

Teme i predlošci za Wordpress

Jelušić, Matea

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:911947>

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

Download date / Datum preuzimanja: **2024-07-13**



Repository / Repozitorij:

[University North Digital Repository](#)





Sveučilište Sjever

Završni rad br. 755/MM/2021

Teme i predlošci za WordPress

Matea Jelušić, 2891/336

Varaždin, rujan 2021. godine



Sveučilište Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 755/MM/2021

Teme i predlošci za WordPress

Studentica

Matea Jelušić, 2891/336

Mentor

mr.sc. Vladimir Stanislavljević, dipl. ing.

Varaždin, rujan 2021. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za multimediju		
STUDIJ	preddiplomski stručni studij Multimedija, oblikovanje i primjena		
PRISTUPNIK	Matea Jelušić	JMBAG	0130332888
DATUM	10.09.2021.	KOLEGIJ	Programski alati 2
NASLOV RADA	Teme i predlošci za Wordpress		
NASLOV RADA NA ENGL. JEZIKU	Wordpress themes and templates		
MENTOR	Vladimir Stanisavljević	ZVANJE	viši predavač

ČLANOVI POVJERENSTVA	
1.	doc.dr.sc. Andrija Bernik - predsjednik povjerenstva
2.	Dražan Crčić, dipl.ing. predavač - član povjerenstva
3.	mr.sc. Vladimir Stanisavljević, v.pred. - mentor
4.	doc.dr. sc. Domagoj Frank - zamjenski član
5.	

Zadatak završnog rada

BROJ	755/MM/2021
OPIS	<p>Mnoge manje tvrtke i institucije trebaju fleksibilno izrađene web stranice, ali samo rijetke pristupe njihovoj implementaciji samostalno. Za takve potrebe web stranice mogu se relativno jednostavno izraditi primjenom sustava za upravljanjem sadržajem (CMS) i odgovarajućim dodacima. Jedna popularana CMS platforma je WordPress (WP).</p> <p>U radu je potrebno:</p> <ul style="list-style-type: none">- opisati čimbenike potrebne za uspješnu implementaciju web stranice kroz WordPress i sve što je potrebno kako bi se napravila vlastite teme kao polaznog predloška za daljnju izradu pojedinačnih stranica- detaljno opisati mogućnosti koje nudi WP i arhitekturu njegovih tema te glavne programske konstrukte koji se koriste u temama- instalirati WordPress na testnom poslužitelju i prilagoditi sustav za vlastite potrebe- izvesti analizu za složeniju web stranicu i napraviti dizajn s vlastitim HTML-om, CSS-om i JavaScriptom te sve potrebno ugraditi u vlastitu WP temu- demonstrirati rad izrađene WP teme na primjeru web mjesta <p>Detaljno opisati izvedbu web stranice te programski kod korišten u radu. Dokumentirati sve potrebne korake pri izvedbi cjelovitog rješenja i opisati iskustva stečena pri uspostavi sustava web trgovine. U zaključku razmotriti moguće varijante izvedbe sustava i eventualna naknadna poboljšanja.</p>

ZADATAK URUČEN

15.09.2021.



POTPIS MENTORA

Mateo Jelušić

Predgovor

Zahvaljujem se mentoru prof. mr.sc. Vladimiru Stanisavljeviću, dipl. ing. na stručnim savjetima, strpljenju i vodstvu tijekom pisanja ovog završnog rada. Velika hvala svim djelatnicima Sveučilišta Sjever koji su svojim radom pomogli u stjecanju moga znanja. Želim se zahvaliti svim kolegama i prijateljima koji su svojim prisustvom uljepšali moje studentske dane. Najveća hvala ide dragom Bogu i obitelji koja mi je omogućila život i pružala bezuvjetnu podršku i razumijevanje tijekom cijelog studiranja.

Sažetak

Temeljni cilj ovog završnog rada je predstaviti mogućnosti i prednosti korištenja WordPressa za izgradnju specifičnog web rješenja. WordPress je jedan od najpopularnijih sustava za upravljanje i uređivanje web sadržaja koji nudi širok spektar mogućnosti – od osnovnog korištenja CMS sustava za upravljanje web sadržajem pomoću jednostavnog sučelja do kompleksnijih radnji kao što su programiranje vlastitih predložaka koji sadrže prilagođene elemente. U praktičnom dijelu rada izrađena je funkcionalna web stranica za potrebe jedne župe i njene zajednice te su na temelju toga objašnjeni svi osnovni postupci potrebni za izradu vlastite WordPress web stranice. Ovaj rad će se moći koristiti u edukacijske svrhe za pomoć svima onima koji žele naučiti osnove kodiranja i korištenja WordPress predložaka.

Ključne riječi: CMS, WordPress, prilagođen WordPress predložak, WordPress dodaci, web sjedište

Summary

The main goal of this undergraduate thesis is to introduce the possibilities and advantages of using WordPress for building a specific web solution. WordPress is one of the most popular Content Management Systems (CMS) offering a range of possibilities - from the basic use of the system with a simple interface to more complex actions such as programming WordPress themes that contain custom elements. In the practical part of this thesis, a functional website was created for the needs of a Catholic parish and its community. Based on that, all the basic procedures needed for a functional WordPress website were explained. This bachelor paper will be available for educational purposes to help anyone who wants to learn the basics of coding and using WordPress templates and themes as such.

Keywords: CMS, WordPress, custom WordPress template, WordPress plugins, website

Popis korištenih kratica

CMS	Content Management System Sustav za upravljanje sadržajem web mjesta.
HTML	HyperText Markup Language Sintaksa za obilježavanje hipertekstualnih dokumenata.
CSS	Cascading Style Sheets Stilski jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću HTML jezika.
JS	JavaScript Skriptni programski jezik koji se izvršava u pregledniku na strani korisnika.
PHP	PHP: Hypertext Preprocessor Programski jezik namijenjen programiranju dinamičnih web stranica.
GPL	General Public License Licenca za softver otvorenog koda.
XAMPP	Cross-Platform, Apache, MariaDB/MySQL, PHP Pearl Platforma koja služi za testiranje i razvoj dinamičkih web stranica na lokalnom računaru.
MVC	Model–View–Controller Obrazac softverske arhitekture.
SEO	Search Engine Optimization Optimizacija za tražilice čiji je cilj povećati vidljivost određene web stranice u rezultatima internetskog pretraživača.
API	Application programming interface Skup specifikacija koje programeri slijede kako bi se služili uslugama operacijskog sustava ili nekog drugog složenog programa kao standardne biblioteke rutina, struktura podataka, objekata i protokola.

Sadržaj

1. Uvod.....	1
2. Wordpress kao CMS sustav	3
2.1. Razvoj WordPress-a kroz povijest	3
2.2. Generiranje web stranice.....	4
2.3. Upravljanje sadržajem.....	5
3. Instalacija WordPressa na lokalnom računalu.....	7
3.1. Instaliranje XAMPP-a.....	7
3.2. Pripremanje baze podataka.....	8
3.3. Konfiguracija WordPressa	8
3.4. Prijavljanje u WordPress.....	10
3.5. Osnovno kretanje kroz administratorsko sučelje.....	11
3.6. Izrada objave i stranice.....	12
4. Struktura i oblikovanje WordPress predloška	14
4.1. Vrste datoteka predložaka	15
4.2. Hijerarhija predložaka	15
4.3. Funkcija petlje (The Loop).....	16
4.4. WordPress dodaci i programčići	18
5. Razvoj vlastite teme	20
5.1. Postavljanje dizajnerskog rješenja	20
5.2. Bootstrap	21
5.3. index.php.....	22
5.4. style.css	22
5.5. template.css	24
5.6. functions.php	24
5.7. header.php	25
5.7.1. Ikone društvenih mreža i obrazac za pretraživanje.....	26
5.7.2. Opcija za dodavanje logotipa	28
5.7.3. Glavni izbornik.....	29
5.7. footer.php	31
5.8. page-pocetna.php.....	33
5.8.1. Postavljanje klizača	33
5.8.2. Petlja s objavama.....	36
5.8.3. Dodavanje bočnog stupca.....	39
5.9. page-o-zupi.php i page-zajednice.php.....	41
5.10. page-Materijali.php	43
5.11. page-kontakt.php	44
5.11.1. Kontakt obrazac.....	44
5.11. 2. Dodavanje Google karte	47
6. Zaključak.....	49
Literatura	50
Popis slika	54
Popis tablica	56

1. Uvod

Komunikacija putem Interneta postala je sastavni dio života. Danas više nego ikada, ljudi se obraćaju Internetu i web preglednicima kako bi u djeliću sekunde pronašli sve informacije koje su im u tom trenutku potrebne. Bilo da se radi o nekom proizvodu, usluzi ili organizaciji, web stranice su postale neophodne u prezentiranju sadržaja i informiranju korisnika. Postavljanje vlastite web stranice omogućuje oglašavanje i širenje informacija cijelome svijetu na brz i isplativ način kako kod poslovnih tako i kod neprofitnih organizacija. Napretkom tehnologije mijenjaju se i trendovi u stvaranju modernih web stranica koje danas zahtijevaju učestalu izmjenu dinamičnog sadržaja. U tu svrhu mnoga se poduzeća i udruge oslanjaju na popularne sustave za upravljanje sadržajem (*engl. CMS – Content Management Systems*). Radi se o specijalnim programima koji omogućuju krajnjim korisnicima, dakle osobama bez znanja programerskih vještina ili programskih jezika (*HTML, CSS, JS i PHP*), izradu i oblikovanje sadržaja na web stranicama. Postoje mnogi CMS sustavi koji nude različite oblike administracije – od onih s jednostavnim sučeljima koja omogućuju brzo dodavanje, brisanje i oblikovanje sadržaja do složenijih sustava kod kojih je ipak potrebno osnovno poznavanje koda.

WordPress je jedan od danas najpopularnijih CMS sustava upravo zbog svoje jednostavnosti i fleksibilnosti u instalaciji i korištenju. Radi se o sustavu za upravljanje sadržajem otvorenog koda pod *GPL licencom* (*engl. General Public Licence*), što znači da je izvorni kod sustava dostupan s licencom kojom nositelj autorskih prava dopušta svim korisnicima da proučavaju, mijenjaju i distribuiraju softver. [1] WordPress ima mogućnost nadogradnje sustava u obliku *dodataka* (*engl. plugins*) i *programčića* (*engl. widgets*) te tako omogućuje gotovo beskonačne prilagodbe u svrhu izrade optimalnog rješenja, odnosno nadopunjavanju izgleda i tehničkih pojedinosti stranice. Iako je u početku služio kao sustav za objavljivanje *blogova*, vrsta stranica kod kojih pojedinac ili poduzeće javno i redovito objavljuje različite sadržaje [2], danas WordPress isporučuje razne tipove web lokacija – od običnih statičnih web stranica i podstranica do korporativnih web lokacija i web trgovina. Činjenica je da danas više od četvrtine cijelog Weba objavljuje svoj sadržaj upravo preko WordPress-a. [1] Osnova za WordPress su programski jezik PHP i MySQL baza podataka te jednostavno sučelje kojim WordPress klijentu omogućava brzo i lako upravljanje sadržajem. Uz kvalitetan dizajn i širok spektar dodataka i programčića, WordPress očituje i izvrsna podrška od strane korisnika, klijenta i programera što omogućuje brz napredak programa kao CMS sustava.

U ovom završnom radu biti će opisani najvažniji dijelovi i izrada WordPress teme, odnosno predložaka na primjeru izrade funkcionalne web stranice za potrebe jedne župe i njene zajednice. Za početak će se opisati najvažnije mogućnosti i ključne značajke WordPressa kao CMS sustava – njegov razvoj kroz povijest, način generiranja web stranica i upravljanje sadržajem. Nakon toga slijedi poglavlje u kojem se detaljno opisuje instalacija WordPressa na lokalno računalo. Za tu potrebu koristiti će se XAMPP program koji sadrži odgovarajuću bazu podataka (MySQL) i web poslužitelja (Apache). Objasniti će se kako ispravno pripremiti bazu podataka te kako se uspješno prijaviti i kretati kroz sučelje WordPress sustava. WordPress svojom fleksibilnošću omogućuje izradu kvalitetnog web sjedišta i jednostavnog sustava za upravljanje sadržajem koji može ispuniti skup zahtjeva specifičan za udruge i neprofitne organizacije. Težnja je na konkretnom primjeru detaljno opisati postupak izrade jedne WordPress teme i korištenje raznih dodataka (engl.*plugins*) i programčića (engl.*widgets*). WordPress tema se sastoji od više datoteka predložaka, no u hrvatskom jeziku se ta dva pojma često koriste kao sinonimi, stoga se u nastavku rada umjesto pojma WordPress teme spominje pojam WordPress predložak. Zadatak je stvoriti jednostavno strukturiranu web stranicu koja će sadržavati osnovne informacije potrebne za uspješnu komunikaciju između župe i njenih župljana. U tu će se svrhu kroz poglavlja opisati sam proces dizajniranja i strukturiranja stranice, kodiranje pojedinih .php datoteka predložaka potrebnih za ispravno funkcioniranje WordPress stranice te dodavanje različitih php funkcija, dodataka i programčića kojima se teži unaprijediti web stranicu, odnosno sam CMS sustav stranice.

2. Wordpress kao CMS sustav

Sustavi za upravljanje sadržajem (*engl. CMS - Content Management Systems*) su softverske aplikacije koje se koriste za stvaranje, objavljivanje i uređivanje web sadržaja izravno kroz web sučelje. [3] U suštini, glavni zadatak CMS sustava je prikazati slikovne medije, audio i video datoteke, elektroničke dokumente i ostali web sadržaj u prezentacijskom sloju pomoću predložaka te pohraniti sav taj sadržaj unutar određene baze podataka. Pritom korisnik ne mora nužno poznavati programske jezike. Na taj način CMS sustavi omogućavaju pretvaranje „statičnih“ stranica u dinamičke kod kojih se redovito mijenja sadržaj. Upravo zbog svoje jednostavnosti u korištenju, fleksibilnosti samog dizajna te velike i aktivne zajednice koja pruža podršku, WordPress se ubrzo popeo i zauzeo prvo mjesto najpopularnijeg sustava za upravljanje sadržajem te ga danas koristi više od 40% svih mrežnih stranica na internetu. [4] Napisan je PHP skriptnim jezikom (*engl. PHP: Hypertext Preprocessor*) koji se najčešće koristi za generiranje dinamičkog sadržaja za web stranice. Radi se o poslužiteljskom jeziku, što znači da se izvršava prije nego što sadržaj stigne do preglednika, odnosno krajnjeg korisnika. [5] WordPress je također poduprt i MySQL relacijskom bazom podataka, jednom od najpopularnijih baza otvorenog izvornog koda. MySQL (*engl. SQL - Structured Query Language*) koristi se za spremanje podataka kod web aplikacija. Upravljanje podacima u MySQL bazi vrši se preko popularnog sučelja pod nazivom phpMyAdmin. WordPress je na samom početku bio razvijen kao softver za objavljivanje blogova [3], no danas pruža sve potrebne alate za upravljanje sadržajem složenijih stranica. To znači da je danas uz WordPress moguće jednostavno održavati vlastiti blog kao i web stranicu velike korporacije, udruge, crkve, škole, fakulteta itd.

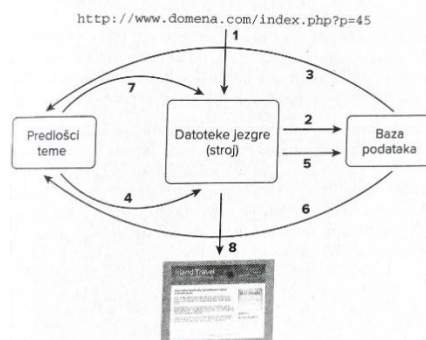
2.1. Razvoj WordPress-a kroz povijest

Michel Valdrighi je 2001. godine pokrenuo projekt *b2/cafelog*, jednostavnu platformu za objavljivanje blogova napisanu u PHP programskom jeziku. Na temelju napisanog koda ove platforme, Matthew Mullenweg i Mike Little su početkom 2003. godine najavili izradu novog, fleksibilnog, dobro arhitekturi ranog sustava pod nazivom WordPress. [6] Prva stabilna verzija (Verzija 1.0.1.) objavljena je 3. siječnja 2004. godine. Od verzije 1.5 (takozvana „*Strayhorn*“ verzija), [7] WordPress podupire upravljanje statičnim stranicama čime je stvorena osnova za izradu jednostavnog sustava za upravljanje sadržajem. 2005. godine osnovali su tvrtku Automattic čiji je cilj bio bolje koordiniranje razvoja WordPressa te pokretanje nove usluge *hostinga* blogova pod nazivom *WordPress.com*, bazirana na višekorisničkoj verziji

WordPressa. Neprestanim razvijanjem i poboljšanjem sustava, WordPress CMS sustav se krajem 2015. godine koristio na četvrtini svih web stranica u svijetu [8]. Prema podacima W3techsa, 2021. godine 64% svih web stranica izrađenih pomoću CMS sustava pripada WordPress-u. Analizirajući tu statistiku i statistiku svih mrežnih stranica [9], dolazi se do zaključka da su približno 40% svih web stranica na internetu nastale upravo pomoću WordPress-a i njegovih alata. Najnovija verzija WordPressa, (verzija 5.8. „Tatum“), nazvana po poznatom jazz pijanistu, izašla je u srpnju 2021. Uz podršku *WebP* formata slike (znatno manji format koji služi kao zamjena za JPEG i PNG formate), objavljen je i *uređivač predložaka* (engl. *template editor*) koji omogućuje kreiranje i uređivanje predložaka pomoću takozvanih *blokova* (engl. *blocks*), što je velik korak u smjeru potpune samodostatnosti WordPressa kao programa za razvoj web stranica. [1] Korisnici sada mogu jednostavno dodavati logotip, naslov, sadržaj, kategorije i oznake na svoje predloške. Novost je i *Duotone Color* efekt koji omogućuje primjenjivanje filtera na slike te blok za pregled PDF-a kojim se jednostavno može ubaciti PDF datoteka bilo gdje na stranicu.

2.2. Generiranje web stranice

Web stranice se u WordPressu generiraju pomoću jezgre, teme i baze podataka te se te tri komponente mogu nezavisno podešavati i ažurirati. Kad se govori o jezgri misli se o skupu mapa i datoteka koje su preuzete sa službene stranice WordPressa (www.wordpress.org) i koje služe za preuzimanje i sastavljanje sadržaja spremljenog unutar baze podataka. Temu čine predlošci koje sadrže informacije potrebne jezgri da sastavi sadržaj koji treba u određenom trenutku preuzeti i prikazati. [5] Sam proces sastavljanja web stranice traje relativno kratko, svega djelić sekunde u kojoj upisana URL adresa vodi korisnika do WordPress jezgre, a ona zatim u svojoj bazi podataka pronalazi odgovarajuće predloške i dijelove sadržaja koje treba uzeti, sastaviti i prikazati krajnjem korisniku u HTML pregledniku (Slika 1.).



Slika 1. Shematski prikaz sastavljanja stranice

Izvor: G.Plumley: Wordpress, Dobar plan,

Zagreb 2012.

Prednost odvajanja dizajna i strukture od samog sadržaja (tekst i grafika) je u tome da se na taj način uz samo jedan predložak mogu generirati na tisuće stranica. Četvrti element u strukturi WordPressa su takozvani *dodaci* (engl. *plugins*), dijelovi koda koji svojom instalacijom omogućuju dodatne funkcionalnosti za specifične potrebe (primjerice ocjenjivanje sadržaja, određen način prikazivanja fotografija, prikazivanje obavijesti s drugih društvenih mreža itd.). Razlog zašto je i sama jezgra odvojena od forme i sadržaja je u tome da se stvori jednostavan i fleksibilan sustav u kojem je lakše odvojeno ažurirati sadržaj od oblikovanja. Čim je jezgra složenija, postoji veća opasnost da će se srušiti sustav te je stoga važno ne opterećivati ju raznim dodacima koji nisu od prevelike važnosti za ispravan rad web stranice. Ovakva struktura WordPress aplikacije u mnogo čemu podsjeća na *MVC arhitekturu*. MVC obrazac dijeli aplikaciju na tri međusobno povezana dijela – *Model – Pogled* (engl. *View*) – *Kontroler* (engl. *Controller*) te omogućuje razdvajanje programiranja obrade podataka od programiranja sučelja i korisničkih zahtjeva. Aplikacija građena prema takvoj arhitekturi omogućava korisniku da preko *Pogleda* unese zahtjev koji se šalje *Kontroleru*, a zatim *Kontroler* od *Modela* traži da obavi radnju i pošalje rezultat. Rezultat se na posljetku šalje u *Pogled* gdje ga korisnik može vidjeti. [10] Iako slični, WordPress softver ipak nije građen prema MVC arhitekturi, no unutar njega se mogu graditi teme i predlošci koji su orijentirani prema takvoj vrsti obrasca. [11]

2.3. Upravljanje sadržajem

Važan dio svake web stranice jest njezin sadržaj koji mora posjetiteljima pružati sve informacije koji oni u zadanom trenutku traže. Osnovna podjela sadržaja na web stranici bila bi na tekst, slike, zvuk, video, animacije, programski kod itd. Upravljanje sadržajem je prema tome proces unosa takve vrste sadržaja u bazu podataka, omogućavanje njegovog uređenja i čuvanje uputa o načinu na koji se taj sadržaj odnosi prema drugom sadržaju. [5] WordPressov sadržaj se dijeli na dvije vrste – objave (engl. *Posts*) i stranice (engl. *Pages*). Radi se o blokovima sadržaja i pridruženih uputa spremljenih u bazi podataka. Važno za napomenuti je da se pojam „*post*“ ne odnosi samo na blok teksta već obuhvaća i ostali željeni sadržaj (slike, video, događaji, proizvodi itd.). Objave su unošene obrnutim kronološkim redoslijedom te se mogu shvatiti kao članci ili ažuriranja koja administrator stranice dijeli svojim čitateljima. [12] Objave se mogu organizirati pomoću kategorija. Redoslijed prikazivanja objava u kategorijama može se odrediti prema više različitih parametara, kao na primjer datum objavljivanja, naslov, autor itd. [5] Mogu se hijerarhijski poredati na kategorije roditelja i potomaka, odnosno jedna kategorija može biti nadređena drugoj kategoriji. [1] Također, jedna objava može sadržavati više

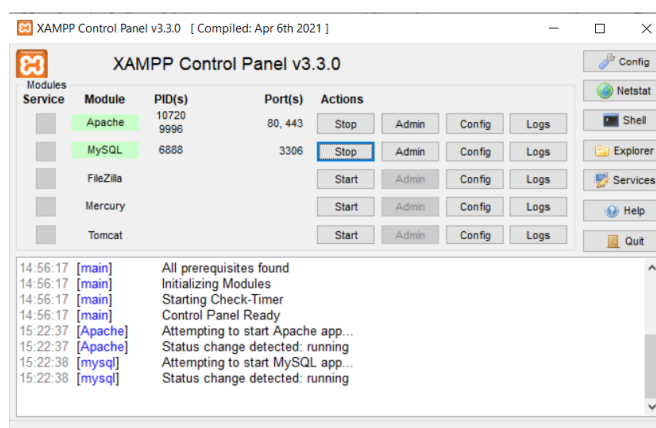
kategorija istovremeno. To omogućuje veliku fleksibilnost u prikazu objava unutar WordPress predloška. Osim u kategorijama, objave se mogu grupirati i prema *oznakama* (engl. *tags*). Kategorije određuju općenitu temu sadržaja objave i tako označavaju skup objava koje su na neki način povezane jedna s drugom, dok oznake služe za detaljniji opis tema obrađenih u određenim objavama. [13] Uobičajeno je da se svakoj objavi dodaje između tri i deset oznaka. Dodavanje i uređivanje kategorija i oznaka detaljnije je opisano u poglavlju 3.6. *Izrada objave*. U suštini, kategorije i oznake pomažu posjetiteljima u pronalasku onog sadržaja koji traže, odnosno služe za bolju SEO optimizaciju WordPress stranice. Stranice sadrže „*statičan*“ sadržaj koji se ne mijenja često – npr. stranica „*O nama*“ ili „*Kontakt*“. Suprotno objavama koje su organizirane prema raznim kategorijama, stranice se ne mogu kategorizirati niti im se mogu pridružiti bilo kakve oznake. One se samo mogu organizirati u hijerarhiju roditeljskih stranica i stranica potomaka, odnosno omogućeno im je postavljanje redoslijeda prema važnosti. [2] Takozvani *slug* je još jedan način organiziranja i označavanja web sadržaja. Radi se o nizu riječi ispisanih malim slovima, brojkama i povlakama bez razmaka koji su sastavni dio URL-a (*permalinka*, dolazi nakon domene) te opisuju sadržaj objave, stranice, kategorije ili oznake.[5]

3. Instalacija WordPressa na lokalnom računalu

Za potrebe izrade vlastite web stranice, odnosno WordPress teme, najbolja polazišna točka je instaliranje programa na osobnom računalu kako bi se lakše mogle provoditi izmjene prilikom kodiranja datoteka predložaka i dodataka. Na taj način sve izmjene ostaju vidljive samo autoru, tako dugo dok se ne odluči postaviti stranicu na javni web poslužitelj. U ovom poglavlju završnog rada detaljno je opisan proces instaliranja WordPress-a na vlastitom računalu.

3.1. Instaliranje XAMPP-a

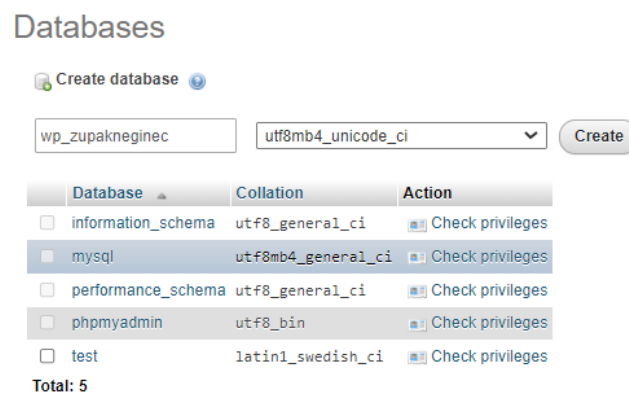
Za izradu web stranice unutar WordPressa potrebno je postaviti odgovarajuću bazu podataka i lokalni poslužitelj. Za to se najčešće koristi besplatan web poslužitelj otvorenog koda pod nazivom XAMPP (*engl. Cross platform, Apache, MariaDB, PHP, Perl*) koji se sastoji od Apache web poslužitelja, MariaDB baze podataka i tumači skripte napisane u programskim jezicima PHP i Perl. Razvijen od tvrtke Apache Friends 2002. godine, XAMPP je danas dostupan na različitim platformama (Windows, Linux, MacOS) i uglavnom se koristi za testiranje web mjesta, pregledavanje istih na World Wide Webu te pomaže u stvaranju i manipuliranju bazama podataka u SQL-u. [14] MySQL je 2015. godine zamijenjen MariaDB bazom podataka. Ovaj sustav, temeljen na strukturi i sintaksi svog prethodnika, MySQL-a, sadrži sve njegove elemente (podržava relacijski model podataka, koristi tablice, ograničenja, okidače, uloge, pohranjene procedure, poglede i ostale karakteristike sustava za upravljanje bazama podataka). Osnovna razlika između ta dva sustava je u performansama – MariaDB ima veću memoriju, bržu optimizaciju i izvođenje upita te mogućnost paralelnog izvršavanja više upita. [15] XAMPP se može skinuti sa službene stranice www.apachefriends.org, te se jednostavno instalira na računalo. Ovdje je važno za napomenuti da WordPress jedino funkcionira s novijim verzijama XAMPP-a (od verzije 7.0, pa nadalje). Za ispravan rad web mjesta potrebno je, unutar XAMPP-a, uključiti Apache web server i MySQL bazu podataka. [16] (Slika 2).



Slika 2. XAMPP Kontrolna ploča, uključeni Apache i MySQL

3.2. Pripremanje baze podataka

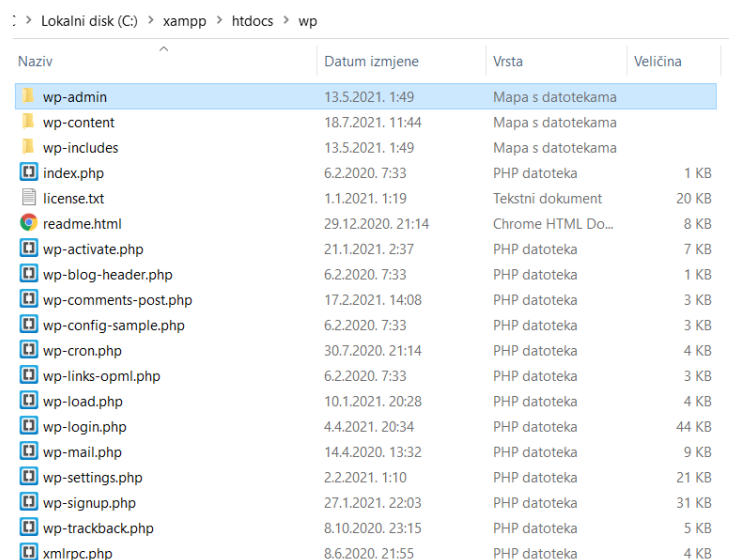
Samim instaliranjem web servera XAMPP-a, automatski je dobiven i phpMyAdmin (besplatni alat za administraciju s otvorenim kodom za MySQL i MariaDB). Za pristup je potrebno u web pregledniku upisati *localhost/phpmyadmin*. U izborniku se odabere opcija *Databases* koja otvara novi prozor unutar kojeg se može napraviti nova baza podataka. Unosi se naziv baze podataka (u ovom primjeru to je *wp_zupakneginec* te se za WordPress odabire kolacija pod nazivom *utf8mb4_unicode_ci*. (Slika 3).



Slika 3. Kreiranje nove baze podataka u phpMyAdmin alatu

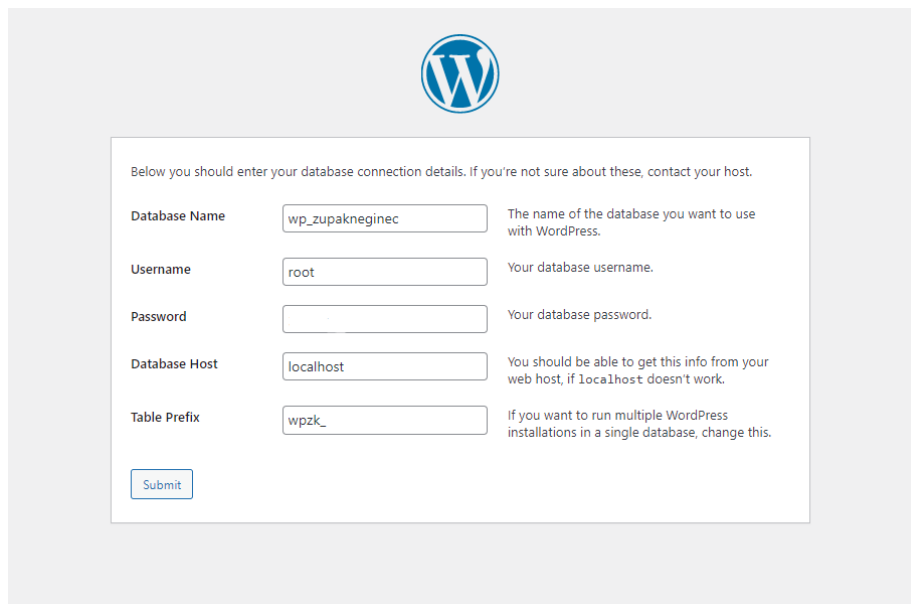
3.3. Konfiguracija WordPressa

WordPress datoteke se preuzimaju s adrese www.wordpress.org. Pritiskom na preuzimanje, dobiva se .zip arhiva s najnovijom verzijom sustava WordPress. Kako bi WordPress uspješno radio, potrebno je otvoriti mapu *htdocs* slijedeći putanju *C disk/xampp/htdocs*. Unutar nje se napravi nova mapa pod nazivom *wp* unutar koje će se raspakirati prethodno preuzeta .zip arhiva WordPress-a. (Slika 4).



Slika 4. Sve raspakirane datoteke sadržane u wp mapi

Nadalje, potrebno je unutar web preglednika u tražilicu upisati *localhost/wp*, dakle naziv prethodno napravljene mape unutar koje su sada spremljeni svi materijali za rad WordPress sustava. Pojavljuje se uvodna poruka u kojoj je potrebno odabrati jezik pod kojim se želi izvesti instalacija. Nakon toga otvara se novi zaslon u kojem se ukratko objašnjava koji su detalji potrebni za uspješnu instalaciju (Slika 5). Potrebno je znati ime same baze podataka, korisničko ime i lozinku za bazu podataka, pri čemu se savjetuje da lozinka bude složenija radi veće sigurnosti te računalo s poslužiteljem baze podataka (adresa računala na kojem se nalazi poslužitelj baze podataka za lokaciju). Postoji dodatno polje pod nazivom *Table Prefix* koje određuje podrazumijevani prefiks koji će imati svaka tablica unutar baze podataka. Preporuča se izmjena prefiksa zbog dodatne mjere sigurnosti. [1]



Slika 5. Potrebni podaci za ispravnu WordPress instalaciju

Klikom na *Submit*, otvara se novi zaslon za finalnu potvrdu gdje se treba pritisnuti gumb *Run the Install*. Sada je potrebno popuniti instalacijski obrazac u kojem se postavljaju osnovni detalji nove Web lokacije kao što su ime stranice, korisničko ime i lozinka (WordPress sam nudi opcije koje su dosta sigurne), E-mail adresa i opcija kojom se uključuje ili isključuje lokacija iz rezultata pretraživanja Weba (preporuča se da se ipak isključi kako bi lokaciji mogli pristupiti Google i druge tražilice). Važno za napomenuti je da se sve te opcije mogu kasnije po potrebi promijeniti. (Slika 6). Nakon uspješne instalacije otvara se prijava za ulazak u administrativno sučelje WordPressa.

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password
Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email
Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

Slika 6. Instalacijski obrazac WordPress-a

3.4. Prijavljivanje u WordPress

Na kraju instalacije pojavljuje se zaslon koji nudi prijavu u WordPress s administratorskim korisničkim imenom i lozinkom (Slika 7). Potrebno je unijeti prethodno odabrane podatke. Ukoliko bi se zaboravila lozinka, stranica za prijavljivanje sadrži i vezu za promjenu lozinke, ali je pritom važno da je adresa elektroničke pošte ispravna.

You are now logged out.

Username or Email Address

Password

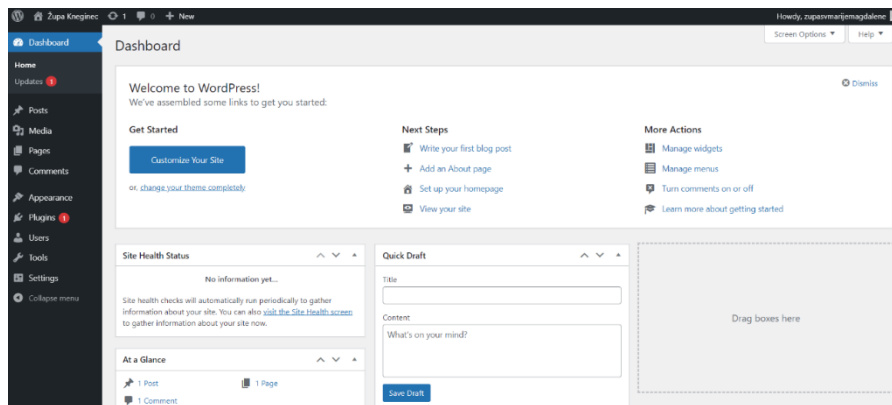
Remember Me

[Lost your password?](#)

Slika 7. Prijava u WordPress

3.5. Osnovno kretanje kroz administratorsko sučelje

Prijavom na WordPress otvara se nadzorna ploča preko koje se upravlja cjelokupnim sadržajem web mjesta. Na samom vrhu ploče nalazi se administratorska vrpca koja između ostalog sadrži vezu do početne stranice WordPress lokacije u obliku ikone s kućom kao i ikona sa znakom plus koja otvara padajući izbornik s vezama za novi post, stranicu, korisnika itd. [1] Najvažnije mjesto na ploči je bočni izbornik koji se nalazi na lijevoj strani (Slika 8). Glavne stavke izbornika su: Postovi, Medij, Stranice, Komentari, Izgled, Dodaci, Korisnici, Alati i Postavke. Stavke i njihove mogućnosti su ukratko opisane u dolje navedenoj tablici (Tablica 1). Lebdjenjem iznad svake od njih prikazuju se podizbornici sa srodnim opcijama. [17] Za potrebe ovog završnog rada, veća pažnja će se posvetiti na stavku Izgled (*engl. Appearance*), s obzirom na to da se unutar nje nalazi opcija za izmjenu i oblikovanje WordPress tema.



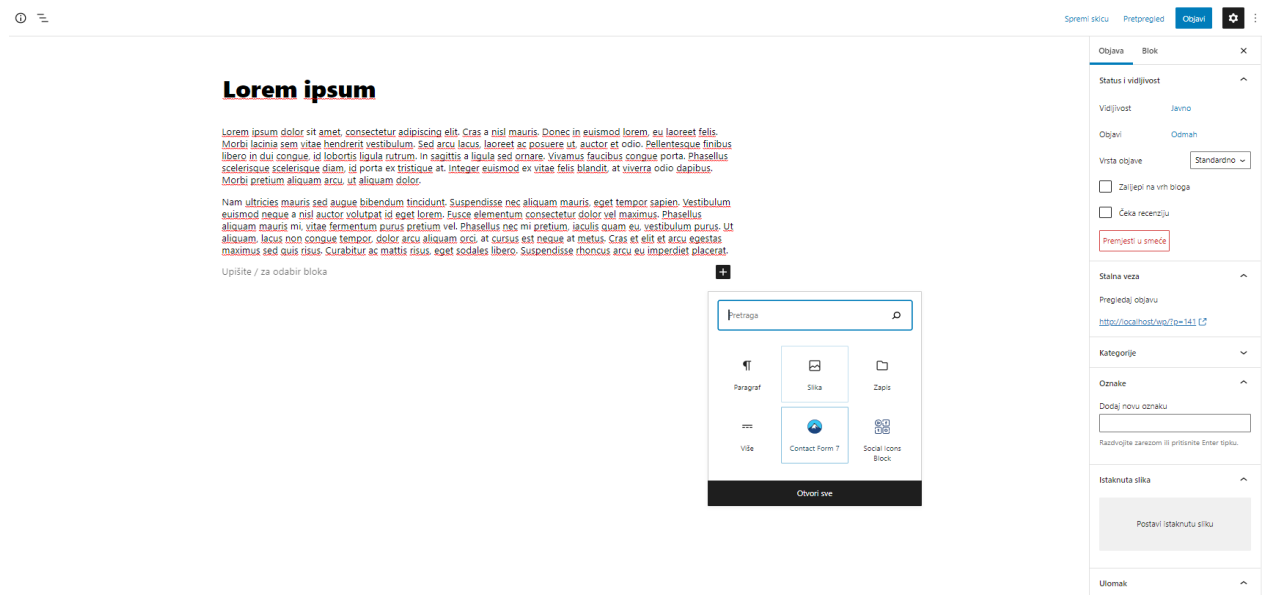
Slika 8. Početna stranica administratorskog sučelja WordPress programa

Postovi (Posts)	Stavka koja služi za izradu, objavu i izmjenu članaka (posts), kategorija (categories) i oznaka članaka (tags).
Medij (Media)	Koristi se za upravljanje medijskim datotekama (slike, dokumenti, glazbene datoteke itd.) koje se dodaju u članke ili na samu stranicu.
Stranice (Pages)	Radi se o upravljanju „statičnim“ stranicama koje se koriste za sadržaje koji se rijetko mijenjaju (stranice kontakta, „O nama“ i sl.).
Komentari (Comments)	Stavka koja služi za upravljanje komentarima ukoliko su oni omogućeni.
Izgled (Appearance)	Služi za upravljanje izgledom same stranice – dakle ovdje se biraju i mijenjaju teme (themes), dodaju dodaci (widgets) i uređuju sami izbornici (menus).
Dodaci (Plugins)	Radi se o aktiviranju i deaktiviranju dodataka koji proširuju funkcionalnost WordPressa.
Korisnici (Users)	Stavka koja služi za dodavanje i uklanjanje korisnika kao i upravljanje pojedinim korisničkim ulogama.
Alati (Tools)	Omogućava pristup raznim alatima, između ostalog služi i za dodavanje vlastitog sadržaja na WordPress web lokaciju.
Postavke (Settings)	Stavka koja sadrži većinu konfiguracijskih alata za web lokaciju. Pomoću nje se mijenjaju razni parametri kao što su naziv web lokacije, dostupni fontovi, vidljivost web lokacije itd.

Tablica 1. Opis pojedinih stavaka WordPress-a i njihovih mogućnosti

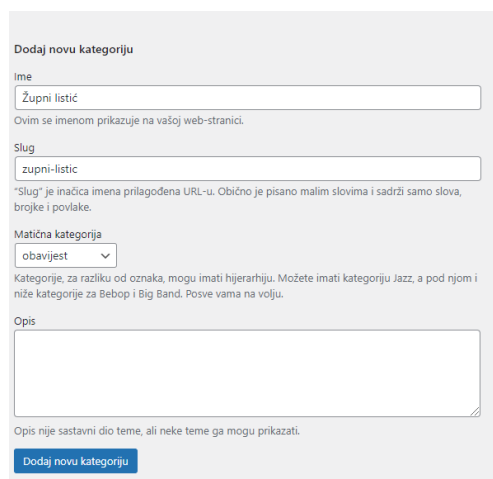
3.6. Izrada objave i stranice

Za stvaranje nove objave potrebno je u administratorskom sučelju kliknuti stavku *Objave>Dodaj novu (Posts>Add New)*. Otvara se novi prozor u kojem se nalaze polja za unos naslova i sadržaj (Slika 9). Najnovija verzija WordPressa omogućuje dodavanje raznog sadržaja pomoću blokova (paragrafi, slike, dokumenti, tablice itd.). S desne strane se nalaze opcije dodavanja kategorija i oznaka.



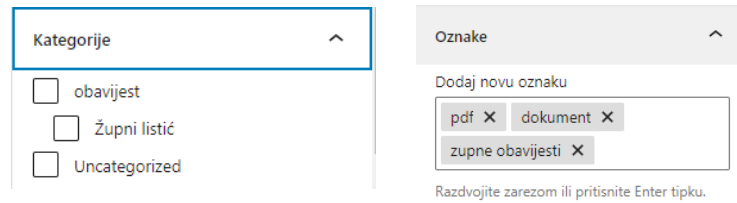
Slika 9. Izrada nove objave

Da bi neka objava bila kategorizirana, najprije je potrebno stvoriti pojedine kategorije. One se dodaju pod stavkom *Objave>Kategorije (Posts>Categories)*. Opcije omogućuju odabir naziva kategorije i takozvani *slug* kategorije (Slika 10)., u desnoj traci s postavkama pod stavkom kategorije, može se vrlo lako označiti kojim sve kategorijama nova objava pripada (Slika 11).



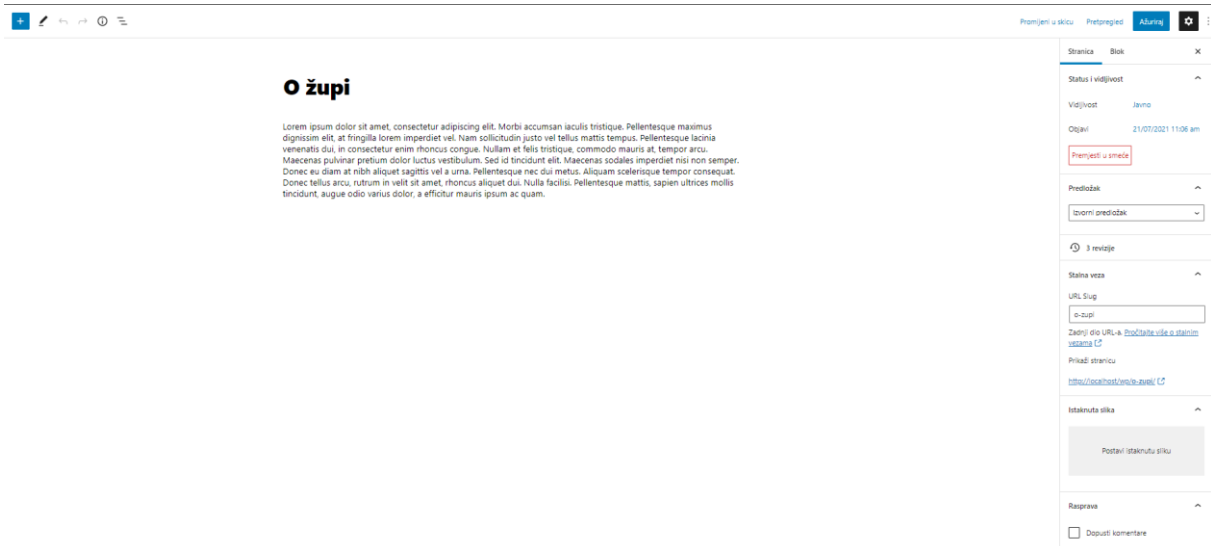
Slika 10. Izrada nove kategorije

Oznake, odnosno pojmovi se mogu jednostavno upisati pod stavkom *Oznake* prilikom pisanja nove objave (Slika 11). Pritiskom na gumb Objavi (Publish) sadržaj objave postaje vidljiv posjetiteljima.



Slika 11. Pridruživanje kategorije i dodavanje oznaka novoj objavi

Nova se stranica može dodati pomoću stavke *Stranice > Dodaj novu (Pages > Add New)*. Tada se otvara novi prozor u kojoj administrator može unijeti naslov i sadržaj stranice, slično kao kod pisanja objave (Slika 12). S desne strane se ovoga puta nalazi opcija koja definira koji predložak će se koristiti za određenu stranicu – on može biti izvorni (*default*, najčešće *index.php*) ili vlastiti (*custom*). Ukoliko se stvori vlastita datoteka predloška za određenu stranicu te se unutar nje definira njeno ime (u obliku komentara – npr. */*Template Name: O župi*/*), onda će se pod opcijom Predložak moći odabrati upravo taj predložak.



Slika 12. Dodavanje nove stranice u WordPressu

4. Struktura i oblikovanje WordPress predložka

WordPress teme služe za definiranje izgleda pojedine web stranice. Teme su sastavljene od većeg broja datoteka predložaka koji uključuju programski kod i grafiku (razne web elemente – npr. slike, boje, tipografiju itd.) kao i njihov raspored na samoj stranici (Slika 13). WordPress, između ostalog, sadrži repozitorij pun besplatnih i plaćenih gotovih tema i predložaka razvijenih od strane kompanija ili individualnih web developera. Njih je vrlo jednostavno aktivirati i upotrijebiti za vlastitu web lokaciju što omogućava veliku fleksibilnost kod promjene izgleda stranice.[18] Svaka dobro osmišljena WordPress tema poštuje razliku između prezentacije, funkcije i samog sadržaja. U suštini, svaki se predložak sastoji od datoteka PHP koda koji za strukturu i grafičko oblikovanje teme, odnosno pojedinog predložka, generira HTML i CSS kod te pomoću PHP jezika definira funkcije koje određuju koji će se sadržaj prikazati posjetitelju u određenom trenutku. [2] S obzirom na to da se u posljednje vrijeme WordPress sve češće koristi kao cjeloviti sustav za upravljanje sadržajem, same teme i njihni predlošci postali su puno složeniji i u strukturi [19].



Slika 13. Raspored pojedinih datoteka predložaka na jednoj WordPress stranici

Izvor: M. MacDonald: *WordPress – the Missing Manual*, O'Reilly Media, Inc., Sebastopol, 2014.

4.1. Vrste datoteka predložaka

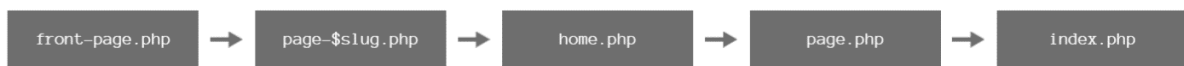
Svaka WordPress tema koristi nekolicinu datoteka predložaka. To su datoteke koje sadrže kod pisan HTML i PHP jezikom te se svaka datoteka koristi za određenu radnju na web stranici. Kvantiteta datoteka predložaka ovisi o kompleksnosti teme. Datoteka mora sadržavati točno određen naziv i završavati sa .php nastavkom kako bi ju WordPress uspješno prepoznao i pokrenuo. [20] Njihov izgled je opisan u .css datotekama (style.css i template.css). U nastavku su navedene osnovne datoteke predložaka koje svaka WordPress tema mora sadržavati da bi uspješno funkcionirala.

- ***index.php*** je WordPress datoteka koja sadrži osnovni predložak. Bez ove osnovne datoteke ne postoji WordPress tema. Koristi se uglavnom za prikaz postova na početnoj stranici i za svaki prikaz predloška koji nije definiran. Ukoliko za određenu stranicu ne postoji definirani predložak, WordPress će elemente, odnosno postove, na toj stranici prikazati u obliku koji je zadan datotekom *index.php*.
- ***header.php*** sadrži kod za elemente koji se pojavljuju na vrhu svake stranice. Elementi koji se često pojavljuju u *header* datoteci su: logotipovi, naslovi stranice, ikone društvenih mreža i navigacijska traka.
- ***footer.php*** je datoteka koja se također pojavljuje na više stranica i sadrži elemente svojstvene podnožju – glavne kontakt informacije, copyright oznaka, godina izrade, navigacija itd.
- ***functions.php*** datoteka služi za definiranje funkcija odnosno programskog koda koji će uvijek biti na raspolaganju predlošku ili web stranici [13] te uključuje registriranje raznih stilova, skripti, navigacijskih izbornika, widgeta i sl. Pomoću funkcija se mijenjaju mogućnosti WordPress-a.
- ***sidebar.php*** datoteka sadrži pomoćne izbornike i widgete.
- ***single.php*** definira izgled i prikazuje jednu pojedinačnu objavu iz baze podataka.
- ***page.php*** je predložak koji definira izgled individualnih statičnih stranica.
- ***archive.php*** se koristi za izradu stranice arhive web sjedišta

4.2. Hijerarhija predložaka

Princip rada svake WordPress teme krije se u hijerarhijskom rasporedu predložaka. Pojam hijerarhije predložaka odnosi se na strukturu i redoslijed datoteka predložaka koju WordPress koristi kako bi odredio koje će se sve datoteke predložaka u temi koristiti za generiranje gotovog

HTML koda određene web stranice. Hijerarhija predložaka je uvedena kako bi olakšala komunikaciju WordPressa s predlošcima, odnosno kako bi se olakšala organizacija predložaka. [21] Iako je za funkcionalnu WordPress temu dovoljno imati samo dvije datoteke (*index.php* i *style.css* koja opisuje stilove), ovisno o kompleksnosti web stranice, većina složenijih i dinamičnijih WordPress stranica sadržava velik broj datoteka predložaka od kojih je svaka određena za neku sekciju unutar web stranice. U takvim slučajevima, u procesu traženja uputa za sastavljanje pojedine stranice, WordPress prolazi kroz popis svih datoteka predložaka koje čine tu stranicu i, po točno određenom redosljedju, sastavlja gotovu HTML stranicu. Kada posjetitelj zatraži određenu stranicu na web lokaciji, primjerice početnu stranicu, WordPress će prvo pokušati pronaći datoteku predloška čija je namjena upravo sadržavati sve elemente koji se nalaze na početnoj stranici. Važno je pravilno imenovati datoteke predloška u mapi teme jer će na taj način WordPress automatski prepoznati i početi koristiti određenu datoteku predloška nad svim drugim predlošcima (u ovom primjeru bila bi to datoteka naziva *front-page.php* koja se prema WordPress kodeksu nalazi na samom vrhu hijerarhije predložaka za početnu stranicu). Ako WordPress ne može pronaći datoteku tog naziva, prelazi na onu sljedeću po hijerarhiji, tražeći predložak koji sadrži slug stranice (u ovom slučaju *page-pocetna.php*, slug je 'pocetna'). Ukoliko ni takva datoteka ne postoji, prelazi na hijerarhijski niže poredanu (*home.php*). Ovaj se princip primjenjuje tako dugo dok se ne nađe datoteka koja odgovara upitu ili se dođe do posljednje datoteke u hijerarhiji - *index.php* (osnovna Wordpress datoteka) (Slika 14).

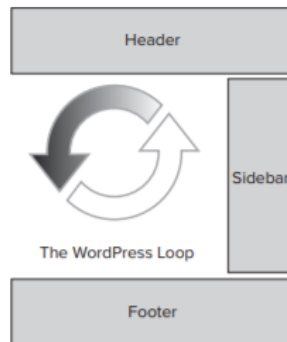


Slika 14. Ilustrirani prikaz hijerarhije predložaka početne stranice

4.3. Funkcija petlje (The Loop)

Petlja (*engl. The Loop*) osnovni je dio svakog WordPress predloška bez koje ne bi postojao dinamični sadržaj. Ona, naime, određuje koji će se sadržaj (jedan post, stranica, sekvenca postova itd.) prikazati na stranici na kojoj se nalazi krajnji korisnik (Slika 15). [19] Radi se o PHP petlji koja može na temelju parametara dohvatiti i prikazati odabranu grupu objava i stranica na način da provodi petlju kroz sadržaj unutar baze podataka. Drugim riječima, petlja predstavlja funkcionalnu vezu između sadržaja koji se nalazi u bazi podataka i izvedenog HTML koda koji prikazuje taj sadržaj krajnjem korisniku. Petlja se modificira korištenjem raznih parametara kako bi dohvatila specifičan sadržaj za određeni dio stranice koji se

pretražuje. Parametri se uglavnom određuju URL-om koji se koristi za pristup WordPress stranici, no mogu biti i proizvoljni. WordPress preslikava sve parametre određenog posta u takozvanu varijablu odabira (engl. *selection variable*) te njome osigurava velik broj različitih načina kojim se mogu promijeniti algoritmi petlje.



Slika 15. Smještaj petlje u osnovnom WordPress predlošku

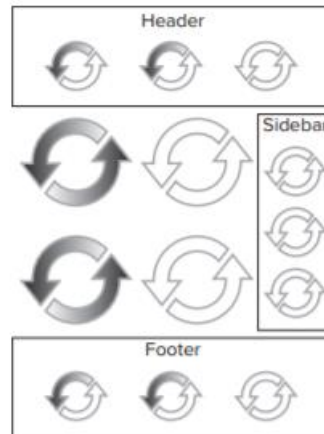
Izvor: B. Williams, D. Damstra, H. Stern: *Professional WordPress, Design and Development*, John Wiley & Sons, Inc., Indianapolis, 2013.

Najosnovnija petlja koja se koristi u WordPress-u je inačica „while“ naredbe te se provodi na sljedeći način. Petlja šalje upit koji provjerava da li postoje određeni postovi, odnosno podaci, unutar baze - izraz *if(have_posts())*: . Ukoliko baza podataka sadrži postove, pokreće se „while“ petlja. Slijedi izraz *while (have_posts()) : the_post()*; kojim se dohvaćaju podaci i ispisuju se unutar predloška tako dugo dok se oni nalaze u bazi podataka (Slika 16). Nakon što su se dohvatili podaci pomoću funkcije *the_post()*, takozvanim *oznakama predloška* (engl. *template tags*) omogućeno je prikazivanje određenih elemenata objave kao što su na primjer *naslov posta* (oznaka *the_title*), *sadržaj posta* (oznaka *the_content*), *kategorija* (oznaka *the_category*), *izvadak članka* (oznaka *the_excerpt*) itd. Postoji više od stotinu ugrađenih oznaka predloška koji se mogu koristiti unutar petlje. ., čime se omogućava potpuna prilagodba WordPress teme za vlastite potrebe. Za zatvaranje petlje i if izraza koriste se izrazi *endwhile* i *endif*. [22]

```
<?php
    if(have_posts() ):
        while(have_posts() ):the_post();
        ...
        endwhile; else:
        ...
        endif:
?>
```

Slika 16. Osnovna petlja za dohvaćanje postova

Petlja se najčešće nalazi između zaglavlja, bočnog stupca i podnožja te tako stvara prostor za dinamičan sadržaj koji će se mijenjati ovisno o dodavanju i modificiranju postova, no nema ograničenja u broju petlji koji se može dodavati unutar predloška (Slika 17). Takve prilagođene petlje mogu se postaviti različitim metodama bilo gdje unutar strukture predloška te se uglavnom koriste za WordPress dodatke (engl. *plugin*) i widgete (engl. *widgets*). [19]



Slika 16. Ilustracija velikog broja petlji unutar WordPress predloška

Izvor: B. Williams, D. Damstra, H. Stern: *Professional WordPress, Design and Development*, John Wiley & Sons, Inc., Indianapolis, 2013.

4.4. WordPress dodaci i programčići

WordPress dodaci (engl. *plugins*) su dijelovi programa koji se naknadno dodaju kako bi se proširila funkcionalnost WordPress stranice. Riječ je o programskim kodovima koje pišu programeri za neke svoje specifične aplikacije ili slučajeve te ih naknadno objavljuju za širu upotrebu. [23] Danas postoji preko 35 000 visokokvalitetnih besplatnih i komercijalnih dodataka koji se lako izabiru, preuzimaju i instaliraju prema potrebama pojedinog web sjedišta. Dodaci mogu obavljati različite poslove, no najčešće se koriste za [24]: optimiziranje sadržaja za tražilice (engl. *SEO - Search Engine Optimization*), prikupljanje podataka od korisnika putem obrasca, prikazivanje statistike posjeta, prikazivanje sadržaja vidljivih posjetiteljima na nov način (npr. popis najčitanijih objava) itd. Korištenje gotovih dodataka štedi na vremenu i trudu koji bi bili potrebni da se dodatak ručno programira u predložak te poboljšava samu kvalitetu i funkcionalnost web stranice. No, prilikom traženja odgovarajućeg dodatka za određeno web rješenje, treba pripaziti na to da je kompatibilan s verzijom WordPress-a koja se koristi (provjeriti datum ažuriranja) i da pruža dovoljnu dobru sigurnost. Važno je i za napomenuti da se u [18] preporuča umjereno korištenje dodataka, jer oni mogu, ukoliko ih je

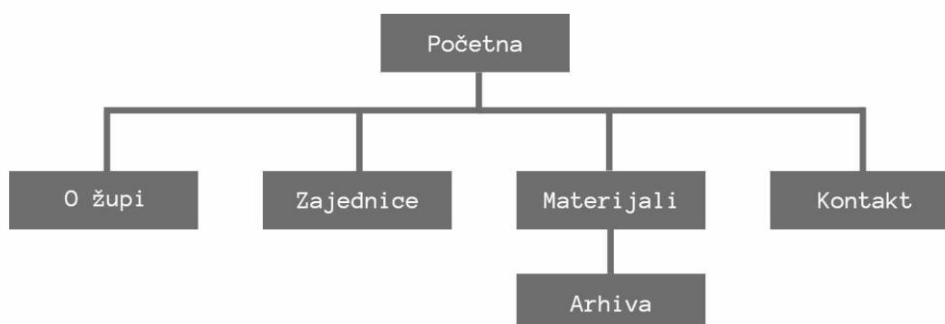
mного ili su neki od njih loše kodirani, znatno usporiti rad web stranice. Programčići (engl. *widgets*) su WordPress dodaci koji prikazuju određene sadržaje te ih korisnici mogu lako dodati, obrisati i premjestiti. Programčići obično sadrže mjesečnu arhivu objava, okvir s metapodacima (veze za prijavljivanje i odjavljivanje), popis kategorija, blokove teksta, polja za pretraživanja itd. [1] Iako programčići mogu obavljati vrlo složene funkcije, na administratorskoj stranici poprimaju oblik okvira kojeg je vrlo jednostavno prilagoditi svojim potrebama. [5] Tradicionalno su smješteni su u bočnom stupcu teme, no programer može lako odrediti i bilo koje drugo područje (zaglavlje, podnožje, središnji dio itd.). U početnu instalaciju WordPress-a je uključeno nekoliko osnovnih programčića dok je za ostale potrebna instalacija. Glavna razlika između dodataka i programčića je u njihovoj osnovnoj funkcionalnosti. Većina dodataka izvodi radnje u pozadini WordPress-a, dakle skriveno od posjetitelja, dok programčići služe za prikaz sadržaja na gotovoj web stranici krajnjih korisnika. Važno za napomenuti je da je neke programčiće nemoguće instalirati bez instalacije odgovarajućih dodataka. [25]

5. Razvoj vlastite teme

U ovom poglavlju se obrađuju osnovni postupci i elementi potrebni za izradu funkcionalne teme za WordPress. Podjelom gotove web stranice na manje cjeline – pojedini predlošci kao i dijelove koji čine jednu stranicu (zaglavlje, glavni i bočni stupac, podnožje) u nastavku je objašnjen ručno pisan HTML i CSS jezik kao i sav potreban PHP kod. Za izradu teme također su korišteni i opisani svi korišteni WordPress programski dodaci i programčići. Za početak je potrebno izraditi mapu teme i prilagoditi ju tako da je WordPress može prepoznati kao temu. [1] Unutar WordPress mape (putanja *C disk/xampp/htdocs/wp*) postoji mapa *wp-content* u kojoj se nalazi podmapa *themes*. Tu su sadržane sve gotove WordPress teme i predlošci. Imenuje se nova mapa pod nazivom nove teme, u ovom slučaju *zupakneginec*. U njoj će se spremati svi predlošci, odnosno .php datoteke potrebne za ispravno funkcioniranje teme.

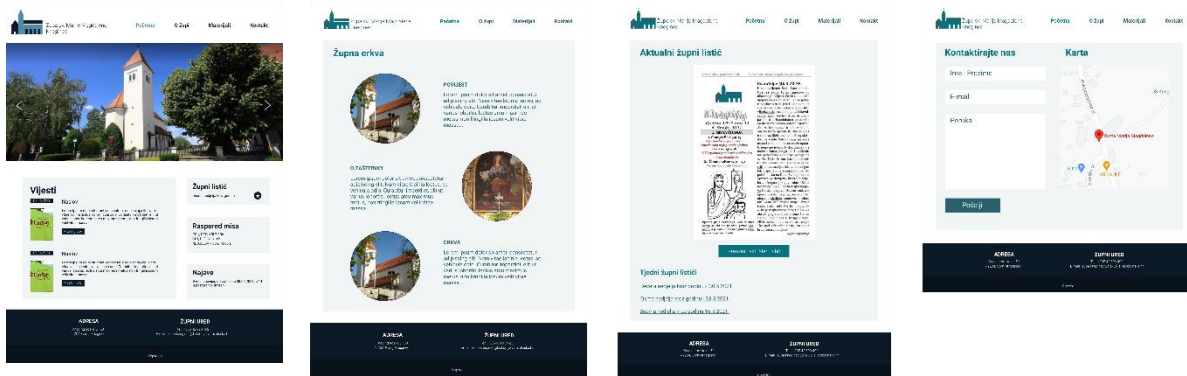
5.1. Postavljanje dizajnerskog rješenja

Za potrebe ovog završnog rada odabrana je izrada funkcionalne web stranice Župe sv. Marije Magdalene u Knegincu. Na zahtjev župnika, stranica mora imati jednostavno dizajnersko rješenje kojim će posjetitelji (bilo da se radi o mlađim ili starijim osobama) vrlo lako i brzo naći sve njima potrebne informacije. Stoga je odlučeno da se struktura stranice sastoji od glavne početne stranice na kojoj bi na jednom mjestu bile sve osnovne informacije (najnovije vijesti, aktualni župni listić, raspored misa, eventualne najave, kontakt i adresa), podstranice koje bi sadržavale osnovne informacije o samoj župi i njenoj zajednici (povijest župe, opis pojedinih zajednica koje djeluju unutar župe itd.), podstranici u kojoj bi bili arhivirani svi materijali koji su objavljeni te kontakt stranice gdje posjetitelji mogu ispuniti obrazac i poslati željenu poruku uredu (Slika 18.).



Slika 17. Ilustrirani prikaz strukture web stranice

Dizajnersko rješenje za web stranicu župe prilagođeno je WordPress sustavu – sadržaj koji će se prikazati raspoređen je prema jednom od standardnih rasporeda sadržaja za blog. Stoga će se predlošci, koji će biti sadržani u temi, sastojati od osnovnih dijelova (zaglavlje, glavni i bočni stupac, podnožje) kako bi se osigurala lakša implementacija postojećih programskih dodataka i programčića. [1] Sam prototip predložaka, odnosno stranica, doručen je u besplatnoj UI i UX dizajnerskoj aplikaciji pod nazivom Figma. Besplatna verzija aplikacije sadrži osnovne alate za oblikovanje sučelja pomoću vektora [26]. Dizajn web stranice temelji se na minimalističkom logotipu župe koji sadrži dvije nijanse plave boje te će se one provlačiti kroz cijeli dizajn svih predložaka (fontovi, linkovi, pozadine i sl.) (Slika 19).



Slika 18. Dizajnersko rješenje za veće zaslone izrađeno pomoću Figma

5.2. Bootstrap

Za izradu strukture predložaka koristit će se Bootstrap. Bootstrap je open source, odnosno besplatan i svima dostupan radni okvir namijenjen kao alat za izrađivanje responzivnih stranica prikladnih za manje ekrane i mobilne uređaje. Razvili su ga Mark Otto i Jacob Thornton 2011. godine te se prva inačica sastojala od samo CSS (*bootstrap.css*) i HTML komponenta, a kasnije se u knjižicu Bootstrapa ubacila i JavaScript biblioteka (*jquery-inačica.slim.js*, *popper.js* i *bootstrap.js*). [1] Ideja je da korisnik pomoću Bootstrapa jednostavno i brzo ubacuje ugrađene CSS klase u HTML elemente kako bi se stvorio osnovan kostur stranice, odnosno da bi se stvorio prostor za inovativna dizajnerska rješenja. U web dizajnu se za izradu konzistentnog rasporeda elemenata koristi takozvana mreža. Takozvani Bootstrap *grid* uključuje responzivanu, mobilnu fluidnu mrežu prilagođenu u 12 stupaca ovisno o veličini ekrana. [2] Kako bi se ispravno pozicionirao sadržaj, Bootstrap koristi određenu strukturu – elementi se moraju smjestiti unutar *redova* (engl. *rows*) pojedinih stupaca koji su određeni div-

om s klasom *spremnika* (engl. *container*). Izradu rasporeda elemenata unutar takvog grida omogućuje sustav četiri ugrađenih CSS klasa: *xs* (mobilni uređaji), *sm*(tableti), *md*(laptopi) i *lg* (veći desktop). Kombiniranjem navedenih CSS klasa stvara se veća fleksibilnost elemenata na web stranici. Moguće je kombinirati i takozvane col klase (skraćena od *column*) čiji zbroj unutar redova mora bit jednak 12 (Slika 20). [3]

```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```

Slika 19. Primjer rasporeda elemenata pomoću Bootstrap grida

5.3. index.php

Kao što je već prije navedeno, da bi WordPress prepoznao mapu predložaka kao ispravnu temu potrebno je da u njoj postoji datoteka *index.php*. Iz tog razloga se jednostavno promjeni naziv datoteke *index.html* u *index.php*. To je dovoljno za početak. Kasnije će se, stvaranjem drugih datoteka (npr. *header.php*, *footer.php*), dodavati i važne funkcije u tu datoteku. *Index.php* koristi se kao zadani predložak ukoliko se ne nađe drugi koji je po hijerarhiji na višoj poziciji. Zbog toga se najčešće unutar datoteke ugrađuje i osnovna petlja upravo da bi se prikazali elementi svakog nedefiniranog predloška. [2]

5.4. style.css

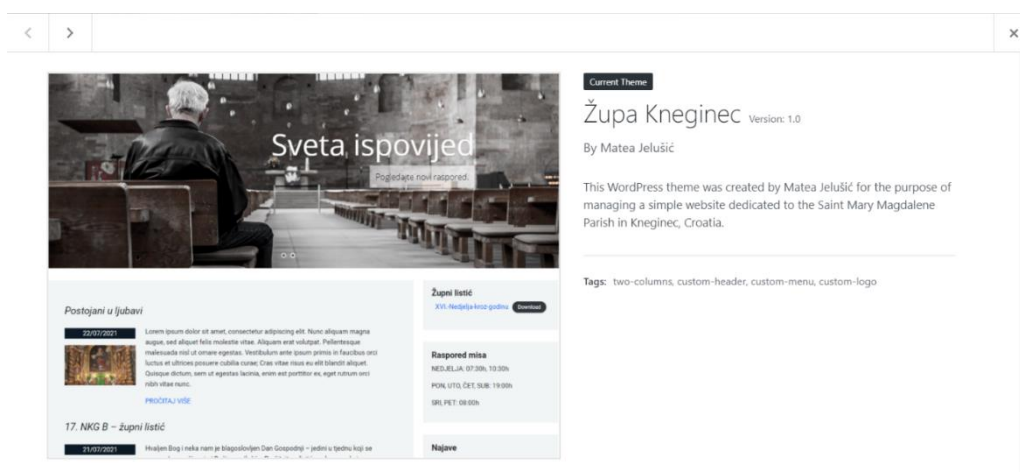
Svaka WordPress tema mora sadržavati dobro formatiranu *style.css* datoteku kako bi bila prepoznata kao ispravna i funkcionalna. WordPress koristi *style.css* kao referencu za temu [19], odnosno kao izvor informacija o predlošku unutar administracijskog sučelja u WordPress-u koji je potreban za čitanje i aktiviranje teme. [2] Unutar *style.css* datoteke nalazi se uvodni komentar koji sadrži osnovne informacije za cijelu temu: naslov teme, URI (URL stranice koja sadrži informacije o temi), autora teme, kratak opis, verzija teme, tagove (riječ i koje omogućuju drugima lakši pronalazak teme ukoliko se ona postavi za javno korištenje), licence, podaci o testiranju itd. (Slika 21).

Redosljied informacija opisanih u style.css datoteci napravljen je na temelju predložka priloženog na adresi https://codex.wordpress.org/Theme_Development#Theme_Stylesheet (kodeks WordPress-a) te je poželjno provjeriti trenutno preporučeni redosljied. [1]

```
/*
Theme Name: Župa Knežinec
Author: Matea Jelušić
Description: This WordPress theme was created by Matea Jelušić for
the purpose of managing a simple website for a Catholic parish.
Version:1.0
Tags: two-columns, custom-header, custom-menu, custom-logo
*/
```

Slika 20. Komentar unutar style.css datoteke

Uz ove podatke koji će biti vidljivi prilikom odabira teme unutar stavke *Izgled>Teme* (*Appearance>Themes*), dodatna datoteka koja usavršava izgled teme jest i sličica dizajna za lakšu vizualnu prepoznatljivost. Screenshot dizajna teme mora biti širok 1200 piksela i visok 900 piksela te se preporuča spremati datoteku u PNG-8 formatu. Za obradu sličice preporuča se korištenje Photoshopa. Datoteka se sprema unutar mape teme pod nazivom screenshot.png kako bi ju WordPress lakše mogao prepoznati. Sada će se pod stranicom *Izgled* (*Appearance*) vidjeti tema sa snimkom zaslona za temu, imenom, opisom i opcijom za aktiviranje (Slika 22). [1] Aktivacijom tema postaje spremna za dodavanje WordPress sadržaja.



Slika 21. Aktivacija vlastite teme

5.5. `template.css`

Za oblikovanje stilova pojedinih elemenata predložaka koristiti će se posebna datoteka pod nazivom `template.css` koju je najbolje smjestiti u podmapu naziva `css` unutar mape teme predložaka (`/zupakneginec/css/template.css`). Sva `css` sintaksa koja će se koristiti u oblikovanju pojedinih elemenata biti će opisana u poglavljima posvećenima predlošcima unutar kojih se ti elementi nalaze.

5.6. `functions.php`

Datoteka `functions.php` sadrži kodove pomoćnih funkcija i postavka za lakšu uporabu predložaka. Kroz funkcije i postavke mogu se dodavati nova korisnička polja, izmijeniti određena zaglavlja, ubaciti razni programčići itd. [2] Ukratko, radi se o datoteci koja omogućava različite modifikacije predložaka - ona definira što se sve u predlošku može učiniti. U ovu datoteku se uglavnom ubacuju funkcije potrebne za stvaranje unaprijed formatiranih odjeljaka sadržaja koji se mogu izmjenjivati preko administrativnog sučelja WordPressa. Ukoliko je prisutan u mapi predložaka teme, datoteka `functions.php` se automatski učitava prilikom WordPress inicijalizacije.

U samoj datoteci, koja se mora nalaziti u mapi predložaka, nakon početne oznake za PHP slijedi funkcija `load_scripts()`; unutar koje se smještaju kuke (engl. *hooks*) `wp_enqueue_style` i `wp_enqueue_scripts`. One omogućuju da se datoteke s opisom stila predložaka (`template.css` i `bootstrap.min.css`) i JS skripte potrebne za ispravan rad Bootstrapa (`bootstrap.min.js` i `popper.min.js`) učitaju jedna za drugom (engl. *enqueueing*) zajedno s WordPress temom. `Template.css` se kreira unutar mape predložaka, dok se datoteke za Bootstrap mogu preuzeti sa službene stranice `getbootstrap.com` (kodovi dostupni pod sekcijom *Quick Start*). Važno za napomenuti je da se kuka za Bootstrap datoteke treba nalaziti iznad kuke za stil predložaka jer će se na taj način omogućiti modificiranje Bootstrap postavka. Unutar kuke dodaje se pet parametara - oznaka skripte/stila (u primjeru `'bootstrap-js'`, `'bootstrap-css'`, `'template'`), putanja do datoteke pomoću direktne veze ili preko funkcije `get_template_directory_uri()`, nakon koje se navodi točna lokacija datoteke (u primjeru `./css/template.css`), niz argumenata koji navode ovisi li datoteka o nekim drugim datotekama (ako je primjerice za ispravan rad potrebno prije učitati neku drugu datoteku, ona se označuje unutar *varijabla* (engl. *array*)), verzija datoteke (u primjeru `'1.14.0'`, `'4.1.0'`, `'1.0'` itd.) te argument koji opisuje za koji je tip medija namijenjena datoteka (`'all'`). Na kraju je potrebno dodati funkcije `add_action` i kuku `wp_enqueue_scripts`

kojima se osigurava da će pridružene skripte i stilovi biti prepoznati i primijenjeni u WordPressu (Slika 23). [23]

```
<?php
/* Dodavanje stylesheet i script datoteka */
function load_scripts(){
    wp_enqueue_script('bootstrap-
js', 'https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js',
    array(), '4.1.0', true);
    wp_enqueue_style('bootstrap-
css', 'https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css',
    array(), '5.0.0', 'all');
    wp_enqueue_style('template', get_template_directory_uri() . '/css/template.css',
    array(), '1.0', 'all');
}
add_action ('wp_enqueue_scripts', 'load_scripts');
```

Slika 22. Dodavanje datoteka stila (.css) i skripti (.js)

Ove datoteke se mogu dodati i na tradicionalan način (ubacivanje koda u *<head>* sekciju zaglavlja), no to znatno usporava učitavanje lokacije. [1] Ostale funkcije koje će se dodavati u datoteku *functions.php* opisane su u narednim poglavljima, ovisno o elementima i programčićima koji se javljaju na pojedinim dijelovima web stranice.

5.7. header.php

Zaglavlje sadrži elemente koji se nalaze na samom vrhu stranice. U zadanom primjeru to su logotip stranice, naslov, navigacija, opcija za pretraživanje i ikone društvenih mreža. Datoteku zaglavlja je, za ispravno funkcioniranje, potrebno pravilno imenovati u *header.php* te na samom vrhu *index.php* datoteke upisati php funkciju *<?php get_header?>* kojom se osigurava uključivanje *header.php* datoteke prilikom učitavanja stranice. [1] *Header.php* sadrži kod za deklaraciju doctype-a te oznake *<head>* i *<header>*. Unutar *<head>* oznake nalaze se meta podaci, naslov stranice i WordPress funkcija *<?php wp_head(); ?>* koja služi kao „zakačka“ WordPress-u i njegovim dodacima za dodavanje raznih funkcija u zaglavlje (Slika 24). [19]

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Župa Sv. Marije Magdalene, Kneginec</title>
    <?php wp_head(); ?>
  </head>
```

Slika 23. Sadržaj unutar *<head>* oznake

Unutar oznake `<body>` dodaje se WordPress funkcija `<?php body_class();?>` kojom se omogućuje kasnije dodavanje raznih CSS klasa elementima unutar tijela stranice. Zaglavlje stranice, označen oznakom `<header>`, podijeljen je u dvije *sekcije* (engl. *sections*). Prva je sekcija u kojoj se nalaze ikone za društvene mreže i opcija pretraživanja stranice (naziv klase 'top-bar'), dok drugu čini logotip župe, naslov i izbornik s podstranicama (naziv klase 'menu-area'). Za prvi dio je potrebno unutar *spremnika* (engl. *container*) dodati *red* (engl. *row*) koji se sastoji od dva *div*-a (jedan za ikone društvenih mreža, drugi za pretraživanje). Drugi dio zaglavlja sastoji se od podsekcija koje su određene klasama ('logo', 'logotext', 'main-menu') (Slika 25).

```
<body <?php body_class();?>
  <header>
    <section class="top-bar">
      <div class="container">
        <div class="row">
          <div class="social-media-icons col-xl-10 col-md-8 col-sm-8 col-
6">
            </div>
            <div class="search col-xl-2 col-md-4 col-sm-4 col-6 text-end">
              </div>
            </div>
          </div>
        </section>
        <section class="menu-area">
          <div class="container">
            <div class="row d-flex align-items-center">
              <section class="logo col-md-2 col-sm-12 col-12 text-end">
                </section>
              <section class="logotext col-md-2 col-sm-12 col-12">
                </section>
              <nav class="main-menu col-md-8 text-end">
                </nav>
            </div>
          </div>
        </section>
      </div>
    </div>
  </div>
</div>
```

Slika 24. Sekcije unutar `<header>` oznake

5.7.1. Ikone društvenih mreža i obrazac za pretraživanje

Za dodavanje ikona društvenih mreža koristio se programčić pod nazivom *Social Icons Widget & Block* autora WPZOOM. U tu svrhu potrebno je najprije pod stavkom *Dodaci>Dodaj novi (Plugins>Add New)* potražiti i aktivirati navedeni programčić te u datoteci *functions.php* registrirati novi *sidebar*. Funkcija se poziva pomoću funkcije *add_action*.

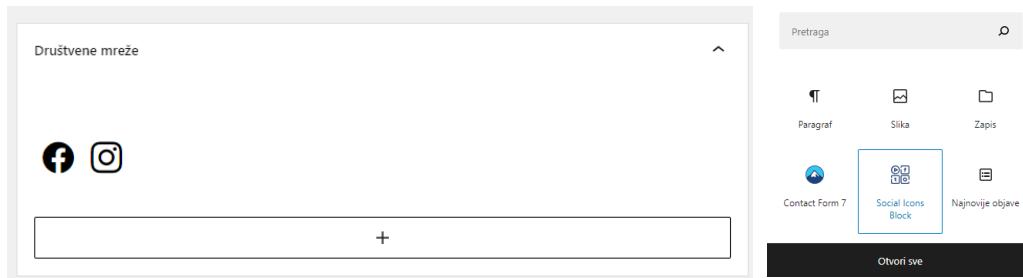
Unutar funkcije se dodaju kuka `'widgets_init'` (služi za registriranje sidebarova i programčića) te oznaka nove funkcije (`'zupa_kneginec_sidebars'`). Funkciju čine argumenti koji se šalju u obliku varijabla te im se dodaje prigodno ime, identifikacija i opis (Slika 26). [30]

```
add_action('widgets_init', 'zupa_kneginec_sidebars');
function zupa_kneginec_sidebars(){
    register_sidebar(
        array(
            'name'=> 'Društvene mreže',
            'id'=>'social-media',
            'description'=>'Social Media icons widget. ',
            'before_widget'=><div class="widget-wrapper">',
            'after_widget'=></div>',
            'before_title'=><h2 class="widget-title">',
            'after_title' => </h2>'
        ));}

```

Slika 25. Registriranje sidebara Društvene mreže

Sada se unutar WordPress administrativnog sučelja pod stavkom *Izgled>Widgeti* (*Appearance>Widgets*) može pronaći nova opcija pod nazivom *Social Icons Block* koja jednim klikom omogućuje dodavanje i oblikovanje ikona društvenih mreža i spajanje na odgovarajuću putanju (Slika 27).



Slika 26. Gotovi programčić Društvene mreže na kojem je primijenjen Social Icons Block

Unutar prvog div-a u `header.php` datoteci ubacuje se if petlja (Slika 28). Kuka `is_active_sidebar` provjerava postoji li aktivan programčić unutar određenog sidebara. Ukoliko je to istina, pokreće se funkcija `dynamic_sidebar` koja osigurava da se sadržaj programčića prikaže na zaslonu.

```
<div class="social-media-icons col-xl-10 col-md-8 col-sm-8 col-6">
    <?php
        if (is_active_sidebar ('social-media')){
            dynamic_sidebar ('social-media');
        }
    ?>
</div>

```

Slika 27. Petlja koja izvodi funkciju `dynamic_sidebar`

Za dodavanje obrasca za pretraživanje koristi se jednostavna funkcija `get_search_form()`; koja se dodaje na predodređeno mjesto unutar `header.php` datoteke (Slika 29). [31]

```
<div class="search col-xl-2 col-md-4 col-sm-4 col-6 text-  
end" style="padding:5px;">  
    <?php get_search_form();?>  
</div>
```

Slika 28. Funkcija za dodavanje obrasca za pretraživanje

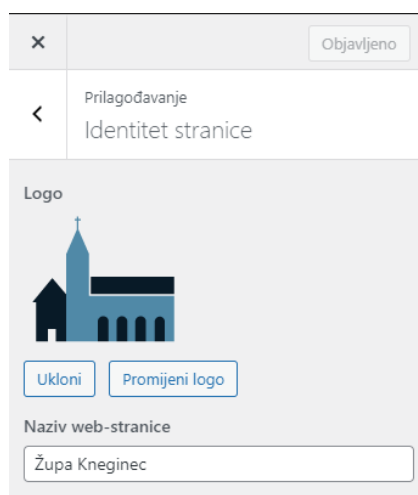
5.7.2. Opcija za dodavanje logotipa

Da bi se omogućila opcija dodavanja logotipa, u datoteci `functions.php` unutar glavne konfiguracijske funkcije ubacuje se funkcija `add_theme_support()`; u koju se upisuje parametar za logotipove (`'custom_logo'`). [32] Dodavanjem varijabla može se odrediti njegova visina i širina (`'110'`, `'200'`) (Slika 30).

```
add_theme_support( 'custom-logo', array('height' => 110,'width' => 200));
```

Slika 29. Funkcija za dodavanje logotipova

Nakon ponovnog osvježavanja stranice u pregledniku, klikom na opciju *Prilagodi* (*Customize*) koja se nalazi u traci na samom vrhu stranice (ukoliko se ona gleda preko WordPress sučelja), pojavljuje se s lijeve strane traka s dodatnim opcijama promjene izgleda predloška. Među njima je i opcija *Identiteta stranice* (Slika 31). Prilikom odabira logotipa (*Select logo*), pojavljuje se napomena koja sugerira da slika bude maksimalne visine od 110px i širine 200px, kako je i postavljeno u `functions.php` datoteci.



Slika 30. Stavka *Identitet stranice* pomoću koje se dodaje logotip

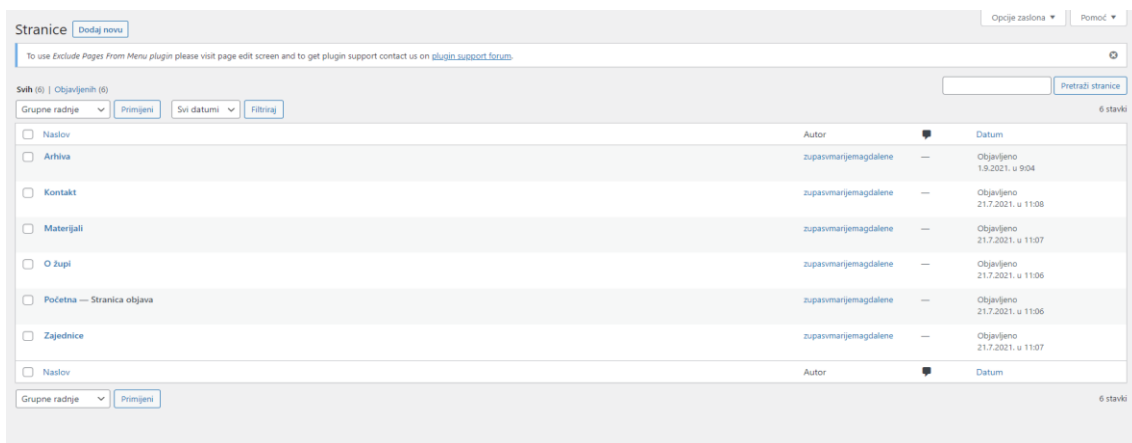
Da bi se logo vidio na stranici potrebno je u *header.php* datoteci dodati funkciju *the_custom_logo()*; koja će povući spremljenu sliku u PNG formatu iz baze podataka (Slika 32).

```
<section class="logo col-md-2 col-sm-12 col-12 px-0 text-end">
    <?php the_custom_logo(); ?>
</section>
```

Slika 31. Funkcija koja prikazuje logotip na predlošku

5.7.3. Glavni izbornik

Za kreiranje glavnog izbornika prvo je potrebno stvoriti stranice u WordPressu. Odabire se stavka *Stranice > Dodaj novu* (*Pages > Add New*) te se svakoj prije objavljivanje mora dodati naziv (u primjeru to su stranice '*Počtna*', '*O župi*', '*Zajednice*', '*Materijali*', '*Arhiva*', '*Kontakt*'). Pregled svih stranica može se vidjeti klikom na stavku *Stranice* (engl. *Pages*) (Slika 33).



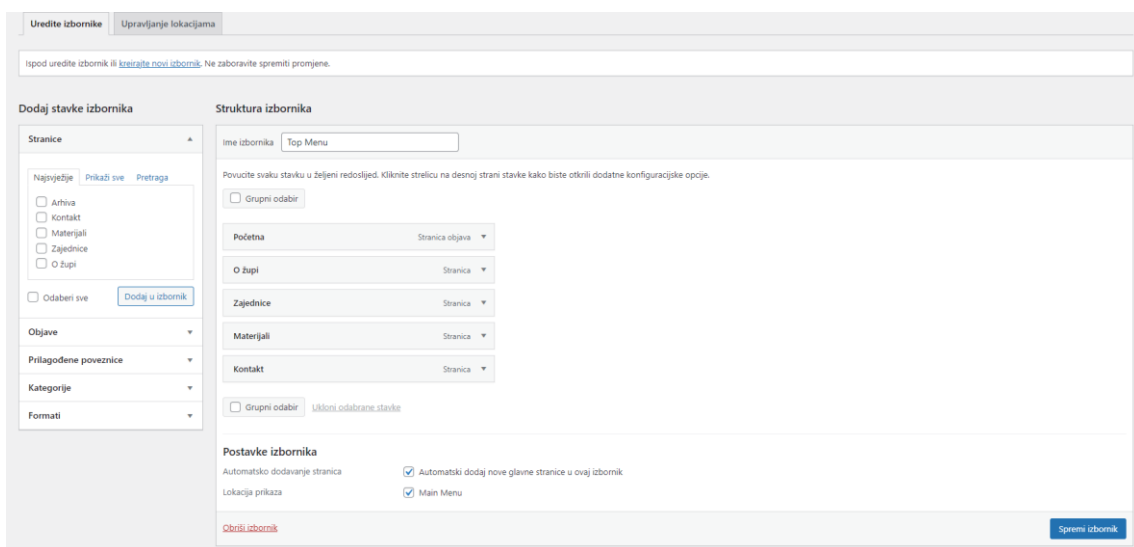
Slika 32. Pregled svih stranica u WordPressu

Nakon toga potrebno je registrirati izbornik u *functions.php* datoteci. Unutar glavne konfiguracijske funkcije upisuje se funkcija *register_nav_menus()*; koja omogućuje dodavanje više lokacija izbornika [1]. U varijablu se upisuje jedinstvena identifikacija za lokaciju izbornika ('*my_main_menu*') i naziv po kojem će se moći odabrati izbornik unutar WordPressa (Slika 34). [33]

```
/*Glavna konfiguracijska funkcija*/
function zupa_kneginec_config(){
    /* Registriranje izbornika */
    register_nav_menus(
        array(
            'my_main_menu' => 'Main Menu'
        )
    );
};
```

Slika 33. Registriranje izbornika u *functions.php* datoteci

Ponovnim osvježavanjem WordPress administratorskog sučelja, trebao bi se pod stavkom *Izgled (Appearance)* pojaviti podstavka *Izbornici (Menus)*. Klikom na podstavku otvara se stranica u kojoj administrator može vrlo jednostavno kreirati izbornike, odnosno dodavati i brisati stranice koje će se pojavljivati u izborniku. Odabir stranica nalazi se s lijeve strane – potrebno ih je označiti kvačicom i pritisnuti gumb *Dodaj u izbornik*. S desne strane je odlomak koji se bavi samom strukturom izbornika. Tu se dodaje naziv izbornika, određuje poredak stranica, automatsko dodavanje novih stranica u izbornik i lokacija prikaza (koja se prethodno odredila registriranjem izbornika i govori o tome da li će ponuđeni izbornik biti glavni ili će se nalaziti negdje sporedno na stranici). Pritiskom na opciju spremi izbornik pohranjene su sve promjene (Slika 35).



Slika 34. Podešavanje strukture izbornika

Sada je samo potrebno postaviti izbornik na određeno mjesto unutar zaglavlja. Unutar *header.php* datoteke ubacuje se funkcija *wp_nav_menu()*; (Slika 36) koja sadrži parametar lokacije izbornika '*theme_location*' za koju se prethodno odredilo da ona glasi '*my_main_menu*'.

```
<nav class="main-menu col-md-8 text-end">
    <?php wp_nav_menu( array( 'theme_location' => 'my_main_menu' ) ); ?
    >
</nav>
```

Slika 35. Funkcija za prikaz izbornika u predlošku

Promjenom pojedinih svojstava elemenata (pozadine, margine, padding, pozicije) pomoću CSS-a u *template.css* datoteci (Slika 37), oblikovan je konačan izgled izbornika i zaglavlja općenito (Slika 38).

```

/* Top Bar */
.top-bar{ background: #508BA9; color: #0B1A26; padding: 5px 0; }
/* Header - Izbornik */
.menu-area{ background: #ffffff; padding: 10px 0; }
.main-menu ul{ list-style: none; padding: 0; margin: 0; font-size: 20px; margin-top:50px; }
.main-menu ul li{ display: inline; margin: 0 30px 0 0;}
.main-menu ul li:last-child{ margin: 0; }
.main-menu ul li a{ color: #0B1A26; text-decoration: none;}
.main-menu ul li.current-menu-item a{ color:#508BA9; }
/* Header - Tekst logotipa */
.logotext {font-size: 20px; margin-top:50px;}
.align {display: inline-block; align-items:center; width: 100%;}

```

Slika 36. CSS kod za oblikovanje zaglavlja



Slika 37. Konačan izgled zaglavlja

5.7. footer.php

Podnožje (engl. *Footer*) je nekad služilo kao prostor za postavljanje kontakt informacija i copyright-a, no posljednjih godina njegova se funkcija znatno proširila, pa danas sadrži dodatne informacije kao što su navigacija, postovi s raznih društvenih mreža, obrasce za newslettere i sl. Zbog toga što je sastavni dio svake podstranice, najbolja praksa je odvajati podnožje u posebnu php datoteku pod nazivom *footer.php* te na dnu *index.php* datoteke predložka dodati WordPress funkciju *php get_footer ()*; kako bi sadržaj podnožja bio vidljiv na stranicama. *Footer.php* datoteka iz primjera sastoji se od dvije kolone jednakih omjera prema Bootstrap grid sistemu u kojima su ispisane osnovne informacije – adresa i kontakt broj. Ispod toga cijeli red ispunjava prostor u kojem je smješten Copyright. Važno je pri kraju datoteke podnožja uključiti funkciju *php wp_footer ()*; kojom se WordPressu omogućuje da na njega primjenjuje svoje dodatke i programčiće (Slika 39). [19]


```

<footer class="text-center footer">
<div class="container p-4 pb-0">
  <div class="row">
    <div class="col-md-6 adresa">
      <h4>ADRESA</h4>
      <p>Knežingradska ul. 50</p>
      <p>42204, Knežinec Gornji</p>
    </div>
    <div class="col-md-6 zupni-ured">
      <h4>ŽUPNI URED</h4>
      <p>Tel: +385 42 690 492</p>
      <p>E-mail: <a class="e-mail-link"
        ref="mailto:zupa-knežinec@biskupija-varazdinska.hr">
        zupa-knežinec@biskupija-varazdinska.hr</a>
      </p>
    </div>
  </div>
  </div>
  <div class="text-center p-3 copyright">
    © 2021 Copyright: Župa Knežinec
  </div>
</footer>
<?php wp_footer();?>
</body>

```

Slika 38. Ispisan kod footer.php datoteke

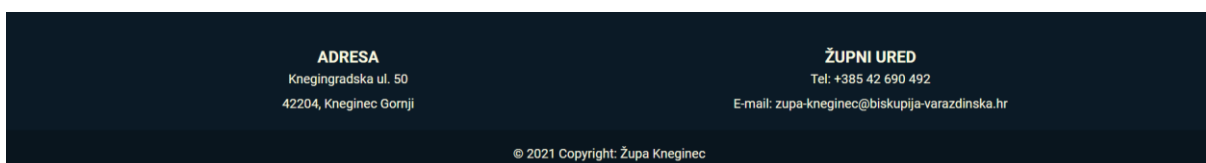
Unutar template.css datoteke navodi se sintaksa koja omogućuje promjenu izgleda podnožja. Dodana je pozadinska boja i svojstvo *padding* na pojedine elemente (Slika 40). Primjenom ovih svojstava dobiva se konačan izgled podnožja (Slika 41).

```

/* Footer */
.footer { background: #0B1A26; padding-top: 20px;
padding-bottom:0; color: #fcfae1;}
.adresa, .zupni-ured {padding-bottom: 10px;}
.adresa p, .zupni-ured p {line-height: 15px;}
.zupni-ured a {text-decoration: none; color: #fcfae1;}
.copyright {background-color: rgba(0, 0, 0, 0.2);}

```

Slika 39. Ispisana CSS sintaksa koji se nalazi u template.css datoteci



Slika 40. Konačan izgled podnožja u pregledniku

5.8. page-pocetna.php

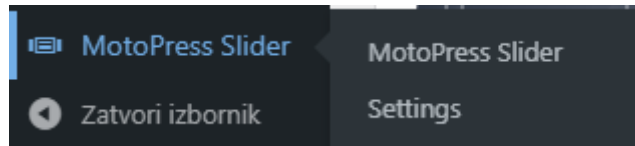
Središnji dio početne stranice sastojati će se od *klizača* (engl. *slider*), glavnog stupca u kojem će se nalaziti petlja s najnovijim objavama i bočnog stupca u kojem se unutar `<aside>` elementa nalaze sekcije s najbitnijim informacijama koje će administrator moći redovito ažurirati preko programčića (tjedni župni listić u PDF formatu, raspored misa, najave). Na samom početku i na kraju datoteke potrebno je pozvati zaglavlje i podnožje pomoću funkcija `<?php get_header();?>` i `<?php get_footer();?>` kako bi se oni ispravno prikazali na početnoj stranici (Slika 42).

```
<?php get_header();?>
<div class="content-area">
  <main>
    <section class="slider">
    </section>
    <section class="middle-area">
      <div class="container">
        <div class="row">
          <div class="news col-md-9"></div>
          <div class="container col-md-3">
            <div class="row">
              <aside class="sidebar"> </aside>
            </div>
            <div class="row">
              <aside class="sidebar"> </aside>
            </div>
            <div class="row">
              <aside class="sidebar"> </aside>
            </div>
          </div>
        </div>
      </div>
    </section>
  </main>
</div>
<?php get_footer();?>
```

Slika 41. Struktura elemenata predloška *page-pocetna.php*

5.8.1. Postavljanje klizača

Za dodavanje klizača koristio se dodatak pod nazivom *Responsive Slider* autora *MotoPress*. Ovaj besplatan dodatak može se pronaći i aktivirati preko WordPressove stavke *Dodaci>Dodaj novi (Plugins>Add New)*. Radi se o responzivnom elementu koji pomoću raznih animacija vrti namješten broj slajdova na kojima se mogu nalaziti slike, naslovi i linkovi.[34] Nakon uspješne instalacije u izborniku administrativnog sučelja WordPressa može se naći nova stavka pod nazivom *MotoPress Slider* (Slika 43).



Slika 42. Stavka MotoPress Slider

Klikom na stavku otvara se nova stranicu u kojoj se može kreirati klizač. Pritiskom na opciju *Create Slider* otvara se prozor u kojem se može odabrati tip klizača (da li će se raditi o prilagođenom klizaču čiji sadržaj određuje sam administrator ili klizaču generiranom na temelju objava). Odabire se opcija *Custom Slider* te se pritisne tipka *Create Slider*. Nakon toga otvara se nova stranica na kojoj se nalaze postavke klizača (Slika 44). Pod opcijom *General* upisuju se potrebne informacije: naslov i kratica klizača koja će se koristiti za njegovo ugrađivanje u stranicu te visina i širina klizača na desktopu. U primjeru su visina i širina postavljene na 1170x600px (zadana veličina stranice nastale pomoću Bootstrap grida).

The image shows the 'Slider Settings' form with the 'General' tab selected. The form contains the following fields:

- Slider title ***: Input field with value 'Početna'. Description: *Required. The title of the slider. Example: Slider1*
- Slider alias ***: Input field with value 'pocetna-slider'. Description: *Required. The alias that will be used in shortcode for embedding the slider. Alias must be unique. Example: slider1*
- Slider Shortcode**: Input field with value '[mpsl pocetna-slider]'. Description: *Copy this shortcode and paste to your page.*
- Slider size on Desktop (default)**:
 - Width**: Input field with value '1170'. Description: *Initial width of the layers*
 - Height**: Input field with value '500'. Description: *Initial height of the layers*

At the bottom, there are four buttons: 'Save Settings', 'Edit Slides', 'Preview Slider', and 'Close'.

Slika 43. Podešavanje postavki klizača

Sljedeća opcija postavki pod nazivom *Size* (Slika 45) omogućava postavljanje veličine klizača za različite uređaje. Tako se može postaviti veličina za laptop, tablet i mobilne uređaje.

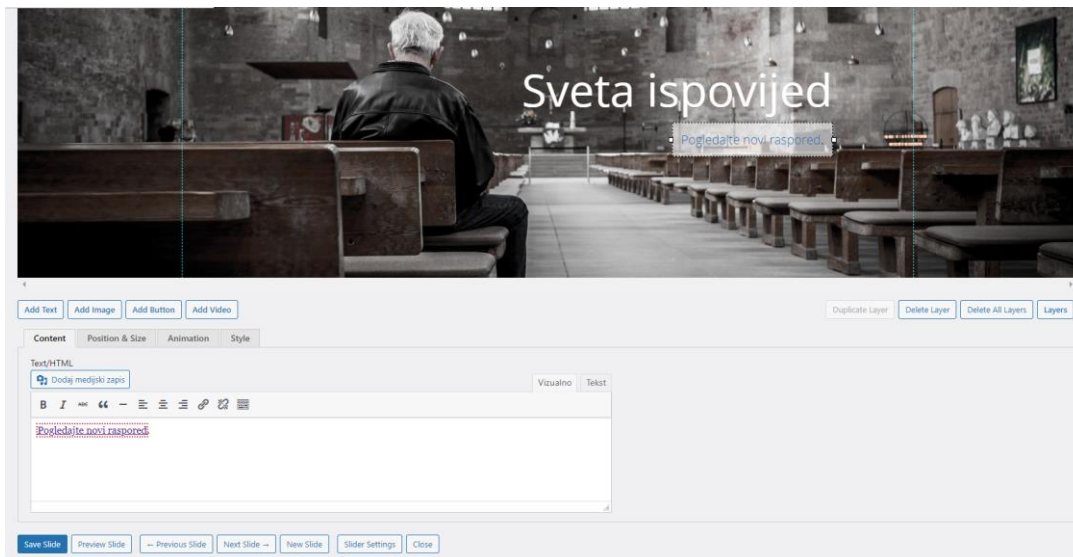
The image shows the 'Slider Settings' form with the 'Size' tab selected. The form contains the following options and fields:

- Full Width**: Make this slider full-width / wide-screen
- Full Height**: Make this slider full-height
- Full Size Grid**: Make grid stretch to parent container. Description: *Even if you select this option you still need to set Grid width and height to define slider size. If you check Full Width and/or Full Height options, the slider will be stretched to screen edges.*
- Slider size on Laptop**: Configure layers size, position and styles manually for Laptop screen size. Initial Laptop dimensions:
 - Width**: Input field with value '1024'
 - Height**: Input field with value '500'
- Slider size on Tablet**: Configure layers size, position and styles manually for Tablet screen size. Initial Tablet dimensions:
 - Width**: Input field with value '768'
 - Height**: Input field with value '500'
- Slider size on Mobile**: Configure layers size, position and styles manually for Mobile screen size. Initial Mobile dimensions:
 - Width**: Input field with value '425'
 - Height**: Input field with value '400'

At the bottom, there are four buttons: 'Save Settings', 'Edit Slides', 'Preview Slider', and 'Close'.

Slika 44. Podešavanje veličine klizača za razne uređaje

Pod opcijom Slideshow određuje se trajanje pojedinih slajdova izraženo u milisekundama, dok opcija Controls omogućuje upravljanje pojedinim elementima klizača (npr. prikaz strelica i bullet pointova). Podešene postavke ažuriraju se pritiskom na gumb *Save Settings*. Odmah pored te opcije nalazi se gumb *Edit Slides* koji omogućuje dodavanje i oblikovanje slajdova. Klikom na opciju *New Slide* otvara se programčić koji omogućuje dodavanje teksta (*Add Text*), slika (*Add Image*), gumba (*Add Button*) i videozapisa (*Add Video*). Dodavanje slika je vrlo jednostavno – pritiskom na Add Image otvara se prozor u kojem je potrebno prenijeti sliku. Odabrana slika može se pomoću miša vrlo lako povećati i pravilno pozicionirati na trenutni slajd. Pritiskom na opciju Add Text generira se tekstualni okvir koji se može oblikovati pomoću opcija koje se nalaze ispod slajda. Tu se može mijenjati pozicija i animacija teksta, njegova boja i veličina fonta (Slika 46). Klikom na gumb *Save Slide* spremaju se promjene te se može prijeći na sljedeći novi slajd (*Next Slide*).



Slika 45. Opcija Edit Slides

Povratkom na stavku *MotoPress Slider* koji sadrži sve generirane klizače, nailazi se na napomenu koja govori o tome kako ubaciti klizač u predložak. (Slika 47) Potrebno je koristiti gotovu funkciju `<?php motoPressSlider('alias') ?>` te umjesto 'aliasa' ubaciti kraticu za generirani klizač. U ovom primjeru to je 'pocetna-slider'.

ID	Name	Shortcode*	Visible from/till	Actions
1	HomePage Slider	[mpsl homepage-slider]	- / -	Settings Edit Template Content Preview Duplicate Delete
2	Pocetna	[mpsl pocetna-slider]	480px / -	Settings Edit Slides Preview Duplicate Delete

* From the page and/or post editor insert the shortcode from the sliders table. From the html use: `<?php motoPressSlider("alias") ?>`

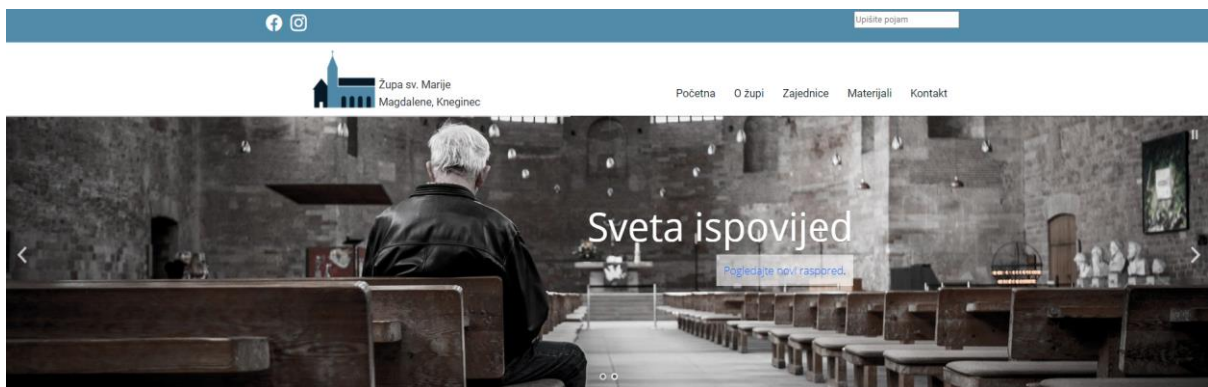
Create Slider Import & Export

Slika 46. Popis kreiranih klizača

Funkcija se ubacuje u predložak na kojem se želi prikazati klizač, dakle u datoteku *page-pocetna.php* (Slika 48). Nakon ponovnog ažuriranja stranice, trebala bi se pojaviti početna stranica s funkcionalnim klizačem (Slika 49).

```
<section class="slider">
  <div class="container">
    <div class="row"> <?php motoPressSlider( "pocetna-
slider" ) ?> </div>
  </div>
```

Slika 47. Ubacivanje funkcije za prikaz klizača



Slika 48. Konačan izgled klizača na početnoj stranici

5.8.2. Petlja s objavama

U glavnom stupcu početne stranice nalaziti će se petlja koja će prikazivati tri najnovije objave obrnutim kronološkim redoslijedom (od novijih prema starijim). Takva petlja dohvatiti će podatke na temelju dogovorenih parametara opisanih unutar prilagođene funkcije te je s toga potrebno, unutar *functions.php* datoteke, registrirati novu funkciju pod nazivom *'latest_posts'*. Važno je također unutar funkcije stvoriti polje argumenata (*'\$args'*), a zatim te argumente predati klasi *WP_Query* koja će omogućiti da se u petlju dohvati sadržaj prema prilagođenim zahtjevima. [23] Lista argumenata je spremište podataka koje navodi određena svojstva, odnosno *ključeve* (engl. *keys*) i njihove *vrijednosti* (engl. *values*) (Slika 50). U primjeru se navode argumenti koji određuju koliko objava će biti na stranici (*'posts_per_page'*), broj objava koji se želi preskočiti (*'offset'*), sortiranje objava prema određenom parametru (*'orderby'*, *'order'*), vrsta objave (*'post_type'*) i status objave (*'post_status'*). [35] Unutar *WP_Query* klase izvršava se prilagođena petlja koja provjerava postoje li objave unutar novog objekta (*'\$query'*) koji slijedi upute dane u listi argumenata (*'\$args'*), ukoliko postoje (*\$query->have_posts():*),

slijedi naredba prikazivanja objava (`$query->the_post();`) i to prema definiranom predlošku koji sadrži ostatak petlje (`content.php`) i koji se poziva preko funkcije `get_template_part();`. [1] Zatvaranjem ove funkcije dodaje se nova funkcija definiranja kratice pomoću koje će se kasnije u predlošku pozvati funkcija (`add_shortcode('latest-post', 'latest-post');`).

```
function latest_post() {
    $args = array(
        'posts_per_page' => 3,
        'offset' => 0,
        'orderby' => 'post_date',
        'order' => 'DESC',
        'post_type' => 'post',
        'post_status' => 'publish'
    );
    $query = new WP_Query($args);
    if ($query->have_posts()) :
        while ($query->have_posts()) : $query->the_post();
            get_template_part('content', get_post_format() );
        ?>
    <?php endwhile; endif;
}
add_shortcode('latest-post', 'latest-post');
```

Slika 49. Prilagođena petlja za prikaz najnovijih objava

Navedeno je da se ostatak petlje nalazi u datoteci `content.php`. S njom se može odrediti način prikazivanja objava u petlji. Za početak je potrebno stvoriti novu datoteku pod nazivom `content.php`. Unutar te datoteke opisuje se izgled jedne objave u petlji (`<article>`). Definira se struktura i izgled pojedinih elemenata (na primjer naslov, linkovi, datum, sličica, sadržaj objave itd.). Za dinamičan prikaz koriste se određene funkcije (Slika 51). Funkcija `<?php the_permalink(); ?>` dohvaća vezu pojedine objave, `<?php the_title(); ?>` naslov objave, a `<?php the_content(); ?>` sam sadržaj. Da bi se prikazao datum potrebno je dodati funkciju `<?php get_the_date();?>` te unutar nje upisati naredbu `echo` kako bi funkcija ispravno ispisala zadani niz znakova. Za sličicu koja bi bila popratni element samog sadržaja objave potrebno je najprije u datoteci `functions.php` ubaciti funkciju `add_theme_support();` koja sadrži varijablu `'post-thumbnails'`. Ova gotova funkcija omogućiti će dodavanje sličice prilikom pisanja nove objave (pojaviti će se stavka pod nazivom *Istaknuta slika (Featured Image)*). Sada se može u `content.php` datoteci dodati funkcija `<?php the_post_thumbnail();?>`. Za definiranje izgleda sličice, odnosno njezine visine i širine, unutar funkcije se ubacuju varijable (170,300).

```

<article>
<div class="container">
  <div class="row">
    <div class="col-md-12" style="margin:20px 0; font-style:italic;">
      <a href="<?php the_permalink();?>" style="text-
      decoration:none; color:#0B1A26;"><h2><?php the_title(); ?></h2></a>
    </div>
  </div>
  <div class="row">
    <div class="col-md-3">
      <div class="container" style="padding:0;">
        <div class="row">
          <div class="col-md-12">
            <p class="datum" style="background:#0B1A26; color:white;
            text-align:center;"><?php echo get_the_date();?></p>
          </div>
        </div>
        <div class="row">
          <div class="col-md-12">
            <?php the_post_thumbnail(array (170,300));?>
          </div>
        </div>
      </div>
    </div>
    <div class="col-md-9" style="margin-top:-20px;">
      <p><?php the_content();?></p>
    </div>
  </div>
</div>
</article>

```

Slika 50. Struktura objave i funkcije koje se koriste za prikaz određenih elemenata

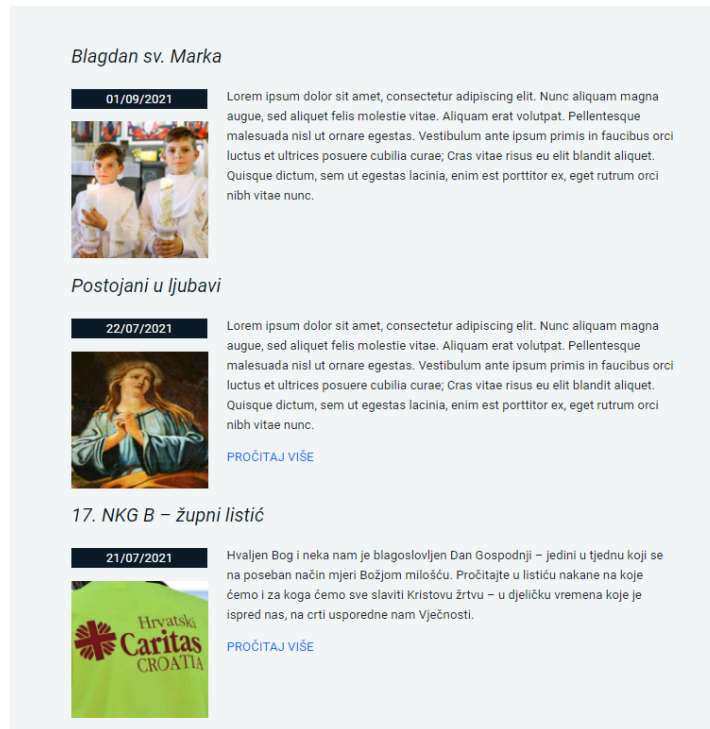
Za kraj je samo potrebno ubaciti kreiranu petlju u sam predložak glavne stranice. Funkcija `<?php echo do_shortcode(); ?>` koristi se za pozivanje kratice (`[latest-posts]`) definirane prethodno unutar `functions.php` datoteke (Slika 52). Ponovnim ažuriranjem stranice može se vidjeti konačan izgled sekcije s najnovijim objavama (Slika 53).

```

<div class="col-md-9 gx-5">
  <div class="news">
    <?php echo do_shortcode('[latest-post]');?>
  </div>
</div>

```

Slika 51. Funkcija za prikaz petlje



Slika 52. Konačan izgled petlje s objavama

5.8.3. Dodavanje bočnog stupca

Sada se može prijeći i na dodavanje dinamičnog sadržaja u bočni stupac. On bi se trebao sastojati od sekcije 'Župni listić', 'Raspored misa' i 'Najave' koje bi administrator preko prilagođenog WordPress programčića mogao vrlo lako izmijeniti i ažurirati. To se može postići dodavanjem novog *sidebarova* čiji je proces registriranja prethodno opisan u poglavlju 5.6.1. *Ikone društvenih mreža i obrazac za pretraživanje*. Unutar funkcije `zupa_kneginec_sidebars()` registriraju se tri nova *sidebarova* pod nazivima 'Župni listić' (Slika 54), 'Raspored misa' i 'Najave' koji imaju posebni naziv i identifikaciju ('id').

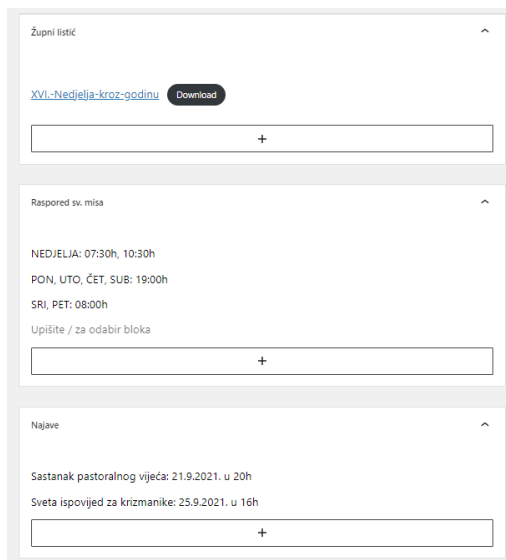
```

register_sidebar(
    array(
        'name'=> 'Župni listić',
        'id'=>'zupnilistic',
        'description'=>'Ovdje se dodaju župni listići u .pdf formatu.',
        'before_widget'=><div class="widget-wrapper">',
        'after_widget'=></div>',
        'before_title'=><h2 class="widget-title">',
        'after_title' => </h2>'
    )
);

```

Slika 53. Registriranje novog programčića pod nazivom 'Župni listić'

Ponovnim ažuriranjem sučelja WordPressa pod stavkom *Izgled>Widgeti* (*Appearance>Widgets*) nalaze se tri nova programčića u koje administrator može lako dodati paragraf teksta, sliku, dokument i mnoge druge elemente (Slika 55).



Slika 54. Gotovi programčići s dodanim dokumentima i tekстом

Da bi se dodani elementi vidjeli na početnoj stranici potrebno je ubaciti kod unutar njenog predloška. U *page-pocetna.php*, unutar `<aside>` oznake (oznaka za sadržaj u sidebarovima), ubacuje se funkcija *dynamic_sidebar()*; (Slika 56). Funkcija na temelju dodijeljenog imena identificira programčić te u pregledniku prikazuje njegov sadržaj. Konačan izgled je postignut oblikovanjem u *template.css* datoteci (Slika 57 & Slika 58). [36]

```

<div class="container col-md-3">
  <div class="row">
    <aside class="sidebar">
      <h4>Župni listić</h4>
      <?php dynamic_sidebar('zupnilistic');?>
    </aside>
  </div>
  <div class="row">
    <aside class="sidebar">
      <h4>Raspored misa</h4>
      <?php dynamic_sidebar('rasporedmisa');?>
    </aside>
  </div>
  <div class="row">
    <aside class="sidebar">
      <h4>Najave</h4>
      <?php dynamic_sidebar('najave');?>
    </aside>
  </div>
</div>

```

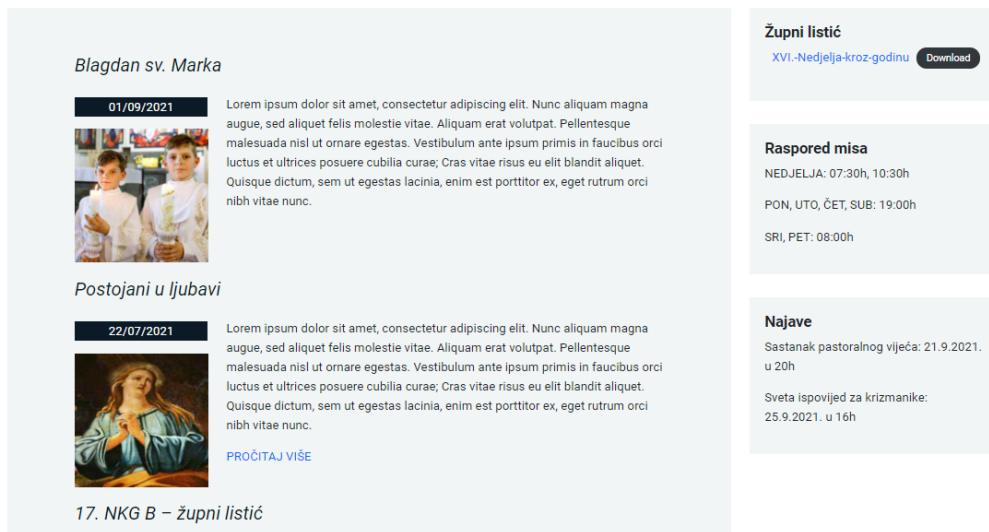
Slika 55. Funkcije za prikaz pojedinih elemenata bočnog stupca

```

/* Sidebar */
.middle-area {margin-top:30px;}
.sidebar{background: #F1F5F5; padding:20px;margin-bottom:30px;}

```

Slika 56. CSS kod za oblikovanje izgleda glavnog i bočnog stupca



Slika 57. Konačan izgled glavnog i bočnog stupca

5.9. page-o-zupi.php i page-zajednice.php

Stranice 'O župi' i 'Zajednice' su zamišljene kao statične stranice na kojima se sadržaj neće toliko često mijenjati. Dakle, budućem administratoru biti će omogućeno da, ukoliko će to biti potrebno, sam ažurira sadržaj (tekst i slike) pomoću programčića. Kao i za sve ostale programčiće prethodno opisane, potrebno ih je prvo registrirati kao nove sidebarove (Slika 59), a nakon toga ubaciti funkciju `dynamic_sidebar()`; u predložak (Slika 60). Isto se ponovi za sve pojedine sekcije slika i teksta koje se želi prikazati na stranici.

```

register_sidebar(
    array(
        'name'=> 'Župna crkva',
        'id'=>'zupnacrkva',
        'description'=>'Ovdje se dodaje sadržaj o župnoj crkvi.',
        'before_widget'=>'<div class="widget-wrapper">',
        'after_widget'=>'</div>',
        'before_title'=>'<h2 class="widget-title">',
        'after_title' => '</h2>'
    )
);

```

Slika 58. Registriranje novog programčića pod nazivom "Župna crkva"

```

<div class="col-md-8 text-frame">

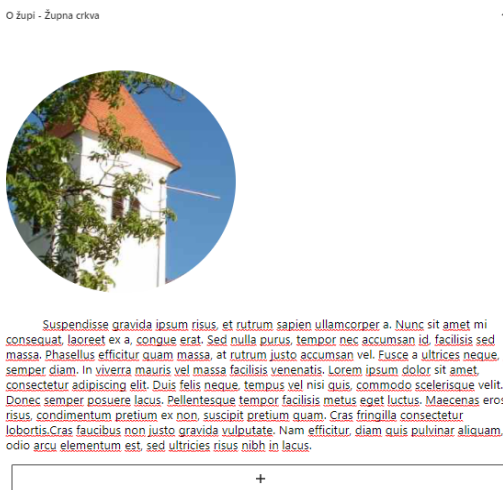
    <h4>Župna crkva</h4>

    <?php
        if( is_active_sidebar('zupnacrkva')){
            dynamic_sidebar('zupnacrkva');}
    ?>
</div>

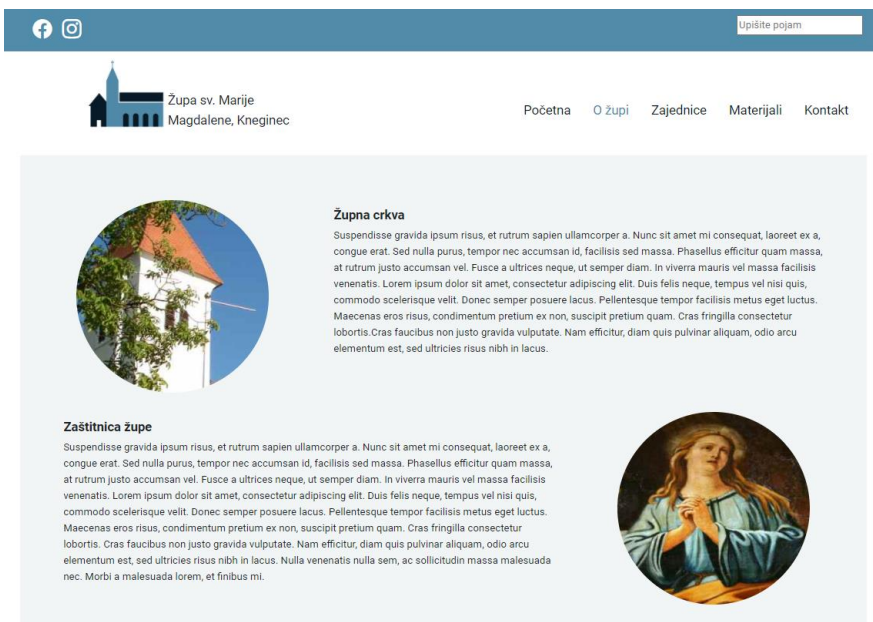
```

Slika 59. Petlja kojom se prikazuje sadržaj programčića "Župna crkva"

Nakon što se registriraju pojedini programčići, otvaranjem stavke *Widgeti(Widgets)* u glavnom izborniku WordPressa dodaje se sav sadržaj u obliku slika i teksta na za to predodređeno mjesto (Slika 61). Ponovnim ažuriranjem može se vidjeti konačni izgled stranice (Slika 62).



Slika 60. Ispunjavanje programčića "Župna crkva"



Slika 61. Konačan izgled stranice "O župi"

5.10. page-Materijali.php

Element koji se pojavljuje u predlošku stranice *'Materijali'* je takozvana slika zaglavlja (engl. *Header image*). Radi se o WordPress dodatku koji omogućuje administratoru odabir reprezentativne slike koja se obično pojavljuje u gornjem dijelu izbornika. [37] Za uključivanje dodatka potrebno je unutar konfiguracijske funkcije *functions.php* datoteke dodati funkciju *add_theme_support('custom-header');* te odrediti parametar visine i širine (*'\$args'*) (Slika 63).

```
$args= array(  
    'height'=>225,  
    'width'=>1920  
);  
add_theme_support('custom-header', $args);
```

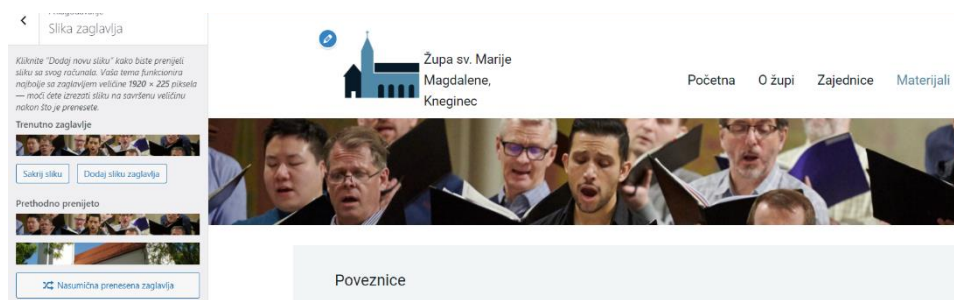
Slika 62. Dodavanje funkcije slike zaglavlja

U predložak *page-materijali.php* ubacuje se gotovi WordPress kod koji sadrži funkcije *header_image();* potrebnu za dohvaćanje slike i *get_custom_header();* odgovornu za pozivanje postavljenih parametara slike (Slika 64).

```
height; ?>" width="<?php echo get_custom_header()->width; ?>" alt="" />
```

Slika 63. WordPress kod za sliku zaglavlja

Nakon toga se u glavnom izborniku WordPressa pod stavkom *Izgled (Appearance)* pojavljuje nova podstavka *Zaglavlje (Header)*. Pritiskom na stavku otvara se WordPress editor koji omogućuje odabir slike zaglavlja. WordPress prilagođava izrezivanje slike prema zadanim parametrima. Također je moguće dodati i birati između više slika zaglavlja, odnosno birati opciju *Nasumična prenesena zaglavlja (Randomize uploaded headers)* kojom će se zaglavlja mijenjati svakim ponovnim dolaskom na stranicu (Slika 65).



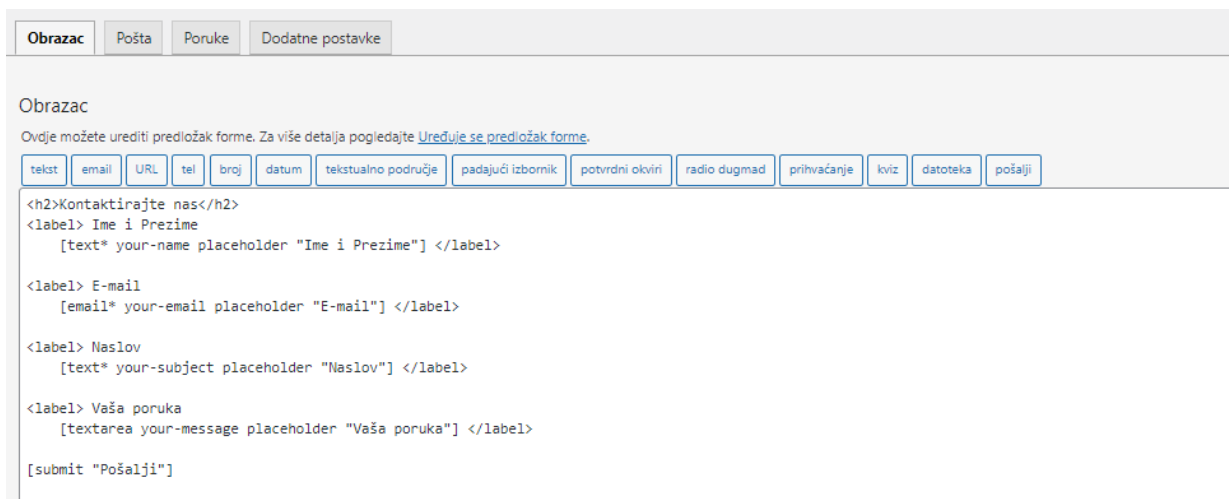
Slika 64. Prilagođavanje slike zaglavlja unutar WordPressa

5.11. page-kontakt.php

Predložak *page-kontakt.php* sastojat će se od dva jednaka stupca unutar kojih će se nalaziti kontakt obrazac kreirani pomoću dodatka *Contact Form 7* te Google karta ugrađena korištenjem jedinstvenog *API ključa*.

5.11.1. Kontakt obrazac

Contact Form 7 jedan je od najpopularnijih i najstarijih dodataka u WordPressu [38] koji omogućuje kreiranje i uređivanje kontaktnih obrazaca te njihovo postavljanje u objave ili za to predviđene stranice. [39] Instalacijom i aktivacijom dodatka pojavljuje se u glavnom izborniku WordPress sučelja novi stavak pod nazivom *Kontakt (Contact)*. Klikom na *Kontakt > Kontakt obrasci (Contact > Contact forms)* omogućeno je jednostavno kreirati novi obrazac pomoću HTML koda. Pritiskom na gumb *Dodaj novi (Add New)* otvara se nova stranica na kojoj se upisuje naziv obrasca i okvir unutar kojeg se mogu dodati razna polja koja će posjetitelji kasnije moći ispuniti. Osnovnim HTML kodom, unutar `<label>` oznake, dodana su određena polja za upis imena i prezimena (`text* your-name`), e-mail adrese (`email* your-email`), naslov poruke (`text* your-subject`) i tekst poruke (`textarea* your-message`) te gumb *Pošalji (submit)* (Slika 66). Za one administratore koji nisu previše upoznati s HTML kodom, na vrhu okvira nalaze se elementi obrasca koje može ubaciti u kod. Također su dodani i nazivi pojedinih polja (`<label>`) i pojedini *placeholderi* (tekst rezerviranog mjesta).



Slika 65. Popunjavanje polja za kontakt obrazac

Pod opcijom *Pošta* prilagođava se obrazac pošte, odnosno kako će izgledati pošta koju će administrator primiti od posjetitelja. Ovdje se može odrediti kamo će se obrazac poslati te kako će pojedini elementi biti formatirani nakon slanja. Ove postavke nisu promijenjene za potrebu izrade ovog obrasca. Opcija *Poruke* omogućuje postavljanje povratnih poruka posjetitelju ukoliko odluči ispuniti obrazac. Ove kratke poruke služe za informiranje posjetitelja o tome da li su ispravno popunjena polja te da li je ispravno poslana poruka (Slika 67). Klikom na gumb *Spremi (Save)* pohranjuje se novi obrazac te je sada samo potrebno dodati gotov obrazac na stranicu na kojoj se želi prikazati.

Slika 66. Prilagođene poruke kontakt obrasca

Da bi se odgovarajući kontakt obrazac ubacio na stranicu *'Kontakt'*, potrebno je dodati novi programčić koji će ga sadržavati. Taj novi programčić pod nazivom *'Kontakt obrazac'* registrira se kao novi sidebar unutar *functions.php* datoteke (Slika 68). Nakon toga se na željeno mjesto unutar datoteke *page-kontakt.php* ubacuje if petlja s funkcijom *dynamic_sidebar()*; u koju se dodaje argument identifikacije za dohvaćanje programčića (Slika 69).

```

register_sidebar(
    array(
        'name'=> 'Kontakt obrazac',
        'id'=>'kontaktobrazac',
        'description'=>'Programčić za ubacivanje kontakt obrasca.',
        'before_widget'=><div class="widget-wrapper">',
        'after_widget'=></div>',
        'before_title'=><h2 class="widget-title">',
        'after_title' => </h2>'
    )
);

```

Slika 67. Registriranje kontakt obrasca

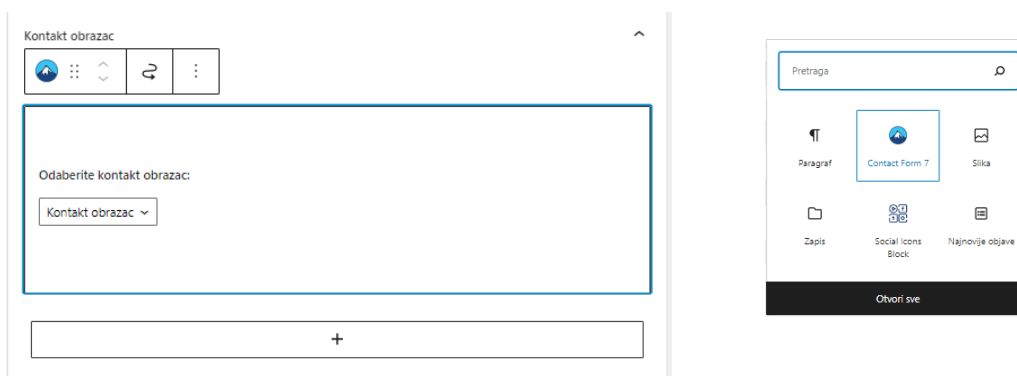
```

<div class="col-md-6">
  <?php
    if( is_active_sidebar('kontaktobrazac')){
      dynamic_sidebar('kontaktobrazac');
    }
  ?>
</div>

```

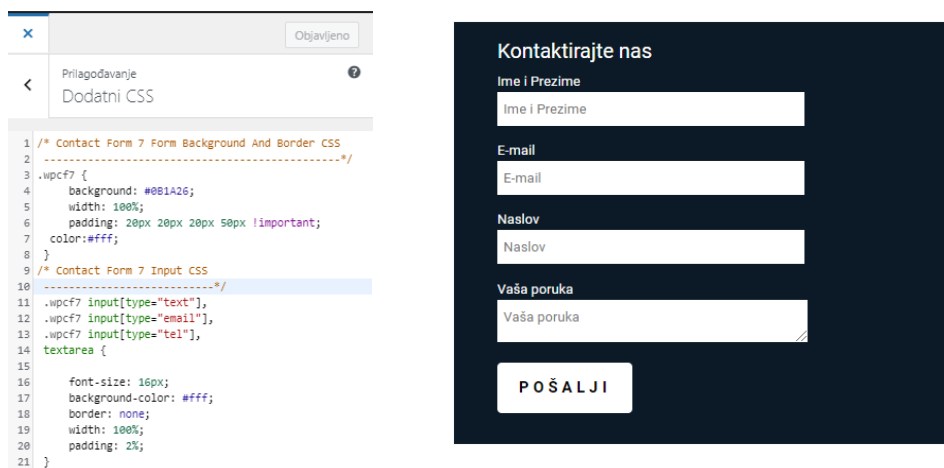
Slika 68. Petlja kojom se prikazuje kontakt obrazac

Unutar sučelja WordPressa pod stavkom *Izgled > Widgeti (Appearance > Widgets)* pod novo registriranim programčićem 'Kontakt obrazac' dodaje se blok *Contact Form 7* (Slika 70) te se odabire koji se od kreiranih obrazaca želi postaviti na web stranicu.



69. Ubacivanje kontakt obrasca u programčić

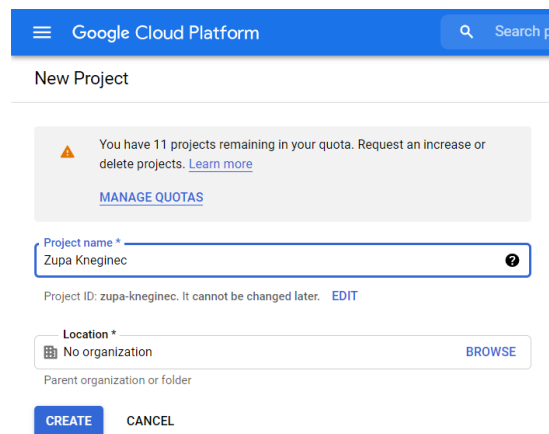
Za oblikovanje obrasca ovog puta se koristila WordPressova opcija dodatnog CSS-a. Prilikom pregledavanja stranice 'Kontakti', u pregledniku se pri samom vrhu nalazi opcija *Prilagodi (Customize)* koja otvara lijevi izbornik. Tu se nalazi opcija *Dodatni CSS (Custom CSS)* (Slika 70). Ubacuje se željeni kod te se pritiskom na gumb *Objavi (Publish)* spremaju promjene. Ova opcija je korisna jer omogućava i administratoru laku i brzu izmjenu izgleda pojedinih elemenata na bilo kojem predlošku bez mijenjanja postavki originalnog koda (Slika 71).



Slika 70. Dodatni CSS i konačan izgled kontakt obrasca

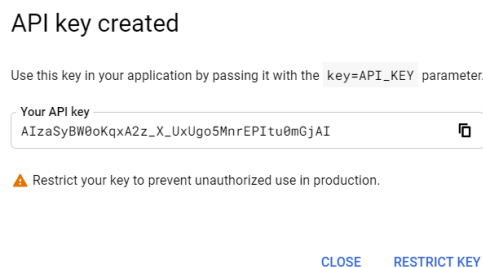
5.11. 2. Dodavanje Google karte

Za ugrađivanje Google karte sa specifičnom lokacijom potreban je Googleov poseban softverski ključ, takozvani *API ključ* (engl. *Application Programming Interface Key*) i odgovarajući JavaScript kod koji se dodaje u predložak *page-kontakt.php*. Registracijom na *Google Cloud platformu* (*console.cloud.google.com*), omogućeno je razvijati i primjenjivati razne usluge bazirane na istoj infrastrukturi kojom se koristi i sam Google. [40] Nakon uspješne registracije pomoću Google korisničkog računa može se kreirati novi projekt (Slika 72).

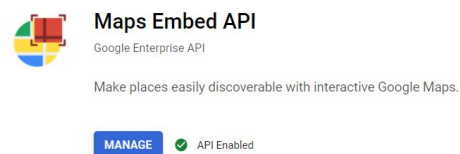


Slika 71. Kreiranje novog projekta na Google Cloud Platformi

Otvora se novi prozor projekta s izbornikom u kojem se nalazi stavka *APIs & Services*. Klikom na stavku odabire se opcija *Credentials*. Tu se generiraju jedinstveni API ključevi (Create Credentials>API key) (Slika 73). S obzirom da se API ključ u ovom slučaju koristi za popunjavanje geokoordinata, potrebno ga je definirati kao takvog. U tu svrhu se pod glavnim izbornikom platforme odabire stavka *Library* koja sadrži razne API dokumentacije. U tražilicu se upisuje '*Maps Embed API*', te klikom na gumb *Embed* aktivira se posebni softver koji omogućuje ugrađivanje interaktivne karte na web lokaciju (Slika 74). [41]



Slika 72. Generiranje novog API ključa

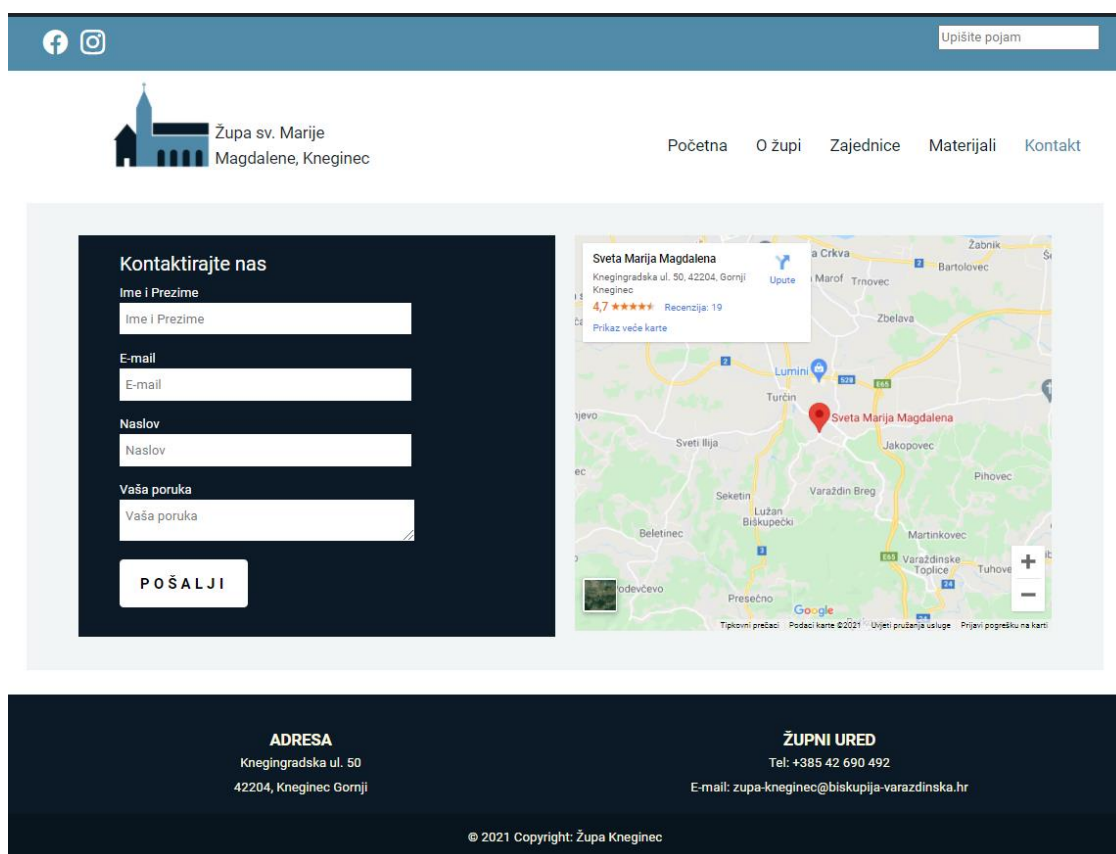


Slika 73. Ugrađen API za interaktivne karte

Na službenoj Google platformi (<https://developers.google.com/maps/>) dostupan je gotov JavaScript kod kojeg je potrebno ubaciti u predložak *page-kontakt.php*. U taj kod se, na za to određeno mjesto, upisuje generirani API kod i naziv lokacije koja se želi prikazati na karti (razmake u nazivu lokacije potrebno odvojiti znakom +) (Slika 75). Spremanjem promjena u pregledniku se pojavljuje karta na kojoj je označena lokacija prethodno navedena u kodu (Slika 76).

```
<div class="col-md-6">
  <iframe width="100% height="480" style="border:0" loading="lazy"
allowfullscreen
src="https://www.google.com/maps/embed/v1/place?key=AIzaSyBB8ioHwDo1fYNFy155Hpe
YEpBkd_Mr4_A&q=Župa+svete+Marije+Magdalene,Kneginec+Gornji&zoom=12">
  </iframe>
</div>
```

Slika 74. Ubacivanje JavaScript koda u predložak



Slika 75. Konačan izgled 'Kontakt' stranice

6. Zaključak

WordPress je s vremenom postao vodeći sustav za upravljanje sadržajem koji omogućava jednostavno korištenje sučelja i brz razvoj web rješenja. Prednosti korištenja WordPressa kao CMS sustava su njegova laka i brza instalacija, jednostavno administratorsko sučelje, dvije vrste sadržaja kojim se upravlja (objave i stranice), ima velik izbor dodataka za proširivanje i prilagođavanje, sadrži veliku bazu gotovih besplatnih i plaćenih tema i predložaka te se lako izrađuju i izmjenjuju vlastiti predlošci. Također, formirana je i zajednica korisnika i programera koji iz dana u dan nastoje poboljšati korisničko iskustvo i performanse WordPressa i svih njegovih web stranica. U ovom radu opisane su osnovne značajke WordPressa kao CMS sustava te na primjeru razvoja vlastitih predložaka objašnjeno njihovo korištenje i funkcija u stvaranju web lokacije. Nadalje, demonstrirani su osnovni WordPress alati i dodaci koji omogućuju razvoj dinamične web stranice te je, kroz jednostavne primjere, dokazana WordPressova mogućnost potpune kontrole nad Web lokacijom. Ovakav pristup razvoja web lokacije od „nule“ ima izrazito veliku obrazovnu vrijednost. WordPress omogućuje apsolutnu prilagodbu svih svojih elemenata te kao platforma otvorenog izvornog koda pruža beskrajne mogućnosti proširenja. Stoga ne čudi zašto se sve više ljudi odlučuje upravo za izradu i održavanje web lokacije na ovoj modernoj platformi. Ove odlike jednostavnosti i brzine prilagođavanja i proširivanja su posebno važne za Web, gdje su tehnološke promjene sve češće.

Literatura

- [1] C. Krol: WordPress kompletan priručnik, Dobar plan, Zagreb 2017.
- [2] M. Janković, M. Tomiša, D. Valdec: Izrada predložaka pomoću WordPress-a, sustava za uređivanje Web sadržaja, Veleučilište u Varaždinu, Varaždin, 2011.
- [3] A. Brazell: WordPress Bible, Wiley Publishing, Inc., Indiana, 2010.
- [4] WordPress: Philosophy. Dostupno na: <https://wordpress.org/about/philosophy/> (pristupljeno 2.8.2021.)
- [5] G.Plumley: Wordpress, Dobar plan, Zagreb 2012.
- [6] M. Olinik, R. Armitage: The Wordpress Anthology, SitePoint, Melbourne, 2011..
- [7] M. Mullenweg: Announcing WordPress 1.5, 2005. Dostupno na: <https://wordpress.org/news/2005/02/strayhorn> (pristupljeno: 2.8.2021.)
- [8] E. Protalinski: WordPress now powers 25% of the Web, 2015. Dostupno na: <https://venturebeat.com/2015/11/08/wordpress-now-powers-25-of-the-web/> (pristupljeno: 2.8.2021.)
- [9] Usage statistics of content management systems, 2021. Dostupno na: https://w3techs.com/technologies/overview/content_management (pristupljeno:2.8.2021.)
- [10] M. Zekić-Sušac, S. Mitrović: Razvoj MVC web aplikacija, Ekonomski fakultet u Osijeku, Osijek, 2013. Dostupno na: http://www.efos.unios.hr/upravljanje-marketingom/wp-content/uploads/sites/228/2013/04/RPA_P4_MVC-web-aplikacije.pdf (pristupljeno:2.9.2021.)
- [11] An MVC Approach to WordPress Theme Development, Dream Production, 2018. Dostupno na: <https://dreamproduction.com/en/mvc-approach-wordpress-theme-development/> (pristupljeno:2.9.2021.)
- [12] Support WordPress.com - Post vs. Page Dostupno na: <https://wordpress.com/support/post-vs-page/> (pristupljeno:2.9.2021.)
- [13] A. Tuca: Categories and Tags in WordPress? How to Use Them Properly, 2019. Dostupno na: <https://themeisle.com/blog/categories-and-tags-in-wordpress/> (pristupljeno:2.9.2021.)

- [14] M. Pintarić: Izrada web trgovine u Wordpressu, Završni rad, Sveučilište Sjever, Varaždin, 2018.
- [15] I. Ilišević: Sustav za upravljanje bazama podataka MariaDB, Završni rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2019.
- [16] M. Kasa: Web aplikacija za fotografski studio, Završni rad, FERIT, Osijek, 2017.
- [17] WordPress: Reference Guide for Students, Office of Technology Services, Towson University, 2019. Dostupno na: <https://www.towson.edu/technology/training/resources/documents/individualwebsites/wp03-wordpress-reference-guide-for-students.pdf> (pristupljeno:2.8.2021.)
- [18] A. Helander: Building a WordPress site for Sports Academy of Kouvola, završni rad, Haaga-Helia University of Applied Sciences, Helsinki, 2015.
- [19] B. Williams, D. Damstra, H. Stern: Professional WordPress, Design and Development, John Wiley & Sons, Inc., Indianapolis,2013.
- [20] M. MacDonald: WordPress – the Missing Manual, O'Reilly Media, Inc., Sebastopol, 2014.
- [21] D. Hayes: The WordPress Template Hierarchy: What, Why and How. 2017. Dostupno na: <https://wphierarchy.com> (pristupljeno: 2.8.2021.)
- [22] K. Muldoon: The WordPress Loop Explained For Beginners. 2014. Dostupno na: <https://www.elegantthemes.com/blog/tips-tricks/the-wordpress-loop-explained-for-beginners> (pristupljeno: 2.8.2021.)
- [23] A. Šimec, B. Grilec, L. Tepeš Golubić: Razvoj prilagođenog predloška za WordPress sustav, Tehničko veleučilište u Zagrebu, Zagreb 2019.
- [24] M. Cvitković: Osnove rada u Wordpressu, Sveučilišni računski centar, Sveučilište u Zagrebu, Zagreb 2021.
- [25] A. Hercigonja: WordPress Widgets vs. Plugins: What's the Difference?, 2020. Dostupno na: <https://codeinteractive.com/magazine/wordpress-widgets-vs-plugins/> (pristupljeno 2.8.2021.)
- [26] E. Wallace: Introducing Vector Networks, Figma, 2016. Dostupno na: <https://www.figma.com/blog/introducing-vector-networks/> (pristupljeno 2.9.2021.)

- [27] M. Fumić: Primjena tehnologija web programiranja u izradi web stranice rukometnog kluba, Završni rad, Sveučilište Josipa Jurja Strossmayera u Osijeku, FERIT, Osijek, 2020.
- [28] A. Hrastić: Programski dodaci za Bootstrap, Završni rad, Sveučilište Sjever, Varaždin, 2016.
- [29] D. Hmorinski: Razvoj web stranice korištenjem Bootstrapa, Završni rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2018.
- [30] J. Casabona: Building WordPress Themes from Scratch, Rockable Press, New York 2012.
- [31] WordPress: Functions – get_search_form,
Dostupno na:
https://developer.wordpress.org/reference/functions/get_search_form/
(pristupljeno 2.9.2021.)
- [32] WordPress: Functionality – Custom Logo,
Dostupno na: <https://developer.wordpress.org/themes/functionality/custom-logo/> (pristupljeno 2.9.2021.)
- [33] Info&Apps: How do I Enable Navigation Menu in WordPress, 2020.
Dostupno na: <https://www.infoandapps.com/blog/how-do-i-add-a-menu-to-appear-in-wordpress/> (pristupljeno 2.9.2021.)
- [34] S. Price: How to Add Sliders to Your WordPress Site With MotoPress Slider Plugin, 2019. Dostupno na: <https://blog.hubspot.com/website/motopress-slider-plugin>, (pristupljeno 2.9.2021.)
- [35] G. Romanato: WordPress: the get_posts() function in depth, 2012. Dostupno na: https://gabrieleromanato.name/wordpress-the-get_posts-function-in-depth, (pristupljeno 2.9.2021.)
- [36] J. Locke: How To Add Dynamic Sidebars To WordPress, 2020. Dostupno na: <https://www.lockedownseo.com/add-dynamic-sidebars-to-wordpress/>, (pristupljeno 3.9.2021.)
- [37] wptom: WordPress Custom Header theme feature, 2020. Dostupno na: <http://wptom.com/chapter/wordpress-custom-header/>, (pristupljeno 3.9.2021.)
- [38] M. Jones: How to Set Up Contact Form 7 (For Beginners), 2020. Dostupno na: <https://themeisle.com/blog/how-to-set-up-contact-form-7/>, (pristupljeno 3.9.2021.)

- [39] M. Zvonković: Kontaktni obrazac (Contact Form 7), 2015. Dostupno na: <http://knjiznice.nsk.hr/kontaktni-obrazac-contact-form-7/>, (pristupljeno 3.9.2021.)
- [40] K. Knox: What is Google Cloud Platform (GCP)?, 2021. Dostupno na: <https://acloudguru.com/blog/engineering/what-is-google-cloud-platform-gcp>, (pristupljeno 3.9.2021.)
- [41] Google Cloud Platform: Maps Embed API, 2021. Dostupno na: <https://console.cloud.google.com/apis/library/maps-backend.googleapis.com>, (pristupljeno 3.9.2021.)

Popis slika

Slika 1.	Shematski prikaz sastavljanja stranice, Izvor: G.Plumley: Wordpress, Dobar plan, Zagreb 2012.	4
Slika 2.	XAMPP Kontrolna ploča, uključeni Apache i MySQL.....	7
Slika 3.	Kreiranje nove baze podataka u phpMyAdmin alatu.....	8
Slika 4.	Sve raspakirane datoteke sadržane u wp mapi.....	8
Slika 5.	Potrebni podaci za ispravnu WordPress instalaciju	9
Slika 6.	Instalacijski obrazac WordPress-a	10
Slika 7.	Prijava u WordPress.....	10
Slika 8.	Početna stranica administratorskog sučelja WordPress programa.....	11
Slika 9.	Izrada nove objave	12
Slika 10.	Izrada nove kategorije	12
Slika 11.	Pridruživanje kategorije i dodavanje oznaka novoj objavi	13
Slika 12.	Dodavanje nove stranice u WordPressu	13
Slika 13.	Raspored pojedinih datoteka predložaka na jednoj WordPress stranici. Izvor: M. MacDonald: WordPress – the Missing Manual, O’Reilly Media, Inc., Sebastopol, 2014.....	14
Slika 14.	Ilustrirani prikaz hijerarhije predložaka početne stranice.	16
Slika 15.	Smještaj petlje u osnovnom WordPress predlošku , Izvor: B. Williams, D. Damstra, H. Stern: Professional WordPress, Design and Development, John Wiley & Sons, Inc., Indianapolis,2013.....	17
Slika 16.	Osnovna petlja za dohvaćanje postova	17
Slika 17.	Ilustracija velikog broja petlji unutar WordPress predloška , Izvor: B. Williams, D. Damstra, H. Stern: Professional WordPress, Design and Development, John Wiley & Sons, Inc., Indianapolis,2013.	18
Slika 18.	Ilustrirani prikaz strukture web stranice.....	20
Slika 19.	Dizajnersko rješenje za veće zaslone izrađeno pomoću Figma	21
Slika 20.	Primjer rasporeda elemenata pomoću Bootstrap grida	22
Slika 21.	Komentar unutar style.css datoteke	23
Slika 22.	Aktivacija vlastite teme.....	23
Slika 23.	Dodavanje datoteka stila (.css) i skripti (.js)	24
Slika 24.	Sadržaj unutar <head> oznake	25

Slika 25.	Sekcije unutar <header> oznake	26
Slika 26.	Registriranje sidebara Društvene mreže.....	27
Slika 27.	Gotovi programčić Društvene mreže na kojem je primijenjen Social Icons Block	27
Slika 28.	Petlja koja izvodi funkciju dynamic_sidebar	27
Slika 29.	Funkcija za dodavanje obrasca za pretraživanje	28
Slika 30.	Funkcija za dodavanje logotipova.....	28
Slika 31.	Stavka Identitet stranice pomoću koje se dodaje logotip	28
Slika 32.	Funkcija koja prikazuje logotip na predlošku	29
Slika 33.	Pregled svih stranica u WordPressu	29
Slika 34.	Registriranje izbornika u functions.php datoteci	29
Slika 35.	Podešavanje strukture izbornika	30
Slika 36.	Funkcija za prikaz izbornika u predlošku	30
Slika 37.	CSS kod za oblikovanje zaglavlja.....	31
Slika 38.	Konačan izgled zaglavlja	31
Slika 39.	Ispisan kod footer.php datoteke	32
Slika 40.	Ispisana CSS sintaksa koji se nalazi u template.css datoteci	32
Slika 41.	Konačan izgled podnožja u pregledniku	32
Slika 42.	Struktura elemenata predloška page-pocetna.php.....	33
Slika 43.	Stavka MotoPress Slider	34
Slika 44.	Podešavanje postavki klizača	34
Slika 45.	Podešavanje veličine klizača za razne uređaje.....	34
Slika 46.	Opcija Edit Slides.....	35
Slika 47.	Popis kreiranih klizača	35
Slika 48.	Ubacivanje funkcije za prikaz klizača.....	36
Slika 49.	Konačan izgled klizača na početnoj stranici	36
Slika 50.	Prilagođena petlja za prikaz najnovijih objava	37
Slika 51.	Struktura objave i funkcije koje se koriste za prikaz određenih elemenata	38
Slika 52.	Funkcija za prikaz petlje	38
Slika 53.	Konačan izgled petlje s objavama.....	39
Slika 54.	Registriranje novog programčića pod nazivom 'Župni listić'.....	39
Slika 55.	Gotovi programčići s dodanim dokumentima i tekstem	40
Slika 56.	Funkcije za prikaz pojedinih elemenata bočnog stupca.....	40
Slika 57.	CSS kod za oblikovanje izgleda glavnog i bočnog stupca.....	41
Slika 58.	Konačan izgled glavnog i bočnog stupca.....	41

Slika 59.	Registriranje novog programčića pod nazivom "Župna crkva"	41
Slika 60.	Petlja kojom se prikazuje sadržaj programčića "Župna crkva"	42
Slika 61.	Ispunjavanje programčića "Župna crkva"	42
Slika 62.	Konačan izgled stranice "O župi"	42
Slika 63.	Dodavanje funkcije slike zaglavlja	43
Slika 64.	WordPress kod za sliku zaglavlja	43
Slika 65.	Prilagođavanje slike zaglavlja unutar WordPressa	43
Slika 66.	Popunjavanje polja za kontakt obrazac	44
Slika 67.	Prilagođene poruke kontakt obrasca	45
Slika 68.	Registriranje kontakt obrasca	45
Slika 69.	Petlja kojom se prikazuje kontakt obrazac	46
Slika 70.	Ubacivanje kontakt obrasca u programčić	46
Slika 71.	Dodatni CSS i konačan izgled kontakt obrasca	46
Slika 72.	Kreiranje novog projekta na Google Cloud Platformi	47
Slika 73.	Generiranje novog API ključa	47
Slika 74.	Ugrađen API za interaktivne karte	47
Slika 75.	Ubacivanje JavaScript koda u predložak	48
Slika 76.	Konačan izgled 'Kontakt' stranice	48

Popis tablica

Tablica 1.	Opis pojedinih stavaka WordPress-a i njihovih mogućnosti	11
------------	--	----

Sveučilište
Sjever

MARKON
AA188BAINO



SVEUČILIŠTE
SJEVER

**IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU**

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Matea Jelović (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Temu i predlozi za WordPress (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Matea Jelović
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Matea Jelović (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Temu i predlozi za WordPress (upisati naslov) čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

Matea Jelović