtičniki so torej lahko zelo enostavni in omogočajo samo eno funkcionalnost, lahko pa so manjši sistem, ki omogoča veliko več, kot je na primer WooCommerce. Prednosti vtičnikov so, da jih lahko uporabimo na več spletnih straneh, da se ob posodabljanju WordPress jedra ne spremenijo in ohranijo svojo funkcionalnost, jih je enostavno posodobiti in enostavno dodajati ter odstranjevati.

### 3.6 Zanke

Termin zanke (angl. hook) se nanaša na mesto v WordPress kodi, kamor lahko dodamo svojo kodo, ali spremenimo privzeto delovanje WordPressa. "Hooke" delimo na akcije in filtre.

Akcija je zanka, ki se izvede v določenem času izvajanja WordPressa in nam omogoča dodajanje svoje funkcije, ki se takrat izvede. Tako lahko dodamo svojo kodo v točno določeno mesto na spletni strani. Filtri nam omogočajo spreminjanje podatkov, preden se ti pošljejo spletnemu brskalniku ali v bazo. WordPress vtičniki in predloge ponujajo številne zanke, ki jih lahko uporabimo. Z njimi lahko spremenimo delovanje platforme, pri tem pa pustimo vse datoteke sistema nespremenjene. Vse zanke zapišemo v datoteko `funci-`

ons.php, ki je obvezna datoteka child teme. Tako si zagotovimo nemoteno posodabljanje sistema. [13]

## 3.7 Shortcodes

Zaradi varnosti je prepovedano izvajanje PHP kode znotraj Wordpress vsebine. Shortcodes so t.i. makri, ki jih lahko uporabimo za dinamično interakcijo z vsebino. Črtne kode so dragocen način ohranjanja vsebine čiste in semantične, hkrati pa končnim uporabnikom omogočajo, da programsko spremenijo predstavitev svoje vsebine. Shortcodes lahko sprejmejo tudi parametre, ki uporabnikom omogočajo spreminjanje načina, kako deluje koda. Ko končni uporabnik dodeli galerijo fotografij v svojo objavo z uporabo kratke kode, uporabljajo najmanjše možne podatke, ki prikazujejo, kako naj bo galerija predstavljena. [14]

## 3.8 Osnovne funkcije platforme WooCommerce

### 3.8.1 Kategorije

Pri WooCommerce lahko izdelke ločimo po kategorijah, oznakah in atributih. Pri kategorijah lahko izberemo nadrejeno kategorijo in s tem ustvarimo hierarhijo. To hierarhijo lahko uporabimo v mega meniju in pri filtriranju izdelkov. Tako izboljšamo uporabniško izkušnjo in pohitrimo iskanje željenega izdelka.

Oznake lahko poljubno dodajamo izdelkom in tako prikažemo skupine izdelkov, ki imajo skupne nekatere lastnosti.

Tretji in pomemben način združevanja izdelkov je uporaba atributov. Obstajata dva načina uporabe atributov, ki sta zelo pomembna za WooCommerce. Prvi način je z uporabo gradnika "WooCommerce Layered Nav", ki deluje na podlagi atributov. Če dodate ta gradnik v stransko vrstico, lahko stranke filtrirajo izdelke na podlagi atributa. Drugi način je za uporabo spremenljivih izdelkov. Če želimo ustvariti spremenljiv izdelek, moramo najprej določiti

atribute za izdelek. Ti atributi se nato lahko uporabijo za razlikovanje med različicami.

### 3.8.2 Izdelki

Ko vnesemo atribute in kategorije lahko začnemo z vnosom izdelkov. WooCommerce omogoča vnos šestih različnih tipov izdelkov:

- enostavni – v to skupino spada večina izdelkov, katerim lahko določimo zalogo, razred dostave in atribute;
- združeni – to je zbirka sorodnih izdelkov, ki jih je mogoče kupiti tudi posamično in so sestavljeni iz enostavnih izdelkov;
- virtualni – to so predvsem storitve, ki ne zahtevajo pošiljanja;
- prenosljivi – ti izdelki ne zahtevajo fizičnega pošiljanja, ampak aktivirajo dodatna polja, kjer lahko zagotovite prenos datoteke. Taki izdelki so: elektronske knjige, PDF revije, fotografije, računalniški programi itn.;
- zunanji ali partnerski izdelki – to so izdelki, ki jih imamo navedene v spletni trgovini, prodaja pa jih neko drugo podjetje;
- spremenljivi – to so izdelki z različnimi atributi, od katerih ima lahko vsak drugačen SKU, ceno, zalogo, sliko itn.

Vsakemu izdelku lahko dodamo naziv, opis, kratek opis, kategorije, oznake, predstavitveno sliko, galerijo in podatke o izdelku. Podatke o izdelku delimo na:

- splošni podatki; to so cena, akcijska cena, status davka in davčni razred,
- zaloga; sem vpišemo šifro za vodenje zaloge (angl. Stock keep unit - SKU), status in količino zaloge,
- dostava vsebuje podatke o teži, dimenziji in razredu dostave,

- povezani izdelki,
- atributi,
- različice, ki jih dodamo glede na izbrane attribute,
- napredni podatki, kot so obvestilo ob nakupu, vrstni red in omogočanje ocenjevanja,
- dodatni podatki, kot so dodatna vsebina in video posnetek.

### 3.8.3 Zaloga

WooCommerce omogoča vodenje zaloge. V nastavitvah lahko onemogočimo vodenje zaloge, nastavimo obvestila za nizko in prazno zalogo ter nastavimo mejo za nizko in prazno zalogo. V spletni trgovini lahko skrijemo izdelke, ki jih nimamo na zalogi, in kupcu prikažemo zalogo. Vsakemu izdelku lahko določimo količino zaloge in omogočimo prodajo tudi, če izdelka nimamo na zalogi.

### 3.8.4 Dostava

WooCommerce omogoča različne načine dostave, kot so: Pošta Slovenije, hitra pošta, brezplačna dostava, osebni prevzem itd. Za pravilno delovanje dostave je potrebno nastaviti območja in za vsako območje izberati načine dostave. Nastavimo lahko tudi razrede dostave za različno velike in težke pakete ter jim nastavimo različne cene.

### 3.8.5 Kuponi

WooCommerce omogoča uporabo kuponov, ki prinašajo popuste na nakupe. Kuponu moramo najprej vpisati kodo, nato mu lahko dodamo še opis. Kuponi omogočajo tri tipe popustov:

- procentualni popust,



- fiksni popust na celoten nakup in
- fiksni popust na izbran izdelek.

Kuponom lahko določimo datum izteka veljavnosti ali pa omejimo število uporabljenih kuponov. Za veljavnost kupona lahko omejimo minimalen in maksimalen znesek nakupa. Kupon pa lahko omejimo tudi na določene izdelke, kategorije in kupce.

### 3.8.6 Darilni boni

WooCommerce ne omogoča avtomatske uporabe darilnih bonov. Za ročno upravljanje z darilnimi boni lahko dodamo izdelek "darilni bon", ki ga nato pošljemo po pošti ali kot kodo pošljemo na elektronski naslov. Za avtomatizirano delovanje darilnih bonov pa je potrebno naložiti brezplačen vtičnik, ki omogoča njihovo uporabo.

### 3.8.7 Plačila

WooCommerce omogoča vse osnovne načine plačila, kot so: bančno nakazilo, plačilo s čeki, plačilo po prevzetju, kreditne kartice in PayPal. Če uporabljamo kreditne kartice in PayPal, je priporočljiva uporaba ponudnika plačilnih storitev kot sta Braintree in Stripe.

### 3.8.8 Davek

WooCommerce omogoča izračun cen izdelkov z davkom in brez davka. Nastavimo lahko različne stopnje davke in jih prilagodimo za vsako območje posebej.

### 3.8.9 Naročila

Tudi pri sistemu WooCommerce ima lastnik trgovine celoten pregled nad naročili in kupci. Pri vsaki spremembi statusa naročila se avtomatsko pošlje elektronsko sporočilo trgovcu in kupcu. Skrbnik spletne trgovine lahko

nastavi obliko elektronskih sporočil in katere osebe prejmejo elektronsko sporočilo ob določeni akciji.

# Poglavje 4

## Magento

### 4.1 Kaj je Magento?

Magento je platforma za izdelavo spletne trgovine (angl. e-commerce platform), ki temelji na odprtokodni tehnologiji, ki spletnim trgovcem ponuja fleksibilnost ter nadzor nad izgledom, vsebino in funkcionalnostjo njihove spletne trgovine. Magento sistem je sistem za upravljanje vsebin, izdelan v programskem jeziku PHP. Temelji na MVVM arhitekturi in je razdeljen na skupine modulov. Vsak modul ločuje področja funkcionalnosti in poskuša biti neodvisen od drugih modulov.

Magento je največji odprtokodni sistem za spletno trgovanje, napisan v programskem jeziku PHP. Platformo je prvotno razvilo zasebno podjetje Varien s sedežem v Culver Cityju v Kaliforniji s pomočjo prostovoljcev. Varien je objavil prvo splošno dostopno različico sistema 31. marca 2008. Nekdanji izvršni direktor podjetja Varien, Roy Rubin je kasneje prodal večji del podjetja e-Bayu, ki je sčasoma odkupil celotno podjetje.

17. novembra 2015 je bil izdan Magento 2, s ciljem zagotoviti nove načine za povečanje uporabniškega posla, nemoteno navigacijo, izboljšanje stopnje konverzij in skupno ustvarjanje prihodkov za lastnike trgovin. Ima dobro organizirana poslovna uporabniška orodja, ki izboljšujejo uporabniško izkušnjo programske opreme. Znatno zmanjšanje težave pri zaklepanju tabel,

izboljšano razširjanje strani, skalabilnost sistema, vgrajeni obsežni odrezki za strukturirane podatke, nova struktura datotek z enostavnim prilagajanjem, poprocesiranje CSS z uporabo programskega jezika LESS, izboljšana zmogljivost in boljša kodna baza so le nekatere izmed prednosti novejših verzij Magenta.

Magenta uporablja sistema MySQL ali MariaDB za upravljanje relacijske podatkovne baze podatkov, programski jezik PHP in ogrodje Zend framework. Uporablja objektno usmerjeno programiranje in arhitekturo model-view-controller (MVC). Magenta za shranjevanje podatkov uporablja tudi Entity-attribute-value model (EAV), poleg tega je Magenta 2 predstavil Model-view-viewmodel (MVVM) vzorec za prikaz spletne trgovine z uporabo knjižnice Knockout. [1]

## 4.2 Zgodovina

Razvoj platforme Magenta se je uradno začel v začetku leta 2007. Sedem mesecev kasneje, 31. avgusta 2007, je bila objavljena prva testna različica (angl. beta version). Podjetje Varien, ki je imelo v lasti Magenta je prej razvijalo sistem osCommerce. Prvoten namen podjetja je bil, da uporabijo osnovo sistema osCommerce za razvoj Magenta. Vendar so se kasneje odločili za ločen razvoj platforme Magenta.

Februarja 2011 je eBay javno sporočil, da je konec leta 2010 vložil v razvoj Magenta in s tem kupil 49% lastniškega deleža družbe.

6. junija 2011 je eBay napovedal, da bo odkupil preostali delež Magenta. Izvršni direktor Roy Rubin in Yoav Kutner sta vodil razvoj platforme. Yoav Kutner je aprila 2012 zapustil Magenta, ker se je vizija podjetja zaradi velikih kadrovske sprememb preveč spremenila.

3. novembra 2015 je Magenta prevzela Permira; zasebna družba za upravljanje delniških skladov. [9]

### 4.3 Predstavitev strukture

Tako kot vsak sistem tudi Magento uporablja svojo strukturno logiko. Strukturna logika Magenta je v primerjavi z WooCommerce precej bolj kompleksna in razvejana. [6]

App mapa je namenjena nadaljnjemu razvoju spletne trgovine. Ta mapa vedno vsebuje design in etc, lahko pa ji dodamo podmapo code in i18n, ki je namenjena prevajanju. V mapi code lahko jedru Magenta dodamo nove funkcionalnosti. Mapa etc vsebuje konfiguracijske datoteke, ki vsebujejo preslikave razredov in nastavitve vmesnikov. Mapa design je ločena na administracijo in sprednji del ter vsebuje datoteke tipa JS, CSS in LESS.

Bin mapa vsebuje datoteko magento, skripto CLI, ki prikaže konzolo z več uporabnimi možnostmi. Iz konzole lahko preprosto nastavite predpomnilnik, izberete Module in med drugim vodite indeksiranje.

Dev mapa vsebuje testne skripte in nekaj orodij za razvoj.

Lib mapa vsebuje več knjižnic, ki niso vsebovane v jedru Magenta. Podmapa internal vsebuje vaše knjižnice na strani strežnika, lahko pa tudi datoteke pisav. Podmapa web vsebuje javascript knjižnice na strani odjemalca. Poleg vaših knjižnic vsebuje tudi jquery, modenizr in requirejs.

V mapi pub so javno dostopne datoteke. Ta mapa mora biti nastavljen kot vaš spletni koren na strežniku (Apache ali Nginx). Ta direktorij vsebuje štiri podmape:

- errors – Vsebuje različne procesorske napake in datotečne predloge (podobno kot napaka 404 ali 503).
- media – Vsebuje vse medijske datoteke, ki jih uporabljamo.
- opt – Vsebuje datoteko, ki omeji dovoljene tipe datotek znotraj mape media.
- static – Vsebuje javascript in css datoteke namenjene za administracijo in sprednjo stran.

Var direktorij vsebuje dinamično generirane datoteke. Vsebuje cache in log datoteke. V primeru, da izbrišete vse znotraj mape var, bo Magento ponovno ustvaril vse datoteke. Vendor direktorij vsebuje večino kode. Ta mapa se je prvič pojavila v verziji 2.0 zaradi uporabe orodja composer, ki je namenjen upravljanju odvisnosti (angl. Dependency Manager for PHP). Znotraj tega direktorija se nahaja jedro Magenta, orodji Symfony in ZendFramework ter moduli.

## 4.4 Arhitektura sistema

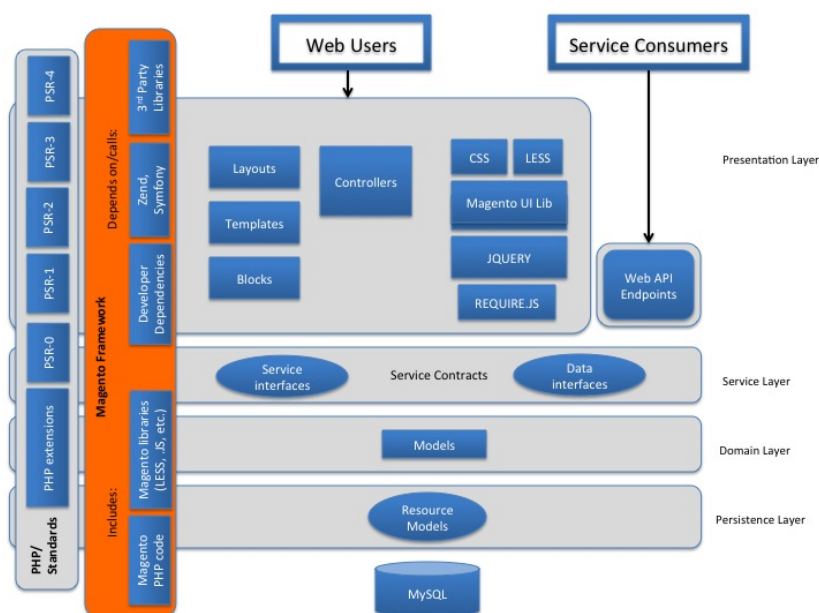
Magento vključuje osnovne arhitekturne principe objektno orientiranega programiranja, ki temelji na programskem jeziku PHP. Ta arhitektura neposredno vpliva na tehnologijo sklada, enostavnost prilagajanja vmesnika, globalne funkcije, ki podpirajo razširljivost, varnost, strategije prilagajanja trgovin, arhitekturne diagrame, razširljivost in modularnost. Arhitektura sistema Magento je sestavljena iz štirih plasti:

- predstavitevna (angl. Presentation layer),
- storitvena (angl. Service layer),
- domenska (angl. Domain layer),
- vztrajnostna plast (angl. Persistence layer).

Predstavitevna plast vsebuje elemente pogleda (postavitev, bloke, predloge) in krmilnike, ki obdelujejo ukaze v uporabniškem vmesniku in iz uporabniškega vmesnika. Predstavitevna koda nadzira interakcijo spletnega uporabnika z izdelkom in njegovim videzom. Za prilagoditev elementov predstavitvene plasti lahko uporabniški vmesnik spremenimo z urejanjem datotek HTML, CSS in PHTML. Storitvena plast zagotavlja povezavo med predstavitveno plastjo in logiko delovanja modela v domenski plasti. To se izvaja s storitvenimi pogodbami (angl. service contracts), ki so definirane z vmesniki PHP.

Domenska plast vsebuje poslovno logiko modula Magento. Običajno ne vsebuje informacij, specifičnih za posamezne vire ali baze podatkov. Njegova primarna funkcija je opredelitev splošnih podatkovnih objektov ali modelov, ki vsebujejo poslovno logiko.

Magento uporablja aktivno strategijo zapisov za vztrajnost. V tem sistemu objekt modela vsebuje model virov, ki zapiše objekt v eno ali več tabel podatkovne baze. Model virov (angl. resource model) izvaja vse zahteve CRUD (create, read, update, delete). Vsebuje tudi SQL kodo za izvajanje teh zahtev.



Slika 4.1: Diagram arhitekturne plasti [2]

Zgornji diagram ilustrira komponente Magenta in prikazuje plasti za vse komponente, dodatne knjižnjice, podprto bazo podatkov in druge tehnologije.

#### 4.4.1 Model View ViewModel (MVVM)

Arhitektura Magenta je bila zasnovana z namenom, da bi bila izvorna koda modularna in razširljiva. Končni cilj tega pristopa je omogočiti enostavno

prilagajanje glede na potrebe vsakega projekta. Prilagajanje pomeni spreminjanje vedenja kode platforme, pri čemer se izvorna koda ne spremeni in ni potrebnih dodatnih spreminjanj kode ob posodobitvah. Arhitektura MVVM zagotavlja bolj robustno ločevanje med sloji Model in View, kot njena nadrejena arhitektura Model View Controller (MVC). Predstavitev posameznih slojev sistema MVVM:

- Model ima poslovno logiko aplikacije in je odvisen od razreda Resource-Model, ki skrbi za povezavo s podatkovno bazo. Modeli s storitvenimi pogodbami (angl. service contracts) omogočajo drugim slojem aplikacije dostop do svojih funkcij.
- View je struktura in postavitve tistega, kar uporabnik vidi na zaslonu - dejanski HTML. To se doseže v datotekah PHTML, ki so razdeljene z moduli. Datoteke PHTML so povezane z vsakim ViewModel v datotekah XML Layout.
- ViewModel komunicira s slojem modela, tako da razkrije samo potrebne podatke sloju View. V Magento 2 to obravnavajo razredi blokov modula. To je bil del vloge krmilnika v arhitekturi MVC. Na MVVM je krmilnik odgovoren samo za ravnanje z uporabniškim tokom, kar pomeni, da sprejema zahteve in sistemu sporoči, da prikaže pogled ali preusmeri uporabnika na drugo pot.

## 4.5 Namestitev

Namestitev Magenta poteka v šestih korakih in ni zelo zahtevna, vendar pa je pred tem potrebno pravilno nastaviti strežnik. Mi smo si pri tem pomagali z uporabo orodja Vagrant, ki pohitri in poenostavi postavitve strežnika. Vagrant je orodje za gradnjo in upravljanje virtualnih okolij. Vse ukaze smo napisali v datoteko Vagrantfile in jo izvršili z ukazom »vagrant up«. Poleg te datoteke smo napisali še skripto bootstrap.sh, ki izvede vse bash ukaze na strežniku. S temi ukazi smo namestili APACHE II, MySQL 5.6, PHP 7.0,



phpMyAdmin, PHP dodatke, nastavili dovoljenja datotek in uredili nastavitve strežnika Apache.

V prvem koraku Magento preveri ali imamo pravilno nastavljen strežnik in nameščene vse komponente. Za delovanje Magento 2 je potrebno imeti nameščen MySQL 5.6 in PHP 5.6 (ali novejšo verzijo), priporočljiva pa je uporaba PHP 7. Namestiti moramo še 15 dodatkov za PHP in Magentu dodeliti pravice za pisanje v direktorije:

- /app/etc,
- /var,
- /pub/media
- in pub/static.

V drugem koraku povežemo Magento s podatkovno bazo, ki jo ustvarimo na strežniku. Pri tem si lahko pomagamo z orodjem phpMyAdmin, ki smo ga namestili na strežnik.

V tretjem koraku določimo spletni naslov naše spletne trgovine in naslov, ki ga bomo uporabljali za dostop do administracije.

V četrtem koraku izberemo časovni pas, valuto in privzeti jezik. Pod dodatnimi nastavitvami lahko izbiramo med 122 moduli, ki nam jih ponudi Magento. Privzeto je dodanih 69 modulov, na izbiro jih imamo še 53.

V petem koraku ustvarimo uporabniški račun skrbnika spletne trgovine.

V šestem koraku samo še potrdimo nastavitve in namestimo Magento.

Po uspešni namestitvi imamo dostop do administracije in do popolnoma delujoče spletne trgovine z dodanimi vzorčnimi izdelki. Magento je v primerjavi z WooCommerce precej večji in bolj kompleksen sistem, zato je v času razvoja priporočljiva povečava omejitve delovnega pomnilnika iz 768 MB na 2048 MB.

## 4.6 Predloga

Predloga je sestavni del sistema Magento, ki zagotavlja dosleden videz spletne trgovine in administracije. Teme so oblikovane tako, da prevzamejo ali prilagodijo izgled, ki ga zagotavljajo moduli in knjižnice.

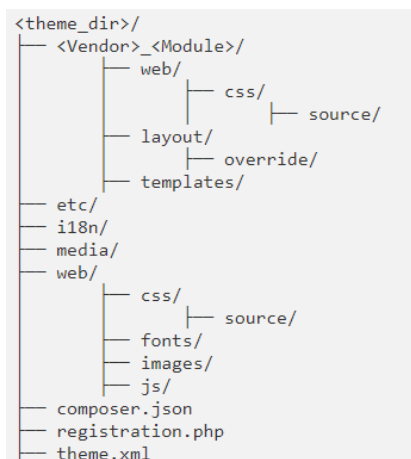
Platforma Magento ponuja dve predlogi: Luma kot predstavitvena predloga in Blank kot podlaga za izdelavo predloge po meri. Ti dve predlogi lahko uporabimo za našo spletno trgovino, ne smemo pa urejati privzetih datotek znotraj predloge, saj se bodo vse spremembe zbrisale ob naslednji posodobitvi. Zato moramo ustvariti novo predlogo, ki je lahko samostojna ali pa podeduje privzeto obliko. Koncept dedovanja v predlogah, ki se izvaja v sistemu Magento, vam omogoča spreminjanje le določenih datotek in dedovanje drugih zahtevanih datotek iz starševske predloge. [7]

### 4.6.1 Struktura predloge

Predloge spletne trgovine se običajno nahajajo v direktoriju `/app/design/frontend/<Vendor>/`. Vgrajeni temi ne bomo našli v tej mapi, ampak ju bomo našli v direktoriju `/Vendor/magento/theme-frontend-<koda_predloge>/`. Vsaka predloga mora biti shranjena v svojem direktoriju, kot prikazuje spodnja struktura. [8]

```
app/design/frontend/<Vendor>/
├── <theme1>
├── <theme2>/
├── <theme3>
└── ...
```

Slika 4.2: Postavitev različnih tem v datottečni strukturi.



Slika 4.3: Osnovna struktura predloge.

Mapa	Opis
/<Vendor>_<Module>	Predloge, postavitve in oblikovanje za določen modul.
/<Vendor>_<Module>/WEB /CSS/SOURCE	Oblikovanje za specifični modul. Splošni slog za modul je v datoteki <code>_Module.less</code> , slog za gradnike pa je v datoteki <code>_widgets.less</code> .
/<Vendor>_<Module>/ LAYOUT	Datoteke za postavitev, ki nadgradijo osnovni modul.
/<Vendor>_<Module>/ LAYOUT/OVERRIDE/BASE	Datoteke za postavitev, ki spremenijo obliko obstoječih postavitev.
/<Vendor>_<Module>/ LAYOUT/OVERRIDE/ <PARENT_THEME>	Datoteke za postavitev, ki spremenijo obliko obstoječih postavitev določenih v nadrejeni temi.
/<Vendor>_<Module> /TEMPLATES	V tej mapi so datoteke, ki urejajo predlogo določeno v modulu ali v nadrejeni temi.
/ETC/VIEW.XML	Ta datoteka vsebuje konfiguracijo slik za vse slike in sličice izdelkov v spletni trgovini.

/I18N	V tej mapi so datoteke tipa csv, ki vsebujejo prevode.
/MEDIA	Ta mapa vsebuje predogled teme.
/WEB	Statične datoteke, ki jih je mogoče naložiti neposredno s spletne trgovine.
/WEB/CSS/SOURCE	Ta mapa vsebuje konfiguracijske datoteke, ki nastavijo globalne spremenljivke iz Magento UI knjižnice.
/WEB/CSS/SOURCE/LIB	Datoteke znotraj te mape spremenijo izgled, določen v Magento UI knjižnicah.
/WEB/FONTS	Pisave, ki se uporabljajo v tej temi.
/WEB/IMAGES	Slike, ki se uporabljajo v tej temi.
/WEB/JS	JavaScript datoteke, ki se uporabljajo v tej temi.
/COMPOSER.JSON	Če vaša tema uporablja Composer, bodo tukaj navedene odvisnosti od modulov.
/REGISTRATION.PHP	Ta datoteka je obvezna, ker registrira temo v sistemu.
/THEME.XML	Datoteka je obvezna, ker razglasi temo kot sistemsko komponento. Vsebuje osnovne meta podatke. Datoteko uporablja sistem Magento, da prepozna temo.

Tabela 4.1: Struktura predloge

## 4.7 Moduli

Modul je skupek programske logike (mapa z bloki, krmilniki, pomočniki, modeli), ki je povezana s specifično poslovno logiko. Za optimalno delovanje sistema modul zajame le eno funkcijo, ki je minimalno odvisna od ostalih

modulov.

Moduli in teme so enote za prilagoditev sistema Magento. Moduli zagotavljajo poslovne funkcije, s podporno logiko, medtem ko teme močno vplivajo na uporabniško izkušnjo in videz trgovin. Obe komponenti imata življenjski cikel, ki jim omogoča namestitve, brisanje in onemogočanje. Z vidika skrbnika in razvijalca so moduli osrednja enota sistema Magento. Magento Framework zagotavlja jedro sistema: PHP kodo, knjižnice in osnovne funkcije, ki jih podedujejo moduli in druge komponente.

Namen vsakega modula je zagotoviti specifične funkcije izdelka z uporabo novih funkcij ali razširitvijo funkcionalnosti drugih modulov. Vsak modul je zasnovan tako, da deluje samostojno, zato vključitev ali izključitev določenega modula običajno ne vpliva na funkcionalnost drugih modulov.

Modul je imenik, ki vsebuje datoteke PHP in XML (bloki, krmilniki, pomočniki, modeli), ki so povezane s specifično poslovno funkcijo.

Moduli se običajno nahajajo v imeniku `app/code` sistema Magento. Standardna postavitev modula `<ModuleName>` v celotno strukturo sistema Magento je `app/code/<Vendor>/<ModuleName>`. Če ustvarjate nov modul za distribucijo, lahko ustvarite nove datoteke in mape znotraj imenika `<ModuleName>`. [3]

### 4.7.1 Struktura modula

Datotečna struktura modula:

- **Api:** Vsebuje vse razrede PHP, ki so izpostavljeni API-ju.
- **Block:** Vsebuje datoteke `ViewModels`, kot del arhitekture MVVM.
- **Controller:** Skrbi za interakcijo uporabnika s sitemom.
- **etc:** Vsebuje konfiguracijske XML datoteke, ki jih modul definira sam.
- **Helper:** Vsebuje datoteke s kodo, ki je uporabljena v več aplikacijskih plasteh. Razredi za pomoč se običajno uporabljajo za pripravo HTML-ja.

- `i18n`: Vsebuje datoteke CSV, ki se uporabljajo za prevajanje.
- `Model`: Za modele in modele virov.
- `Observer`: Vsebuje datoteke, ki opazujejo sistemske dogodke. Običajno, ko je tak dogodek sprožen, opazovalec o tem dogodku obvesti model, ki vsebuje potrebno poslovno logiko za tak dogodek.
- `Setup`: Vsebuje razrede za strukturo baze podatkov modula in nastavitvev podatkov, ki se uporabljajo pri namestitvi ali nadgradnji.
- `Test`: Vsebuje datoteke za testiranje delovanja modula.
- `Ui`: UI elementi, kot so mreže in obrazci, ki se uporabljajo v aplikaciji admin.
- `view`: Datoteke za strukturo (XML) in predloge (PHTML), ki določajo izgled modula v administraciji in spletni trgovini.

Koren modula vsebuje datoteke `composer.json`, `README.md` in `registration.php`. Te datoteke pomagajo pri upravljanju odvisnosti, deklaraciji modula in drugih pomožnih funkcionalnostih.

## 4.8 Sistemski načini delovanja

Magento 2 lahko deluje v treh sistemskih načinih delovanja, ki so namenjeni fazi razvoja in fazi produkcije.

Najpomembnejša sistemska načina sta razvijalski (angl. `Development mode`) in produkcijski način (angl. `Production mode`). Prvi se uporablja v času razvoja in je namenjen hitrejšemu razvoju. Ta način zahteva precej več procesorske moči, saj vedno znova ustvari vse statične datoteke. Vse spremembe, ki jih opravimo so vidne takoj, v primeru izjeme pa izpiše zelo podrobno opisano napako. Drugi se uporablja, ko je razvoj končan in se spletna trgovina prenese na javno spletno mesto. Produkcijski način zagotavlja

	<b>Razvijalski način</b>	<b>Produksijski način</b>	<b>Privzeti način</b>
Predpomnjenje statičnih datotek			X
Prikaz napak	X		
Zapis napak		X	X
Negativni učinek na hitrost delovanja	X		X

Tabela 4.2: Zgornja tabel prikazuje najpomembnejše razlike med sistemskimi načini delovanja.

najvišjo učinkovitost in nikoli ne prikaže napak uporabniku. Ta način je tudi bolj varen, ker ne dovoli pisanja v direktorij docroot. [4]

## 4.9 Osnovne funkcije platforme Magento

### 4.9.1 Izdelki

Magento omogoča dodajanje šestih različnih tipov izdelkov:

- Enostaven izdelek je fizični predmet, ki ga določa koda SKU (angl. Stock Keeping Unit). Enostavni izdelki imajo različno ceno in nadzor vnosa, ki omogoča prodajo različic izdelka. Enostavne izdelke lahko uporabite v povezavi s skupinami, paketi in nastavljivimi izdelki.
- Nastavljiv izdelek izgleda kot en izdelek z izbiro možnosti za vsako različico. Vendar pa vsaka možnost predstavlja ločen preprost izdelek z ločeno kodo SKU, kar omogoča spremljanje inventarja za vsako različico.
- Skupinski izdelek predstavlja več samostojnih izdelkov kot skupina. Ponudite lahko različice posameznega izdelka ali jih združite za promocijo. Izdelke je možno kupiti ločeno ali kot skupino.

- Virtualni izdelki niso opredmeteni izdelki in se navadno uporabljajo za izdelke, kot so storitve, članstva, garancije in naročnine. Virtualne izdelke lahko uporabljamo v povezavi z združenimi izdelki in paketi.
- Skupni izdelki omogočajo strankam, da sestavijo svoje iz različnih možnosti. Snop je lahko darilna košarica, računalnik ali kaj drugega, kar lahko prilagodite. Vsak element v svežnju je ločen, samostojen izdelek.
- Prenosni izdelek je digitalno naložljiv izdelek sestavljen iz ene ali več datotek. Te so lahko naložene na strežniku ali pa jih pošljemo kot URL do datoteke na drugem strežniku.

Vsakemu izdelku moramo nastaviti še osnovne podatke kot so: ime, SKU, cena, slike, davčni razred, količina in kategorije. Poleg cene, lahko vnesemo še akcijsko ceno, ki jo lahko omejimo na določeno obdobje in na izbrane stranke. Za urejanje vsebine in opisa izdelka se uporablja urejevalnik besedila. Ta urejevalnik je po izgledu in po načinu delovanja zelo podoben urejevalniku besedila v programu Microsoft Word 2003. Urejevalnik besedila omogoča celotno manipulacijo z besedilom, izdelavo seznamov in tabel ter dodajanje slik, urejamo pa lahko tudi html in css kodo ter dodajamo attribute in dogodke.

V spletni trgovini Magento lahko poljubno dodajamo skupine atributov, ki vsebujejo skupne lastnosti neke skupine izdelkov. Vsaki skupini atributov nato izberemo attribute, ki jih želimo izpolniti pri tej skupini izdelkov. Attribute je potrebno vnesti ločeno, saj se lahko uporabljajo v večjem številu skupin. Če so vse vrednosti atributa določene, jih lahko nastavimo pri atributu in jih kasneje na strani izdelka le izberemo. Izdelku najprej izberemo eno od nastavljenih skupin atributov in pojavijo se nam vsa polja, ki smo jih predvideli za ta tip izdelka.

Magento uporablja zapleten, a zelo dober način uporabe atributov. Z uporabo skupin atributov izdelke dodatno ločimo po lastnostih, pri tem pa uporabljamo le tiste attribute, ki jih za določen izdelek potrebujemo.



### 4.9.2 Zaloga

Za večje spletne trgovine, ki imajo veliko izdelkov in naročil, je zelo pomembno pravilno upravljanje z zalogo. Magento v primerjavi z drugimi sistemi za spletno trgovanje zelo dobro upravlja z zalogo in omogoča lastniku trgovine popolni nadzor nad zalogo ter omogoča mnogo različnih nastavitev. Magento poleg osnovnih nastavitev zaloge ponuja še možnosti prodaje, ko izdelek ni na zalogi. V tem primeru lahko stranka prejme obvestilo, da bo ta izdelek prispel kasneje. Uporabniku lahko omejimo maksimalno količino za vsak izdelek posebej in minimalno število izdelkov v kompletu. Magento omogoča avtomatsko odstranjevanje izdelkov iz spletne trgovine, ko tega ni več na zalogi, po vnosu nove zaloge pa se izdelek ponovno prikaže v spletni trgovini. V spletni trgovini se vedno prikazuje status zaloge, prikazujemo lahko še število izdelkov na zalogi ali pa prikažemo to število, ko primanjkuje zaloge.

### 4.9.3 Dostava

Študije kažejo, da imajo spletne trgovine, ki strankam na izbiro ponujajo več načinov pošiljanja, večje število naročil od tistih, ki uporabljajo eno samo metodo za vse pošiljke. Magento omogoča vse osnovne načine dostave in izračun stroškov pošiljanja na 4 načine:

- pavšalna stopnja – strošek pošiljanja je za vse enak,
- tabela stopenj – izračuna se minimalen strošek glede na velikost in težo paketa; pri tem načinu dostave se pogosto uporablja več vrst transportov,
- dimenzija in teža – izračuna se strošek glede na volumen in maso paketa,
- brezplačna dostava – stranke lahko paket prejmejo brezplačno pod določenimi pogoji.

Magento sistem lahko enostavno povežemo s sistemom komercialnega prevoznika, kot so: UPS, USPS, FedEx in DHL. Ti prevozniki so v Magento sistemu podprti in jih je potrebno le pravilno nastaviti. Pri dostavi s katerim drugim prevoznikom, je potrebno kupiti oziroma izdelati modul, ki pošilja naročila željenemu prevozniku.

#### 4.9.4 Kuponi

Magento, tako kot ostali večji sistemi za spletno nakupovanje, omogoča uporabo kuponov. Koda kupona lahko zmanjša končno ceno nakupa, če so izpolnjeni določeni pogoji. Kuponi so lahko omejeni na določeno skupino ali pa začnejo veljati, ko je znesek nakupa dovolj visok. Magento omogoča različne načine uporabe kuponov:

- stranka prejme kupon po e-pošti,
- stranka dobi natisnjen kupon,
- kupone prejmejo samo mobilni uporabniki.

Kode kuponov lahko pošiljate po elektronski pošti ali pa so vključene v glasila, kataloge in oglase. Seznam kode kuponov se lahko izvozi in natisne. Kupone v trgovini lahko ustvarite tudi s kodo za hiter odziv, ki jo lahko kupci pregledajo s pametnimi telefoni, to QR kodo pa nato uporabijo za korištenje kupona.

Kupone lahko ustvarimo na dva načina. Prvi način je, da ustvarimo kupon z eno kodo, ki mu nastavimo omejitev uporabljenih kuponov in omejitev uporabljenih kuponov za posameznega kupca. Drugi način je, da ustvarimo množico kuponov, kjer je vsak kupon unovčljiv samo enkrat.

#### 4.9.5 Darilni boni

Magento CE ne omogoča uporabe darilnih bonov kot ločenega tipa izdelkov. Za pravilno delovanje darilnih bonov, bi morali plačati uporabo sistema Magento EE ali pa kupiti modul za uporabo darilnih bonov.

Darilne bone lahko prodajamo tudi brez dodatnih modulov, kot navadne izdelke. Pri tem moramo kupcu poslati darilno kartico, ki vsebuje kodo kupona z željenim zneskom. Kupec pri tem načinu uporabe kuponov ne bo opazil razlike, zato pa bo imel trgovec več dela z upravljanjem kuponov.

#### **4.9.6 Plačila**

Magento CE podpira različne načine plačila. Prvi in najbolj pogost način je plačilo brez povezave. Sem spadajo spletna plačila s čekom, denarno nakazilo in plačilo ob dostavi. Drugi in vedno bolj pogost način je spletno plačilo. To so spletna plačila s kreditnimi karticami in PayPal plačila. Magento ponuja uporabo orodja Braintree, ki prevzame izvajanje transakcije in zagotavlja varno spletno nakupovanje.

#### **4.9.7 Davek**

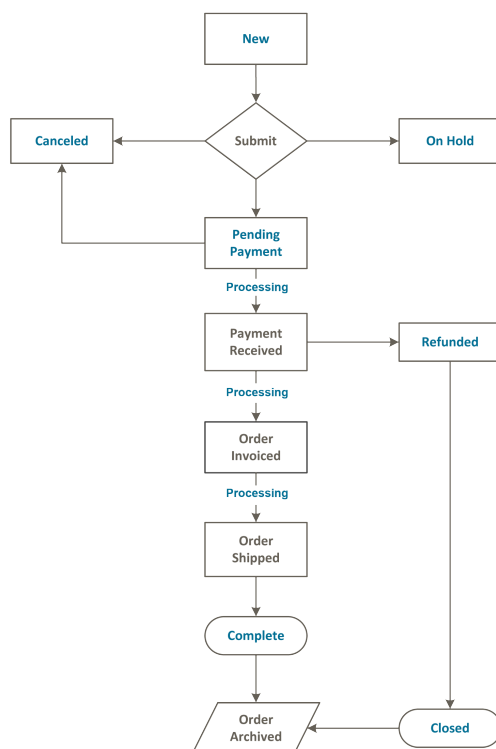
Magento omogoča nastavitve davkov v skladu z zahtevami naše države. Nastavimo lahko davčne razrede za izdelke in skupine strank ter ustvarimo davčna pravila, ki združujejo razrede izdelkov in strank, davčne cone in tarife.

Magento ponuja različne možnosti za določitev davkov. Davčni razredi se uporabljajo za opredelitev "davčnih pravil". Davčna pravila so kombinacija razreda izdelkov, razreda kupcev in davčne cone ter stopnje. Magento prav tako podpira izračun davka na dodano vrednost (DDV) za posle med podjetji v Evropski uniji.

#### **4.9.8 Naročila**

Tako kot vsi ostali sistemi tudi Magento v administraciji prikazuje vse opravljene nakupe. Ker je nakupov pogosto zelo veliko, ima Magento napredno filtriranje po identifikacijski številki, datumu, statusu, znesku, prejemniku in naslovu prejemnika. Vse nakupe je možno tudi izvoziti v formatu CSV in XML.

Vsa naročila imajo status, ki predstavlja stanje naročila v danem trenutku. Magento vsebuje 12 različnih statusov: v obdelavi, sumljivo naročilo, plačilo v teku, pregled plačila, v teku, na čakanju, zaključeno, zaprto, preklicano, preklicano plačilo PayPal, plačilo PayPal v teku in zavrnjeno plačilo PayPal.



Slika 4.4: Proces naročila v Magentu

## 4.10 Magento 1 vs Magento 2

Prva različica Magenta 2 je bila objavljena leta 2015. Za ekipo Magento je bil to nov začetek, ki je temeljil na dobro premišljenem delu. Z novo različico platforme je bila uporabnikom predstavljena povsem nova struktura. Vmesnik je bolj uporabniku prijazen, saj je postal bolj intuitiven. Poleg tega so razvijalci dodali celoten nabor najnovejših spletnih tehnologij, vključno s HTML5, CSS, PHP 5.5, PSR itd. Privzeto JavaScript knjižnico je na-

domestil jQuery. Za izboljšanje varnostnih težav je ekipa Magento dodala podporo plačilnim sistemom kot so Braintree, PayPal in Authorize.net, ki tako postanejo privzeto dostopni. Enterprise Edition podpira tudi CyberSource in WorldPay.

Z izdajo Magenta 2.1 so se vprašanja o uporabnosti in varnosti izboljšala še bolj. Razvijalci so spremenili vmesnike upravljanja, zaradi česar so bile operacije ustvarjanja, spreminjanja kategorij izdelkov in iskanja enostavnejše. Uporabniki PayPal-a ne potrebujejo več zapustiti trgovine, da opravijo spletni nakup. [5]

Struktura imenika Magenta 2 se precej razlikuje od prejšnjih različic Magenta

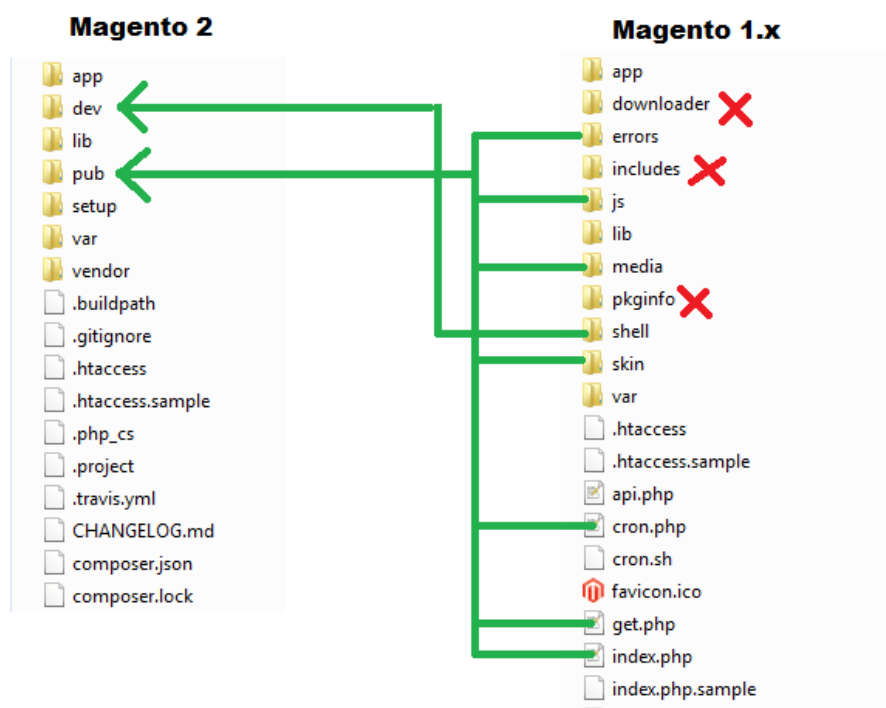
1. Glavne spremembe strukture imenika se prikažejo že v korenski mapi:

- mapi media in errors sta prestavljeni v mapo pub,
- mapi skin in js sta prestavljeni v mapo pub/static, kar omogoča enostavno vzdrževanje statične vsebine prek CDN-ja (Content Delivery Network),
- js, css in slike so razdeljene v mapi pub/static/adminhtml in pub/static/frontend,
- mapa dev je nadomestila mapo shell, vsebuje pa različna orodja za razvijalce, kot so orodja za migracije in teste,
- mapi downloader in includes sta odstranjeni,
- dodana je mapa setup, ki je namenjena procesu namestitve.

#### 4.10.1 Mageto 2 – Prednosti

Magento 2 je poleg novih funkcionalnosti tudi izboljšal šibke točke iz prejšnje različice, kot so hitrost, avtomatski testi, uporabniški vmesnik, modularnost in število konfliktov.

Magento 2 naj bi deloval 20% hitreje od njegovega predhodnika. To je najbolj



Slika 4.5: Datotečna struktura Magenta 1 in Magenta 2.

vidno pri dodajanju izdelkov v košarico in pri zaključku nakupa, kar izboljša tudi zadovoljstvo kupcev. Proces nakupa je optimiziran in poenostavljen. Izboljšal se je uporabniški vmesnik za trgovca, ki omogoča prilagoditev administracije in hitrejše dodajanje in urejanje izdelkov. Magento 2 je postal tudi bolj prijazen uporabnikom mobilnih naprav, saj je vedno več nakupov opravljenih preko mobilnih naprav in tablic. Maksimalno število prikazanih izdelkov se je iz 500.000 zvišalo na 10 milijonov. Izboljšala se je tudi podatkovna baza, ki zdaj omogoča istočasno urejanje izdelkov kar 25 uporabnikom.

#### 4.10.2 Magento EE (Enterprise Edition)

Magento poleg odprtokodne različice ponuja tudi plačljivo verzijo imenovano Enterprise Edition. Ta vsebuje precej funkcionalnosti, ki so pogosto nujno

potrebne pri večjih spletnih trgovinah. Za večino Enterprise funkcionalnosti je mogoče kupiti Community modul. V primeru, da potrebujete le nekaj modulov, se vedno bolj izplača uporabiti odprtokodni sistem in mu dodati module. Problem nastane pri uporabi več modulov, ker je treba združljivost z ostalimi nameščenimi moduli preverjati in testirati vsakič, ko nek modul posodobimo ali naložimo novega. Magento EE vsebuje tudi napredne funkcionalnosti, kot so:

- predpomnilnik strani Varnish,
- porazdeljena podatkovna baza,
- uporaba MySql Cluster,
- boljša organizacija javascript datotek,
- mehanizem čakalnih vrst,
- odložene posodobitve zaloge, ki omogoča hitrejši postopek nakupa.

Končno odločitev med verzijama Enterprise in Community je najbolje prestaviti na konec, ko imamo dobro dodoločeno, kaj točno potrebujemo. V vsakem primeru je možno Magento Community nadgraditi na Magento Enterprise. Možno pa je tudi obratno, torej iz Magento Enterprise narediti downgrade na Magento Community.





## Poglavje 5

# Primerjava najpopularnejših platform za spletno trgovanje

### 5.1 Splošna primerjava

Spodnja tabela podaja splošno primerjavo izbranih treh platform:

	<b>Shopify</b>	<b>WooCommerce</b>	<b>Magento</b>
Programski jezik	Ruby, Liquid	PHP	PHP, Zend
Odprtokodni/ zaprtokodni sistem	zaprtokodni sistem	odprtokodni sistem	odprtokodni sistem
Atributi	omejeno število	neomejeno število	neomejeno število
Vrste izdelkov	enostavni, nastavljeni izdelki	enostavni, združeni, virtualni, spremenljivi, prenosljivi, zunanji ali partnerski izdelki	enostavni, sestavljeni, prenosljivi, virtualni, nastavljeni izdelki, paketi
Nadzorna plošča za skrbnike	enostavna, preprosta za upravljanje trgovin	srednje kompleksna, veliko funkcionalnosti je prilagodljivih	kompleksna, celoten nadzor nad spletno trgovino

Struktura predloge	Spreminjanje delovanja teme zahteva urejanje osnovnih datotek predloge.	Omogoča prilagoditev predloge brez spreminjanja osnovnih datotek predloge.	Omogoča prilagoditev predloge brez spreminjanja osnovnih datotek predloge.
Zmogljivost	Hitro delovanje brez dodatnega dela.	Za hitro delovanje je potrebna dodatna optimizacija.	Za hitro delovanje je potrebna zahtevna dodatna optimizacija in dober strežnik.
Funkcionalnost	Funkcionalnost dodajamo z vtičniki, ki uporabljajo API.	Funkcionalnost dodajamo z vtičniki.	Funkcionalnost dodajamo z moduli.
Zemljevid spletnega mesta	samodejno ustvarjeno	Za samodejno ustvarjen zemljevid potrebujemo brezplačen vtičnik.	samodejno ustvarjen
Podatkovna baza	nimamo dostopa	celoten nadzor	celoten nadzor
Varnost	Za varnost skrbi Shopify, bolj varno kot odprtokodni sistemi, saj je varnostne luknje težje odkriti.	Varnost izboljšamo z rednimi nadgradnjami in dodatnimi vtičniki za varnost.	Varnost izboljšamo z rednimi nadgradnjami in dodatnimi moduli za varnost.
Večjezičnost	ne podpira	potrebni dodatni vtičniki	vgrajeno
Skupna administracija za več spletnih trgovin	ne podpira	vgrajeno	vgrajeno

Trgovina na debelo	omejeno	Delno podpira z uporabo dodatnih vtičnikov.	Popolno podpira z uporabo modula B2B.
Blog	vgrajeno	vgrajeno	z uporabo brezplačnega vtičnika
Ciljna skupina	Male in srednje velike spletne trgovine.	Male in srednje velike spletne trgovine.	Srednje velike in velike spletne trgovine.
Gostovanje	z gostovanjem upravlja Shopify	gostovanje izberemo sami	gostovanje izberemo sami
Navigacija in filtriranje	delno prilagodljivo	popolnoma prilagodljivo	popolnoma prilagodljivo
Upravljanje zalog	Preprost vmesnik za upravljanje zaloge.	Preprost vmesnik za upravljanje z zalogo in izdelki.	Popolnoma prilagodljiv vmesnik, kjer lahko urejate izdelke, njihove kategorije itd.
Licenca	21,5 €/mesec Basic 68 €/mesec Professional 154 €/mesec Unlimited	brezplačna	Magento CE – brezplačna Magento EE – 15.550 €
Zahtevnost razvoja	srednje zahteven	srednje zahteven	zelo zahteven
Podpora	Odlična podpora je dostopna ves čas, zagotavlja jo Shopify.	Podpora je na voljo iz ogromne skupnosti razvijalcev WordPressa.	manjša podporna skupnost
Priljubljenost/delež na trgu	8 % spletnih trgovin uporablja Shopify	24,8 % spletnih trgovin uporablja WooCommerce	20,3 % spletnih trgovin uporablja Magento

Zadnja verzija	Shopify Plus	WooCommerce 3.2.3	Magento 2.2.0
----------------	--------------	----------------------	---------------

Tabela 5.1: Splošna primerjava platform

Spodnja tabela pa podaja primerjavo platform s poslovnega vidika:

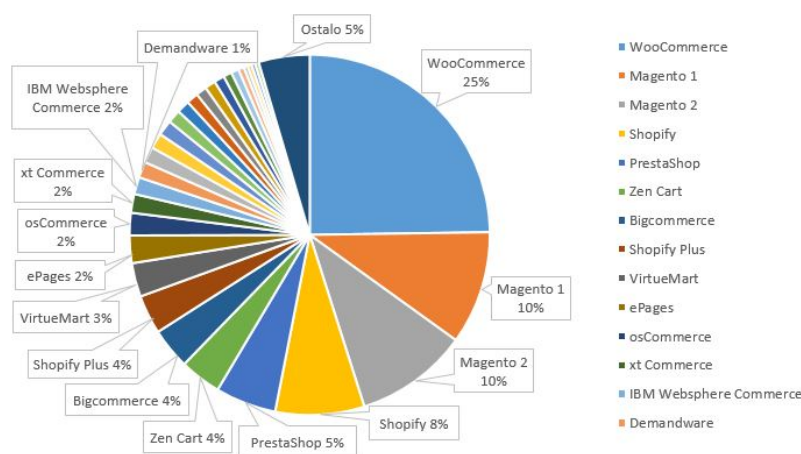
	Shopify	WooCommerce	Magento CE	Magento EE
Obiskovalci/dan	< 10.000	< 10.000	> 10.000	> 50.000
Naročila/dan	< 100	< 100	< 1.000	> 1.000
Prodaja/mesec	< 50.000 €	< 50.000 €	> 20.000 €	> 1.000.000 €
Čas razvoja	2-5 tednov	4-10 tednov	8-15 tednov	15-20 tednov
Stroški razvoja	2.000-5.000 €	3.000-10.000 €	10.000-100.000 €	> 50.000 €
Stroški gostovanja/mesec	> 30 €	> 20 €	> 100 €	> 150 €
Stroški licence	21.5 €/ mesec Basic 68 €/ mesec Professional 154 €/ mesec Unlimited	/	/	15.500 €

Tabela 5.2: Poslovna primerjava platform

## 5.2 Tržni delež

Pri izbiri platforme je zelo pomemben podatek tržni delež. Bolj priljubljene platforme imajo več vtičnikov, modulov in predlog ter precej večjo podporo. Te platforme so tudi bolj varne, saj več ljudi skrbi za varnost, varnostne luknje pa so hitro odpravljene. Spodnji graf prikazuje najpogosteje upora-

bljene platforme za spletno trgovanje. Ta analiza je bila opravljena avgusta 2017 s strani podjetja SimilarTech. [10]



Slika 5.1: Delež posameznih platform za spletno trgovanje.

### 5.3 Optimizacija spletnih strani (SEO)

WooCommerce je zelo prilagodljiv na SEO področju. Z uporabo dobre predloge in dodatnih vtičnikov za SEO optimizacijo, zelo lahko dosežemo dobro SEO oceno. Dodatno prednost nam daje dostop do izvorne kode spletne trgovine. Po zadnjih statističnih podatkih, ki jih je objavila spletna stran ecommerce platforms, ima WooCommerce 97 točk od 100.

Magento je zelo prilagojen za SEO in ponuja veliko obsežnih SEO funkcionalnosti. Po zadnjih statističnih podatkih, ki jih je objavila spletna stran ecommerce platforms, ima Magento 95 točk od 100.

Shopify je odlično optimiziran za SEO. Zagotavlja veliko odličnih SEO funkcij. Po zadnjih statističnih podatkih, ki jih je objavila spletna stran ecommerce platforms, ima Shopify 98 točk od 100.

## 5.4 Cena in stroški

WooCommerce je brezplačen vtičnik za spletno trgovino z brezplačnimi posodobitvami. Poleg tega ponuja WooCommerce cenovno ugodne razširitve. Uradna cena WooCommerce dodatkov se začne pri 42 evrih.

Magento razvoj ponuja dve strukturi cen. Prva je Magento CE, ki jo je mogoče naložiti brezplačno, druga pa Magento EE za podjetja. Cena te izdaje Magenta so začne pri 15.500 €.

Shopify najprej ponuja 14-dnevni preizkus, nato pa se osnovni paket prične pri 21,5 evrih na mesec, napredni Shopify pa pri 154 evrih na mesec. Poleg tega Shopify prejme dobra 2 % pri vsakem opravljenem nakupu.

## 5.5 Varnost

Za varnost v WooCommercu morate imeti certifikat SSL, ki stane od 100 € naprej. Varnost v WooCommerceu poleg tega zahteva močna gesla, vtičnik iThemes in omejene prijave, ki so običajne za WordPress.

Za varnost v Magentu potrebujete varno spletno gostovanje, SSL certifikat in PPI za varne transakcije.

Za varnost v Shopifyju potrebujete le močno geslo, ker so potrdila SSL dodana v naročnino Shopify.

## 5.6 Prijaznost do uporabnika

WooCommerce se ponaša z visoko stopnjo prijaznosti do uporabnika in omogoča enostavno dnevno uporabo, vendar pa je pri postavitvi spletne trgovine bolj zahtevna platforma kot Shopify.

Magento, za razliko od WooCommerca, ni zelo prijazen do uporabnika. To ni najboljša možnost za tiste, ki imajo manjše tehnično znanje.

Shopify je zelo uporabniku prijazen. Ima veliko funkcij, kot so povleci in spusti, preprosta orodja in enostavno urejanje predlog, ki naredijo odlično uporabniško izkušnjo.

## 5.7 Predloge

WooCommerce omogoča, da vsaka WordPress predloga deluje s spletno trgovino. Obstaja preko 1000 predlog, ki jih podjetja lahko izberejo za svojo spletno stran. Poleg tega lahko predloge najdemo na spletnih mestih, kot je ThemeForest. Veliko predlog je brezplačnih, povprečna cena dobre predloge pa je 50 evrov.

Magento ponuja predloge na portalu Magento Connect. Cene predlog se začnejo pri nekaj evrih, povprečna cena pa znaša 80 evrov.

Shopify ponuja 9 brezplačnih predlog in za plačljivo temo pa je potrebno plačati 120 evrov. Shopify nam ponuja 51 plačljivih predlog, mnogo več predlog pa lahko najdemo na portalu ThemeForest.

## 5.8 Pomoč strankam

Zaradi arhitekture odprtega sistema WooCommerce je podpora strankam omejena. V pomoč nam je lahko tudi ogromna skupnost, ki jo najdemo na spletu.

Magento nima podpore strankam, vendar pa se lahko povežete z obsežno skupnostjo razvijalcev in od njih dobimo odgovore na naša vprašanja, povezana s platformo.

Shopify ima precej boljšo podporo, saj je podpora njihovim uporabnikom ves čas dostopna. Skupina za podporo strankam je dostopna po telefonu, elektronski pošti ali preko video pogovora.

## 5.9 Zaprtokodni ali odprtokodni programi za izdelavo spletne trgovine

Na voljo imamo 2 možnosti, ko izbiramo platformo za spletno trgovino – bodisi so to odprtokodni sistemi (npr. Magento in WooCommerce) ali zaprtokodni sistemi (npr. Shopify). Glavna razlika se skriva v gostovanju spletnih

trgovin in možnosti urejanja. Zaprtokodni sistemi gostujejo na oddaljenem strežniku do katerega nimamo dostopa, nadgradnje se delajo avtomatsko, pri urejanju pa nimamo prostih rok. Na drugi strani omogočajo odprtokodni sistemi gostovanje na našem strežniku, določene prilagoditve, nadgradnje in module si lahko ustvarimo sami.

Prednosti zaprtokodnega programa za izdelavo spletne trgovine:

- Manj dela s programiranjem – pri izdelavi spletne trgovine z zaprtokodnim sistemom nimamo nobene potrebe po programerskem znanju.
- Manj stroškov z gostovanjem (spletna trgovina gostuje na tujem strežniku).
- Manj kompleksna izdelava – večina programov za izdelavo spletnih trgovin ima izjemno enostaven uporabniški vmesnik, s katerim spreminjate vsebino oziroma gradite spletno trgovino.

Prednosti odprtokodnega sistema (tj. CMS sistem za izdelavo spletne trgovine):

- Boljše funkcionalnosti – izjemno pestra izbira tem, vtičnikov in modulov, ki nam služijo za nadgradnje so na voljo na spletu, vnaprej pripravljene za namestitev in vklop v CMS sistemu.
- Možnost večjih prilagoditev, modifikacij in konfiguracij – CMS sistem za izdelavo spletne trgovine nam omogoča svobodo pri dodajanju in konfiguriranju dizajna spletne trgovine; če za tem stoji ekipa razvijalcev, ste popolnoma svobodni pri nadgradnjah in dodajanju vtičnikov.
- Program za izdelavo spletne trgovine v CMS obliki nam ponuja redne nadgradnje in posodobitve, ki uporabniško izkušnjo tako lastnika trgovine kot tudi stranke na spletni trgovini močno izboljšujejo.
- Izdelava spletne trgovine s CMS sistemom nam omogoča gostovanje na svojem strežniku, zaradi česar smo lahko svobodni pri velikosti spletne trgovine. Odvisno od našega strežnika lahko na spletno trgovino dodamo neomejeno število izdelkov in nadgradenj. [11]



## 5.10 Zakaj izbrati Shopify?

Shopify je v tej primerjavi platform najmanjši in najenostavnejši sistem, po zmožnostih pa ne zaostaja veliko za WooCommerce. Shopify omogoča, da si spletno trgovino izdelamo sami, tudi če nimamo veliko tehničnega znanja.

Shopify je odlična platforma za majhne trgovine, ki želijo prodajati tudi na spletu in si ne morejo privoščiti večjih investicij v spletno trgovino. Shopify je primeren za trgovine s specifičnimi izdelki, ki jih redkeje vidimo v trgovskih centrih. Primarno je bil Shopify razvit za ameriški trg, ki se precej razlikuje od evropskega. Zdaj je Shopify primeren za uporabo po vsem svetu.

Vse več spletnih trgovin je večjezičnih in podpira nakupovanje z različnimi valutami. To je sicer mogoče doseči z dodatnimi aplikacijami, a je vseeno bolje uporabiti platformo, ki za to ne prebuje nadgradnje. Zato priporočamo Shopify samo tistim, ki trgujejo znotraj lastne države.

## 5.11 Zakaj izbrati WooCommerce?

WooCommerce je najpogosteje uporabljena in zelo hitro razvijajoča se platforma. Razlog za to je WordPress, ki je še vedno vodilni CMS in poganja približno 25 % celotnega spleta. Ta platforma omogoča lastnikom trgovin in razvijalcem celoten nadzor nad sistemom. Glavna prednost WooCommerce so predloge in vtičniki, ki jih je na spletu skoraj neomejeno. Z njimi je ta platforma primerna za skoraj vsako spletno trgovino.

Platforma WooCommerce je primerna za večjezične spletne trgovine z možnostjo izbire valute. WooCommerce omogoča izdelavo srednje velike spletne trgovine z nekaj tisoč izdelki in veliko dodatnimi funkcijami, vendar se bo to poznalo na hitrosti delovanja spletne trgovine.

WordPress omogoča tudi izdelavo več spletnih strani na enotni namestitvi platforme (angl. multisite). Tako lahko izdelamo več spletnih trgovin in jih upravljamo v skupni administraciji.

Z vsemi možnimi razširitvami se lahko WooCommerce primerja s precej večjimi sistemi, ki imajo vse te storitve že vgrajene, vendar taka nadgranja

zahteva precej dodatnega razvoja in testiranja.

## 5.12 Zakaj izbrati Magento?

Magento je največji odprtokodni sistem, kar lahko opazimo že ob namestitvi sistema in pregledu administracije. Obstaja ogromna skupnost razvijalcev, programerjev, vodij projektov in blogerjev, ki skrbijo za varnost platforme in redne posodobitve. Osnovna namestitev podpira napredne funkcije, kot so večjezičnost, nakupovanje v različnih valutah, napredne metode plačevanja in nadzor več spletnih trgovin znotraj enotne administracije (angl. multistore). Magento ima vgrajene splene storitve (angl. Web Services) SOAP v1, SOAP v2, XMLRPC in REST z zaščito OAuth 1.0a, s pomočjo katerih je mogoče spletno trgovino povezati z drugimi sistemi in CRM-ji.

Magento je torej primeren za spletne trgovine, ki poslujejo v večih državah, imajo veliko prometa in delujejo v povezavi z drugimi sistemi.

# Poglavje 6

## Sklep

V diplomski nalogi sem analiziral in primerjal trenutno tri najpogosteje uporabljene platforme za izgradnjo spletne trgovine. Vse tri platforme so se od nastankov do danes zelo razvile in so zelo primerne za izdelavo spletne trgovine.

Pri izdelavi nove spletne trgovine je bolje izbrati večji sistem, ker vsebuje mnoge dodatne funkcionalnosti in ga je lažje nadgraditi. Poleg tega, ga je možno povezati z drugimi sistemi in s tem avtomatizirati precej procesov. Magento je med omenjenimi tremi platformami najboljši, zato ga priporočamo za izdelavo nove spletne trgovine.

V primerih, ko potrebujemo enostavno spletno trgovino v kratkem časovnem roku in imamo na voljo majhen proračun, je bolje izbrati platformi Shopify in WooCommerce. Prednost Shopifyja je ta, da ne potrebujemo tehničnega znanja za postavitve spletne trgovine. Če želimo spletno trgovino po meri, je čas razvoja krajši kot pri WooCommerce. Prednost WooCommerca so izjemno nizki stroški vzdrževanja, predloge in vtičniki, ki so relativno poceni, pogosto tudi brezplačni, ter dejstvo, da je izdelava nove spletne trgovine cenejša kot pri uporabi enakovrednih sistemov.



# Literatura

- [1] Jonathan Bownds Bret Williams. *Mastering Magento 2*. Packt Publishing, 2 edition, 2016.
- [2] Diagram arhitekturne plasti. Dosegljivo: [http://devdocs.magento.com/guides/v2.3/architecture/archi\\_perspectives/arch\\_diagrams.html](http://devdocs.magento.com/guides/v2.3/architecture/archi_perspectives/arch_diagrams.html), 2017. [Dostopano: 21. 10. 2017].
- [3] Magento arhitekture. Dosegljivo: [http://devdocs.magento.com/guides/v2.0/architecture/archi\\_perspectives/components/Modules/mod\\_intro.html](http://devdocs.magento.com/guides/v2.0/architecture/archi_perspectives/components/Modules/mod_intro.html), 2017. [Dostopano: 27. 10. 2017].
- [4] Magento arhitekture. Dosegljivo: <http://rebeccabrocton.com/switch-magento-2-system-modes/>, 2017. [Dostopano: 13. 10. 2017].
- [5] Magento community edition. Dosegljivo: <https://blog.aheadworks.com/magento-commerce-magento-open-source/>, 2017. [Dostopano: 13. 10. 2017].
- [6] Magento directory structure. Dosegljivo: <https://www.andrewversalle.com/blog/magento-2-directory-structure>, 2017. [Dostopano: 8. 10. 2017].
- [7] Magento predloga. Dosegljivo: <http://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-overview.html>, 2017. [Dostopano: 23. 10. 2017].

- 
- [8] Struktura predloge magento. Dosegljivo: <http://devdocs.magento.com/guides/v2.2/frontend-dev-guide/themes/theme-structure.html>, 2017. [Dostopano: 24. 10. 2017].
- [9] Wikipedia - magento. Dosegljivo: <https://en.wikipedia.org/wiki/Magento>, 2017. [Dostopano: 7. 10. 2017].
- [10] Market share. Dosegljivo: <https://www.similartech.com/categories/ecommerce-platforms>, 2017. [Dostopano: 21. 10. 2017].
- [11] Spletnik cms sistemi. Dosegljivo: <https://spletnik.si/blog/program-za-izdelavo-spletne-trgovine-cms-sistem/>, 2017. [Dostopano: 21. 10. 2017].
- [12] About woocommerce. Dosegljivo: <https://woocommerce.com/features/>, 2017. [Dostopano: 25. 9. 2017].
- [13] Woocommerce hooks. Dosegljivo: <https://developer.wordpress.org/plugins/hooks/>, 2017. [Dostopano: 26. 9. 2017].
- [14] Woocommerce shortcodes. Dosegljivo: <https://developer.wordpress.org/plugins/shortcodes/>, 2017. [Dostopano: 26. 9. 2017].