

Razvoj prilagođenog predloška za WordPress sustav

Alen Šimec

Tehničko veleučilište u Zagrebu, Vrbik 8, 10 000 Zagreb, Hrvatska

alen@tvz.hr

Borna Grilec

Veleučilište Velika Gorica, Zagrebačka ul. 5, 10410 Velika Gorica, Hrvatska

bornagrilec@gmail.com

Lidija Tepeš Golubić

Tehničko veleučilište u Zagrebu, Vrbik 8, 10 000 Zagreb, Hrvatska

ltepes2@tvz.hr

Ključne riječi

WordPress ideja i razvoj, mogućnosti prilagođenog WordPress predloška, WordPress dodaci, razvoj WordPress predloška

Sažetak

U ovom radu objasnit će se ideja i razvoj WordPress sustava za upravljanje sadržajem. Ukazat će se na prednost i mogućnosti korištenja prilagođenog predloška za specifično web rješenje. Spomenut će se na koji način dodaci (plugins) mogu olakšati i poboljšati web stranicu. Prikazat će se osnovne faze razvoja prilagođenog predloška koji se sastoji od prilagođenih polja, bloga i navigacije kojima će u krajnosti upravljati klijent za kojega se izrađuje web stranica i to putem WordPress sučelja.

1. Uvod

U razdoblju u kojem se svako poduzeće i način poslovanja mora digitalizirati zbog konkurentnosti na tržištu, potrebno je stvarati web rješenja koja će poduzećima i korisnicima to zbilja i omogućiti. Mogućnosti koje otvara Internet primjenjive su za svakoga, no još uvijek mnogi ne znaju kako i s čime nastupiti na webu.

Internet je često od presudne važnosti za uspješnost poslovanja, ali nekvalitetnim i polovičnim rješenjima dešava se da ulaganje u Internet poslovanje ne daje očekivane rezultate. Česta pogreška pojedinaca i poduzeća je u tome što traže izradu rješenja u brzom vremenskom roku, a traži li se nešto specifično onda je nemoguće cijelu strategiju nastupa na Internetu stvoriti u par dana, a pri tome dizajnirati, programirati i optimizirati rješenje.

Za slučajeve malih i srednjih poduzeća često je moguće stvoriti kvalitetna rješenja korištenjem nekih već gotovih sustava u kombinaciji s pažljivom prilagodbom rješenja. Isto tako, takvim rješenjima moguće je postići smanjenje vremena izrade rješenja, a da pritom kvaliteta ne strada, a na kraju se postigne optimalno rješenje za klijenta. Primjer takvog sustava je WordPress sustav za upravljanjem sadržajem čije gotovo beskonačne mogućnosti prilagodbe mogu poslužiti u svrhu izrade optimalnog rješenja u kraćem vremenskom periodu, na način da se samom klijentu omogući upravljanje sadržajem putem sučelja koje je user-friendly.

U prvom dijelu rada ukratko će se opisati kako je došlo do razvoja WordPressa te koja je kompletna ideja sustava za upravljanje sadržajem. Također će se pričati o ključnim značajkama WordPress sustava. Drugi dio pokrit će temu važnosti izrade prilagođenih predložaka i

njihove prednosti i mogućnosti. Zatim će se ukazati na iznimnu korist dodataka (plugins). U predzadnjem poglavlju pokušat će se pobliže objasniti razvoj prilagođenog predloška, ali od faze kada već postoji osnovna stranica. Dakle, osnovna stranica prilagodit će se na način da može funkcionirati s WordPressom te dodatno prilagođavati putem sučelja WordPressa. Završetak ovog rada opisat će se zaključkom na temelju ovog rada.

2. Ideja i razvoj WordPress sustava

WordPress sustav stvoren je s ciljem da korisnicima olakša način objavljivanja sadržaja online. Vraćanjem u razdoblje kad sustavi za upravljanje sadržajem nisu bili toliko moćni i pojednostavljeni za rad, nailazilo se na ne baš funkcionalne sustave. Tako se je u razdoblju 2000. godine radilo na razvoju različitih blog platformi od strane individualaca. U lipnju 2001. godine Michael Valdrighi počeo je razvijati blog sustav pod nazivom b2 koji je bio PHP i MySQL alternativa tadašnjim poznatim platformama Blogger i Greymatter.¹ Želja je bila integrirati ugrađeni sustav za komentiranje, dobro upravljanje korisnicima s kompletnim korisničkim profilima te drugačiji načini arhiviranja blogova. Također, ideja je bila stvoriti sustav s jednostavnom instalacijom koji će biti jednostavan za korištenje i brz po pitanju objavljivanja sadržaja. Razvoj blog sustava b2 uspio je, no ubrzo se moglo primijetiti mnogo mana. Zbog jednostavnosti je sustav bio poprilično zastupljen, čak se je stvorila i zajednica korisnika koji su konstantno davali povratne informacije kako bi se radilo na poboljšanju sustava. Michael Valdrighi stalno je usavršavao b2, sve dok nije dobio otkaz u firmi u kojoj je bio zaposlen. Nakon toga gotovo da mu se gubi svaki trag, što je utjecalo na prestanak razvoja b2 blog platforme.

Kako bilo, b2 platforma pomogla je spojiti dvije osobe koje su zaslužne za razvoj WordPressa, Mike Little i Matt Mullenweg. Oni su iskoristili otvoreni kod b2 platforme i konstantno radili na poboljšanjima kako bi svoje blogove učinili boljima. Prva inačica WordPressa

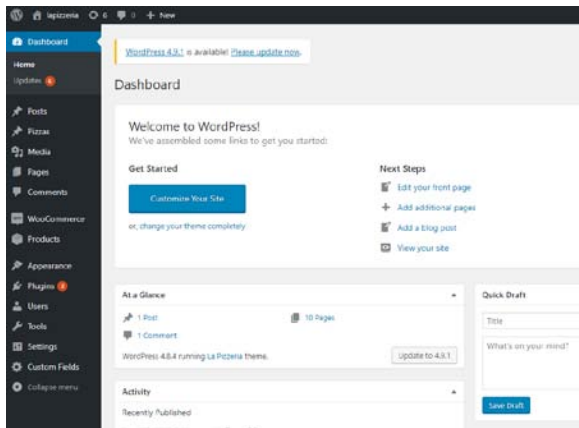
nastala je 1. travnja 2003. godine i objavio ju je Matt putem SourceForgea.²

Ubrzo se WordPress zajednica počela širiti, a samim time i WordPress slijedeći principe jednostavnosti, lakog korištenja i usklađenosti s web standardima. Nakon određenog vremena opet se pojavio Michael Valdrighi, osnivač b2 blog platforme koji je svoj doprinos dao i WordPressu. Od tada pa još uvijek traje razvoj WordPressa, zato je i popularnost i korisnost ove platforme toliko velika. Konstantnim se unaprjeđenjima stvorila platforma čije su mogućnosti nadišle početnu ideju bloga.

Tokom godina WordPress je prerastao u sustav za upravljanje sadržajem (CMS-Content Management System) korišten na milijunima stranica i viđen od desetaka milijuna korisnika svakodnevno. WordPress je omogućio jednostavnost brzog objavljivanja, fleksibilnost u smislu kreiranja osobnog bloga, foto bloga, poslovne web stranice, profesionalnoga portfolija, medijskih portala, čak i web trgovina korištenjem dodataka (*plugins*). Dao je veliki doprinos u smislu širenja informacija i stvaranja online zajednica te na taj način otvorio vrata malom čovjeku da se njegove ideje i misli prošire diljem planete. Jedne od ključnih značajki WordPressa svakako su odlične mogućnosti objave sadržaja, optimizacija, upravljanje korisnicima te različitim ulogama korisnika, upravljanje medijskim zapisima, korištenje sustava za komentiranje i prijevod na više od 70 jezika kako bi se rad sa sustavom maksimalno pojednostavio svakome. Mogućnosti personalizacije WordPressa visoke su, tako je moguće koristiti već gotove teme (predloške), instalirati dodatke (*plugins*) koji jednim klikom mogu izmijeniti određene funkcionalnosti i web stranicu učiniti još boljom uz samo par klikova. Može se primijetiti kako su mogućnosti gotovo neograničene, no uvijek treba uzeti u obzir pruža li WordPress dovoljne performanse za korisničke potrebe, oko čega treba obratiti pažnju prilikom početka razvoja web rješenja. Kako bilo, može se pretpostaviti kako će WordPress još jedno dugo vrijeme biti zastupljen ako se nastavi razvijati u smjeru kao dosad. Slika 1 prikazuje izgled početnog sučelja WordPress sustava.

¹ Milestones: The Story of WordPress; str. 21; Wordpress.

² Milestones: The Story of WordPress; str. 34; Wordpress.



Slika 1. Izgled početnog korisničkog sučelja WordPressa.

3. Prednost i mogućnosti prilagođenog predloška

U prethodnom poglavlju spomenuto je kako WordPress omogućava korištenje već gotovih predložaka, od čega su dva standardna unutar temeljne instalacije. Osim toga moguće je pristupiti repozitoriju prepunom gotovih predložaka (besplatnih i onih s plaćanjem) koji mogu odgovarati potrebama određenog poduzeća, grupe ili pojedinca. Jedno od čestih pitanja je: „Kako odlučiti odgovara li određenom korisniku jedan od gotovih ili besplatnih predložaka?“. Kako bi se došlo do odgovora potrebno je upoznati brand koji će svoju priču ispričati online korisnicima, a to iziskuje dobro poznavanje ciljane skupine, prirode onoga što će se prezentirati na webu te budžetu. Ako korisnik želi ispričati svoju jedinstvenu priču, velika je šansa da mu gotov predložak neće davati najbolje rezultate za definirane ciljeve. Osim toga, kada se priča o prilagođenom predlošku mnogi razmišljaju samo o izgledu predloška, a ne i o mogućnostima povezivanja određenog dijela predloška sa samim sustavom za upravljanje sadržajem. To znači da je određenu sekciju ili element prilagođenog predloška moći uređivati i modificirati putem WordPress sučelja bez da administrator stranice mora ulaziti u programski kod.

Znanje o razvoju prilagođenog predloška danas je od ključne važnosti. Popularizacija Interneta dignula je cijelu populaciju i omogućila da svako plasira nešto svoje. Bez prilagođenih rješenja mogućnost uspješnosti određenog projekta drastično se smanjuje. Uzme li se za primjer poduzeće koje želi ponuditi svoje usluge putem Interneta, prezentirati svoje projekte te

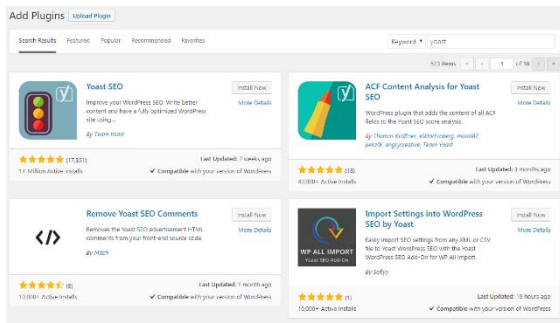
pričati o svojim projektima u svrhu educiranja i prikupljanja klijenata, potrebno je kreirati prilagođeni predložak. Krene li se s pretpostavkom da će poduzeće kroz određeno vrijeme nuditi još drugačijih usluga, da će se slike u određenim sekcijama mijenjati novima ili broj završenih projekata povećati, potrebno je prilagoditi cijelo rješenje kako bi vlasnik poduzeća kroz *user-friendly* način sam mogao raditi potrebne izmjene, bez potrebe za poznavanjem kodova.

Dakle, prednosti prilagođenog predloška svakako su mogućnost dizajniranja specifičnog predloška za određeni brand i mogućnost upravljanja određenim dijelom predloška pomoću sučelja WordPress sustava za upravljanje sadržajem. Na taj način postiže se maksimalna prilagodba koju može pružiti sustav poput WordPressa. Nadalje, to utječe na mogućnosti povećanja svijesti o brandu, prodaje usluga i proizvoda, komunikacije s online korisnicima te uštede vremena za vlasnika poduzeća, što u krajnosti donosi veće prihode ako se rješenje izvede kvalitetno.

4. Važnost korištenja dodataka (plugins)

Osnovnu priču vezanu uz prilagodbu web rješenja definitivno zaokružuje korištenje dodataka (*plugins*). Dodaci zapravo predstavljaju dodatne komponente koje se mogu implementirati u određeno web rješenje. Te dodatne komponente nisu ništa drugo nego programski kodovi koje pišu programeri za neke svoje specifične aplikacije ili slučajeve. Vrlo često se na webu koriste slične komponente na više stranica ili aplikacija, a WordPress je u tom pogledu napravio platformu koja svojim korisnicima omogućava korištenje tuđih komponenti ako su izrađene prema standardima i arhitekturi koju zahtjeva WordPress.

Dodaci se kao i predlošci nalaze u repozitoriju kojega je vrlo jednostavno pretraživati, a sama instalacija i aktivacija doslovno zahtijeva samo nekoliko klikova. Kao i predlošci, tako i dodaci postoje plaćeni i besplatni. U većini slučajeva korisnik će pronaći besplatan dodatak (*plugin*) koji odgovara zahtjevima. Slika 2 prikazuje dio dostupnih dodataka (*plugins*).



Slika 2. Prikaz dodatka (plugins) unutar WordPressa.

Cijelu priču vezanu uz dodatke moguće je izbjeći, no bilo bi šteta razvijati rješenja koja već postoje i na taj način trošiti vrijeme. Neki od neizostavnih dodataka svakako su dodaci za optimiziranje sadržaja za tražilice, gotovi kontakt obrasci te dodaci za kreiranje prilagođenih polja unutar sučelja kako bi se ostvarilo lako povezivanje s predložkom. Naravno, postoji ih još, te njihov izbor ovisi o konkretnom rješenju. Važno je imati u vidu da više dodataka rezultira usporavanjem i zakrčenjem stranice, tako da je uvijek važno odabrati optimalan broj dodataka.

5. Faze razvoja prilagođenog predloška

5.1. Osposobljavanje WordPressa

Kako bi se mogao koristiti WordPress, prvo je potrebno skinuti WordPress sustav sa službene web stranice. WordPress je besplatan i moguće ga je skinuti s navedene poveznice: <https://wordpress.org/download/>. U ovom projektu koristit će se xampp kao okruženje za razvoj web stranice. Nakon osposobljenog xampp okruženja, potrebno je raspakirati skinutu zip datoteku WordPressa i premjestiti je u „htdocs“ mapu unutar xamppa. U ovom primjeru raspakiran WordPress nazvat će se „elkopprojekt“. Nakon toga potrebno je kreirati bazu podataka unutar phpmyadmina. Naziv baze podataka bit će „elkop_wp“, naziv naravno može biti proizvoljan. Sljedeći korak je instalacija WordPressa koja se pokreće tako da se unutar polja za pretraživanje u pretraživaču upiše: „localhost/elkopprojekt“, u slučaju da je naziv raspakirane WordPress datoteke unutar mape „htdocs“ drugačiji tada se umjesto „elkopprojekt“ upisuje neki drugi naziv. Sada kreće postupak instalacije unutar pretraživača, gdje se u prvom koraku odabire željeni jezik instalacije. Zatim se upisuju naziv prethodno stvorene baze podataka, korisnik i lozinka baze podataka te server i željeni prefiks tablica. Sve prikazuje slika 3.

Below you should enter your database connection details. If you're not sure about these, contact your hosting provider.

Database Name The name of the database you wish to use with WordPress.

Username Your database username.

Password Your database password.

Database Host You should be able to get this information from your hosting provider. If localhost doesn't work, you may need to enter the IP address of your database host.

Table Prefix If you want to run multiple WordPress installations in a single database, you can specify a table prefix. The prefix should be at least 4 characters long and should not contain any spaces or special characters.

Slika 3. Povezivanje baze podataka s WordPressom.

Nadalje, potrebno je unijeti podatke o naslovu stranice, korisničkom imenu i lozinki za pristup sučelju WordPressa i upisati email adresu. Važno je da se svi podaci u postupku instalacije upamte jer će biti potrebni za daljnji rad. Slika 4 prikazuje podatke za ovaj projekt, oni se mogu razlikovati ovisno o projektu koji se razvija.

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

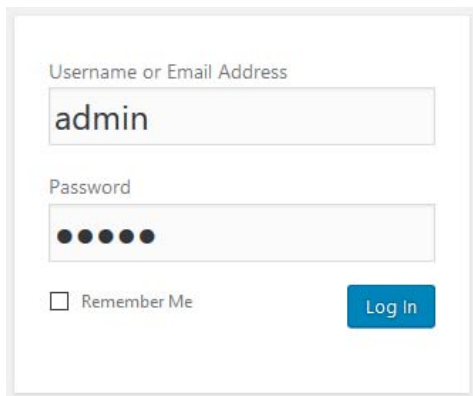
Confirm Password Confirm use of weak password

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 4. Unos detalja web stranice i definiranje korisnika sučelja WordPressa.

Po uspješnoj instalaciji moguće se je prijaviti u sučelje WordPressa pomoću podataka koju su prethodno definirani. Slika 5 prikazuje obrazac za prijavu, kojemu se uvijek može pristupiti ako se unutar polja za pretraživanje upiše: „localhost/elkopprojekt/wp-admin“.



Slika 5. Obrazac za prijavu u administratorsko sučelje WordPressa.

5.2. Kreiranje novog predloška

Kreiranje novog predloška vrši se unutar instalacije WordPressa na serveru. Potrebno se je pozicionirati u „wp-content/themes“ i tamo kreirati novu mapu proizvoljnog naziva predloška, u ovom slučaju „elkop“. Nakon toga slijedi stvaranje dva osnovna dokumenta, a to su style.css i index.php. Kod 1 prikazuje izgled style.css dokumenta. Svaki style.css dokument mora imati: definiran naziv teme (predloška), poveznicu do teme na webu, autora, poveznicu do profila autora, opis predloška, verziju teme, licencu i poveznicu do licence, oznake (tagove) teme i domenu teme.

```

/*
Theme Name: Elkop projekt
Theme URI:
Author: Borna Grilec
Author URI:
Description: Predlozak za Elkop
projekt
Version: 1.0
License: GNU General Public License
v2 or later
License URI:
http://www.gnu.org/licenses/gpl-2.0.html
Tags: elektroinstalacije, automatika,
responzivnost
Text Domain: elkop
*/

```

Kod 1. Neizostavni podaci na početku style.css dokumenta.

5.3. Stvaranje zaglavlja i podnožja

Budući da će više stranica koristiti isto zaglavlje i podnožje, WordPress omogućuje kreiranje zasebnih datoteka header.php i

footer.php koje se po potrebi mogu pozivati na mjestima gdje je to potrebno. Kod 2 prikazuje izgled header.php datoteke koja se u dijelu stranice koja treba koristiti kod zaglavlja može pozvati pomoću već ugrađene funkcije get_header().

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Elkop projekt</title>
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <?php wp_head(); ?>
  </head>
  <body>
    <nav class="nav-container">
  </nav>

```

Kod 2. Mogući izgled zaglavlja (header.php).

Osim zaglavlja, još jedan specifičan dokument koji se najčešće koristi više puta na web stranici je podnožje. Podnožje se sprema u footer.php dokument. Kod 3 prikazuje footer.php dokument koji se u dijelovima stranice koje ga zahtijevaju uključuje pomoću funkcije get_footer().

```

<footer>
  <p>Copyright - 2018. Elkop
projekt</p>
</footer>
<?php wp_footer(); ?>
</body>
</html>

```

Kod 3. Mogući izgled podnožja (footer.php).

Važno je napomenuti kako trenutni izgled zaglavlja i podnožja nije finalni, već služi kao primjer što sve mogu sadržavati ovi dokumenti. S vremenom, kako se projekt razvija dodavat će se novi strukturni elementi.

5.4. Korištenje functions.php datoteke

U slučaju kada je potrebno definirati programski kod koji će uvijek biti na raspolaganju predlošku ili web stranici potrebno je koristiti functions.php datoteku.³ Unutar nje

³ Williams, B; Damastra, D; Stern, H: Professional WordPress: Design and Development, Second Edition, John Wiley & Sons, 2013.

najčešće se sprema programski kod koji može uključivati različite stilove, skripte, navigacijske izbornike i slično. Kod 4 prikazuje functions.php datoteku i primjer uključivanja različitih stilova i skripti. Unutar funkcije `elkop_styles()` definiraju se različiti stilovi i to pomoću funkcije `wp_register_style()`, kojoj se redom moraju dodijeliti proizvoljno ime, poveznica do datoteke na serveru, polje u kojem se definira stil koji će se uključiti prije te verzija dokumenta koji se uključuje. Kako bi se uspješno dodao stil potrebno je funkciji `wp_enqueue_style()` predati parametar naziva prethodno definiranog stila. Sličan postupak je i za skripte, samo se kod njih koriste funkcije `wp_register_script()` i `wp_enqueue_script()`. Za uspješno korištenje funkcije `elkop_styles()` na kraju je potrebno dodati funkciju `add_action()`.

```
<?php
function elkop_styles(){
    wp_register_style('normalize',
get_template_directory_uri() .
'/css/normalize.css', array(), '7.0.0');
    wp_register_style('layout',
get_template_directory_uri() .
'/css/layout.css', array('normalize'),
'1.0');
    wp_register_style('style',
get_template_directory_uri() .
'/style.css', array('layout'), '1.0');

    wp_enqueue_style('font',
'https://fonts.googleapis.com/css?family=
Palanquin:100,300,400,500', false);
    wp_enqueue_style('normalize');
    wp_enqueue_style('layout');
    wp_enqueue_style('style');

    wp_register_script('script',
get_template_directory_uri() .
'/js/script.js', array('jquery'),
'1.0.0', true);

    wp_enqueue_script('jquery');
    wp_enqueue_script('script');
}
add_action('wp_enqueue_scripts',
'elkop_styles');
?>
```

Kod 4. Mogući izgled functions.php datoteke.

5.5. Dodavanje navigacijskog izbornika

Navigacijski izbornik moguće je povezati sa sučeljem WordPressa. Na taj način izbornik je moguće izmjenjivati unutar samog sučelja bez

potrebe za direktnim modifikacijama u HTML kodu. Kako bi se to ostvarilo prvi korak je registracija novog izbornika unutar functions.php dokumenta. Kod 5 prikazuje registraciju novog izbornika za željenu web stranicu. Na početku se definira funkcija `elkop_menus()` i na kraj se dodaje funkcija `add_action()` koja prilikom inicijalizacije uključuje izbornik. Parametar `init` definira inicijalizaciju. Potom se pomoću ugrađene funkcije `register_nav_menus()` omogućava kreiranje izbornika unutar polja. Parametar `'header-menu'` proizvoljni je naziv izbornika, zatim se definira uljepšani naziv `'Header Menu'` koji se prikazuje unutar WordPress sučelja i domena predložka `'elkop'`.

```
function elkop_menus(){
    register_nav_menus(
        array(
            'header-menu' => __('Header Menu',
'elkop')
        )
    );
}
add_action('init', 'elkop_menus');
```

Kod 5. Registracija navigacijskog izbornika unutar functions.php dokumenta.

U slučaju ovog projekta postojat će samo jedna zasebna stranica na koju će upućivati navigacija, a ostale poveznice usmjeravat će na određene sekcije na početnoj stranici. Za ostvarenje toga, prvo je potrebno dodati novu stranicu unutar WordPress sučelja. To je moguće učiniti klikom na **Pages**, zatim **Add New** i nazvati stranicu željenim nazivom. U slučaju ovog projekta nova stranica bit će „Naši projekti“ koja će sadržavati blog članke o provedenim projektima poduzeća. Nakon što se taj dio odradi i spremi se izmijenjen functions.php dokument unutar WordPress sučelja pod izbornikom **Appearance** pojavit će se opcija **Menus**. Ovdje je potrebno kreirati novi izbornik u kojega se dodaje prethodno dodana stranica „Naši projekti“ i nekoliko prilagođenih poveznica (*Custom Link*). Slika 6 prikazuje krajnji izgled izbornika unutar Wordpress sučelja.



Slika 6. Izgled navigacijskog izbornika unutar WordPress sučelja.

Još uvijek izbornik nije vidljiv u samom dizajnu web stranice. Da bi se to ostvarilo potrebno je napisati dodatni kod unutar HTML koda (dijela stranice u kojem se želi ispisati). Kod 6 prikazuje kod pomoću kojega je moguće ispisati navigacijski izbornik. Za početak je potrebno stvoriti PHP varijablu (u ovom slučaju \$args) koja će sadržavati polje. Parametru 'theme_location' predaje se naziv izbornika iz functions.php dokumenta, zatim se stvara željeni HTML element i dodjeljuje mu se klasa za kasnije dodatno stiliziranje. Pomoću funkcije wp_nav_menu() i predane varijable \$args ispisuje se izbornik.

```
<?php
$args = array(
    'theme_location' => 'header-menu',
    'container' => 'div',
    'container_class' => 'col-nine'
);
wp_nav_menu($args);
?>
```

Kod 6. Ispis navigacijskog izbornika pomoću PHP koda stranice.

Kod 7 prikazuje kako izgleda HTML kod za definirani izbornik kad se pogleda izvor web stranice. Elementima automatski se dodjeljuju još dodatne klase, no zbog skraćivanja koda nema potrebe da se ovdje prikazuje budući da ih WordPress sam dodjeljuje.

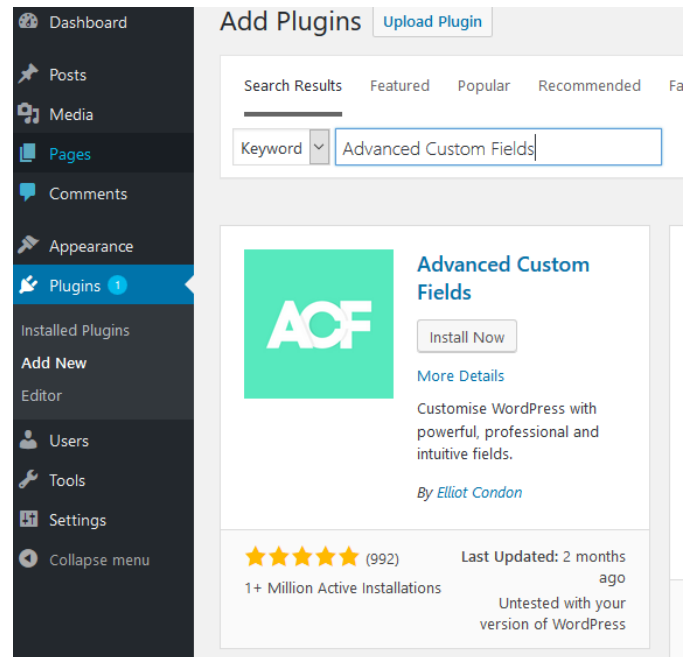
```
<div class="col-nine">
    <ul id="menu-header-menu"
    class="menu">
        <li>...</li>
        <li>...</li>
    </ul>
</div>
```

Kod 7. Izgled HTML koda definiranog izbornika.

5.6. Integracija prilagođenih polja

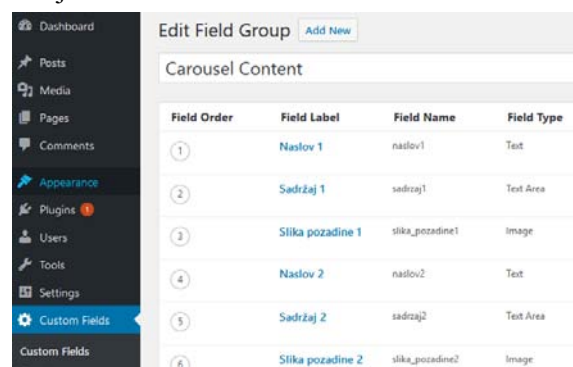
Često se predložak želi prilagoditi na način da administrator sam može upravljati sadržajem određenih dijelova web stranice bez da mora ulaziti u HTML kod. Pomoću WordPressa to je ostvarivo. U ovom poglavlju koristit će se dodatak (plugin) Advanced Custom Fields. Prvi korak je instalacija dodatka unutar WordPress sučelja tako da se klikne na izbornik **Plugins**, zatim **Add New** te se u tražilicu unutar sučelja upiše naziv željenog dodatka (Advanced Custom Fields). Slika 7 prikazuje sučelje za pretraživanje

i instalaciju dodataka. Klikom na **Install Now** pokreće se instalacija dodatka kojega je moguće koristiti nakon što se klikne na **Activate**.



Slika 7. Sučelje za pretraživanje i instalaciju dodataka (plugins).

Nakon aktivacije dodatka, unutar izbornika sučelja pojavit će se opcija **Custom Fields** na koju je potrebno kliknuti, a zatim odabrati **Add New**. Slika 8 prikazuje kreiranje novih polja tako da se na početku definira generalni naziv prilagođenih polja (u ovom slučaju Carousel Content), a zatim se odabiru nazivi i tipovi polja kakvi su potrebni za određeni slučaj.



Slika 8. Kreiranje novih polja korištenjem dodatka Advanced Custom Fields.

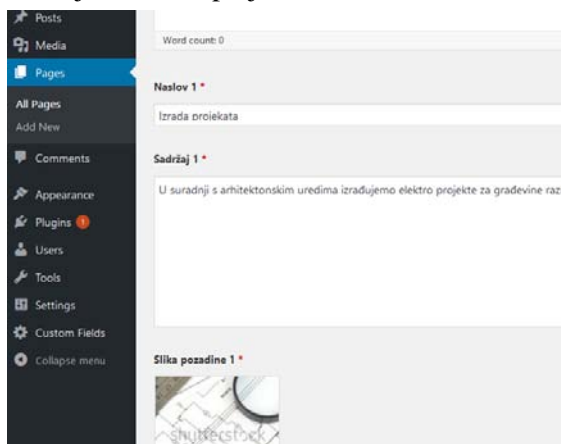
Ispod definiranih polja potrebno je odabrati gdje će se sadržaj prikazivati. Slika 9 prikazuje lokaciju novo stvorenih polja koja će biti smještena na početnoj stranici, odnosno na

stranici „O nama“. Nakon toga je potrebno spremati promjene.



Slika 9. Advanced Custom Fields lokacija polja.

Kako bi se dodao sadržaj pomoću stvorenih polja, u WordPress sučelju potrebno je izabrati opciju Pages iz glavnog izbornika, zatim kliknuti na uređivanje stranice „O nama“. Nakon otvaranja uređivanja stranice „O nama“ bit će vidljiva prethodno stvorena polja. Slika 10 prikazuje stvorena polja na stranici „O nama“.



Slika 10. Izgled stvorenih polja unutar uređivanja stranice „O nama“.

Zadnji dio uključuje ispis sadržaja i slika unutar HTML koda stranice „O nama“. Da bi se sadržaj i slike doista ispisali potrebno je pomoću ugrađenih WordPress funkcija dohvatiti unesene podatke. Tekstualni sadržaj moguće je dohvatiti pomoću funkcije `the_field()` kojoj se kao parametar predaje definirani naziv polja dodijeljen prilikom kreiranja polja, a slika se dohvaća pomoću funkcije `get_field()`, gdje se kao parametar također dohvaća naziv polja. Kod 8 prikazuje korištenje funkcija za dohvaćanje sadržaja i slika.

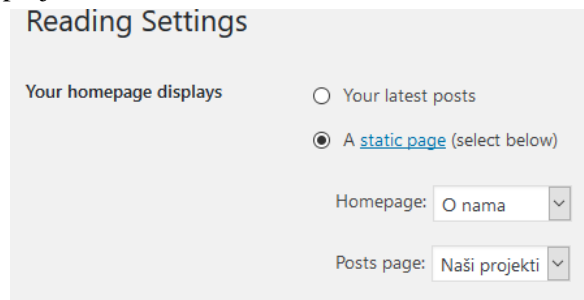
```

  <p class="carousel-text-
heading"><?php
the_field('naslov1');?></p>
  <p class="carousel-text-
content"><?php
the_field('sadržaj1');?></p>
```

Kod 8. Funkcije za dohvaćanje sadržaja i slika.

5.7. Osposobljavanje bloga

Blog ove web stranice bit će na stranici „Naši projekti“ na kojoj će se objavljivati detalji o provedenim projektima. Kako bi se to ostvarilo u WordPress sučelju potrebno je odabrati opciju **Settings**, zatim **Reading** te označiti *radio button* na „A static page (select below)“. Nakon toga potrebno je odabrati stranicu za postove (*Post page*). Slika 11 prikazuje odabir stranice za postove, u ovom slučaju odabire se „Naši projekti“.



Slika 11. Odabir stranice za postove.

Nakon toga potrebno je stvoriti nekoliko postova unutar WordPress sučelja. Kako bi se postovi doista mogli prikazati na web stranici, u `index.php` potrebno je napisati dodatni kod. Kod 9 prikazuje kod za ispis objavljenih članaka. Prvo se kreira `while` petlja koja pomoću ugrađene funkcije `have_posts()` učitava članke. Zatim se pomoću funkcije `the_post_thumbnail()` učitava glavna slika članka. Funkcija `the_title()` dohvaća naslov, potom `the_excerpt()` skraćeni sadržaj članka, a `the_permalink()` poveznicu na članak.

```
<section class="section-blog">
  <h2><?php echo
get_the_title($blog_page); ?></h2>
  <?php while
(have_posts()):the_post(); ?>
    <article class="col-six article-
container">
      <?php the_post_thumbnail(); ?>

      <h3 class="article-title"><?php
the_title(); ?></h3>
      <p class="article-
sazetak"><?php the_excerpt(); ?></p>
      <a href="<?php
the_permalink(); ?>">PROČITAJTE VIŠE</a>
    </article>
  <?php endwhile; ?>
</section>
```


Kod 9. Kod za ispis objavljenih članaka.

Osim ispisa svih blog članaka potrebno je moći ispisati i sadržaj zasebnog članka na zasebnoj stranici. Kako bi se to ostvarilo, potrebno je kreirati stranicu pod nazivom `single.php` jer nju WordPress tumači kao zasebnu stranicu za određeni članak. Nakon što se kreira stranica `single.php`, dodaje se kod za ispis sadržaja određenog članka. Kod 10 prikazuje na koji način se može ispisati sadržaj članka. Ispis je sličan ispisu svih članaka. Naslov se opet ispisuje pomoću funkcije `the_title()`, sadržaj korištenjem funkcije `the_content()`, a slika funkcijom `the_post_thumbnail()`.

```
<?php while (have_posts()):the_post(); ?>
  <article class="projekt-container">
    <h1><?php the_title(); ?></h1>
    <p class="opis-projekta"><?php
the_content(); ?></p>
<?php the_post_thumbnail(); ?>
  </article>
<?php endwhile; ?>
```

Kod 10. Ispis sadržaja zasebnog članka.

6. Zaključak

Temeljem ovog rada zaključuje se da WordPress doista omogućava visoku razinu prilagodbe web rješenja. Glavne prednosti svakako su jednostavnost korištenja sučelja od strane administratora web stranice i brzina razvoja web rješenja od strane programera. Prilikom izrade ovog rada moglo se naići na brojnu dokumentaciju koju nudi sam WordPress te stručnu literaturu koja objašnjava gotovo svaki korak razvoja predloška. U radu nisu objašnjene sve mogućnosti WordPressa, već je fokus na osnovnim značajkama jer WordPress može ponuditi puno više od samog bloga. Osobi koja poznaje tehnologije kao što su HTML, PHP, CSS i JavaScript, poznavanje WordPressa može biti od ključne važnosti za razvoj manjih i srednjih web rješenja upravo zbog brzine razvoja i odličnih mogućnosti optimizacije sadržaja za tražilice kao i korištenja već gotovih dodataka.

7. Literatura

[1] Milestones: The Story of WordPress; Wordpress.

- [2] Williams, B; Damastra, D; Stern, H: Proffesional WordPress: Design and Development, Second Edition, John Wiley & Sons, 2013.
- [3] Šimec A.; Nožica B.; Virtualizacija uredskog poslovanja; Tiskarstvo & Dizajn 2014.; ISBN 9789537064235
- [4] Filipova Olga; Learning Vue.js 2; Packt Publishing, 2016.
- [5] Šimec Alen, Davor Lozić; Extending PHP with modules; Polytechnic & Design; Vol. 3, No. 1, 2015.; ISSN: 1849 – 1995
- [6] Djirdeh, H; Murray, N; Lerner, A: Fullstack Vue: The Complete Guide to Vue.js; Fullstack.io, 2018.
- [7] Statcounter; URL: <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>; 20.1.2019.
- [8] Webopedia; URL: <https://www.webopedia.com/TERM/C/CDN.html>; 20.1.2019.
- [9] Wordpress; URL: <https://wordpress.com/notable-users>; 20.1.2019

