

Izrada i primjena CMS tehnologija u web okruženju

Vodopija, Neven

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:958816>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-26**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Neven Vodopija

Izrada i primjena CMS tehnologija u web okruženju

Diplomski rad

Mentor: izv. prof. dr. sc. Božidar Kovačić

Rijeka, rujan 2019.

Sažetak

Ovaj diplomski rad prikazuje svrhu, primjenu i značajke sustava za upravljanje sadržajem odnosno CMS-ova u web okruženju, te izradu jednostavnog primjera CMS-a. Rad je podijeljen na devet cjelina koje opisuju funkcioniranje CMS-ova, njihove prednosti i nedostatke a navedeni su i najpoznatiji CMS sustavi i njihove funkcionalnosti.

U uvodnom dijelu, kratko su opisani sustavi za upravljanje sadržajem te njihova povijest i nastanak. Glavni dio rada sastoji se od dva djela. Prvi dio govori o svrhi i značajkama sustava za upravljanje sadržajem u web okruženju, te njihove prednosti i nedostaci. Tu su navedeni i najkorišteniji web CMS sustavi te je u kratko opisana njihova uloga na tržištu. Drugi dio glavnog djela ovog rada opisuje izradu jednog jednostavnog primjera CMS sustava od početka do kraja. Posljednji dio rada govori o aktualnoj paradigmi i novoj tehnologiji CMS-ova i završen je zaključkom.

Ključne riječi

CMS (engl. *Content Management System*), Web sjedišta, WordPress, Headless CMS

Sadržaj

Sažetak	2
Ključne riječi	2
Sadržaj.....	3
Uvod	5
1. Sustav za upravljanje sadržajem ili CMS (engl. <i>Content Management System</i>).....	6
2. Povijest i nastajanje CMS-a	7
2.1. Web 1.0 – Upravljanje statičkim web sadržajem	8
2.2. DOM i dinamička HTML revolucija	8
2.3. Web 2.0 i uloga CMS-a	9
2.4. Početak monolitnog CMS-a.....	9
2.5. CMS otvorenog koda i okviri (engl. <i>Frameworks</i>).....	10
2.6. Web 3.0.....	10
2.7. Headless CMS i prelazak paradigme na Omnikanal	11
3. Svrha, struktura i značajke CMS-a.....	12
3.1. Struktura CMS-a.....	13
3.1.1. Content Management Application - CMA	13
3.1.2. Content Delivery Application - CDA	13
3.2. Značajke CMS-a.....	14
3.2.1. Automatizirani predlošci	14
3.2.2. Kontrola pristupa	14
3.2.3. Skalabilna ekspanzija.....	15
3.2.4. Jednostavno uređivanje sadržaja.....	15
3.2.5. Skalabilni skup značajki.....	15
3.2.6. Nadogradnje	15
3.2.7. Upravljanje tijekovima rada	15
3.2.8. Suradnja	15
3.2.9. Delegacija.....	16
3.2.10. Upravljanje dokumentima	16
3.2.11. Virtualizacija sadržaja.....	16
3.2.12. Udruživanje sadržaja.....	16
3.2.13. Višejezičnost.....	16
3.2.14. Verzije	16
4. Prednosti i nedostaci CMS-a.....	17
4.1 Prednosti	17
4.2 Nedostaci	18

5.	Najkorišteniji web CMS sustavi.....	19
5.1.	Wordpress	20
5.2.	Joomla!	22
5.3.	Drupal.....	24
6.	Alati za razvoj CMS-a	26
6.1.	HTML.....	26
6.2.	CSS.....	27
6.3.	PHP.....	27
6.4.	MySQL	29
6.5.	JavaScript.....	29
7.	Izrada CMS sustava.....	31
7.1.	Kreiranje strukture web sjedišta.....	31
7.2.	Kreiranje baze i povezivanje sa bazom podataka.....	32
7.3.	Dohvaćanje članaka iz baze podataka	34
7.3.1.	Prikazivanje članaka na index.php.....	35
7.3.2.	Dohvaćanje na stranici članka	36
7.4.	Dohvaćanje slika iz baze podataka.....	37
7.5.	Upravljačko sučelje	38
7.5.1.	Prijava	38
7.5.2.	Struktura upravljačkog sučelja	40
7.5.3.	Dodavanje članaka.....	41
7.5.4.	Brisanje članaka.....	43
7.5.5.	Dodavanje slika	45
7.5.6.	Brisanje slika	46
7.5.7.	Odjava	46
8.	Headless CMS.....	47
9.	Zastarjelost standardnih CMS sustava u odnosu na Headless CMS sustave	49
9.1.	Standardni CMS ograničava kreativnost.....	49
9.2.	Standardna CMS sučelja previše su složena	49
9.3.	Standardni CMS je preskup	49
9.4.	Standardni CMS nije siguran	50
9.5.	Standardni CMS ne odgovara arhitekturi.....	50
	Zaključak.....	51
	Popis slika	52
	Izvori sa WWW-a.....	53
	Literatura.....	54

Uvod

Zadatak ovog rada je bio opisati primjenu sustava za upravljanje sadržajem u web okruženju, te izraditi jednostavan primjer jednog sustava za upravljanje sadržajem odnosno CMS-a. Kod objašnjavanja i opisivanja značajki CMS-a korištenja je razna literatura sa web-a te osobna mišljenja iz vlastitog iskustva. Primjer jednostavnog CMS sustava napravljen je pomoću literature i video materijala na web-u te je smješten na testni server tvrtke Blockcode Studio j.d.o.o. Slikovni primjeri priloženi uz rad prikazuju tijek izrade CMS-a te njegovu funkcionalnost. Osim raznih stranica i članaka korišteni su i razni Youtube tutoriali za prikupljanje što više korisnih informacija. Prikupljene informacije omogućile su detaljnije obrazloženje svrhe, povijesti, načina funkcioniranja i korištenja sustava za upravljanje sadržajem. Zadnji dio rada govori o njihovoj ulozi u skoroj budućnosti te prelazak na neke nove paradigme.

Motivacija za odabir ove teme proizlazi iz želje za novim znanjem i boljim razumijevanjem uloge CMS sustava na web tržištu. Vizualno reprezentativna web stranica ili web trgovina je u današnje vrijeme „must have“ za svako poduzeće ili obrt. Radeći u tom području godinu dana shvatio sam da korisnici sve više žele biti uključeni u proces izrade i dorade web stranica i web trgovina. CMS-ovi tu nude brojne mogućnosti koje olakšavaju posao programera a i korisnika.

1. Sustav za upravljanje sadržajem ili CMS

CMS (engl. *Content management system*) je softverska aplikacija koji omogućuje kreiranje, upravljanje i modificiranje digitalnog sadržaja. Skraćenica za „Content Management System“ u prijevodu znači sustav za upravljanje sadržajem.¹ U najširem smislu odnosi se na svako rješenje koje omogućuje klasifikaciju, organizaciju, povezivanje i svaki drugi oblik uređivanja sadržaja. Iako se pojam može koristiti za manualne procese upravljanja sadržajem, danas se u prvom redu primjenjuje za različita programska rješenja koja omogućuju napredno upravljanje velikim brojem informacija. CMS sustavi se koriste pri sinkronizaciji podataka iz više izvora, za izvršavanje kolaborativnih projekata, za organizaciju rada u korporacijskim okruženjima i slično.

CMS se također koristi za upravljanje sadržajem web sjedišta. Takozvani „**Web content management system**“ ili **WCMS**. WCMS je CMS dizajniran za upravljanje sadržajem web sjedišta. Web sadržaj uključuje tekst, grafiku, slike, video, zvuk, mape i programski kod koji prikazuje sadržaj ili vrši interakciju sa korisnikom. Korisnici mogu pohranjivati slike pomoću CMS-a, kao i upravljati dokumentima. U prošlosti su većina web sjedišta bila statična. Ako su bile potrebne promjene u sadržaju, programeri su morali ručno napraviti te promjene. Danas postoji potreba za brzim promjenama, a CMS-ovi to omogućuju.

CMS-ovi pomažu krajnjim korisnicima koji nemaju profesionalne vještine poput razvijanja web sjedišta ili poznavanja programskih jezika da učinkovito upravljaju sadržajem bez traženja profesionalne pomoći. Korisnici mogu izraditi web stranice te ih mijenjati po svojim potrebama. Naprednim funkcijama i brzim razvojem, CMS sustavi nametnuli su se kao najvažniji nositelji razvoja u web-industriji. Glavne osobine dobrih CMS sustava su jednostavnost korištenja, preglednost podatkovne strukture, prilagodljivost za različite svrhe i visok nadzor nad cjelokupnim procesom objave sadržaja

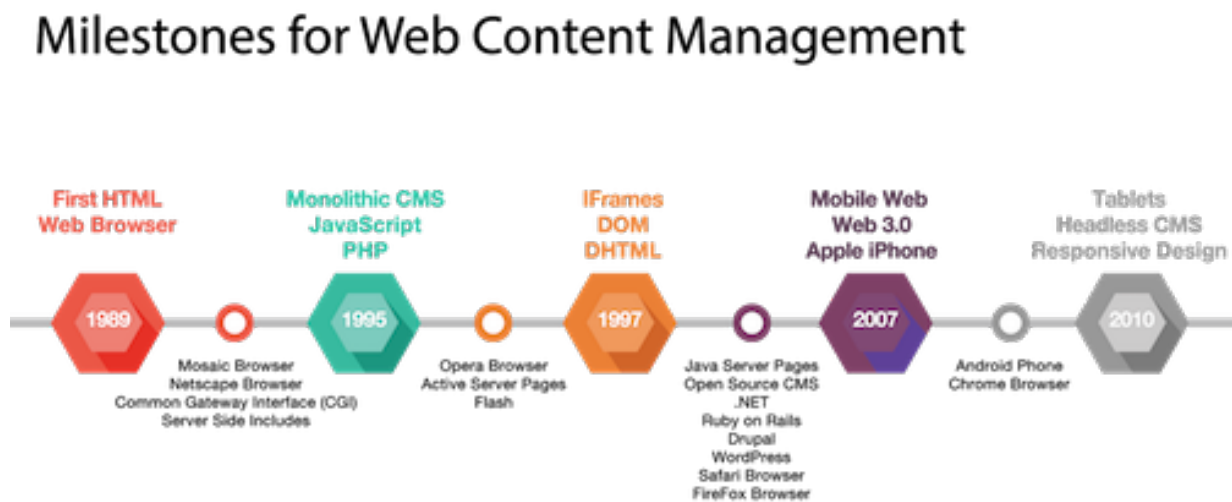
Najpopularniji CMS-ovi su također WCMS-ovi. Na temelju statistike tržišnih udjela, najpopularniji CMS je WordPress, koji koristi preko 28% svih web stranica na internetu, a 59% svih web stranica koristi neki poznati sustav za upravljanje sadržajem. Među najpopularnije CMS-ove spadaju još Joomla i Drupal.

¹ Prema <https://hr.wikipedia.org/wiki/CMS>

2. Povijest i nastajanje CMS-a

²Kaže se da je povijest sustava za upravljanje sadržajem duga je kao i povijest samoga "sadržaja", a ona nastaje u isto vrijeme kao i ljudsko znanje. Kao posljedica želje za efikasnim upravljanjem sadržajima razvijale su se posebne tehnike organizacije i pohranjivanja podataka. Početke toga nalazimo u prvim svitcima, knjigama i zbirkama, a nešto kasnije u razvoju knjižnica i arhiva. Informatičkim razvojem u potpunosti se promijenio pristup upravljanju sadržajem. Umjesto ručne i mehaničke obrade, računala su omogućila iznimno brz i precizan način obrade podataka.

Da bi mogli razumjeti kako su sustavi za upravljanje sadržajem prvi put došli na scenu i zašto postoje različite vrste, osvrnuti ću se na to kako je sadržaj evoluirao na webu (slika 1).



Slika 1 - Vremenska crta razvoja WCMS-a

² Prema <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms>, History of Content Management Systems and Rise of Headless CMS

2.1. Web 1.0 – Upravljanje statičkim web sadržajem

Web 1.0 je pojam koji se odnosi na prvu fazu razvoja na World Wide Webu koju karakteriziraju jednostavne statične web stranice. Godine 1989. Tim Berners-Lee predložio je internetski baziran hipertekstualni sustav HTML te je krajem 1990. kreirao preglednik i poslužiteljski softver. HTML potječe iz SGML-a (engl. *Standard Generalized Markup Language*), koji predstavlja Standardni generalizirani označni jezik, a kreirali su ga Charles F. Goldfarb, Ed Mosher i Ray Lorie 1970-ih IBM-u. Prve web stranice bile su jednostavne HTML tekstualne datoteke. Koristio se FTP program za kopiranje datoteka u direktorij pod pokrenutim web poslužiteljem. Godine 1993. preglednici Mosaic počeli su podržavati slike koje bi se mogle pojaviti zajedno s tekstom, kao i statične stranice slične brošuri koje dijele informacije o tvrtki i proizvodima.

Početakom 1990-ih, prvi korak prema upravljanju sadržajem na web stranici došao je sa SSI-om (engl. *Server Side Includes*). SSI je omogućio da dijelovi web sjedišta ostanu odvojeni od glavnog sadržaja, kao što je izbornik web sjedišta ili podnožje. Otprilike u isto vrijeme, Common Scene Interface je došao na scenu koji je omogućio kreiranje interaktivnih obrazaca.

Već 1990. Tim Berners-Lee rekao je da je odvajanje strukture dokumenta od izgleda dokumenta cilj HTML-a. Håkon Wium Lie je vidio potrebu za programskim jezikom stila za web. Kasnije se Lie-u pridružio i Bert Bos koji je izradio prilagodljiv preglednik sa stilskim listovima. Do 1995. World Wide Web (W3C) je pokrenut, a Lie i Bos su zajedno radili na W3C na preporukama prvih stilova.

U kolovozu 1996. dolazi Microsoftov Internet explorer 3, prvi komercijalni preglednik koji podržava CSS (engl. *Cascading Style Sheets*). Godine 1997. Microsoft je uveo iframe-ove koji su omogućili podjelu HTML prozora preglednika na segmente, pri čemu svaki okvir prikazuje drugačiji dokument koji se može koristiti za prikazivanje sadržaja s drugih web sjedišta te je bio popularan za prikazivanje oglasa i banneri. Oznaka iframe donijela je sigurnosne, navigacijske i optimizacijske probleme koji su na kraju riješeni.

2.2. DOM i dinamička HTML revolucija

Do prekretnice dolazi 1997. pojavom dinamičkog sadržaja uvođenjem DOM-a (engl. *Document Object Model*). DOM definira logičku strukturu dokumenata koja omogućuje identificiranje i programsko kontroliranje dijelova dokumenta. DOM je sučelje za programiranje aplikacija (API) za HTML i XML dokumente. Na primjer, DOM omogućuje pristup stilovima HTML elemenata kao što je cijelo tijelo (body) ili sekcija (div) na stranici i upravlja njima.

Dinamički HTML koji koristi asinkroni JavaScript i XML, koji se obično naziva Ajax, bio je revolucionarni proboj koji je dopuštao programerima da traže i primaju podatke za ažuriranje web stranice bez ponovnog učitavanja stranice.

2.3. Web 2.0 i uloga CMS-a

Dinamička isporuka sadržaja donijela je nove načine predstavljanja i interakcije sa sadržajem na webu, s naglaskom na društvena web sjedišta. Pojam Web 2.0 pomogao je u definiranju onoga što se naziva i participativnom društvenom mrežom. Web 2.0 se također odnosi na porast sadržaja koji generiraju korisnici i jednostavnost korištenja kako bi web-sjedišta radila s drugim proizvodima i sustavima.

Kako je web prešao iz statičkih brošura na interaktivna web sjedišta s dinamičnim sadržajem, želja za suradnjom i svježim, relevantnim sadržajem rasla je, a potreba za upravljanjem sadržajem došla je u prvi plan. Web sjedišta je bilo potrebno svakodnevno ažurirati, a različiti ljudi željeli su dodavati i uređivati sadržaj. Na primjer, odjel za marketing želi ažurirati promotivni materijal, ljudski resursi moraju objavljivati nova radna mjesta, odjel za odnose s javnošću treba objaviti priopćenja za javnost, odjel za dokumente mora objaviti dokumentaciju o proizvodu, odjel za podršku želi komunicirati s korisnicima na mreži, i tako dalje. Uloga sustava za upravljanje sadržajem bila je pružanje mogućnosti za više korisnika s različitim razinama dopuštenja da upravljaju sadržajem za web sjedišta.

2.4. Početak monolitnog CMS-a

Bilo je očito da je potreban sustav koji će omogućiti grupama ljudi da upravljaju i dostavljaju sadržaj na web. Monolitni CMS je sustav koji uključuje sve što je potrebno za upravljanje i objavljivanje sadržaja na webu. Ovaj tip CMS-a je povezan sustav, što znači da je sveobuhvatno rješenje za upravljanje sadržajem.

Osnovan 1985. godine, FileNet se smatra prvim sustavom koji je bio pravi sustav za upravljanje sadržajem. Godine 1995. FileNet je uveo kompletan integrirani paket za upravljanje dokumentima s prikazom dokumenata i radnim procesom. Tvrtka Vignette je došla na scenu krajem 1995. s ciljem da objavljivanje na webu postane pristupačnije i personalizirano, te se njoj obično pripisuje podrijetlo izraza "sustav za upravljanje sadržajem". Godinu dana kasnije Vignette predstavlja StoryBuilder. Mnogi korporativni CMS-ovi počeli su se pojavljivati u to vrijeme, uključujući Interwoven (1995), Documentum (1996), FatWire (1996), FutureTense (1996), Inso (1996) i EPiServer (1997).

2.5. CMS otvorenog koda i okviri (engl. *Frameworks*)

Do ranih 2000-ih sustavi za upravljanje sadržajem dominirali su webom. Pojavljuju se sustavi i okviri za upravljanje sadržajem otvorenog koda. Okvir ili „Framework“ je programska biblioteka unaprijed napisanog koda. OpenCMS, PHP-Nuke, Mambo, WordPress, Drupal, Plone i Joomla nude besplatne alternative za upravljanje sadržajem. WordPress je stekao popularnost kao open source rješenje fokusirajući se na isporuku sadržaja bloga i dopuštajući razvojnim programerima trećih strana dodavanje prilagodbi i proširenja. U 2006. Alfresco je ponudio besplatnu alternativu za upravljanje sadržajem u korporacijskom svijetu.

Počevši od 2003., CMS-ovi za jednostavnu izradu i korištenje web sjedišta nude predloške za osobe bez programerskog iskustva kao što su WordPress (2003), SquareSpace (2003), a zatim Weebly (2006) i Wix (2006). Iako Weebly i Wix nisu čisti sustavi za upravljanje sadržajem, te su platforme omogućile izradu male, jeftine web lokacije koja ne zahtijeva poznavanje HTML-a, CSS-a i kodiranja.

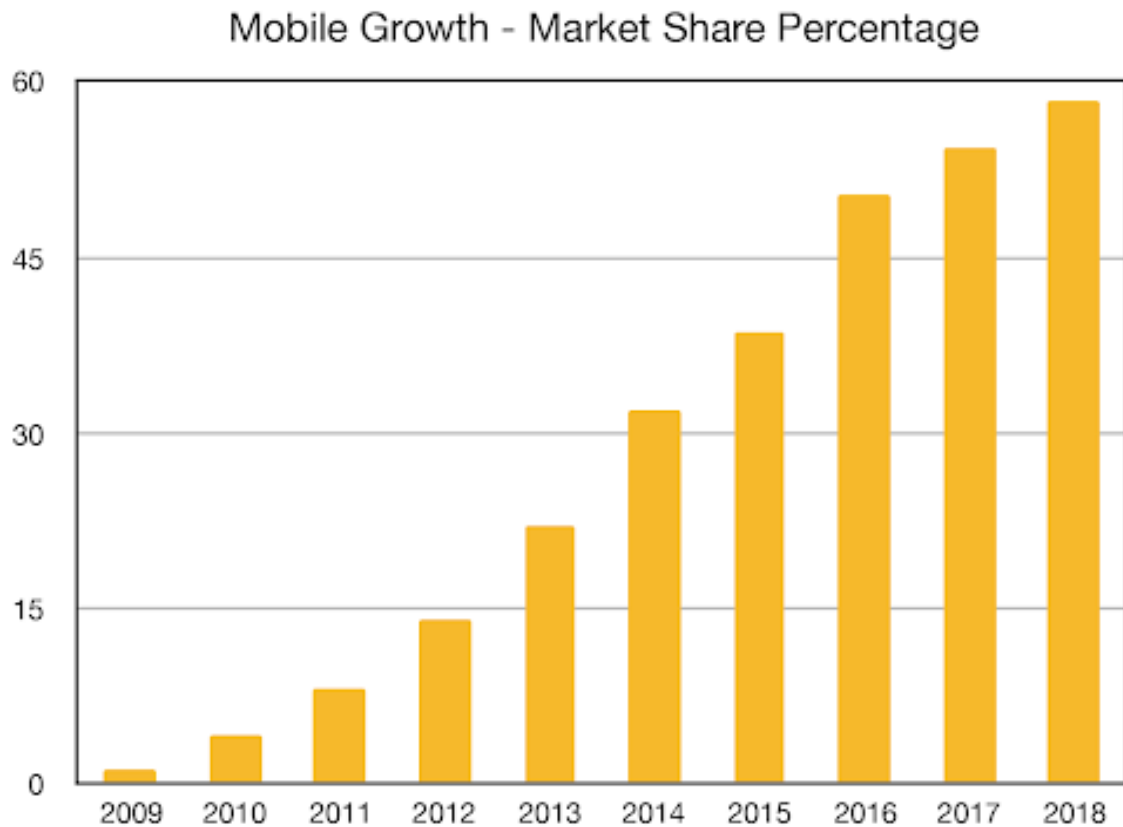
2.6. Web 3.0

Krajem devedesetih i početkom 2000-ih Nokia Symbian, Palm i Blackberry mobilni uređaji omogućili su pristup webu. Međutim, sve do uvođenja iPhone-a 2007. i Android pametnog telefona 2008. godine, mobilni telefoni zaista nisu utjecali na isporuku web sadržaja. 2010. godine pojavili su se pametni tableti. REST API-ji i JSON format podataka bili su od vitalnog značaja za isporuku sadržaja na mobilne uređaje. Ovaj mega trend isporuke sadržaja na mobilne uređaje pokrenuo je mobilnu eru na webu, koja se također naziva i Web 3.0. Do početka 2014. uporaba mobilnog interneta premašila je upotrebu interneta na računalima u SAD-u. Slika 2 prikazuje porast tržišnog udjela u postocima pretrživanja weba na mobilnim uređajima od 2009.-2010. godine.

Ovaj porast potrošnje sadržaja mobilnim uređajima predstavljao je problem za monolitni CMS koji je izričito stvoren za isporuku web sadržaja stolnim i prijenosnim računalima. Nije bilo moguće pouzdano isporučiti sadržaj za desktop i mobilne uređaje. Kako bi se odgovorilo na porast korištenja mobilnog weba, programeri su počeli stvarati desktop i mobilne verzije svojih web sjedišta. Mobilne verzije su bile u suštini verzije za stolna računala bez nekih pojedinih dijelova.

Web sjedišta za mobilne uređaje bile su na zasebnoj poddomeni i nazvane su "m.dot" budući da su poddomene završavale s ".m".

Godine 2010. Ethan Marcotte predstavio je pojam “responzivni dizajn” koji je promicao promjenu mišljenja od fiksnog dizajna za web sjedišta do fluidnih i prilagodljivih izgleda. Kako bi ispunio obećanje o odgovornom dizajnu, W3C je stvorio medijske upite kao dio specifikacije CSS3. Medijski upit omogućuje programerima da utvrde vrstu uređaja i pregledaju fizičke karakteristike uređaja, kao što je veličina zaslona. Na primjer, pomoću CSS-a moguće je koristiti pravilo @media kako bi odredili koja se veličina zaslona koristi i uključiti blok svojstava CSS-a za taj uređaj.



Slika 2 - Porast tržišnog udjela pretraživanja weba na mobilnim uređajima od 2009. do 2018. godine

2.7. Headless CMS i prelazak paradigme na Omnikanal

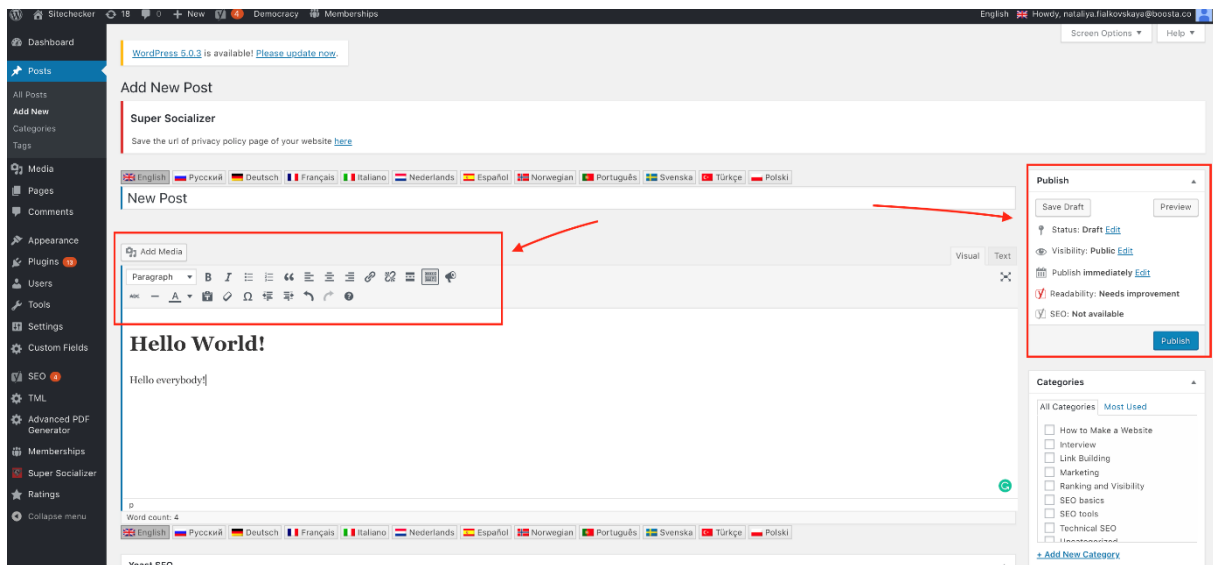
Riječ "omni" znači "sve stvari" na latinskom, tako da se omnikanal odnosi na sve moguće kanale. Baš kao što je mobilni kanal ometao isporuku web sadržaja prije 10 godina, novi kanali, kao što su pametni satovi, igraće konzole, glasovno aktivirani uređaji kao što su Amazonovi Echo (Alexa) i Google početna stranica, stalno se pojavljuju i predstavljaju probleme u isporuci sadržaja za tradicionalni CMS. Pomak paradigme, od isporuke sadržaja za nekoliko kanala do prave isporuke omnikanal sadržaja koji je dovoljno fleksibilan da podrži bilo koji kanal sutrašnjice, zahtijevaju bolje rješenje, a to je „Headless CMS“. Headless CMS je detaljnije opisan u nastavku ovog rada.

3. Svrha, struktura i značajke CMS-a

³CMS-ovi su korisni iz sljedećih perspektiva:

- Koristite specifične upravljačke programe za rješavanje određenih problema ovisno o vrsti web sjedišta i njegovim funkcijama;
- Korištenje CMS-a pomaže vlasnicima web sjedišta pri izradi i brisanju dijelova web sjedišta, uređivanju sadržaja web sjedišta bez sudjelovanja stručnjaka. To je i glavna prednost u odnosu na statične web stranice.
- Rad web sjedišta kontinuirano testira više korisnika. Pogreške i ranjivosti web sjedišta brzo se eliminiraju, a web sjedište koristi najnovija i provjerena rješenja.
- Troškovi vremena na izradi web sjedišta znatno su smanjeni jer se programer ne mora usredotočiti isključivo na tehničke zadatke kao što su “kako stvoriti novosti” ili “kako naučiti CMS da pretražuje robu u katalogu”, nego isključivo na informacijske i vizualne komponente.

Dobar primjer CMS-a je WordPress. Na primjer, prilikom stvaranja dijela sadržaja bez korištenja sustava za upravljanje sadržajem, moramo ručno kreirati HTML datoteku. Za običnog korisnika koji nije upoznat za HTML-om to može biti dosta komplicirano. WordPress omogućuje stvaranje sadržaja korištenjem sučelja koje je više prilagođeno korisnicima jer izgleda kao stranica Microsoft Word (slika 3). Stoga je stvaranje sadržaja uz pomoć CMS-a mnogo lakše.



Slika 3 - WordPress sučelje za objavu članka

Sličan pristup koristi se i kod učitavanja slika ili upravljanjem medijima. U tom slučaju korisnik ne mora izravno komunicirati sa svojim poslužiteljem nego to obavlja WordPress.

³ Prema <https://sitechecker.pro/what-is-cms/>, What Is CMS and How to Choose It for Your Website?

3.1. Struktura CMS-a

Dekompozicija CMS-a na komponente pomaže bolje razumjeti značenje CMS-a. Iz tehničke perspektive, podijeljen je u dva dijela: aplikaciju za upravljanje sadržajem CMA (engl. *Content Management Application*) i aplikaciju za isporuku sadržaja CDA (engl. *Content Delivery Application*). Ta dva sustava pomažu u održavanju web sjedišta.

3.1.1. Content Management Application - CMA

CMA se koristi za dodavanje sadržaja na web stranici i upravljanje istima. To je front-end korisničko sučelje koje omogućava korisniku, čak i sa ograničenim znanjem web developmenta i dizajna, dodavanje, mijenjanje te brisanje sadržaja sa web stranice bez pomoći stručnjaka.

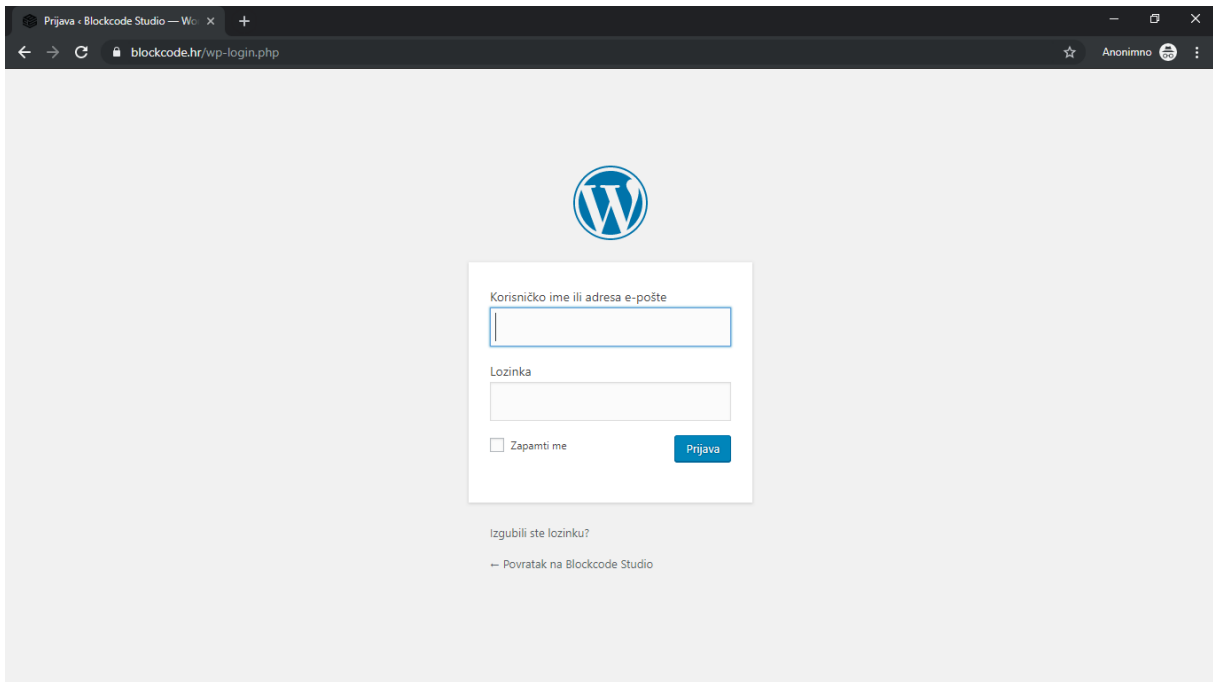
3.1.2. Content Delivery Application - CDA

CDA je odgovoran za skrivene back-end procese kao što je pohranjivanje sadržaja i isporuka istih posjetiteljima web sjedišta.

Glavno načelo CMS-a je sadržaj web stranice, a njegov dizajn je odvojen. Dizajn web stranica rijetko se modernizira, dok su promjene sadržaja potrebne kod mnogih web sjedišta na dnevnoj bazi. Stoga mnogi CMS-ovi koriste predloške. Predložak je prazna stranica s ugrađenim komponentama dizajna. Za dodavanje i promjenu sadržaja, potrebno je samo ispuniti predložak potrebnim informacijama. CMS ne zahtijeva instalaciju dodatnog softvera jer se nalazi izravno na poslužitelju. Sučelju CMS-a se pristupa pomoću preglednika, a način pristupanja ovisi o CMS-u.

Načini pristupanja CMS-u web sjedišta kod različitih CMS-ova:

- **WORDPRESS** - website.com/wp-admin ili [wp-login.php](http://website.com/wp-login.php) (slika 4)
- **DRUPAL** - website.com/user
- **JOOMLA** - website.com/administrator
- **HOST CMS** - website.com/admin
- **WEBASYST** - website.com/login
- **INSTANTCMS** - website.com/login
- **MAGENTO** - website.com/login
- **MODX** - website.com/manager



Slika 4 - Prijava u Wordpress CMS stranice blockcode.hr

3.2. Značajke CMS-a

⁴Sustav za upravljanje web sadržajem kontrolira dinamičku zbirku web materijala, uključujući HTML dokumente, slike i druge oblike medija. Web CMS olakšava kontrolu dokumenata, reviziju, uređivanje i upravljanje vremenskim okvirima. Ovdje su navedene neke od značajki web CMS-a:

3.2.1. Automatizirani predlošci

Kreiranje standardnih predložaka (obično HTML i XML) koje korisnici mogu primijeniti na novi i postojeći sadržaj, mijenjajući izgled cjelokupnog sadržaja s jednog središnjeg mjesta.

3.2.2. Kontrola pristupa

Neki CMS sustavi podržavaju grupe korisnika koje kontroliraju način na koji registrirani korisnici komuniciraju s web-lokacijom. Stranica na web-lokaciji može biti ograničena na jednu ili više grupa. To znači da je anonimnom korisniku (netko tko nije prijavljen) ili prijavljenom korisniku koji nije član grupe na koje je stranica ograničena, zabranjen pristup.

⁴ Prema https://en.wikipedia.org/wiki/Web_content_management_system

3.2.3. Skalabilna ekspanzija

Mogućnost proširenja jedne implementacije (jedna instalacija na jednom poslužitelju) na više domena, ovisno o postavkama poslužitelja. CMS web sjedišta mogu stvoriti mikro stranice / web portale unutar glavne stranice. Ta mogućnost je dostupna u većini modernih CMS-ova.

3.2.4. Jednostavno uređivanje sadržaja

Kada je sadržaj odvojen od vizualnog prikaza web sjedišta, obično je mnogo lakše i brže njime manipulirati i uređivati ga. Većina CMS softvera koristi WYSIWYG (engl. *What You See Is What You Get*) alate za uređivanje koji omogućuju običnim korisnicima stvaranje i uređivanje sadržaja.

3.2.5. Skalabilni skup značajki

Većina CMS softvera koristi dodatke ili module koji se mogu lako instalirati kako bi se proširila funkcionalnost postojećeg web sjedišta.

3.2.6. Nadogradnje

Aktivni CMS softver obično prima redovite obavijesti o ažuriranjima koja uključuju nove skupove značajki i održavaju sustav ažurnim prema web standardima.

3.2.7. Upravljanje tijekovima rada

Tijek rada je proces stvaranja ciklusa sekvencijalnih i paralelnih zadataka koji se moraju ostvariti u CMS-u. Na primjer, jedan ili više stvaratelja sadržaja mogu poslati priču, ali ona se ne objavljuje dok je urednik kopije ne očisti i glavni urednik to ne odobri.

3.2.8. Suradnja

CMS softver može djelovati kao platforma za suradnju u kojoj mnogi korisnici preuzimaju i rade na sadržaju. Promjene se mogu pratiti i autorizirati za objavljivanje ili zanemariti te vraćati na stare verzije. Drugi napredni oblici suradnje omogućuju višestrukim korisnicima istovremeno mijenjanje (ili komentiranje) stranice u sesiji suradnje.

3.2.9. Delegacija

Neki CMS softveri omogućuju različitim skupinama korisnika ograničene privilegije nad određenim sadržajem na web sjedištu.

3.2.10. Upravljanje dokumentima

CMS softver može osigurati način zajedničkog upravljanja životnim ciklusom dokumenta od početnog vremena stvaranja, preko revizija, objavljivanja, arhiviranja i uništavanja dokumenata.

3.2.11. Virtualizacija sadržaja

Softver CMS-a može osigurati način da se svakom korisniku omogući rad unutar virtualne kopije cijelog web sjedišta, skupa dokumenata i/ili baze koda. To omogućuje pregled promjena na više međuovisnih resursa u kontekstu prije objavljivanja.

3.2.12. Udruživanje sadržaja

CMS često pomaže u distribuciji sadržaja generirajući RSS i Atom izvore podataka u druge sustave. Također omogućuje slanje e-pošte korisnicima kada ažuriranja postanu dostupna.

3.2.13. Višejezičnost

Mnogi CMS-ovi mogu prikazivati sadržaj na više jezika.

3.2.14. Verzije

Poput sustava za upravljanje dokumentima, CMS softver može implementirati kontrolu verzija, pomoću koje korisnici provjeravaju stranice unutar i izvan CMS-a. Ovlašteni urednici mogu dohvatiti prethodne verzije i raditi s odabrane točke. Verzije su korisne za sadržaj koji se mijenja i zahtijeva ažuriranje, ali može biti potrebno i početi s prethodne verzije ili se pozivati na nju.

4. Prednosti i nedostaci CMS-a

Iako svaki CMS sustav ima svoje posebne prednosti i nedostatke, postoji nekoliko općenitih prednosti i nedostataka koje mogu opisati većinu CMS sustava.

4.1 Prednosti

- **Lakoća ažuriranja sadržaja**

Dodavanje novih sadržaja u mnogome je olakšano putem CMS sustava. Prednost se ogleda u tome što je mnogo lakše mijenjati sadržaj u pregledniku nego postavljati stand-alone aplikaciju preko koje možete pristupiti HTML kodu vaše web stranice.

- **Samostalno održavanje sadržaja**

Putem CMS-a sadržaj možemo osvježavati, mijenjati ili brisati bilo kad poželimo i sa bilo kog računala koji ima vezu sa internetom.

- **Razdvojenost sadržaja od dizajna**

Izmjena template-a na tradicionalan način je dug i zamoran posao i najčešće se naplaćuje kao dodatna usluga. Koristeći CMS možemo promijeniti template i samim tim i izgled naše stranice kad god poželimo u samo nekoliko klikova.

- **Stalna ažuriranja CMS-a**

Za razliku od CMS-a, završetkom razvoja web sjedišta na tradicionalan način dolazimo do gotovog proizvoda. CMS posjeduje široku bazu korisnika koji ga unapređuju sa novim zakrpama i dodacima koji se mogu naknadno dodavati.

- **Dodaci koji unapređuju funkcionalnost**

Instaliranjem plugin-ova i drugih dodataka možemo povećati funkcionalnost web sjedišta. Postoji mnogo različitih dodataka koje možemo sa lakoćom uklopiti na našu stranicu bez potreba usluga profesionalnog programera.

- **Podrška ostalih korisnika**

Ukoliko se desi da ne razumijemo određenu funkciju unutar CMS-a uvijek možemo postaviti pitanje na nekom od foruma gdje se okupljaju korisnici tog CMS sustava.

- **Sadržaj postavljen unutar baze podataka**

Baza podataka je zaslužna za lako održavanje sadržaja. Ona omogućava lako dodavanje, mijenjanje ili brisanje sadržaja. Kao dodatna mjera opreza, baza podataka omogućava očuvanje svih podataka unutar nje i tako dodatno osigurava postojanost podataka.

4.2 Nedostaci

- **Smanjena fleksibilnost**

Iako se sa CMS sustavima može napraviti bezbroj različitih web sjedišta, kad govorimo o web sjedištima po narudžbi sa točno specifičnim osobinama, tu dolazimo do problema. Što je više specifičnih osobina potrebno na stranici, to je veća mogućnost da će nam CMS sustav biti od vrlo male ili nikakve koristi.

- **Limitiranost alata**

Iako je izbor velik i konstantno se širi, izbor alata za izradu web sjedišta u nekom besplatnom CMS-u je još uvijek limitiran. Nije dostupno uvijek sve što nam je potrebno.

- **Učenje**

Iako je lak za korištenje, ipak je potrebno neko vrijeme i treninga da se efektivno savladaju sve značajke CMS-a. Također bez osnovnih poznavanja HTML, CSS i PHP-a korisnikove mogućnosti izmjene su ograničene na konfiguracije pojedinog predloška.

- **Sigurnost**

⁵CMS-ovi su ranjivi po prirodi jer su programi otvorenog koda (barem većina). Različiti CMS premium dodaci i teme koje se mogu naći besplatno na webu izvan službenog Wordpress direktorija, obično su zaraženi i vrlo opasni za stranicu. Ako CMS nije ažuriran može postati laka meta za spammer-e i hakere.

⁵ Prema <https://www.imperva.com/blog/cms-security-tips/>, Why CMS Platforms Are Common Hacking Targets (and what to do about it)

5. Najkorišteniji web CMS sustavi

⁶Prema raznim procjenama, anketama i eksperimentima, vjeruje se da danas postoji gotovo dvije milijarde internetskih stranica odnosno web sjedišta a od toga aktivnih 200 milijuna. To otvara puno prostora za mnoge web platforme za izradu web sjedišta. Za dostizanje te brojke u veliko su zaslužni CMS-ovi otvorenog koda. CMS-ovi otvorenog koda WordPress, Joomla ! i Drupal trenutno su tri najpopularnija standardna CMS rješenja za profesionalno izrađivanje web sjedišta. No, svaki sustav, naravno, nije jednako prilagođen svakoj pojedinačnoj svrsi. Open source rješenja su popularna širom svijeta. Slika 5 prikazuje 10 najkorištenijih web CMS sustava na tržištu 2018. godine. Najpopularnije rješenje za brzo i ekonomično objavljivanje sadržaja je WordPress CMS sustav.

#	Websites Using	Market Share %	Active Sites	# of Websites in Million
1	 WordPress	59.7 %	22,671,100	313,050
2	 Joomla	6.7 %	1,837,862	21,952
3	 Drupal	4.7 %	577,549	27,863
4	 Magento	2.3 %	214,455	14,188
5	 Blogger	1.9 %	343,944	7,797
6	 Shopify	1.7 %	739,176	17,285
7	 TYPO3	1.5 %	303,333	6,580
8	 Bitrix	1.5 %	183,346	6,127
9	 Squarespace	1.4 %	1,727,900	9,390
10	 Prestashop	1.3 %	247,350	5,770

Slika 5 - 10 najkorištenijih web CMS sustava na tržištu i njihov udio na tržištu 2018. godine

⁶ Prema <https://websitesetup.org/popular-cms/>, Popular CMS by Market Share

5.1. Wordpress

WordPress je open source Content Management System poduprt PHP-om i MySQL-om. ⁷Prema najnovijim podacima iz 2019. godine, WordPress kontrolira oko **60% CMS tržišta**. Izvorno zamišljen kao blog sustav, sada je dostupno više proširenja za osnovnu instalaciju, što omogućuje nadogradnju softvera na potpuno funkcionalan sustav za upravljanje sadržajem.



Slika 6 - Wordpress.org

Prvi put je objavljen 27. svibnja 2003. Objavio ga je Matt Mullenweg. WordPress je započeo samo kao blog sustav, ali je evoluirao u to da se može koristiti kao potpuni sustav za upravljanje sadržajem i puno više od toga koristeći tisuće dodataka, widgeta i tema.

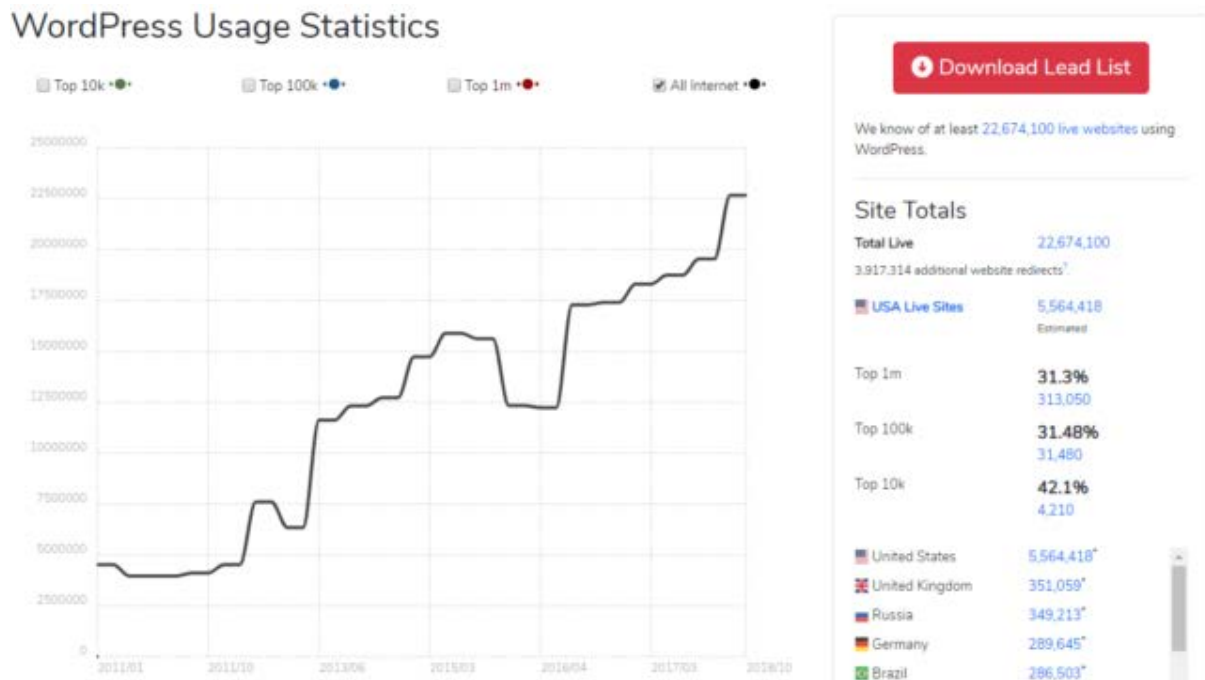
Na stranici www.wordpress.org (slika 6) postoji WordPress zip. datoteka sa potrebnim kodovima koja se može preuzeti i instalirati lokalno ili na poslužitelju. Za to je potrebno imati web poslužitelj koji podržava minimalne zahtjeve te malo znanja i vremena. WordPress je potpuno podesiv i može se koristiti skoro za sve. Također postoji i servis nazvan WordPress.com koji omogućuje stvaranje novih i besplatnih WordPress baziranih blogova u par sekundi, ali razlikuje se u nekoliko stvari i manje je fleksibilan od WordPressa kojeg sami preuzmete i instalirate.

⁷ Prema https://w3techs.com/technologies/overview/content_management/all

WordPress se može primjenjivati primjenom različitih metoda na hosting okruženju. ⁸Korisnici softvera imaju ogroman izbor tema, dodataka i widgeta. Kritičari kažu da sustav nije samo previše složen, nego i zbunjujući i neugodan za korištenje. WordPress dijeli iste probleme kao i svaki uspješni razvoj softvera: zbog svoje visoke cirkulacije, to je popularna meta za hakere. Zbog toga korisnici moraju redovito ažurirati svoj WordPress softver s sigurnosnim ažuriranjima. Ta se ažuriranja pojavljuju brzo nakon sigurnosnih incidenata, ali nisu uvijek kompatibilna sa svim proširenjima koja su potrebna za učinkovito korištenje softvera. Kao rezultat toga, ažuriranja softvera mogu se povezati s znatnim administrativnim naporima, posebno u slučaju velikih projekata.

Zbog svoje male instalacije, WordPress je dobar za male i jednostavno strukturirana web sjedišta. Pogotovo za blogere, koji svojim posjetiteljima žele ponuditi sadržaj koji se često mijenja u atraktivnim izgledima. WordPress je učinkovit CMS s intuitivnim web sučeljem, međutim, kako se složenost online projekta povećava, prilagođavanje softvera znatno se smanjuje. Projekti velikih poduzeća također se mogu provesti teoretski, ali u takvom slučaju veći broj funkcija neizbježno je popraćen povećanjem administrativnih napora. Za stabilno funkcioniranje web sjedišta, WordPress postavlja razmjerno visoke zahtjeve poslužitelju kako se vaš promet web sjedišta povećava.

Broj aktivnih WordPress korisnika konstanto se povećava. Slika 7 nam prikazuje statistiku korištenja Wordpress CMS sustava od 2011. do 2018. godine.



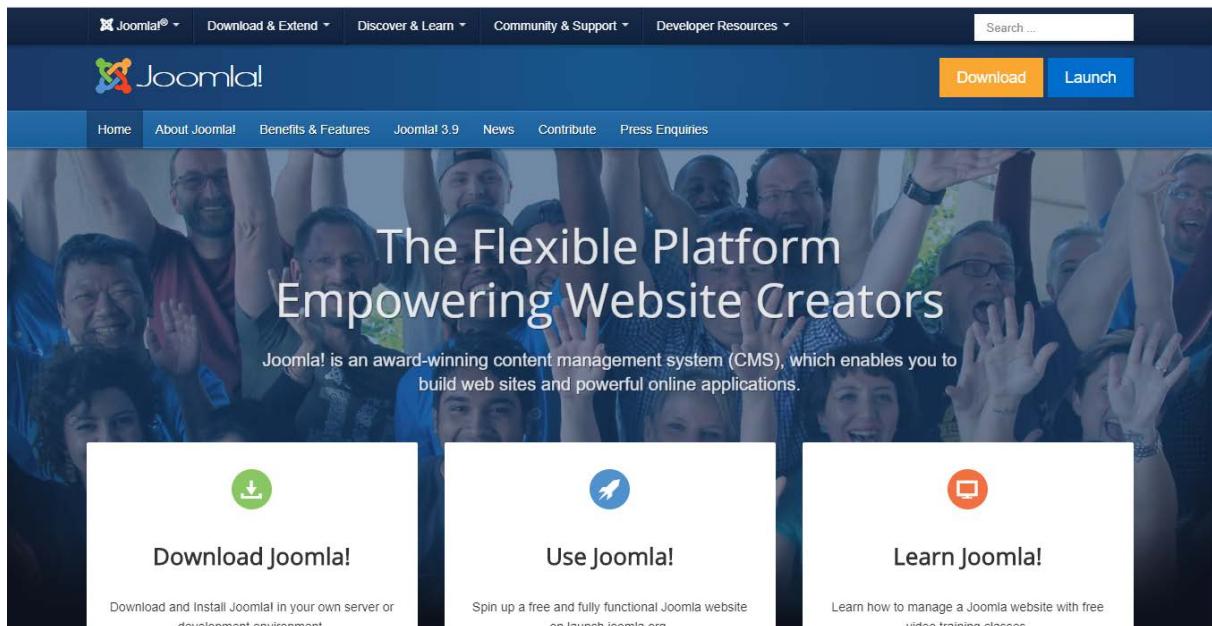
Slika 7- Statistika korištenja Wordpress CMS sustava od 2011. do 2018. godine

⁸ Prema <https://www.ionos.com/digitalguide/hosting/cms/cms-comparison-a-review-of-the-best-platforms/>, CMS comparison 2019: The most popular open source systems

5.2. Joomla!

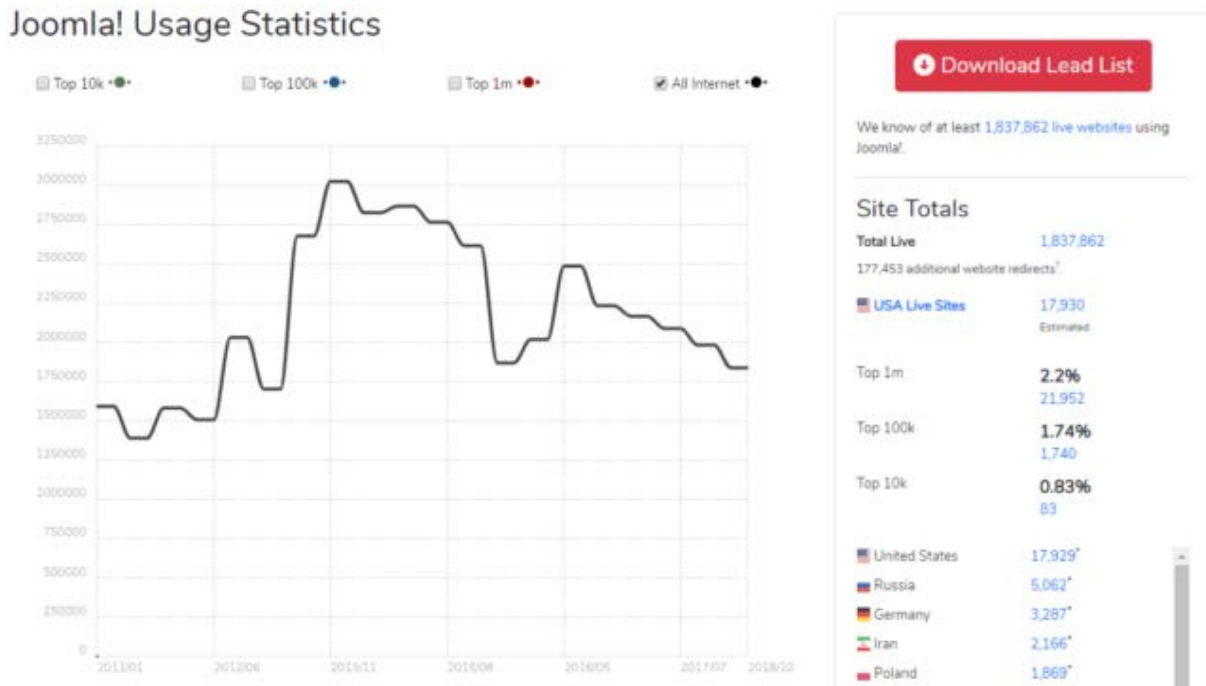
⁸**Joomla!** je također CMS sustav za upravljanje web sadržajem otvorenog koda, koji služi objavljivanju sadržaja na internetu. Nalazi se na web adresi joomla.org (slika 8). Sa skoro 2 milijuna instalacija širom svijeta (prema websitesetup.org), Joomla! je drugi najveći web CMS tržištu. Joomla! sustav je namijenjen početnicima i naprednim korisnicima, ali je zahtjevniji za korištenje od WordPressa. Za razliku od WordPressa, sveobuhvatne CMS funkcionalnosti uključene su bez instaliranja proširenja i dodataka. Bonus za ne-stručnjake koji nemaju profesionalnu IT pozadinu je dobra dokumentacija u obliku online priručnika, kao i aktivnih korisničkih foruma.

Softver je pisan u programskom jeziku PHP, a za spremanje podataka koristi bazu podataka MySQL. Joomla dolazi s opcijama za upravljanje tekstem i slikama, prikaz sadržaja u RSSu, PDFu i verziji za printanje, raznim modulima kao što su "posljednje novosti" i "popularni sadržaji", upravljanje banerima i anketama, tražilicom i kontakt formom. Osim navedenih opcija dostupne su tisuće specijaliziranih dodataka koji mogu Joomla! stranicu pretvoriti npr. u web dućan, multimedijalnu galeriju ili on-line zajednicu. Još jedna odlika ovog sustava je i podrška za strane jezike pa je tako moguće preuzeti hrvatski jezični paket koji će je u potpunosti prevesti na hrvatski jezik.



Slika 8 - Joomla.org

Joomla! sustav je posebno popularan u SAD-u, a karakterizira ga potpuno objektno-orijentiran dizajn softvera koji se temelji na samostalnom MVC okviru. Predložci se mogu kreirati u modularnom Joomla! modulu kombiniranjem unaprijed definiranih elemenata sadržaja. Joomla! dijeli proširenja na dodatke, komponente i module. Oni se mogu koristiti u back-endu kao i u front-endu. Slika 9 nam prikazuje statistiku korištenja Joomla! CMS sustava od 2011. do 2018. godine.



Slika 9 - Statistika korištenja Joomla! CMS sustava od 2011. do 2018. godine

Joomla! je nastala 17. kolovoza 2005. kao fork sustava Mambo od strane razvojnog tima Joomla!. U to vrijeme Mambo je bio pod zaštitom tvrtke Miro International Pty Ltd koji su osnovali neprofitnu organizaciju koja je imala cilj financirati projekt Mambo i pravno ga štiti. Razvojni tim Joomla! tvrdi da je dolazilo do niza neslaganja, nedostatka komunikacije i djela koja su kršila vrijednosti otvorenog koda. Vođa tima, Andrew Eddie, je uskoro objavio otvoreno pismo zajednici u kome je najavio odvajanje. Timovi su se ubrzo reorganizirali i okupili oko stranice OpenSourceMatters te počeli raditi na novom projektu.

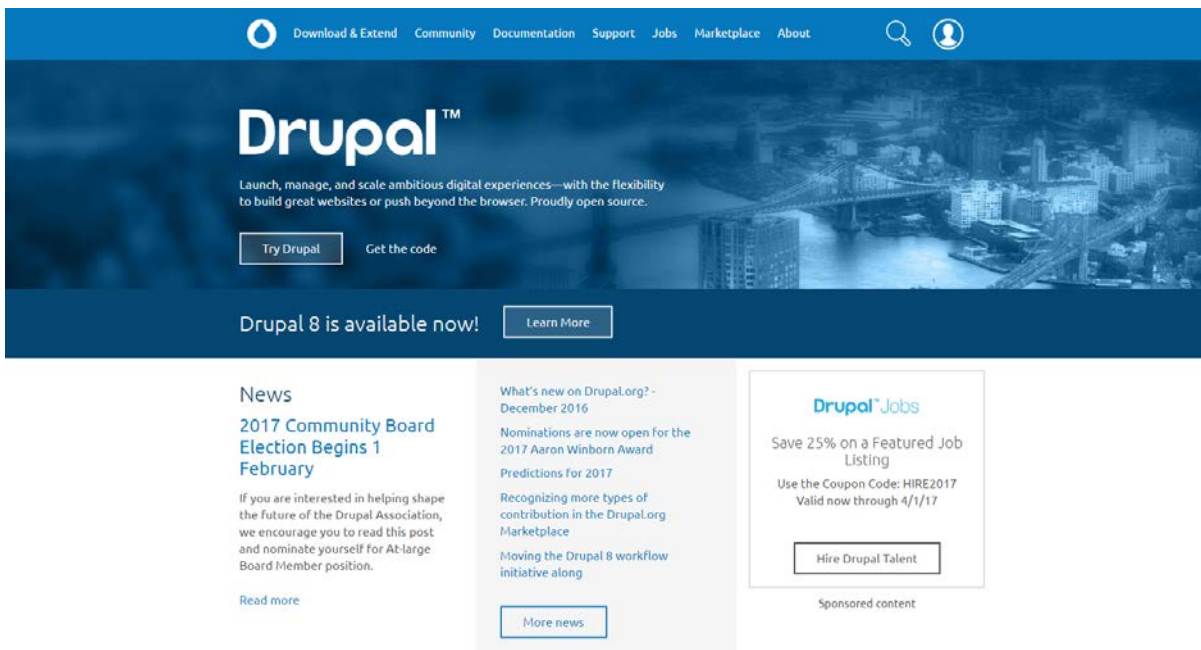
Prva verzija Joomla! (Joomla 1.0.0) je objavljena 16. rujna 2005., a to je bio malo prerađeni i preimenovani Mambo CMS. Najnovija inačica je Joomla 3.

Joomla je 2006. i 2007. dobila nagradu "Open Source Content Management System Award" koju dodjeljuje izdavačka kuća Packt Publishing, Te je svake godine od 2014. do danas proglašena najboljim besplatnim CMS sustavom po "Critic People's Choice Awards-u".

5.3. Drupal

⁹Treći CMS je i najstariji od navedenih. Prva verzija **Drupala** izašla je 2001. i sada je treće najpopularnije rješenje za izgradnju web sjedišta. Nalazi se na web adresi drupal.org (slika 10). Njegov ukupni tržišni udio svih web sjedišta na internetu iznosi oko 2%. Zanimljivo je da je ovaj sustav posebno popularan među većim web sjedištima. Među milijun najpopularnijih web-mjesta, on je zapravo popularniji od Joomlae.

Drupal je i tehnički najnapredniji CMS od ove trojke. Kao i ostali, njegova glavna tehnologija je PHP, a ujedno je i softverski projekt otvorenog koda.



Slika 10 – Drupal.org

Drupal je napravljen za brzu izvedbu. Zbog toga je manje hardverski zahtjevan od konkurencije, što je dobra vijest za troškove poslužitelja.

S druge strane, osim ako ste sami developer, izgradnja web sjedišta s Drupalom jamči da ćete nekome morati platiti. To je daleko najsloženije rješenje na ovom popisu i nije pogodno za početnike. Iz tog razloga, troškovi razvoja gotovo sigurno moraju biti dio vašeg proračuna.

⁹ Prema <https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>, WordPress vs Joomla vs Drupal (Comparison)

Drupal je izvorno je razvijen kao rješenje za studentske zajednice. Softver dolazi s iznimno siromašnom osnovnom instalacijom, koju moduli mogu prilično proširiti. Danas snaga projekta leži u njegovoj zajednici. Osnovni modul već ima brojne web 2.0 i društvene funkcije kao što su weblogovi, forumi i tag clouds. Slika 11 nam prikazuje statistiku korištenja Drupal CMS sustava od 2011. do 2018. godine.

Drupal Usage Statistics



Slika 11 - Statistika korištenja Drupal CMS sustava od 2011. do 2018. godine

Sa svojim fokusom na društveno izdavaštvo, Drupal je posebno prikladan za male i srednje velike projekte. Veliki spektar proširenja omogućuje implementaciju portala tvrtki sa složenim multi-domenskim strukturama. Međutim, upravljanje jezgrom CMS-a sa sveobuhvatnim proširenjima zahtijeva stručnost i instinkt. Ako osnovne funkcije softvera nisu dovoljne, dodatni moduli morat će se ručno instalirati putem FTP-a. To može biti složen proces. Ažuriranje proširenja ponekad je teško, jer nisu uvijek kompatibilne sa back-endom.

6. Alati za razvoj CMS-a

Alati koji su korišteni za razvoj većine postojećih web sjedišta, pa tako i web CMS-ova su: HTML, CSS, PHP, MySQL i JavaScript. U ovom dijelu rada ukratko je objašnjen svaki od tih alata.

6.1. HTML

¹⁰HTML je kratica za **HyperText Markup Language**, što znači prezentacijski jezik za izradu web sjedišta. Hipertekst dokument stvara se pomoću HTML jezika. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. To je jezik koji bilježi tekst na način koji je sintaktički prepoznatljiv tako da računalo može manipulirati njime. Osnovni element za izradu svake stranice su oznake (engl. *tags*) koje opisuju kako će se nešto prikazati u web pregledniku. HTML je skup tih oznaka za označavanje koje se koriste za opisivanje web stranica. Oznake odvajaju normalni tekst od HTML koda. To su riječi između znakova `<>`.

HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik niti su ljudi koji ga koriste programeri. Njime ne možemo izvršiti nikakvu zadaću, pa čak ni najjednostavniju operaciju zbrajanja ili oduzimanja dvaju cijelih brojeva. On služi samo za opis naših hipertekstualnih dokumenata. Html datoteke su zapravo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Povezice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranica.

Različite oznake izvršavat će različite funkcije, poput prikazivanja slika ili tablica. HTML je kombinacija riječi i simbola koji daju upute o tome kako će dokument biti predstavljen. Te nisu vidljive kada pregledavate svoju stranicu putem preglednika, ali njihovi učinci jesu.

Označavanje je to što HTML oznake rade s tekстом unutar njih. Oni ga označavaju kao određeni tip teksta (npr. kurzivni tekst). HTML dokumenti sadrže HTML oznake i običan tekst. Sadržaj na HTML stranici bit će statičan. Kako bi promijenili sadržaj, urednik mora imati neko znanje o HTML-u i mijenjati sadržaj u skladu s tim.

¹¹HTML5 je peta i trenutna glavna verzija HTML-a koja sadrži XHTML. XHTML je kratica za **eXtensible HyperText Markup Language** te je dio obitelji XML jezika.

¹⁰ Prema <https://hr.wikipedia.org/wiki/HTML>

¹¹ Prema <https://en.wikipedia.org/wiki/HTML5>

HTML 5 uključuje detaljne modele obrade kako bi se potaknule interoperabilnije implementacije; proširuje, poboljšava i racionalizira oznaku dostupnu za dokumente i uvodi sučelja za označavanje i aplikacijsko programiranje (API) za složene web-aplikacije. Iz istih razloga, HTML 5 je također kandidat za mobilne aplikacije s više platformi jer uključuje značajke osmišljene s malim uređajima na umu.

6.2. CSS

CSS je skraćenica od **Cascading Style Sheets**. Koristi se za kontrolu stila i izgleda više web stranica odjednom. Stilovi definiraju način prikaza HTML elemenata. CSS nadjačava zadane postavke preglednika za tumačenje načina prikazivanja HTML oznaka, što omogućuje korištenje bilo koji HTML element označen početnom i završnom oznakom za primjenu definiranih atributa stila.

Eksterni stilski listovi mogu uštedjeti mnogo posla. Oni se pohranjuju u CSS datoteke. Stilovi sadrže pravila koja se sastoje od selektora i deklaracija koje definiraju kako će stilovi biti primijenjeni. Selektor (redefinirani HTML element, ime klase ili ime ID) je veza između HTML dokumenta i stila. Postoje dvije različite vrste selektora: vrste ili tipovi (Oznake HTML elemenata) i attribute (kao što su imena klasa i ID-ova).

¹²CSS3 je najnovija evolucija jezika CSS i ima za cilj proširiti CSS2.1. Donosi mnogo noviteta, kao što su zaobljeni kutovi, sjene, gradijenti, prijelazi ili animacije, kao i novi rasporedi poput višestrukih stupaca, fleksibilnog okvira ili rasporeda mreže.

6.3. PHP

¹³PHP je rekurzivni akronim za **Hypertext Preprocessor**. Izvorno značenje skraćenice je bilo Personal Home Page. To je programski jezik opće namjene, koji se orijentira po C i Perl sintaksi, izvorno dizajniran za razvoj dinamičkih web stranica. Osmislio ga je Rasmus Lerdorf, 1994. godine.

¹⁴PHP kod može se kompajlirati pomoću sučelja komandne linije (CLI), ugrađenog u HTML kod, ili koristiti u kombinaciji s različitim sustavima web predložaka, sustavima za upravljanje web sadržajem i web okvirima. PHP kod obično obrađuje PHP interpreter implementiran kao modul u web poslužitelju ili kao izvršni program „Common Gateway Interface“ (CGI). Web poslužitelj šalje rezultate interpretiranog i izvršenog PHP koda, koji može biti bilo koji tip podataka, kao što je generirani HTML kod ili binarni slikovni podaci. PHP se može koristiti za mnoge zadatke programiranja izvan web konteksta, kao što su samostalne grafičke aplikacije i robotska kontrola dronova.

¹² Prema <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>

¹³ Prema <https://hr.wikipedia.org/wiki/PHP>

¹⁴ Prema <https://en.wikipedia.org/wiki/PHP>

Standardni PHP interpreter, kojeg pokreće Zend Engine, besplatan je softver objavljen pod PHP licencom. PHP je vrlo rasprostranjen i može se besplatno koristiti na većini web poslužitelja i na gotovo svakom operativnom sustavu i platformi.

PHP interpreter izvršava samo PHP kod unutar svojih razdjelnika. Sve što je izvan razdjelnika ne obrađuje PHP, iako je tekst koji nije PHP kod još uvijek podložan kontrolnim strukturama opisanim u PHP kodu. Najčešći razdjelnici su: `<?php` - otvaranje PHP odjeljka i `?>` - zatvaranje PHP odjeljka. Ovaj kratki razdjelnik otežava prenosivost datoteke skripte, budući da se podrška za njih može onemogućiti u lokalnoj PHP konfiguraciji i stoga se ne preporuča. Svrha tih razdjelnika je odvojiti PHP kod od ne-PHP sadržaja, kao što je JavaScript kod ili HTML oznake.

Ti razdjelnici u XHTML-u i drugim XML dokumentima, stvaraju ispravno oblikovane XML upute za obradu. To znači da je rezultirajuća mješavina PHP koda i drugih oznaka u datoteci na strani poslužitelja sama po sebi dobro oblikovan XML.

Varijable imaju predznak simbola dolara (\$), a tip ne mora biti unaprijed određen. Za razliku od imena funkcija i klasa, imena varijabli su osjetljiva na velika i mala slova. I dvostruko citirani ("") i heredoc (<<<) nizovi pružaju mogućnost interpolacije vrijednosti varijable u niz. PHP tretira nove retke kao praznine u obliku jezika slobodnog oblika, a naredbe se završavaju točkazarezom. PHP ima tri vrste sintakse komentara: `/* */` označi blok i inline komentare; `//` ili `#` koriste se za komentare u jednom retku. Echo izjava je jedan od nekoliko objekata koje PHP pruža za izlaz teksta.

Što se tiče ključnih riječi i sintakse jezika, PHP je sličan sintaksi programskom jeziku C. Uvjetovani izrazi **if**, petlje **for** i **while**, te return funkcije slični su sintaksi programskih jezika kao što su C, C ++, C #, Java i Perl.

Tijekom 2014. i 2015. godine razvijena je najnovija velika verzija PHP-a koja je numerirana PHP-om 7. Temelj PHP-a 7 je grana PHP-a koja je izvorno nazvana PHP sljedeća generacija (engl. *PHP next generation, phpng*). Njegovi autori su Dmitry Stogov, Xinchun Hui i Nikita Popov, a cilj mu je optimizirati performanse PHP-a refactoringom Zend Enginea uz zadržavanje gotovo potpune jezične kompatibilnosti. Od 14. srpnja 2014., mjerila temeljena na WordPress-u, koja su poslužila kao glavni standard za phpng projekt, pokazala su gotovo 100% povećanje performansi. Od promjena u phpngu se također očekuje da će olakšati poboljšanje performansi u budućnosti, budući da se kompaktnije strukture podataka i druge promjene smatraju bolje prilagođenim za uspješnu migraciju na kompilaciju JIT (engl. *just in time*). Zbog značajnih promjena, prerađeni Zend Engine naziva se Zend Engine 3, kojem prethodi Zend Engine 2 koji se koristi u PHP-u 5.

6.4. MySQL

¹⁵MySQL je sustav za upravljanje relacijskom bazom podataka otvorenog koda (engl. *Relational Database Management System, RDBMS*). Njegovo ime je kombinacija "My", ime kćeri suosnivača Michaela Wideniusa, i "SQL", kratica za "Structured Query Language".

MySQL je besplatan softver otvorenog koda pod uvjetima GNU opće javne licence, a dostupan je i pod različitim vlasničkim licencama. MySQL je bio u vlasništvu i sponzoriran od strane švedske tvrtke MySQL AB, koju je kupio Sun Microsystems (sada Oracle Corporation). 2010. godine kada je Oracle kupio Sun, Widenius je iskoristio MySQL projekt otvorenog koda i stvorio MariaDB. MySQL je napisan u C i C ++. Njegov SQL parser je napisan u yacc, ali koristi vlastiti leksički analizator. MySQL radi na većini platformi operativnih sustava. MySQL je komponenta LAMP web aplikacijskog softvera (i drugih), što je akronim za Linux, Apache, MySQL, Perl / PHP / Python. MySQL koriste mnoge web-aplikacije, uključujući Drupal, Joomla i WordPress. MySQL koriste i mnoge popularne web stranice, uključujući Facebook, Flickr, MediaWiki, Twitter, i YouTube.

6.5. JavaScript

JavaScript ili JS, je programski jezik visoke razine koji je u skladu s ECMAScript specifikacijama. JavaScript ima sintaksu kovrčavih zagrada ({}), dinamičko tipkanje, objektanu orijentaciju temeljenu na prototipu i funkcije prve klase.

¹⁶Uz HTML i CSS, JavaScript je jedna od temeljnih tehnologija World Wide Weba. JavaScript omogućuje interaktivnost web sjedišta i bitan je dio web aplikacija. Velika većina web sjedišta ga koristi, a glavni web preglednici imaju namjenski JavaScript mehanizam za njegovo izvršavanje. Kao multi-paradigmatski jezik, JavaScript podržava stilove programiranja temeljene na događajima, funkcionalne i imperativne (uključujući objektno-orijentirane i prototipne). Ima API-je za rad s tekstom, nizovima, datumima, regularnim izrazima i DOM-om, ali sam jezik ne uključuje I / O, kao što su umrežavanje, pohrana ili grafički sadržaji. On se oslanja na okruženje domaćina u kojem je ugrađeno kako bi pružilo te značajke.

Od svibnja 2017. godine, 94,5% od 10 milijuna najpopularnijih web stranica koristilo je JavaScript. Najčešća upotreba JavaScripta je dodavanje dinamičko ponašanje na strani klijenta HTML stranice, poznate i kao Dinamički HTML (DHTML). Skripte su ugrađene ili uključene sa HTML stranica i međusobno djeluju s Object Modelom dokumenta (DOM) stranice. Neki jednostavni primjeri ove upotrebe su:

¹⁵ Prema <https://en.wikipedia.org/wiki/MySQL>

¹⁶ Prema <https://en.wikipedia.org/wiki/JavaScript>

- Učitavanje novog sadržaja stranice ili slanje podataka na poslužitelj putem Ajaxa bez ponovnog učitavanja stranice (na primjer, društvena mreža može dopustiti korisniku da objavljuje ažuriranja statusa bez napuštanja stranice).
- Animiranje elemenata stranice, njihovo zatamnivanje, izmjena veličine, premještanje itd.
- Interaktivni sadržaj, na primjer igre, i reprodukcija zvuka i videozapisa.
- Provjera valjanosti ulaznih vrijednosti web obrasca kako bi bili sigurni da su prihvatljivi prije slanja poslužitelju.
- Prijenos informacija o korisnikovim navikama čitanja i aktivnostima pregledavanja na različite web-lokacije. Web stranice to često čine za web analizu, praćenje oglasa, personalizaciju ili u druge svrhe.

JavaScript kod može se lokalno izvoditi u korisnikovom pregledniku (a ne na poslužitelju), čime se povećava cjelokupna aktivnost aplikacije na korisničke akcije. JavaScript kod može također otkriti radnje korisnika koje sam HTML ne može, kao što su pojedinačni pritisci na tipke. Aplikacije kao što je Gmail koriste ovu prednost: mnogo logike korisničkog sučelja zapisano je u JavaScriptu, a JavaScript šalje poslužitelju zahtjeve za informacijama (kao što je sadržaj poruke e-pošte). Širi trend Ajaxovog programiranja isto tako iskorištava tu snagu.

Iako postoje sličnosti između JavaScripta i Jave, uključujući ime jezika, sintaksu i odgovarajuće standardne knjižnice, ta dva jezika su različita i značajno se razlikuju u dizajnu. JavaScript je pod utjecajem programskih jezika kao što su Self i Scheme. JSON-format za serijalizaciju, koji se koristi za spremanje struktura podataka u datotekama ili za prijenos preko mreža, temelji se na JavaScriptu.

7. Izrada CMS sustava

Za potrebe ovog rada opisana je izrada jednostavnog sustava za upravljanje sadržajem odnosno CMS-a koji upravlja sadržajem web sjedišta. Glavne funkcionalnosti sustava sastoje se od dva modula:

- Dodavanje i brisanje članaka
- Dodavanje i brisanje slika

Web sjedište i CMS napravljeni za potrebe ovog rada, postavljeni su na testnom serveru tvrtke Blockcode Studio j.d.o.o.

7.1. Kreiranje strukture web sjedišta

U prvom koraku potrebno nam je osnovno poznavanje HTML-a i CSS-a. sa kojima kreiramo glavnu strukturu web sjedišta odnosno izgled web stranice. Pošto je taj dio prilično trivijalan, nema potrebe previše u njega ulaziti. Svaki HTML dokument se sastoji od 3 glavna djela označenih oznakama `<html>`, `<head>` i `<body>`. `<html>` označuje početak i kraj HTML dokumenta, `<head>` označava zaglavlje dokumenta u kojem se nalaze meta podaci, skripte, stilovi itd., a u `<body>` se nalazi sadržaj te stranice. Osim tih glavnih HTML oznaka, ubačen je naslov stranice i podijeljen je dokument na dva djela od kojih će jedan prikazivati članke a drugi slike (slika 12). Unutar `<head>` oznaka uključen je i stilski dokument `style.css` koji nam pomaže u postavljanju vizualnog djela web sjedišta.



CMS - Web Stranica

Članci:

Slike:

Slika 12 - Struktura početne stranice web sjedišta

7.2. Kreiranje baze i povezivanje sa bazom podataka

Za kreiranje, povezivanje i općenito rad sa bazom podataka koristimo MySQL. Bazu podataka kreirana je u phpMyAdmin sučelju te se sastoji od 3 tablice: **clanci**, **slike**, **users** (slika 13).

Tablica clanci sadrži 4 stupca:

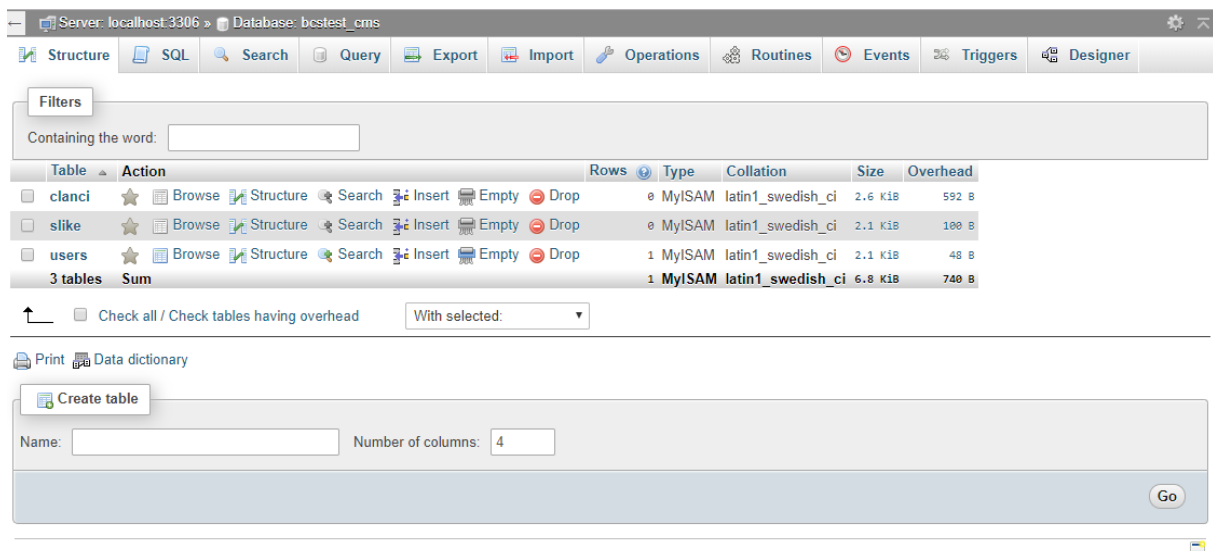
- **clanak_id** – primarni ključ tablice clanci, jedinstveni identifikator, automatski dodaje ID broj svakom novom clanku
- **clanak_naslov** – tip varchar, sprema naslov svakog članka
- **clanak_sadrzaj** – tip text, sprema sadržaj svakog članka
- **clanak_timestamp** – tip int, sprema datum objave svakog članka

Tablica slike sadrži 2 stupca:

- **slika_id** – primarni ključ tablice slike, jedinstveni identifikator, automatski dodaje ID broj svakoj novoj slici
- **slika_img** – tip blob koji se koristi za spremanje slika i drugih multimedijjskih datoteka u bazu podataka

Tablica users sadrži 3 stupca:

- **user_id** – primarni ključ tablice users, jedinstveni identifikator, automatski dodaje ID broj svakom novom korisniku
- **user_name** – tip varchar, sprema ime korisnika
- **user_password** – tip varchar, sprema lozinku korisnika sa md5 kriptografskom hash funkcijom



Slika 13 - PhpMyAdmin sučelje sa tri tablice u bazi podataka

Nakon toga potrebno je povezati kreiranu bazu podataka sa našim CMS-om. Za povezivanje sa bazom podataka kreirana je nova datoteka **connection.php** koja sadrži jednostavan PDO kod za povezivanje (slika 14). ¹⁷**PDO** (engl. *PHP data objects*) je PHP proširenje prvi put uvedeno u PHP 5 te se koristi kao elegantno i jednostavno rješenje u povezivanju i dohvaćanju baze podataka.

```
1 |<?php
2 |
3 | try{
4 | $pdo = new PDO('mysql:host=localhost;dbname=bcstest_cms', 'bcstest_cms', 'customcms01');
5 | }catch (PDOException $e){
6 |     exit('Database error.');
```

Slika 14 - PHP pdo kod za spajanje sa bazom podataka

Kreirana datoteka connection.php uključuje se sa funkcijom **include_once('/connection.php')** u zaglavlju naslovne stranice za dohvaćanje podataka iz baze.

¹⁷ Prema <https://www.php.net/manual/en/intro.pdo.php>

7.3. Dohvaćanje članaka iz baze podataka

Za dohvaćanje članaka iz baze kreirana je nova datoteka **articles.php** te u njoj klasa “Article” koja sadrži dvije metode za dohvaćanje članaka iz baze. Prva metoda **fetch_all()** služi za dohvaćanje svih članaka u bazi, a druga **fetch_data()** služi za dohvaćanje pojedinih članaka prema njegovom ID-u (slika 15).

```

1 | k?php
2 |
3 | class Article{
4 |     public function fetch_all(){
5 |         global $pdo;
6 |
7 |         $query = $pdo->prepare("SELECT * FROM clanci");
8 |         $query->execute();
9 |
10 |        return $query->fetchAll();
11 |    }
12 |
13 |
14 |    public function fetch_data($article_id){
15 |        global $pdo;
16 |
17 |        $query = $pdo->prepare("SELECT * FROM clanci WHERE clanak_id = ?");
18 |        $query->bindValue(1, $article_id);
19 |        $query->execute();
20 |
21 |        return $query->fetch();
22 |    }
23 | }
24 |
25 |
26 | ?>

```

Slika 15 - Klasa Article u articles.php

Kako bi unutar funkcije bilo moguće koristiti varijablu **\$pdo** kreiranu u connection.php koja nam služi za povezivanje sa bazom, potrebno ju je napraviti globalnom. Nakon toga se koristi **\$pdo->prepare** koji priprema SQL upit za izvršenje i vraća objekt upita. Predložak upita može sadržavati nijedan ili više imenovanih parametara (: ime) ili upitnika (?) za koje će se stvarne vrijednosti zamijeniti kada se upit izvrši.

7.3.1. Prikazivanje članka na index.php

Za pozivanje tih metoda i prikazivanje članka na početnoj web stranici, na glavnoj stranici index.php uključuju se datoteke za povezivanje i dohvaćanje članka connection.php i articles.php, instancira se klasa Article te se poziva prva metoda za dohvaćanje svih članaka i sprema se u varijablu (slika 16).

```

1  <?php
2
3  include_once('includes/connection.php');
4  include_once('includes/article.php');
5
6
7  $article = new Article;
8  $articles = $article->fetch_all();
9
10
11
12
13  ?>
14

```

Slika 16 - Povezivanje sa bazom, instanciranje klase i pozivanje metode za dohvaćanje svih članaka

Nakon toga se foreach petljom ispisuje naslov i datum objave svih članaka spremljenih u bazi (slika 17). Poveznica uključena na naslov članka koja uzima id svakog članka vodi na stranicu svakog pojedinog članka, što je objašnjeno u nastavku.

```

24  <div class="clanci">
25  <h5>Članci:</h5>
26  <ol><?php foreach($articles as $article){ ?>
27
28  <li><a href="article.php?id=<?php echo ($article['clanak_id']);
29  ?> "> <?php echo $article['clanak_naslov']; ?></a> -
30  <small>Objavljeno: <?php echo date('l jS',
31  $article['clanak_timestamp']); ?> </small></li>
32  <?php } ?></ol>
33  </div>

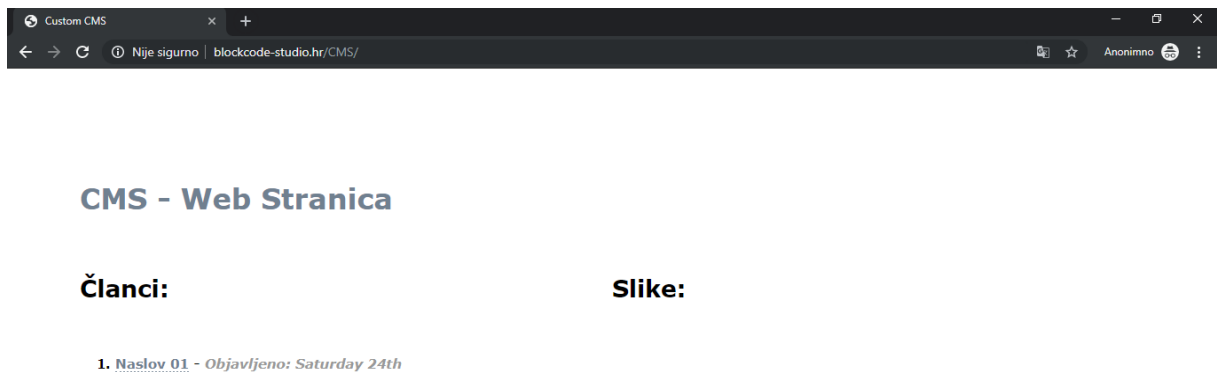
```

Slika 17 - Kod za prikaz liste članaka u index.php

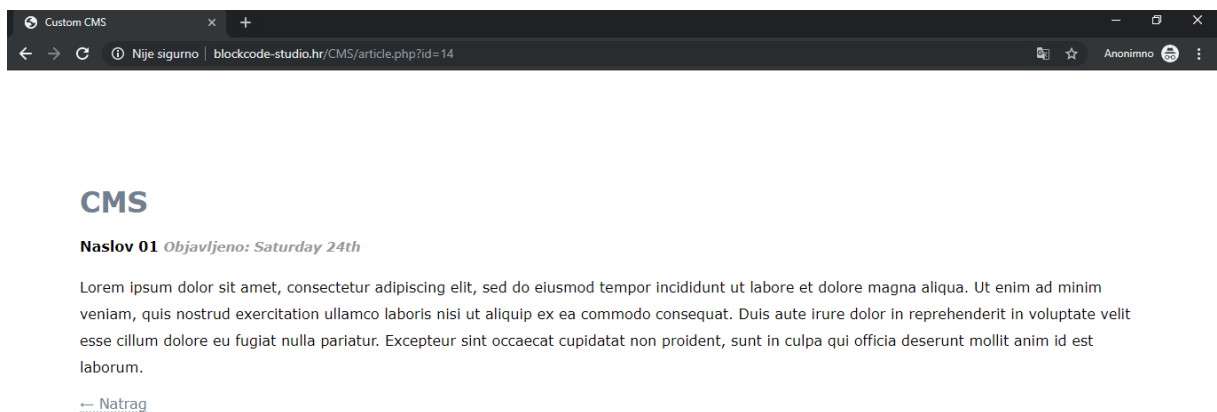
7.3.2. Dohvaćanje na stranici članka

Kako bi sa naslovne stranice mogli otvoriti svaki članak, kreirana je datoteka **article.php** u glavnoj mapi web sjedišta. U njoj su također uključene datoteke `connection.php` i `article.php` za povezivanje sa bazom, te instanciranje klase `Article`.

Nakon toga se funkcijom `isset` provjerava da li je prosljeđen ID koji se sprema u varijablu te koristi kao parameter u drugoj metodi klase `Article`. Ako nije, program vraća korisnika na `index.php`. U HTML-u se jednostavno ispisuju podatci članka sa prethodno prosljeđenim ID-om. Slike 18 i 19 prikazuju rezultat ispisa svih članaka i sadržaja pojedinog članka.



Slika 18 - Početna stranica sa svim člancima u bazi



Slika 19 - Prikaz pojedinog članka

7.4. Dohvaćanje slika iz baze podataka

Kod dohvaćanja i prikazivanja slika iz baze korišten je nešto drugačiji pristup nego kod prikazivanja članaka. Upload slika funkcionira na način da se slike spremaju u bazu te u mapu **uploads** kreiranu na serveru. Stoga za prikazivanje slika koristimo URL adresu slike u mapi uploads čiji naziv se podudara sa nazivom slike u spremljene u bazi podataka (slika 20). Na taj način osiguravamo da će biti prikazane samo one slike koje se nalaze u bazi i u mapi uploads.

```

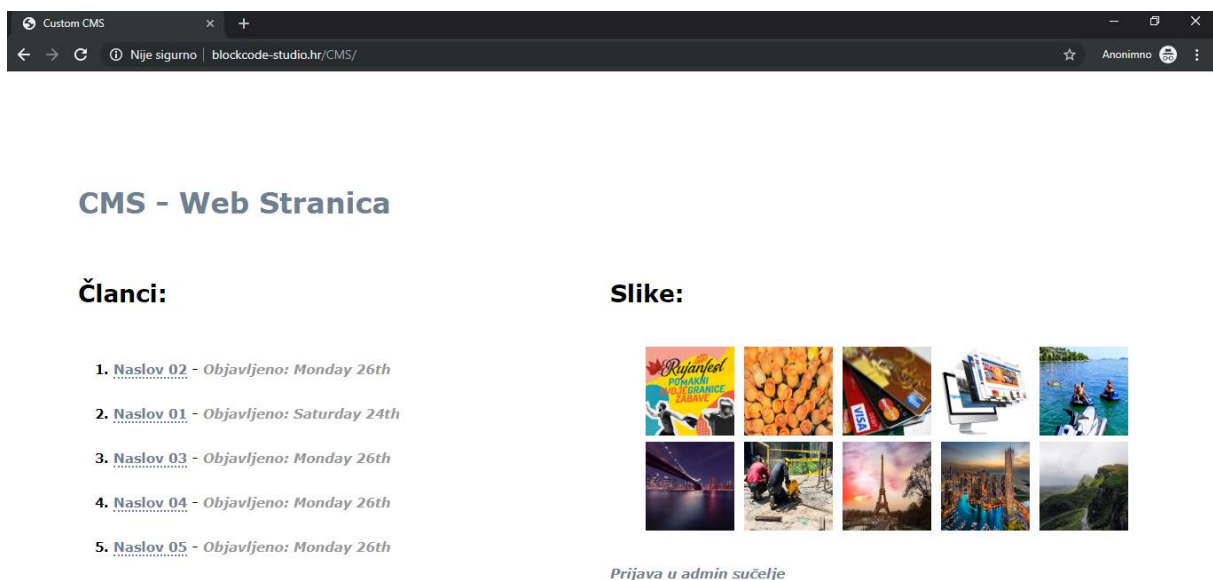
36 <h5>Slike:</h5>
37 <ol>
38 <?php
39
40 $select = $pdo->prepare("SELECT * FROM slike ");
41 $select->setFetchMode(PDO::FETCH_ASSOC);
42 $select->execute();
43 while($data=$select->fetch()){
44     ?>
45
46 
47 </div>
48

```

Slika 20 - Prikazivanje slika iz baze i mape uploads

Mapa uploads kreirana je unutar mape admin čija svrha je objašnjena u nastavku. Dohvaćenim slikama određene su visina i širina prikaza od 100px. Na kraju je dodana i poveznica za prijavu u upravljačko sučelje koja vodi u mapu admin/index.php.

Konačna struktura početne stranice web sjedišta prikazana je na slici 21.



Slika 21 - Početna stranica sa svim slikama i člancima iz baze

7.5. Upravljačko sučelje

U sljedećem koraku potrebno je napraviti upravljačko sučelje odnosno CMS koji će služiti za unos novih članaka i slika te brisanje postojećih. Za tu svrhu kreirana je posebna mapa **admin** te u njoj datoteka `index.php` koja sadrži obrazac za prijavu te glavnu strukturu CMS sučelja. U mapi `admin` su spremljene i ostale `php` datoteke za dodavanje i brisanje članaka i slika te datoteka za odjavu iz sustava. Mapa `admin` također sadržava i mapu `uploads` u koju se spremaju slike.

7.5.1. Prijava

U zaglavlju datoteke `index.php` pokreće se sesija sa `session_start()`; te se uključuje datoteka `connection.php` za povezivanje sa bazom. Nakon toga se vrši provjera da li je korisnik prijavljen. Ako korisnik nije prijavljen, prikazuje se obrazac za prijavu.

```

1  <?php
2
3  session_start();
4
5  include_once('../includes/connection.php');
6
7  if(isset($_SESSION['logged_in'])){
8
9      // CMS sučelje
10
11 }else{
12
13
14  ?>
15
16  <html>
17  <head>
18      <title>Custom CMS</title>
19      <link rel="stylesheet" href="../assets/style.css"/>
20  </head>
21  <body>
22  <div class="container">
23      <a href="index.php" id="logo"><h1>CMS - Prijava</h1></a>
24
25      <br><br>
26
27  <form action="index.php" method="post">
28
29      <input type="text" name="username" placeholder="korisničko ime"/>
30      <input type="password" name="password" placeholder="lozinka"/>
31      <input type="submit" value="Prijava" />
32  </form>
33  </div>
34
35  </body>
36  </html>
37
38
39

```

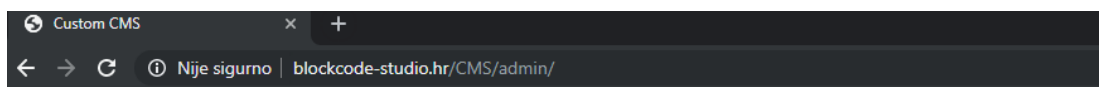
Slika 22 - Obrazac za prijavu u upravljačko sučelje front-end

Obrazac za prijavu se sastoji od dva HTML polja za upis, korisničko ime i lozinka, te gumba za prijavu (slika 22). Prije toga je napravljen novi korisnik u phpMyAdmin sučelju. Nakon izrade obrasca potrebno je napraviti validaciju. Provjerava se da li je korisnik kliknuo na prijavu te da li su sva polja ispunjena. Ako barem jedno od polja korisničko ime ili lozinka ostane prazno, program vraća poruku **“Sva polja su obavezna!”**. Ako su sva polja ispunjena provjerava se da li su uneseni točni podaci. Kod provjere točnosti podataka uzimaju se svi podaci iz tablice **users** u bazi te se uspoređuju uneseni podaci sa podacima u tablici. Broj vraćenih redaka sprema se u varijablu **\$num**. Ako je broj vraćenih redaka nije 1, program ispisuje poruku **“Krivi podaci!”**. Ako je broj vraćenih redaka 1, postavlja se **\$_SESSION['logged_in']** na TRUE te se osvježava stranica sa **header('Location: index.php')**; kako bi se prikazalo glavno CMS sučelje (slika 23).

```
11 }else{
12
13 if(isset($_POST['username'], $_POST['password'])){
14     $username = $_POST['username'];
15     $password = md5($_POST['password']);
16
17     if(empty($username) or empty($password)){
18         $error = 'Sva polja su obavezna!';
19     }else{
20         $query = $pdo->prepare('SELECT * FROM users WHERE user_name = ? AND user_password = ?');
21
22         $query->bindValue(1, $username);
23         $query->bindValue(2, $password);
24
25         $query->execute();
26
27         $num = $query->rowCount();
28
29         if($num==1){
30             $_SESSION['logged_in'] = true;
31             header('Location: index.php');
32             exit();
33         }else{
34             $error = 'Krivi podaci!';
35         }
36     }
37 }
```

Slika 23 - Obrazac za prijavu u upravljačko sučelje back-end

Obrazac za prijavu prikazan je na slici 24.



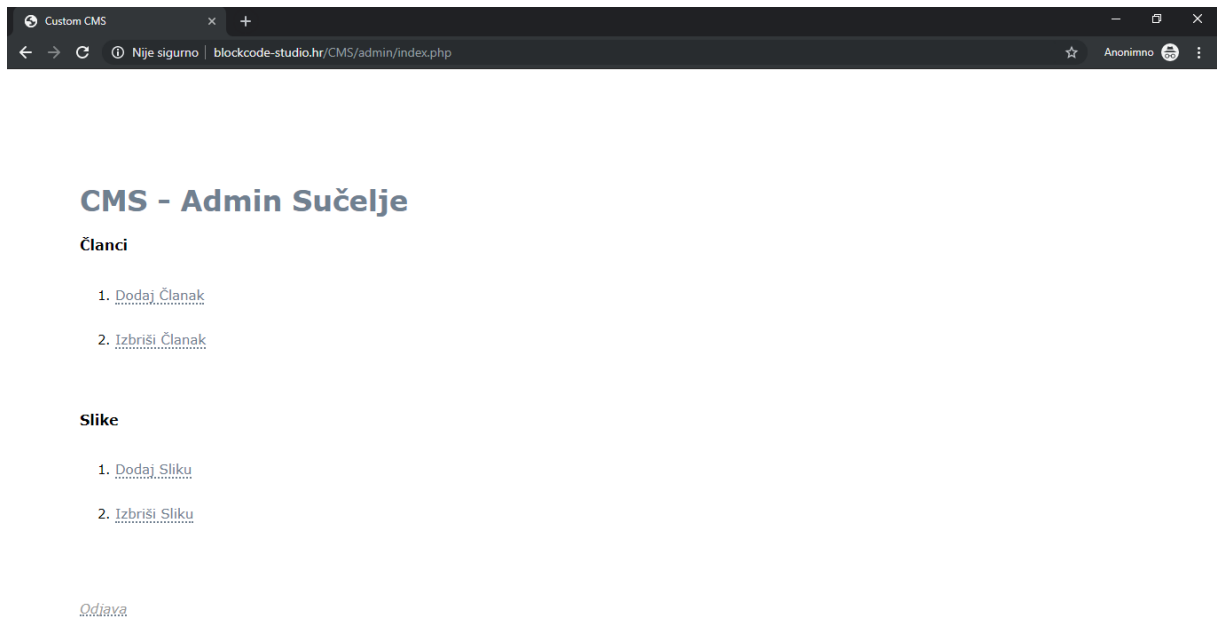
CMS - Prijava

<input type="text" value="korisničko ime"/>	<input type="text" value="lozinka"/>	<input type="button" value="Prijava"/>
---------------------------------------------	--------------------------------------	----------------------------------------

Slika 24 - Izgled obrasca za prijavu

7.5.2. Struktura upravljačkog sučelja

Početna strana upravljačkog sučelja u suštini je napravljena kao jednostavan meni sa četiri funkcije i odjave iz sustava. Sučelje je podjeljeno na dva djela od kojih prvi dio sadrži opcije za dodavanje i brisanje članaka, a drugi za dodavanje i brisanje slika. Jednostavan HTML kod sa dvije liste čije poveznice pozivaju datoteke za unos i brisanje koje ću objasniti u nastavku. Ispod te dvije liste umetnuta je poveznica na datoteku **logout.php** koja služi za odjavu iz sustava. Početna stranica upravljačkog sučelja prikazana je na slici 25.



Slika 25 - Upravljačko sučelje

7.5.3. Dodavanje članka

Za dodavanje članka napravljena je nova datoteku **add.php** u mapi admin. U zaglavlju datoteke add.php pokreće se sesija te se uključuje datoteka connection.php za povezivanje sa bazom. Nakon toga provjerava se da li je korisnik prijavljen što je vrlo bitno, inače bi svatko mogao dodavati članke iako nisu prijavljeni. Ako korisnik nije prijavljen, stranica nas preusmjerava na obrazac za prijavu u index.php.

U front-endu datoteke nalazi se jednostavan obrazac za unos članka. Obrazac se sastoji od polja naslov i sadržaj te gumba za dodavanje članka (slika 26).

```

38 <h4>Dodaj članak</h4>
39
40 <?php if(isset($error)){ ?>
41
42 <small style="color:#aa0000;"><?php echo $error;?></small>
43 </br></br>
44
45 <?php } ?>
46
47 <form action="add.php" method="post" autocomplete="off">
48 <input type="text" name="title" placeholder="Naslov" /> </br></br>
49 <textarea rows="15" cols="50" name="content" placeholder="Sadržaj"></textarea></br></br>
50 <input type="submit" value="Dodaj članak" />
51 </form>
52

```

Slika 26 - Dodavanje članka front-end

U back-endu datoteke nakon provjere da li je korisnik prijavljen, vrši se validacija koja provjerava da li su sva polja ispunjena prije slanja. Ako nisu ispisuje se poruka **“Sva polja su obavezna!”**. Ako su sva polja ispunjena, unešeni podaci se spremaju u varijable te šalju u odgovarajuća polja u bazi podataka. Osim naslova i sadržaja članka, kao treću vrijednost šaljemo i trenutno vrijeme objave sa funkcijom **time()**; i spremamo je u polje **clanak_timestamp** u tablici **clanci**. Kompletan back-end kod prikazan je na slici 27.

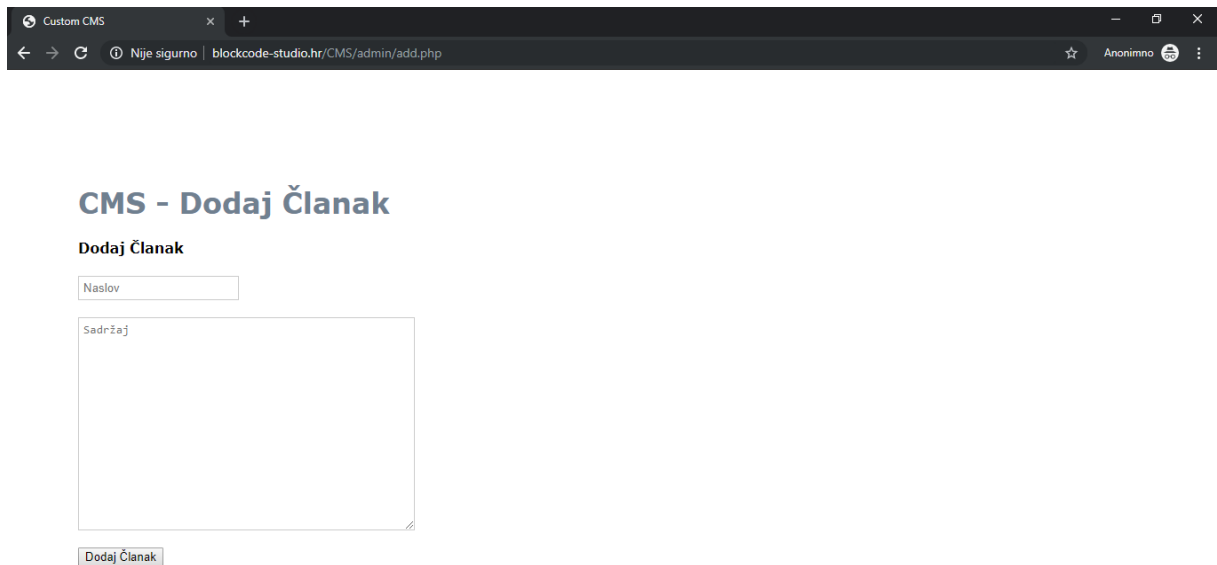
```

3 session_start();
4 include_once('../includes/connection.php');
5
6
7 if(isset($_SESSION['logged_in'])){
8     if(isset($_POST['title'], $_POST['content'])){
9         $title = $_POST['title'];
10        $content = nl2br($_POST['content']);
11
12        if(empty($title) or empty($content)){
13            $error = 'Sva polja su obavezna!';
14        }else{
15            $query = $pdo->prepare('INSERT INTO clanci (clanak_naslov, clanak_sadrzaj, clanak_timestamp) VALUES (?,?);');
16
17            $query->bindValue(1, $title);
18            $query->bindValue(2, $content);
19            $query->bindValue(3, time());
20
21            $query->execute();
22
23            header('Location: index.php');
24        }
25    }
26

```

Slika 27 - Dodavanje članka back-end

Nakon izvršenja SQL upita, Korisnik se preusmjerava natrag na početnu stranicu upravljačkog sučelja. Stranica za dodavanje članka prikazana je na slici 28.



The screenshot shows a web browser window with the address bar displaying 'blockcode-studio.hr/CMS/admin/add.php'. The page content includes a heading 'CMS - Dodaj Članak', a sub-heading 'Dodaj Članak', a text input field labeled 'Naslov', a large text area labeled 'Sadržaj', and a button labeled 'Dodaj Članak'.

Slika 28 - Dodavanje članka u upravljačkom sučelju

7.5.4. Brisanje članaka

Za brisanje članaka napravljena je nova datoteka **delete.php** u mapi admin. U zaglavlju datoteke kao i kod add.php pokreće se sesija, uključuje se datoteka connection.php za povezivanje sa bazom ali i article.php jer je potrebno dohvaćanje podataka iz baze. Isto tako je potrebno opet instancirati klasu Article za dohvaćanje podataka.

U front-endu datoteke za brisanje članaka nalazi se HTML oznaka <select> odnosno padajući izbornik koji izlistava naslove svih članaka u bazi pomoću foreach petlje. Obrazac funkcionira na način da kada korisnik odabere članak iz padajuće liste, odabrana vrijednost odnosno id članka se automatski šalje pomoću JavaScript-a i članak se briše iz baze (slika 29).

```

37 <h4>Odaberi članak za brisanje:</h4>
38 <form action="delete.php" method="get">
39 <select onchange="this.form.submit();" name="id">
40 <option disabled selected value> -- Odaberi -- </option>
41 <?php foreach($articles as $article){ ?>
42 <option value="<?php echo $article['clanak_id'];?>">
43 <?php echo $article['clanak_naslov'];?>
44 </option>
45 <?php } ?>
46 </select>
47 </form>

```

Slika 29 - Brisanje članaka front-end

U back-endu opet provjeravamo da li je korisnik prijavljen, dohvaćamo članke iz baze sa metodom fetch_all(); te spremamo odabrani id članka u varijablu \$id. Tu varijablu ubacujemo kao vrijednost u SQL upit za brisanje članka u tablici clanci prema stupcu clanak_id (slika 30).

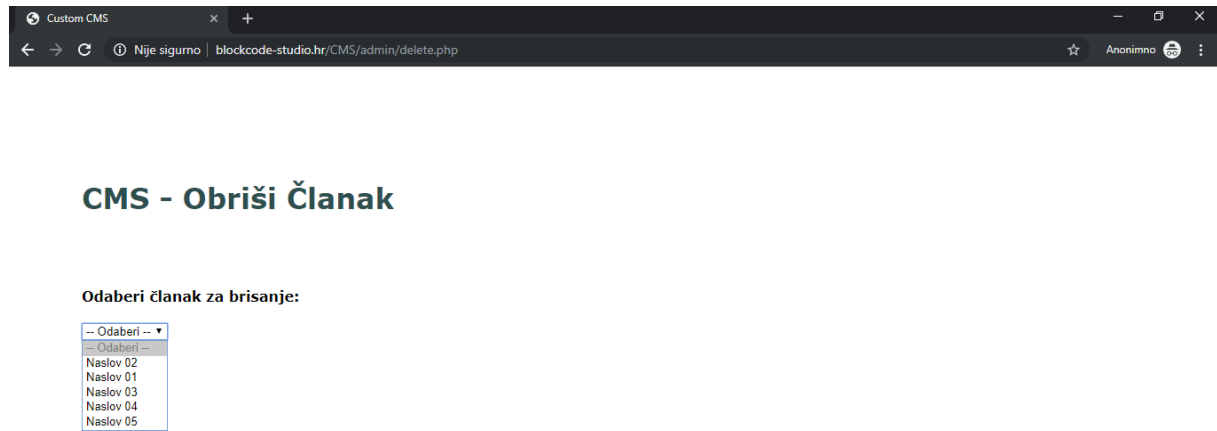
```

11 if(isset($_SESSION['logged_in'])){
12     if(isset($_GET['id'])){
13
14         $id = $_GET['id'];
15         $query = $pdo->prepare('DELETE FROM clanci WHERE clanak_id = ?');
16
17         $query->bindValue(1, $id);
18         $query->execute();
19
20         header('Location: delete.php');
21     }
22     $articles = $article->fetch_all();
23
24     ?>

```

Slika 30 - Brisanje članaka back-end

Nakon izvršavanja upita, korisnik je preusmjeren na istu stranicu odnosno stranica se osvježava. Slika 31 prikazuje izgled stranice za brisanje članaka.



Slika 31 - Brisanje članaka u upravljačkom sučelju

7.5.5. Dodavanje slika

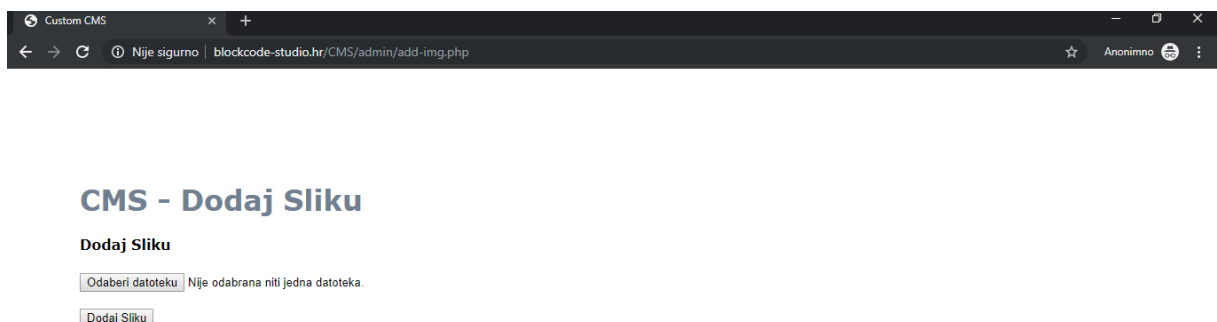
Za dodavanje slike napravljena je nova datoteka **add-img.php** u mapi admin. U zaglavlju datoteke add-img.php pokreće se sesija te se uključuje datoteka connection.php za povezivanje sa bazom. Nakon toga se provjerava da li je korisnik prijavljen.

U front-endu datoteke nalazi se jednostavan obrazac upload slika (slika 32).

```
62 <n4>Dodaj Sliku</n4>
63
64 <?php if(isset($error)){ ?>
65     <small style="color:#aa0000;"><?php echo $error;?></small>
66     </br></br>
67
68
69 <?php } ?>
70
71 <form action="add-img.php" method="post" enctype="multipart/form-data">
72
73     <input type="file" name="image" />
74     </br></br>
75     <input type="submit" name="submit" value="Dodaj Sliku" />
76 </form>
77
```

Slika 32 – Obrazac za upload slike

U back-endu datoteke nakon provjere da li je korisnik prijavljen, vrši se validacija koja provjerava nakon što je korisnik pritisnuo gumb za slanje obrasca, da li je datoteka koju šalje formata .jpg, .png, .jpeg ili .gif . Ako nije ispisuje se poruka **“Dopuštene vrste datoteka: JPG, JPEG, PNG i GIF”**. Ako je, slika se sprema u mapu uploads i u tablicu slike u bazi. Izgled obrasca za upload slike prikazuje slika 33.

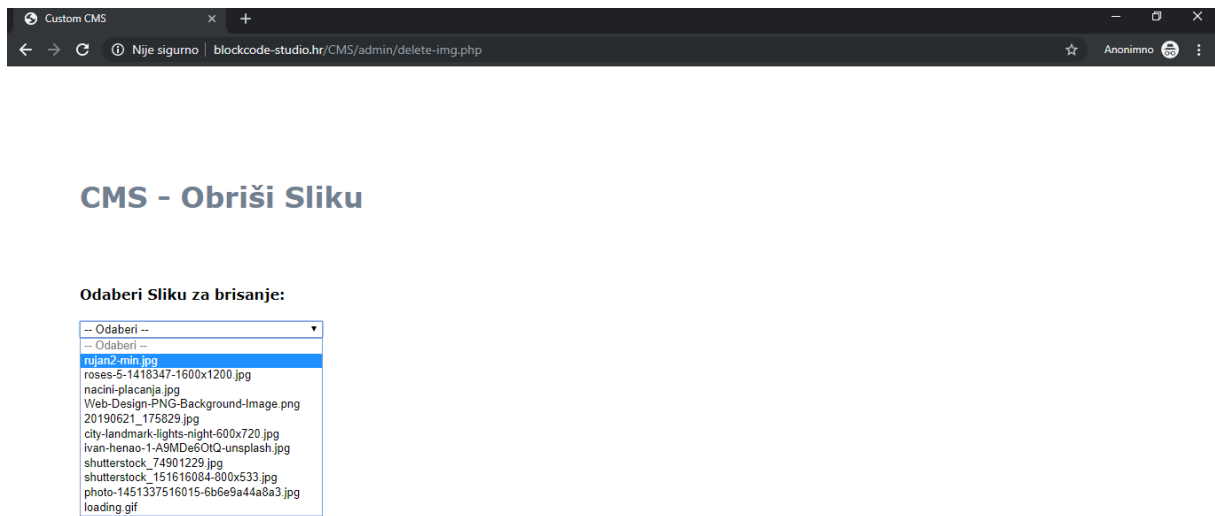


Slika 33 - Dodavanje slika u upravljačkom sučelju

7.5.6. Brisanje slika

Za brisanje slika napravljena je nova datoteka **delete.img.php** u mapi admin. Sadržaj datoteke je skoro identičan sadržaju u datoteci za brisanje članka koja je objašnjena u prethodnoj cjelini.

U datoteci za brisanje slika nalazi se padajući izbornik koji izlistava imena svih slika u bazi pomoću foreach petlje. Odabirom slike pomoću JavaScripta šalje se id slike u SQL upit koji briše sliku sa tim id-om u bazi. Izgled stranice za brisanje slika prikazan je na slici 34.



Slika 34 - Brisanje slika u upravljačkom sučelju

7.5.7. Odjava

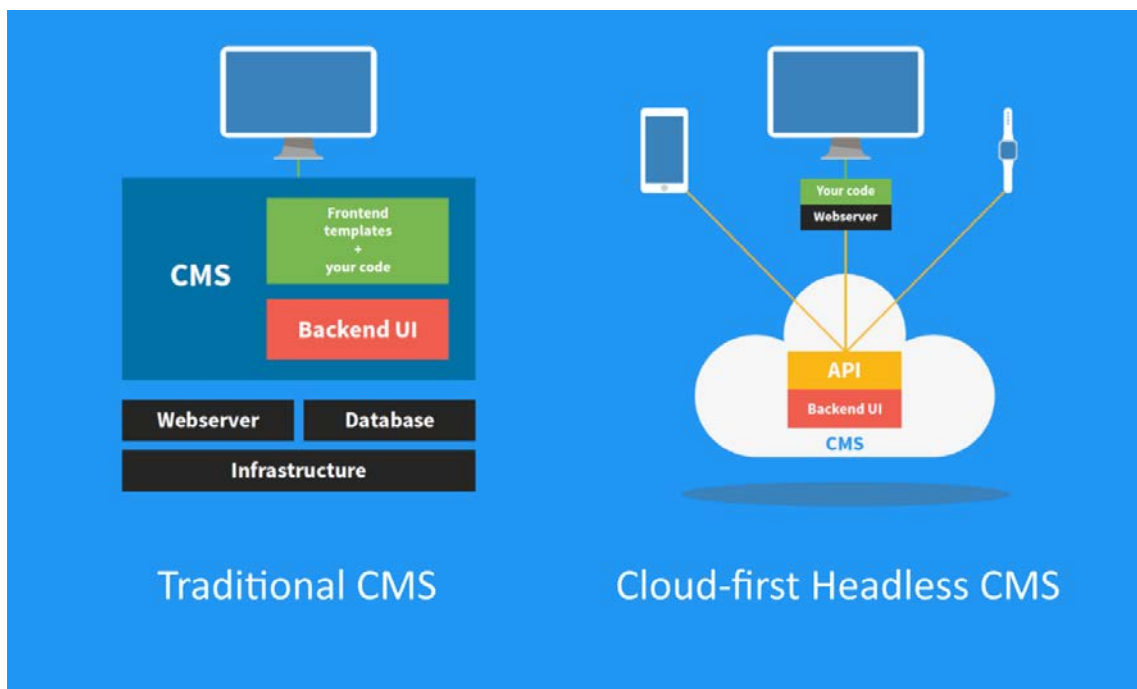
Odjava iz sustava vrši se pomoću kreirane datoteke **logout.php** u mapi admin. Datoteka sadrži jednostavan php kod za početak sesije te uništavanje sesije (slika 35). Nakon uništavanja sesije korisnik se preusmjerava na početnu stranicu upravljačkog sučelja index.php koja, pošto je sesija završena i korisnik više nije prijavljen, prikazuje obrazac za prijavu.

```
1 <?php
2
3 session_start();
4
5 session_destroy();
6
7 header('Location: index.php');
8
9 ?>
```

Slika 35 - Logout.php kod

8. Headless CMS

Spoment je ranije Headless CMS kao novi val u web CMS području. Naime Headless CMS arhitektura čini prilagođene CMS-ove zastarijelima. ¹⁸U posljednjih 15 godina, tržište i tehnologija CMS-a transformirali su se u korak s promjenama digitalnog okruženja i očekivanjima korisnika u smislu funkcionalnosti na više uređaja. Sada se nova generacija CMS tehnologije – „**cloud-first headless CMS**“ - sprema za revoluciju u industriji upravljanja sadržajem. ¹⁹Izraz "headless" odnosno „bez glave“ dolazi od koncepta sjećanja „glave“ (front-enda, tj. izgleda web sjedišta) od "tijela" (back-enda, tj. spremišta sadržaja). Dok tradicionalni CMS obično kombinira sadržajne i prezentacijske slojeve web stranice, CMS bez glave je samo komponenta sadržaja i u potpunosti se usredotočuje na administrativno sučelje za tvorce sadržaja, olakšavajući tijek radnje, suradnje i organizaciju sadržaja. Ne bavi se prezentacijskim slojevima, predlošcima, strukturom web mjesta ili dizajnom, već pohranjuje sadržaj u čistom formatu i omogućuje pristup drugim komponentama (npr. alatima za analitiku) putem API-a. CMS koncept bez glave rođen je prema zahtjevima digitalne ere i poslovnim potrebama za privlačenje korisnika personaliziranim sadržajem putem više kanala u svim fazama kupčevog puta. Kako se sadržaj u CMS-u bez glave smatra čistim (jer nema priloženi prezentacijski sloj), samo se jedan primjerak može koristiti za prikaz na bilo kojem uređaju: web, mobilni, tablet, pametni satovi itd. Slika 36 prikazuje razliku u strukturi standardnih CMS rješenja i Cloud first Headless CMS-a.



Slika 36 - Standardni CMS vs Cloud-first Headless CMS

¹⁸ Prema <https://hackernoon.com/how-i-built-a-cms-and-why-you-shouldnt-daff6042413a>, How I built a CMS, and why you shouldn't

¹⁹ Prema https://en.wikipedia.org/wiki/Headless_content_management_system

Za razliku od tradicionalnih CMS sustava, CMS s bez „glave“ fokusiraju se samo na upravljanje sadržajem i čine sadržaj dostupnim putem API-a bilo kojoj aplikaciji. Kako im nedostaje „glava“ koja bi normalno diktirala kako sadržaj treba prikazati, CMS bez glave ostavlja ovaj dizajnerski element u potpunosti na programeru.

U nastavku je navedeno nekoliko razloga zašto se standardni CMS sustavi smatraju zastarjelima u odnosu na headless CMS sustave

9. Zastarjelost standardnih CMS sustava u odnosu na Headless CMS sustave

9.1. Standardni CMS ograničava kreativnost

Broj 1 prigovor većine programera o CMS-u je što zabrlja njihov HTML kod i tjera ih da traže zaobilazno rješenje. CMS bez glave omogućava apsolutnu slobodu i ne ostavlja trag u rezultirajućem HTML kodu. Sve što je potrebno je pozvati njegov REST API koristeći programski jezik po želji za preuzimanje sadržaja iz repozitorija. Način prikazivanja sadržaja ovisi u potpunosti o programeru.

9.2. Standardna CMS sučelja previše su složena

Mnogi tradicionalni CMS sustavi su znatno narasli u posljednjih deset godina. Iako su svi započeli s idejom pružanja izvrsnog rješenja za upravljanje sadržajem, većina njih nije uspjela izbjeći zaostajanje karakteristika jer su se proširili na područja poput e-trgovine, automatizacije marketinga, sustava rezervacija, marketinga putem e-pošte itd.

Iako je možda prikladno imati sve na jednom mjestu za neke korisnike, za nove korisnike je prilično zahtjevno učenje kako koristiti CMS. Ako sve što žele je upravljanje sadržajem, previše mogućnosti utječe na njihovu produktivnost.

CMS bez glave dolazi iz drugačije perspektive: on je samo jedan dio slagalice mikro usluga i usredotočen je na pružanje znatno prostranijeg korisničkog sučelja usmjerenog upravo na sadržaj. U isto vrijeme, obično pruža API za upravljanje sadržajem koji omogućuje stvaranje vlastitog sučelja za uređivanje na vrhu repozitorija sadržaja.

Ovo može biti korisno u stvaranju pojednostavljenog korisničkog sučelja ili integriracije mogućnosti uređivanja sadržaja unutar vlastite aplikacije, umjesto preusmjerenja korisnika na drugo sučelje.

9.3. Standardni CMS je preskup

„Nismo htjeli platiti X dolara za komercijalni CMS, pa smo odlučili napisati vlastiti.“ To možete čuti od nekih programera. Nema načina da dugoročno uštedite novac prilagođenim CMS-om, osim ako vam treba nešto puno jednostavnije od pravog CMS-a, kao što je upravljanje popisom vijesti ili nešto slično izrađenom primjeru CMS-a ih prethodnog poglavlja.

Danas postoji čitavo mnoštvo besplatnih CMS-a s otvorenim kodom te headless CMS koji nudi cijene temeljene na potrošnji koje će uvijek biti manje od troškova razvoja i pokretanja vlastitog CMS-a.

9.4. Standardni CMS nije siguran

Za mnoge organizacije sigurnost CMS-a je noćna mora. Razmišljanje većine programera je: "Ako napišemo vlastiti CMS, biti će kompliciranije za hakere da nađu nedostatke." Iako je istina da hakeri mogu utjecati na poznati sigurnosni problem, široko korišten CMS obično se intenzivno testira. Zapravo, glavni izvor sigurnosnih problema jesu tvrtke koje ne primjenjuju najnovije hitne popravke na različite upotrijebljene dodatke i module.

Headless CMS je uvijek ažuran. CMS host je izravno dobavljač koji poznaje kod kao i infrastrukturu i može posvetiti odgovarajuću pažnju sigurnosti.

9.5. Standardni CMS ne odgovara arhitekturi

To je nekada bio valjan razlog u određenim scenarijima. Očekivalo se da će se većina tradicionalnih CMS rješenja koristiti kao središnja platforma na kojoj se može graditi, što znači da je aplikacijski kod bio čvrsto povezan s CMS-om, kao što je prikazano na slici 36.

CMS platforma, programski jezik, ciklusi nadogradnje, skalabilnost, sigurnost, sve te komponente ograničavaju programere u razvoju.

Ovo su samo neki od mnogih razloga zbog čega više nije praktično koristiti vlastite prilagođene CMS-ove. Mnoge digitalne agencije i dalje pokreću vlastiti CMS za svoje klijente. Neki od njih čak ih namjerno koriste kako bi zaključili svoje klijente kako ne bi prešli na drugu agenciju. Ubiti, imati vlastiti CMS za digitalnu agenciju više nema apsolutno nikakve prednosti.

Drugi problem je strah od promjena. U nekim slučajevima to je i emocionalna ili politička odluka. CMS je napisao osnivač agencije prije mnogo godina ili je dijete njihovog najboljeg programera koji je, na kraju krajeva, jedina osoba koja zna kako CMS radi.

Naravno uvijek postoje iznimke i slučaju kada je pisanje prilagođenog CMS-a korisno. Ljudi uvijek podcjenjuju količinu napora koji ulaže u izgradnju pravog CMS-a. Odlična je vježba programiranja, ali ako to nije osnovna djelatnost potrebno ga je izbjegavati.

Zaključak

Izradom ovog rada te iz vlastitog iskustva koristeći CMS sustave, shvatio sam da postoji bezbroj mogućnosti korištenja CMS-a te ih je teško obraditi u jednom radu. Nastojao sam obuhvatiti najbitnije i najosnovnije značajke te sam pokušao barem predočiti sliku o CMS sustavima, njihovoj ulozi, korisnosti i svrsi. Proces izrade web sjedišta kroz godine je postao nevjerojatno brz i jednostavan, a brojne mogućnosti sustava za upravljanje sadržajem uveliko su pridonjele tome. Ubrzani napredak tehnologije mobilnih uređaja, društvenih mreža, interneta i dr. utječu i na razvoj web CMS-ova te se i oni sukladno tome neprestano prilagođavaju i nadograđuju. Prije 20 godina bilo je gotovo nezamislivo da običan korisnik bez nekog programerskog znanja ili pomoći može ažurirati svoju web stranicu, dodavati i mijenjati sadržaj a kamoli napraviti funkcionalno i izgledom reprezentativno web sjedište ili web trgovinu. Danas, sa brojim materijalima i video tutorijalima na webu, je čak možda i trivijalno.

Kao i u svim područjima, tako se i CMS paradigme mijenjaju i poboljšavaju. Po svim pokazateljima pred nama je doba Headless CMS-a, a vrijeme će pokazati koja će sve intuitivna i jednostavna rješenja web CMS sustavi nove generacije pružiti.

Popis slika

Slika 1 - Vremenska crta razvoja WCMS-a	7
Slika 2 - Porast tržišnog udjela pretraživanja weba na mobilnim uređajima od 2009. do 2018. godine	11
Slika 3 - WordPress sučelje za objavu članka	12
Slika 4 - Prijava u Wordpress CMS stranice blockcode.hr	14
Slika 5 - 10 najkorištenijih web CMS sustava na tržištu i njihov udio na tržištu 2018. godine	19
Slika 6 - Wordpress.org	20
Slika 7- Statistika korištenja Wordpress CMS sustava od 2011. do 2018. godine	21
Slika 8 - Joomla.org	22
Slika 9 - Statistika korištenja Joomla! CMS sustava od 2011. do 2018. godine	23
Slika 10 – Drupal.org	24
Slika 11 - Statistika korištenja Drupal CMS sustava od 2011. do 2018. godine	25
Slika 12 - Struktura početne stranice web sjedišta	31
Slika 13 - PhpMyAdmin sučelje sa tri tablice u bazi podataka	32
Slika 14 - PHP pdo kod za spajanje sa bazom podataka	33
Slika 15 - Klasa Article u articles.php	34
Slika 16 - Povezivanje sa bazom, instanciranje klase i pozivanje metode za dohvaćanje svih članaka	35
Slika 17 - Kod za prikaz liste članaka u index.php	35
Slika 18 - Početna stranica sa svim člancima u bazi	36
Slika 19 - Prikaz pojedinog članka	36
Slika 20 - Prikazivanje slika iz baze i mape uploads	37
Slika 21 - Početna stranica sa svim slikama i člancima iz baze	37
Slika 22 - Obrazac za prijavu u upravljačko sučelje front-end	38
Slika 23 - Obrazac za prijavu u upravljačko sučelje back-end	39
Slika 24 - Izgled obrasca za prijavu	39
Slika 25 - Upravljačko sučelje	40
Slika 26 - Dodavanje članaka front-end	41
Slika 27 - Dodavanje članaka back-end	41
Slika 28 - Dodavanje članaka u upravljačkom sučelju	42
Slika 29 - Brisanje članaka front-end	43
Slika 30 - Brisanje članaka back-end	43
Slika 31 - Brisanje članaka u upravljačkom sučelju	44
Slika 32 – Obrazac za upload slike	45
Slika 33 - Dodavanje slika u upravljačkom sučelju	45
Slika 34 - Brisanje slika u upravljačkom sučelju	46
Slika 35 - Logout.php kod	46
Slika 36 - Standardni CMS vs Cloud-first Headless CMS	47

Izvori sa WWW-a

SLIKA 1:

<https://images.contentstack.io/v3/assets/blt1264fa9b448be721/blt429cbb9b98c013e5/5c1a96e91036144e3893f243/download>

SLIKA 2:

<https://images.contentstack.io/v3/assets/blt1264fa9b448be721/bltb8edba8974054ac0/5c1a95d2434bb30537207de3/download>

SLIKA 3:

<https://sitechecker.pro/wp-content/uploads/2019/02/wordpress2.png>

SLIKA 5:

<https://websitesetup.org/wp-content/uploads/2018/12/CMS-table-02-02-768x349.png>

SLIKA 7:

<https://websitesetup.org/wp-content/uploads/2018/12/WordPress-BuiltWith-640x363.png>

SLIKA 9:

<https://websitesetup.org/wp-content/uploads/2018/12/Joomla-BuiltWith-640x362.png>

SLIKA 11:

<https://websitesetup.org/wp-content/uploads/2018/12/Drupal-BuiltWith-640x358.png>

SLIKA 36:

https://hackernoon.com/hn-images/1*uDwr6iOPIEM_VyOExBVbHw.png

Literatura

1. <https://hr.wikipedia.org/wiki/CMS>
Pristupljeno 9.8.2019.
2. Brent Heslop, History of Content Management Systems and Rise of Headless CMS
<https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms>,
Pristupljeno 10.8.2019.
3. What Is CMS and How to Choose It for Your Website?
<https://sitechecker.pro/what-is-cms/>
Pristupljeno 12.8.2019.
4. https://en.wikipedia.org/wiki/Web_content_management_system
Pristupljeno 12.8.2019.
5. Orion Cassetto, Why CMS Platforms Are Common Hacking Targets (and what to do about it)
<https://www.imperva.com/blog/cms-security-tips/>
Pristupljeno 12.8.2019.
6. Nick Schaferhoffp, Popular CMS by Market Share
<https://websitesetup.org/popular-cms/>
Pristupljeno 13.8.2019.
7. Usage of content management systems
https://w3techs.com/technologies/overview/content_management/all
Pristupljeno 15.8.2019.
8. CMS comparison 2019: The most popular open source systems
<https://www.ionos.com/digitalguide/hosting/cms/cms-comparison-a-review-of-the-best-platforms/>
Pristupljeno 16.8.2019.
9. Nick Schaferhoffp, WordPress vs Joomla vs Drupal (Comparison)
<https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>
Pristupljeno 16.8.2016.
10. <https://hr.wikipedia.org/wiki/HTML>
Pristupljeno 18.8.2019.
11. <https://en.wikipedia.org/wiki/HTML5>
Pristupljeno 18.8.2019.

12. <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>
Pristupljeno 18.8.2019.
13. <https://hr.wikipedia.org/wiki/PHP>
Pristupljeno 18.8.2019.
14. <https://en.wikipedia.org/wiki/PHP>
Pristupljeno 18.8.2019.
15. <https://en.wikipedia.org/wiki/MySQL>
Pristupljeno 18.8.2019.
16. <https://en.wikipedia.org/wiki/JavaScript>
Pristupljeno 18.8.2019.
17. <https://www.php.net/manual/en/intro.pdo.php>
Pristupljeno 22.8.2019.
18. Petr Palas, How I built a CMS, and why you shouldn't
<https://hackernoon.com/how-i-built-a-cms-and-why-you-shouldnt-daff6042413a>
Pristupljeno 28.8.2019.
19. https://en.wikipedia.org/wiki/Headless_content_management_system
Pristupljeno 28.8.2019.