

Redizajn web stranica tvrtke Zimo digital

Mihalić, Vedran

Undergraduate thesis / Završni rad

2016

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:620782>

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

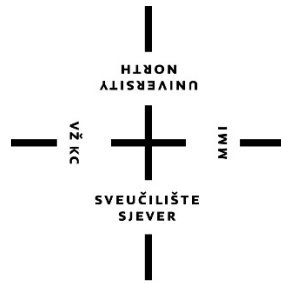
Download date / Datum preuzimanja: **2025-01-02**



Repository / Repozitorij:

[University North Digital Repository](#)





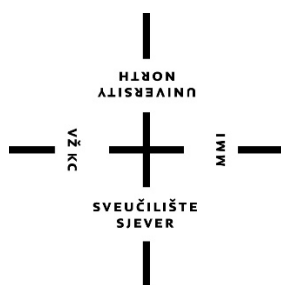
**Sveučilište
Sjever**

Završni rad br. 472/MM/2016

Redizajn web stranica tvrtke Zimo digital

Vedran Mihalić, 2823/601

Varaždin, rujan 2016. godine



Sveučilište Sjever

Odjel za multimediju, oblikovanje i primjenu

Završni rad br. 472/MM/2016

Redizajn web stranica tvrtke Zimo digital

Student

Vedran Mihalić, 2823/601

Mentor

izv. prof. dr. sc. Mario Tomiša

Varaždin, rujan 2016. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za multimediju, oblikovanje i primjenu		
PRISTUPNIK	Vedran Mihalić	MATIČNI BROJ	2823/601
DATUM	21.03.2016.	KOLEGIJ	Web dizajn
NASLOV RADA	Redizajn web stranica tvrtke Zimo digital		
NASLOV RADA NA ENGL. JEZIKU	Redesigning the Zimo digital company website		
MENTOR	dr.sc. Mario Tomiša	ZVANJE	Izvanredni profesor
ČLANOVI POVJERENSTVA	1. doc.dr.sc. Dean Valdec - predsjednik		
	2. pred. Snježana Ivančić Valenko, dipl.ing. - član		
	3. izv.prof.dr. Mario Tomiša - mentor		
	4. izv.prof.dr. Damir Vusić - zamjenski član		
	5. _____		

Zadatak završnog rada

BROJ 472/MM/2016

OPIS
Razlozi za redizajn web stranica uvijek su vezani uz osvježavanje postojećih web stranica, ali i razvoj novih funkcionalnosti, s ciljem povećanja broja posjetitelja web stranica te u konačnici i povećanja broja klijenata. Za dobar rezultat potrebno je stalno osvježavati izgled weba, dodavati nove funkcionalnosti i mogućnosti, ali i konstantno restrukturirati sadržaj, zbog bolje preglednosti web stranica i bolje optimizacije za pretraživače. U ovom radu definirat će se pojmovi koji objašnjavaju sam način izrade web stranice, pojasnit će se način stiliziranja koda te način spajanja na sustav za uređivanje sadržaja web stranice. Izradit će se redizajn web stranica tvrtke Zimo digital i istražiti zastupljenost sustava za uređivanje sadržaja. Nakon lansiranja stranice objavit će se i analiza o mišljenju korisnika glede novog dizajna te jednostavnosti pristupanju informacija na web stranici.

U radu je potrebno:

- Definirati HTML i stilske jezike
- Objasniti upotrebu stilskih jezika
- Definirati pojam sustava za uređivanje sadržaja
- Objasniti korištenje sustava, njegovih dodataka i predložaka
- Istražiti zastupljenost navedenog sustava
- Ispitati mišljenje korisnika o korištenom sustavu
- Izraditi redizajn web stranica tvrtke Zimo digital
- Dokumentirati i analizirati dijelove koda web stranica
- Analizirati prednosti novih web stranica nakon redizajna

ZADATAK URUČEN

30. 09. 2016



Handwritten signature of the mentor.

Predgovor

Razvoj tehnologija omogućio je ljudima sveprisutnost interneta, a samim time i weba, odnosno web stranice. Gotovo svako poduzeće posjeduje vlastitu web stranicu koje se koriste u razne svrhe. Tako se web stranice dijele na portfolio, osobno web mjesto, korporativni web, web shop, portal, društvenu mrežu, blog, ovisno o namjeni web stranice. Jedan od najčešćih i najkvalitetnijih oblika marketinškog oglašavanja su upravo web oglašavanja, odnosno web stranice.

Kako bi se ostvarilo pozitivno korisničko iskustvo te kako bi se web stranica u potpunosti ostvarila i opravdala svoju postojanost, potrebne su česte promjene da bi bila u korak sa standardima i pravilima.

Najveće prednosti čestih unaprjeđenja jesu veća rangiranost na tražilicama, dok se korisnicima daje potpuno novi osjećaj što ih zaintrigira da se duže zadrže na samoj stranici.

Zahvale

Zahvaljujem svom mentoru Mariu Tomiši koji je uložio trud i strpljenje, te svojim stručnim i korisnim savjetima pridonio izradi ovog rada. Također, zahvaljujem se direktoru tvrtke Zimo digital, Goranu Mudrom koji je pratio izradu praktičnog djela rada i dao smjernice za sam redizajn web stranice.

Izjavljujem da sam ovaj rad izradio u potpunosti samostalno, pri čemu su svi navodi iz drugih izvora i tuđih radova jasno označeni. Rad sam izradio na temelju znanja, vještina i kompetencija stečenih tokom trogodišnjeg obrazovanja na Sveučilištu Sjever, a uz pomoć u radu navedene literature.

Sažetak

Britanski znanstvenik Tim Berners Lee u prosincu 1989. godine izrađuje dokument pod nazivom „Information management: A proposal“ putem kojeg iznosi molbu institutu CERN za odobrenjem i financiranjem projekta kojem je cilj kao krajnji rezultat razmjena hipertekstualnih datoteka. Početkom 1990. godine dobiva rješenje o odobrenju projekta i počinje s radom na projektu. Do 6. kolovoza 1991. godine izrađuje sve potrebne alate, tehnologije i infrastrukturu potrebnu za izradu i objavu web stranice te objavljuje prvu web stranicu. 1992. godine kongres Sjedinjenih Američkih Država donosi odluku da se internet može koristiti i u komercijalne svrhe te počinje nagli razvoj web tehnologija i web stranica.

U razdoblju od objave prve web stranice do danas razvijeni su mnogi alati, njihove inačice kao i tehnologije za izradu i objavu web stranica, što je dovelo do objave velikog broja web sadržaja posljednjih godina, a web stranice su postale nezaobilazan dio marketinga i oglašavanja, ali i načina prodaje proizvoda i usluga putem weba. Isto tako razvoj tehnologija putem kojih se može pristupiti web sadržaju omogućuje da se sadržaju pristupa bez obzira na vrijeme i mjesto. Tako se sadržaj, osim putem računala, može pregledavati putem pametnih telefona, tableta, pametnih satova, televizora i slično.

Kako bi se privukao veći broj posjetitelja web stranica mora biti u potpunosti funkcionalna i ugodna za pregledavanje te mora zadovoljiti korisničko iskustvo kako bi posjetitelj ili kupio proizvod ili se ponovo vratio na istu. Osim toga, web stranice moraju biti prilagođene za ranije spomenute uređaje na način da se njihov prikaz i važnost prikazivanja elemenata ne razlikuje znatno među pojedinim uređajima.

Obzirom da danas postoji veliki broj sadržaja i da je postalo gotovo pa pravilo da svako poduzeće posjeduje vlastitu web stranicu bez obzira o kojoj vrsti web stranice se radilo, ona uvijek mora pratiti trend dizajna i funkcionalnosti kako bi ostvarila što bolje korisničko iskustvo i naposljetku opravdala svoju postojanost i isplativost. Upravo iz tog razloga potrebne su redovne promjene i dorade na web stranicama bez obzira radilo se o dizajnerskim promjenama ili o promjenama u funkcionalnosti web stranice.

U daljnjem sadržaju raditi će se upravo o redizajnu web stranice, te će se navesti cjelokupni postupak rada ne takvom projektu.

Ključne riječi: HTML, CSS, PHP, WordPress, CMS, korisničko iskustvo.

Abstract

In December of 1989 British scientist Tim Berners-Lee drafted a document entitled „Information Management: A Proposal“ in which he applied to CERN Institute for approval and financing project which had exchange of hypertextual data as main purpose. In early 1990, he received a decision on approving of the project and he begins to work on it. By August, 6th 1991, he made all of the necessary tools, technology and infrastructure that was needed to produce and publish a website and he had launched the first website. In 1992 Congress of the United States makes the decision that internet can be used for commercial purposes and that was a beginning of a rapid development of web technology.

In the period of the first site was published since today many tools and technologies for creating and publishing web pages have been developed, which has led to the publication of a great number of web content in recent years, and websites have become an indispensable part of the marketing and advertising, as well as selling products and services via web. Development of technologies allowed access to web content through computer, smartphone, tablet, smart watch and television regardless of time and place.

In purpose of gaining a larger number of visitors, website should be fully functional and enjoyable to browse. Also, in purpose of returning the visitor, or to buy a product, website should satisfy the user experience. Web site must be adapted to the previously mentioned devices where the appearance of elements on the website does not differ significantly between individual devices.

Since there is a great number of content on the internet and that it has almost become the rule that each company has its own website, it should always follow the trend of design and functionality in order to achieve the best user experience and ultimately justify its durability and cost effectiveness. This is why the changes on the website are necessary. Whether it is the design changes or changes in the functionality of the site.

Topic of this paper is redesigning of the website, and it indicates the overall process of work on such a project.

Keywords: HTML, CSS, PHP, WordPress, CMS, User experience.

Popis korištenih kratica

HTML	HyperText Markup Language Opisni jezik za opisivanje strukture web stranice.
CSS	Cascading Style Sheets Jezik za stiliziranje web stranice.
CMS	Content Management System Sustav za upravljanje sadržajem web mjesta.
SEO	Search engine optimizatio Optimizacija web stranica za web pretraživače.
SASS	Syntactically Awesome Stylesheets Preprocesorski jezik namijenjen stilizaciji HTML elemenata.
PHP	Hypertext Preprocessor Skriptni jezik koji se izvršava na strani servera.

Sadržaj

1.	Uvod.....	7
2.	HTML.....	9
2.1.	Osnovna struktura HTML prezentacijskog jezika.....	9
2.1.1.	Naslovi.....	12
2.1.2.	Oblikovanje.....	13
2.2.	Multimedija.....	14
2.3.	Meta oznake.....	15
3.	Stiliziranje i oblikovanje.....	16
3.1.	Preprocesori.....	17
3.2.	SASS.....	18
3.2.1.	SASS varijable.....	19
3.2.2.	SASS grananje.....	20
3.2.3.	Mixins.....	21
3.3.	Usporedba CSS-a i SASS-a.....	22
4.	Sustav za upravljanje sadržajem.....	25
4.1.	WordPress CMS sustav.....	26
4.2.	Datoteke u WordPress CMS sustavu.....	26
4.2.1.	Header.php datoteka.....	27
4.2.2.	Index.php datoteka.....	27
4.2.3.	Footer.php datoteka.....	28
4.2.4.	Style.css datoteka.....	28
5.	Dizajn web stranice.....	29
5.1.	Dizajn naslovne stranice.....	30
5.2.	Dizajn standardnih unutarnjih stranica.....	30
5.3.	Dizajn kontakt stranice.....	31
6.	Prijelom statične web stranice.....	32
6.1.	Pisanje HTML kôda.....	32
7.	Integracija statične web stranice na WordPress CMS sustav.....	36
7.1.	WordPress CMS sučelje.....	37
7.2.	Datoteka header.php.....	37
7.3.	Datoteka footer.php.....	39
7.4.	Predlošci stranica.....	40
8.	Objava web stranice.....	44
9.	Istraživanje učinkovitosti redizajna.....	45
10.	Zaključak.....	47
11.	Literatura.....	49

1. Uvod

Razvoj web stranica počinje na CERN-ovom institutu kada su stručnjaci Tim Berners Lee i Robert Caillia 1989. godine objavili dokument „Information management: A proposal“ u kojem opisuju razmjenu hipertekstualnih dokumenata te od instituta traže odobrenje i financiranje projekta. Do kraja godine Tim Berners Lee izradio je cjelokupnu infrastrukturu potrebnu za objavu web stranica kao i alate potrebne za izradu i objavu web stranica kao što su HyperText Transfere Protocol 0.9 (HTTP), prezentacijski jezik za izradu web stranica HyperText Markup Language, web preglednik WorldWideWeb, web server i program za HyperText Transfere Protocol, Cern httpd. 6. kolovoza 1991. Godine objavljena je prva web stranica čiji je autor upravo sam Tim Berners Lee. [1,2]

Američki kongres 1992. godine donio je odluku da se internet može koristiti u komercijalne svrhe zbog čega dolazi do naglog i velikog razvoja tehnologija i alata potrebnih za izradu i objavu web stranica. Brzina razvoja tehnologija dovela je do osnivanja krovne organizacije koja će pratiti i usklađivati tehnologije te koja je kasnije i sama preuzela ulogu usavršavanja tehnologija potrebnih za izradu web stranica kao što su HTML i CSS. Tako je 1994. godine osnovana krovna organizacija pod nazivom W3C (World Wide Web Consortium), a predsjednik od samog osnutka pa sve do danas je i izumitelj web stranica Tim Berners Lee. [3,4]

Od tada se alati i tehnologije potrebne za izradu i objavu web stranica konstantno razvijaju i nadograđuju. U razdoblju od 1994. do 2000. godine razvijene su mnoge tehnologije kao i programski jezici neophodni za izradu i objavu web stranica, od kojih se mnogi i danas koriste kao obavezni alati, a to su primjerice JavaScript, PHP, AJAX i slični. Tehnologije koje se danas svakodnevno koriste omogućavaju korisniku pregledavanje web sadržaja bez obzira na vrijeme i mjesto na kojem se nalaze. Uz tehnologije i alate razvio se i dizajn samih web stranica kojem je svrha vizualno privući i zadržati korisnike, odnosno posjetitelje određenog web mjesta. Osim praćenja suvremenih tehnologija i dizajna, prilikom izrade web stranice potrebno je pratiti i korisnička iskustva kako bi se web stranica po funkcionalnosti u potpunosti približila korisniku. Za pregledavanje web sadržaja danas se koristi široki spektar uređaja kao što su: računala, prijenosna računala, tableti, pametni telefoni, pametni satovi i pametni televizori. Važno je napomenuti kako gotovo svaki od navedenih uređaja izvršava prikaz web stranice na različitoj rezoluciji te kako upravljanje web stranicom od strane korisnika, odnosno ostvarivanje povratne veze ovisi od uređaja do uređaja. Web stranica ima zadaću u potpunosti zadovoljiti korisnikove želje i osigurati maksimalno korisničko iskustvo pri pregledu web stranice bez obzira na kojem se od gore navedenih uređaja web stranica pregledava. Upravo na temelju toga u dizajnu web stranica danas je gotovo pravilo izrada web stranica po načelu „mobile first“. Načelo se temelji

na istraživanjima koja su pokazala da većina posjetitelja web stranica istima pristupa prvo putem pametnih telefona koja posjeduju manju rezoluciju u odnosu na stolna i prijenosna računala.

S druge strane nalaze se klijenti, odnosno naručitelji i vlasnici web stranica koji unose i izmjenjuju sadržaj većinom na dnevnoj bazi te im je potrebno omogućiti brzo i jednostavno obavljanje zadataka, obzirom da se radi o velikoj većini koja ne posjeduje znanje programiranja i izrade web stranica. To je omogućeno razvojem CMS sustava. CMS (eng. Content Management System) ili u prijevodu sustav za upravljanje sadržajem web mjesta je sustav koji omogućava [5,6] unošenje, izmjenu i brisanje sadržaja koji se nalazi na web stranici. Time je omogućeno klijentu da pristupa sadržaju i vrši razne manipulacije nad sadržaje koji se nalazi na web stranici u njihovom posjedu. Jedan od najpopularnijih CMS sustava danas je WordPress, besplatan sustav koji pruža veliki broj mogućnosti kako za programera web stranice, tako i za naručitelja, odnosno klijenta.

Osim izrade i objave web stranice važni su koraci i postupci koji slijede nakon toga. Pod time se smatra održavanje web stranice, redovita izmjena sadržaja, unaprjeđenje funkcionalnosti web stranice te promjena dizajna ili neki veći pothvat kako bi se web stranica osvježila. Ovi postupci vrše se kako bi web stranica pratila trendove u dizajnu i funkcionalnosti, a osim toga, redovno održavanje web stranice osigurava i bolji položaj na tražilicama kao što je Google. Česta izmjena i osvježavanje sadržaja često pomaže u SEO optimizaciji kako objavljena web stranica ne bi izgubila svoje mjesto ili bila odbačena općenito.

Kako bi se izbjegle nepravilnosti rada web stranice prije same objave često se vrše analize, ispitivanja i ankete kako bi se uvidjele eventualne pogreške ili nedostaci, bilo da se radi o nedostacima vezanim za funkcionalnost ili sam dizajn web stranice. Prilikom ovakvih analiza ispituje se responzivnost web stranice, odnosno prilagođavanje web stranice različitim rezolucijama uređaja na kojima se može ista pregledati, ispituje se njena funkcionalnost, odnosno provjerava se da li svi elementi primjerice, rotatori i linkovi web stranice rade bez greške, provjerava se da li je web stranica u potpunosti izrađena po predlošku grafičkog dizajna i na posljatku ispituje se korisničko iskustvo kako bi se mogla provjeriti reakcija posjetitelja web stranice.

2. HTML

HTML je prezentacijski jezik namijenjen za izradu web stranica, a koristi se kao skraćenica za HyperText Markup Language. Pušten je u uporabu 1993. godine nakon što je internet postao i komercijalno dostupan.

Iako je ostao besplatan (eng. open source) program, kasnije su nastale mnoge inačice ovog prezentacijskog jezika koje su donijela mnoga poboljšanja i unaprjeđenja vezana za samu sintaksu. HTML prezentacijski jezik jednostavan je i brzo se uči. Iako posljednja inačica pruža snažnu multimedijску platformu ostao je smislen i jednostavan za uporabu. Osim toga primjena ovog prezentacijskog jezika ostala je zagwarantirana i za buduću primjenu. HTML prezentacijski jezik koristi se za izradu statičnih web stranica te se od 2014. godine kada je izašla posljednja inačica koristi i za izradu mobilnih aplikacija i slično. Međutim, prezentacijski jezik se ne koristi samo pri izradu web stranica i mobilnih aplikacija, već je njegova specifičnost u tome što se svaka web stranica koja se otvara na računalu korisnika prikazuje upravo putem ovog prezentacijskog jezika, bez obzira da li je web stranica statična ili dinamična, odnosno da li sadrži PHP kod. Već je spomenuto da se HTML prezentacijski jezik koristi za izradu statičnih web stranica. Statična web stranica je izrađena isključivo pomoću HTML prezentacijskog jezika i CSS opisnog jezika uz eventualnu primjenu JavaScripta kako bi se pojedini elementi stranice ponašali dinamički. U kombinaciji s PHP jezikom web stranica postaje dinamična, odnosno ostvaruje se veza s web serverom.[2]

2.1. Osnovna struktura HTML prezentacijskog jezika

Iako je od nastanka prve inačice do danas izdano pet verzija ovog prezentacijskog jezika i iako posljednja verzija pruža mnogo više nego bilo koja od prethodnih verzija sama struktura nije se značajnije mijenjala. Osnovna struktura HTML prezentacijskog jezika vrlo je jednostavna, a sastoji se od dva osnovna dijela a to su elementi i atributi, pri čemu su elementi osnovni elementi namijenjeni za izradu strukture web stranice dok se atributi koriste kao elementi namijenjeni za modificiranje elemenata. Osim toga od samog početka HTML pruža uporabu komentara koji se koriste za bolju organizaciju koda, dok se u posljednjoj inačici prezentacijskog jezika nalazi pregršt entiteta, odnosno kodova posebnih oznaka koji služe za razne ispise web stranice u pregledniku.[2,7,8]

HTML dokument sastoji se od dvije osnovne cjeline. Te cjeline se zovu zaglavlje (eng. head) i tijela (eng. body). Unutar zaglavlja nalaze se veze i relacije kojima se povezuje vanjska datoteka u kojoj se nalazi opisni jezik CSS namijenjen stiliziranju elemenata web stranice, razni „meta“ elementi koji omogućavaju SEO optimizaciju, veze s vanjskom bazom fontova korištenih

na web stranici i slično. Unutar tijela nalazi se sav sadržaj koji se prikazuje prilikom renderiranja web stranice u pregledniku. Svi elementi pišu se u posebnim oznakama (eng. tags). Većina elemenata koristi otvoreni, primjerice „<body>“ i zatvoreni „</body>“ tag te se unutar njih nalazi sadržaj, dok se neke vrste sadržaja mogu unositi putem atributa, primjerice „“. [2,8]

```
<!DOCTYPE>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Naslov web stranice</title>
</head>
<body>

</body>
</html>
```

Slika 2.1 osnovna struktura HTML prezentacijskog jezika

Iz viđenog može se zaključiti da je osnovna struktura prezentacijskog jezika HTML zaista jednostavna. Sastoji se od inicijalizacije HTML5 dokumenta s oznakom „<!DOCTYPE>“, početne i završne oznake koji označava početak i kraj cjelokupnog HTML dokumenta oznakom „<head> </head>“ te od dva osnovna dijela HTML dokumenta s oznakama „<head> </head>“ i „<body> </body>“, dok prvi element unutar zaglavlja označava preglednicima ispis svih znakova uključujući i one s dijakritičkim oznakama, a drugi element ispisuje naslov stranice u tabulatoru preglednika. [2]

Iako na primjeru osnovne strukture ne postoji sadržaj unutar tijela, tijelo služi kao nositelj cjelokupnog sadržaja koji se prikazuje u pregledniku. Ne postoji točno definirana struktura koja je nositelj sadržaja, međutim postoje neka pravila kojih se potrebno pridržavati. Isto tako postoje načela koja definira sam Google kao najsnažnija tražilica web stranica, a vezana su za SEO optimizaciju i načine pretraživanja. Sukladno tome potrebno se pridržavati pravilnog načina pisanja naslova, o čemu će biti više u sljedećim poglavljima, takozvanim „containerima“ koji obuhvaćaju pojedine dijelove sadržaja, pravilno korištenje novih tagova koji su došli s novom inačicom HTML-a, što može znatno utjecati na plasman web stranice na tražilici.

Neke od oznaka koje se koriste kao već spomenuti kontejneri (eng. container) nose tagove kao što su primjerice „div, section, article, header, footer“ i njima slični. Div je jedna od najčešće korištenih oznaka kontejnera unutar koje se nalazi neki sadržaj bio on tekstualnog, slikovnog ili

nekog drugog oblika. Spomenuta je jedna od najstarijih oznaka koje postoje u HTML-u, te je vjerojatno upravo to razlog najčešće primjene ove oznake, iako se u posljednjoj inačici HTML5 nalaze oznake koje se preporučaju kao što su „article, header, footer, section“. Tako se primjerice kombinacijom HTML-a i CSS-a odvajaju pojedini dijelovi web stranice kao kontejneri u kojima se nalazi različit sadržaj koji se može oblikovati u raznim oblicima, veličinama, bojama pozadine i slično, čime će se naglasiti određeno polje u kojem se nalazi neki sadržaj.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Naslov web stranice</title>
  <link rel="stylesheet" href="css.css">
</head>
<body>
  <div class="main-wrapper">
    
    <p class="paragraph">Lorem ipsum dolor sit amet, consectetur
dipisicing elit. Possimus porro culpa eum quis dicta cupiditate
eligendi optio unde. Quasi cumque obcaecati et nostrum dignissimos
voluptate perspiciatis ex provident, quis aspernatur ullam sequi
repellendus necessitatibus amet odio voluptas consequatur quidem
placeat earum a, doloremque fuga nulla facilis hic sapiente!
Doloremque, quasi.
    </p>
  </div>
</body>
</html>
```

Slika 2.2 primjer strukture HTML prezentacijskog jezika

U danom primjeru unutar tijela nalaze se oznake „div“ kojoj je pridodana klasa „main-wrapper“, oznaka „img“ putem koje se unosi slika s oznakom „image“ i oznaka „p“ s pridodanom klasom „paragraph“. Pomoću klasa HTML se povezuje s opisnim jezikom CSS pomoću kojeg se isti stilizira, a o kojem će biti više u sljedećem poglavlju. Nakon pokretanja prethodnog primjera u pretraživaču, biti će prikazani sljedeći elementi. Oznaka „div“ biti će poravnata centralno širine 1180 pixela, oznaka „img“ biti će nositelj slike širine 300 pixela poravnata u lijevo pored koje će se s desne strane ispisati tekst koje se nalazi unutar oznake „p“. Upravo ovim postupkom odvajaju se različite sekcije i dijelovi koji se prikazuju na web stranici čime je moguće dizajnerski stilizirati web stranicu te naglasiti bitne ili sporedne dijelove.

2.1.1. Naslovi

Kao što postoje naslovi i podnaslovi u svim tiskanim oblicima knjiga, radova, časopisa i slično, tako postoje naslovi i pri izradi web stranica. Naslovi se označavaju posebnim tagovima, a tagovi se pišu s oznakama od „<h1> </h1>“ do „<h6> </h6>“ pri čemu oznaka „h1“ označava najveći i najvažniji naslov unutar sadržaja, dok „h6“ označava najmanji. Naslovi se mogu pisati u bilo kojem tagu kao što su primjerice „div, section, article, p“ preporuča se korištenje upravo gore navedenih oznaka kako bi tražilica pomoću svojih algoritama pretraživanja što lakše i što brže pronašla upravo onaj dio sadržaja koji korisnik zahtjeva.[2,7,8]

Osim tekstualnog sadržaja koji se sastoji od ključnih riječi određenog paragrafa, a nalazi unutar oznaka naslova, tekstualni sadržaj koji se nalazi kao odjeljak piše se unutar oznaka „article“ ili „p“ čime se tražilici naglašava da se unutar tih oznaka nalazi tekstualni sadržaj većeg obujma, primjerice neki članak na blogu ili neki opisni sadržaj na web stranici.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Naslov web stranice</title>
</head>
<body>
  <article>
    <h1>Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Laborum
aliquid ea odio atque deserunt expedita, quam asperiores!
Beatae, aut, exercitationem. Cumque enim sit architecto id
reiciendis. Excepturi, voluptatem vitae modi.</p>
    <h2>Lorem ipsum dolor</h2>
    <p><b>Lorem</b> ipsum dolor sit amet, consectetur <i>adipiscing</i>
elit. Laboriosam libero tenetur illo rerum repudiandae nihil
omnis, nulla iure optio voluptates tempora modi veritatis praesentium
molestias ipsa magnam, laborum culpa aliquam nam enim!
Quisquam iure distinctio, culpa illo earum similique, doloribus,
tempora sit in molestias sint blanditiis vel, impedit incidunt soluta!</p>
    <h3>Lorem ipsum dolor sit</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quaerat
ab, blanditiis placeat ipsam distinctio, vitae quia ducimus reiciendis!
Dolorum inventore quam quibusdam vitae, <small>quod</small> maiores
veritatis iusto laboriosam nihil cum.</p>
  </article>
</body>
</html>
```

Slika 2.3 struktura naslova HTML prezentacijskog jezika

U navedenoj strukturi naslova nalazi se primjer pisanja naslova unutar odrađene sekcije, pri čemu je korištena pravilna struktura pisanja naslova i paragrafa. Osim već spomenutih oznaka koje se koriste pri pisanju naslova i paragrafa, na primjeru se nalaze i neke od oznaka koje su došle s posljednjom inačicom HTML-a kao što su primjerice oznake „i, b, small“ i slične. Iste se

koriste za stiliziranje određenog dijela teksta unutar paragrafa koji se želi naglasiti, a kako se ne bi to trebalo činiti pomoću CSS-a. Tako se primjerice pomoću oznake „i“ može prikazati dio teksta u kosom, odnosno italic obliku, pomoću oznake „b“ određeni tekst se može podebljati (eng. bold), ili pak se dio teksta može smanjiti pomoću oznake „small“. Iako se u navedenom primjeru nalaze unutar oznake „p“ iste se mogu pisati zasebno i izvan bilo koje oznake i tekst će se ispisati pravilno.

2.1.2. Oblikovanje

Obzirom da sve do 1996. godine nije postojao jezik za oblikovanje, odnosno za stilizaciju elemenata koji se nalaze u HTML datoteci, nije postojalo posebno oblikovanje elemenata. U samim počecima oblikovanja elemenata koristio se takozvani interni CSS, što znači da se oblikovanje vršilo unutar samog HTML dokumenta, što nije znatno utjecalo na brzinu i prijenos web sadržaja obzirom da se radilo o jednostavnim i malim datotekama koju su se objavljivale. Danas je nemoguće zamisliti ovakav način rada zbog veličine datoteka koje sadrže kod, neovisno da li se radi samo o HTML ili CSS datoteci.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Naslov web stranice</title>
</head>
<body>
  <div style="width: 1180px; margin: 0 auto; background: #808080;">
    
    <p style="width: 800px; font-size: 16px; display: inline-block;
vertical-align: top;">Lorem ipsum dolor sit amet, consectetur
adipiscing elit.
  Possimus porro culpa eum quis dicta cupiditate eligendi optio unde.
Quasi cumque obcaecati et nostrum dignissimos voluptate perspiciatis ex
provident, quis aspernatur ullam sequi repellendus necessitatibus
amet odio voluptas consequatur quidem placeat earum a,
doloremque fuga nulla facilis hic sapiente! Doloremque, quasi.</p>
  </div>
</body>
</html>
```

Slika 2.4 HTML unutarlinijska stilizacija

Na predstavljenom nalazi se takozvana unutarlinijska stilizacija elemenata HTML elemenata. Na ovaj su se način stilizirali elementi u samom početku oblikovanja. Unutar oznaka elemenata unosio se dio koda pod oznakom „style“ koji je označavao da se ovdje nalazi dio koda koji se

odnosi na stilizaciju elementa. Danas, iako u veoma rijetkim slučajevima, koristi se ovakav način stilizacije, a poseban je po tome što ima najveću razinu prioriteta u stilizaciji pojedinog elementa.

Sljedeći korak u stilizaciji elemenata bio je takozvana interna ili unutarnja stilizacija elemenata. Ona se izvršavala tako da se u zaglavlju HTML dokumenta nalazila oznaka „<style></style>“ unutar koje se pisao CSS kod, dok su se elementi povezivali putem klasa kao što je to na primjeru „Slika 2.2 Primjer strukture HTML prezentacijskog jezika“.

Posljednji korak je „izvlačenje“ CSS koda koji se nalazi unutar oznaka „style“ u zaglavlju HTML dokumenta i premješta se u posebnu datoteku pod najčešćim nazivom „style.css“ koja se povezuje s HTML dokumentom putem posebnog „link“ elementa koji se piše unutar zaglavlja dokumenta. [2,7,8]

2.2. Multimedija

Kombinacija više vrsta pojedinačnih medija koji tvore jednu cjelinu naziva se multimedija. Najznačajnije komponente u kontekstu informatike su tekst, slike, zvuk, video i animacije. Kao što je već spomenuto, posljednja inačica HTML-a, HTML5 pruža značajne mogućnosti u području prikaza multimedijskog sadržaja. Tako su primjerice gotovo u potpunosti izbačeni neki od donedavno korištenih dodataka (eng. plugins) koji su omogućavali prikaz multimedijskog sadržaja na web stranici kao što su Adobe Flash, Apple QuickTime i Windows Media player. Izbacivanje iz uporabe navedenih dodataka omogućeno je uporabom HTML elemenata kao što su „audio“ ili „video“ koji kao i kod slika posjeduju putanju do određenog video ili audio sadržaja. U nastavku se nalazi primjer uporabe multimedijjskih elemenata namijenjenih reprodukciji video i audio sadržaja na web stranici.

```
<video controls>
  <source src="movie.mpg4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
<audio controls>
  <source src="audio.mpg4" type="audio/mp4">
  <source src="audio.ogg" type="audio/ogg">
</audio>
```

Slika 2.5 HTML audio i video

Osim toga, HTML5 uvodi mogućnost izrade interaktivnih animacija pomoću CSS3 i JavaScript, odnosno jQuery koda koji se mogu reproducirati izravno kroz HTML platno (eng.

canvas). Time je ujedno smanjena uporaba alata za izradu animacija kao što su Adobe Flash, Adobe Flash player i pripadajući ActionScript jezik.[7,8]

Uporaba novih tehnika reprodukcije multimedijalnih elemenata na web stranici može imati veliki utjecaj na brzinu prijenosa web sadržaja i na brzinu učitavanja web sadržaja, a zasigurno će utjecati i na prepoznavanje web stranice od strane web tražilica. Obzirom da su gotovo svi pretraživači u potpunosti prilagođeni novim tehnologijama, uporaba starih tehnologija za reprodukciju multimedijalnog sadržaja može negativno utjecati na brzinu prijenosa sadržaja između web servera i posjetiteljevog uređaja putem kojeg pretražuje sadržaj. Tako će primjerice znatno brže biti učitani sadržaj ukoliko mu posjetitelj pristupa putem mobilnog uređaja ako se koristi nova tehnologija reprodukcije gdje se sadržaj prikazuje pomoću HTML5 elemenata bez korištenja alata kao što su Adobe Flash i slični.

2.3. Meta oznake

Kako bi se web stranica što bolje plasirala na web tražilicama, odnosno kako bi se nakon unesenih ključnih riječi pojavila među prvim ponuđenim rezultatima potrebno je mnogo pažnje obratiti na SEO optimizaciju. SEO je kratica za optimizaciju za web tražilice (eng. Search engine optimization). SEO postupak je zasebno područje u izradi web stranica, dio optimizacije može se odraditi i kroz HTML kod. Tako se pomoću takozvanih „meta“ oznaka može unaprijediti optimizacija. Meta oznake nalaze su unutar zaglavlja, a mogu nositi mnoge podatke kao što su primjerice „keywords, description, author“ kao i podatak „charset“ koji omogućava pravilno renderiranje dijakritičkih znakova. Oznaka „keywords“ označava sve ključne riječi koje se nalaze na pojedinoj stranici bilo da je unutarnja ili takozvana naslovna odnosno index stranica. Meta oznaka „description“ nosi podatke o opisu pojedine stranice, odnosno ukratko opisuje sadržaj koji se nalazi na pojedinoj stranici te je svojevrsan sažetak stranice, dok oznaka „author“ nosi podatke o autoru web stranice.

Meta oznake spadaju u takozvanu „on-page“ optimizaciju u koju spadaju još i naziv domene, naziv stranice, uređivanje slika, navigacija te optimizirani URL. Tako primjerice pravilna uporaba meta oznaka u kombinaciji s pravilnom uporabom naslova, alt i title tagova slika prilikom izrade web stranice mogu biti dovoljni za SEO optimizaciju ukoliko se ne radi o web stranici tvrtke koja ima veliku konkurenciju na webu.

3. Stiliziranje i oblikovanje

CSS je kratica za Cascading Style Sheets što je zapravo opisni jezik za stiliziranje i oblikovanje, odnosno definiranje izgleda web stanice sastavljen od niza unaprijed definiranih svojstva i mogućnosti. Håkon Wium Lie i Bert Bos, djelatnici W3C organizacije 1994. godine iznijeli su ideju o razdvajanju strukture i prezentacije web stranica te su 1996. godine ponudili rješenje pod nazivom CSS1. Uporaba opisnog jezika omogućila je uređivanje sadržaja web stranica te početkom njegove primjene web stranice više ne izgledaju kao tekst na papiru.[10,11]

Od izlaska prve inačice opisnog jezika isti se neprestano razvija, tako je sljedeća inačica objavljena već dvije godine nakon prve verzije, a samo godinu dana nakon toga 1999. godine započinje se izrada skica za posljednju inačicu i verziju koja se trenutno koristi CSS3. Za razliku od prijašnjih inačica, CSS3 se razvija modularno, a najvažniji moduli objavljeni su 2014. godine, što omogućava razvoj međusobno neovisnih standarda. CSS se primarno sastoji od svojstva i njemu pridružene vrijednosti, što se naziva deklaracijom. Deklaracije se grupiraju u blokove kojima se pomoću klase ili identifikatora vežu na pojedine elemente HTML koda. CSS sadrži niz selektora među kojima su najčešće korištene klase, pseudo-klase i kontekstni selektori, dok se identifikatori danas koriste u vrlo rijetkim slučajevima.

```
main-wrapper {  
  width: 1180px;  
  margin: 0 auto;  
  background: #808080;  
}  
img {  
  width: 300px;  
  height: auto;  
  display: inline-block;  
  vertical-align: top;  
}  
:hover {  
  color: #000;  
}  
#paragraph {  
  font-size: 16px;  
  font-family: Helvetica;  
}
```

Slika 3.1 Css

U prethodnom primjeru nalaze se primjeri pisanja CSS deklaracija smještenih u blokove koji se putem klase, pseudo klase, kontekstnih selektora i identifikatora vežu na pojedine elemente iz HTML koda. Prvi blok sastoji se od klase „main-wrapper“ unutar čijeg se bloka nalaze selektori „width“, „margin“ i „background“. Blok s oznakom klase točka (.) veže se na pojedini element

HTML datoteke tako da se elementu doda oznaka „class“ te se unutar navodnika doda naziv klase i CSS datoteke, što u primjeru izgleda ovako: „<div class=“main-wrapper“> ... </div>“. Blok koji se kontekstno veže na pojedini element HTML datoteke, veže se na način da se u CSS datoteci doda naziv elementa kao naziv bloka. Tako će primjerice veza elementa slike „“ u CSS-u biti povezana tako što će naziv bloka biti „img“. Pseudo klasa je zapravo kombinacija kontekstnog selektora s takozvanom pseudo-klasom kao što je u primjeru navedena pseudo-klasa „:hover“ čime se omogućava linku promjena zadane vrijednosti kada korisnik pokazivačem prijeđe preko veze, odnosno linka. Posljednji način veze koji se danas više ne koristi u tolikoj mjeri kao što je to bilo ranije je upravo veza putem identifikatora. Navedena veza u primjeru nalazi se na posljednjem mjestu. Veza se piše vrlo slično vezi klase, ali razlika je što se u CSS datoteci umjesto „.“ piše „#“, dok se u HTML-u umjesto „class“ piše „id“, tako da u primjeru to izgleda ovako „<p id=“paragraph“> Inicijalni sadržaj </p>“. Uporaba identifikatora se smanjila zato što posljednja inačica CSS-a ima znatno veće mogućnosti uporabe klasa i zato što identifikator ima veću razinu prioriteta u odnosu na klasu tako da pri uporabi obje veze može doći do problema.[2,7,10,11]

Osim toga, nove tehnologije takozvane preprocesori (eng. preprocessors) koje se koristi pri izradi web stranica, konkretnije pri izradi CSS-a kao što su Less ili SASS tehnologija, pružaju znatno veće mogućnosti manipulacije klasama nego identifikatorima.

3.1. Preprocesori

Preprocerosi su jezici temeljeni na CSS-u, odnosno, pisanje CSS datoteka pomoću skriptnih jezika koji omogućavaju izvršavanje raznih funkcija u CSS kodu. Prednosti preprocesorskih jezika su „čišći“, odnosno uredniji i pregledniji kod, varijable, petlje, uključivanje stila u stil i veća fleksibilnost te razmjena raznih biblioteka i dodataka. Postoji nekoliko preprocesorskih jezika od kojih su najpopularniji Less, SASS i Stylus, pri čemu svi rade na sličnom principu i sa vrlo sličnom sintaksom koda, a razvijaju se od 2006. godine.

Obzirom da su preprocesorski jezici skriptni jezici i da su njihove ekstenzije „.sass“ ili „.less“, a ne „.css“ oni se ne mogu prikazivati u pretraživaču kao klasičan CSS dokument. Upravo iz tog razloga postoje takozvani kompajleri kojima je zadaća i uloga da preprocesorske jezike prevode u standardnu CSS sintaksu i spremaju je u dokument s ekstenzijom „.css“ kako bi bila vidljiva pretraživačima. Neki od najkorištenijih kompajlera su CodeKit, Less, Koala, Mixture, Compass i LiveReload, od kojih je većina besplatna za preuzimanje i korištenje, odnosno takozvani „open-source“ program.

Uporaba preprocesorskih skriptnih jezika omogućava smanjenje veličine datoteka zato što cjelokupni kod ne mora biti sadržan u jednoj datoteci kao što je to primjer kod CSS-a, već se kod

može podijeliti na razne takozvane module koji se zatim implementiraju u jednu datoteku. Takav proces omogućava veću fleksibilnost u smislu promjene neke od vrijednosti elemenata na samo jednom mjestu umjesto da se promjena vrši na svim mjestima gdje se ta vrijednost nalazi.

[12,13,14]

```
// CORE
@import "core/bourbon/bourbon";
@import "core/grid-settings";
@import "core/neat/neat";
@import "core/normalize";
@import "core/defaults";

// MODULES
@import "modules/template"; // Main template file
@import "modules/home";
@import "modules/single";
@import "modules/category";
@import "modules/mobile";
@import "modules/scripts"; // Scripts extension

// PARTIALS
@import "partials/header";
@import "partials/footer";
@import "partials/nav";

//MOBILE
@import "modules/mobile";
```

Slika 3.2 SASS implementacija modula u jednu datoteku

3.2. SASS

Jedan od prvih preprocesorskih skriptnih jezika koji je bio i ostao besplatan te je jedan od najpopularnijih preprocesorskih jezika danas je upravo SASS. SASS je kratica punog naziva Syntactically Awesome Stylesheets objavljena za uporabu 2006. godine, a izradila ga je Natalie Weizenbaum. Od samog početka do danas razvijeno je nekoliko inačica, a posljednja je 3.4.22 objavljena 28. ožujka 2016. godine.

SASS preprocesorski jezik sastoji se od varijabli, grananja (eng. nesting), takozvanih mixins-a pod što ulaze petlje, argumenti i kombinacije, skupina selektora te pomoćnih biblioteka (eng. library). Obzirom da preprocesorski skriptni jezici omogućuju pisanje standardne CSS sintakse u kombinaciji s određenim funkcijama koje vuku korijene iz programskih jezika čime je omogućena uporaba gore navedenih dijelova od kojih se primjerice SASS sastoji.

Preprocesorski jezici omogućavaju i izradu stilizacije na nekoliko predložaka kako bi se smanjila veličina jedne datoteke, pri čemu se svi predlošci mogu pozvati u jedan zajednički predložak.

3.2.1. SASS varijable

Varijable se kao i u programskom jeziku sastoje od dva osnovna dijela, a to su deklaracija i inicijalizacija. Deklaracija varijable označava navođenje varijable koja se sastoji od znaka dolara „\$“ i imena varijable, primjerice „red“. Kombinacija deklaracije varijable u konačnici izgleda ovako: „\$red“. Inicijalizacija varijable je postupak kada se pojedinoj deklariranoj varijabli pridodaje vrijednost. Ono što je zajedničko SASS-u i CSS-u je dio nakon deklaracije varijable, a to je „:“ koja služi kao svojevrsna granica, odnosno znak pripadanja i vrijednost koja se pridodaje, konkretno za navedeni primjer crvene boje „#ab0100“. Tako u konačnici inicijalizirana varijabla crvene boje izgleda ovako: „\$red:#ab0100;“. Prednost varijable je to što omogućava fleksibilnost u smislu da ukoliko se s vremenom želi promijeniti nijansa crvene boje na web stranici, promjena će se odraditi na jednom mjestu dok će svi elementi koji posjeduju crvenu boju biti automatski promijenjeni.[12]

Varijabla boje koja je gore navedena primjenjuje se u elementima na sljedeći način. Kombinacijom standardne sintakse i uporabe svojstva kojim na nekom elementu dodaje stilizacija i inicijalizirane varijable elementu se dodaje oblikovanje.

```
$sizeNormal: 16px;
$red: #ab0100;
$gray: #808080;

.wrapper {
  font-size: $sizeNormal;
  color: $red;
  background: $gray;
}
```

Slika 3.3 SASS primjer inicijalizacije varijable i njene primjene

U danom primjeru nalazi se oblik uporabe inicijalizirane varijable na pojedini element koji se nalazi u HTML kodu. Tako će element koji nosi klasu „wrapper“ u HTML-u posjedovati sivu pozadinu i crvenu boju slova veličine 16 pixela. Ukoliko nakon određenog vremena korisnik želi promijeniti boju pozadine u tamniju potrebno je samo promijeniti vrijednost varijable „\$gray“ u neku drugu vrijednost koja će dati tamniju nijansu sive boje, umjesto da pretražuje cijeli CSS dokument i da mijenja vrijednost boje na svakom elementu na kojem se ista koristi. Osim toga, varijable omogućavaju promjenu nijanse boje na pojedinom elementu bez promjene heksadekadske vrijednosti boje u globalnoj varijabli. Tako će se pojedini element potamniti na način da se svojstvu „background“ uz postojeću varijablu „\$gray“ doda znak dijeljenja „/“ i određena vrijednost, primjerice „10“, što će značiti da će element koji posjeduje ovaj način definiranja pozadinske boje imati 10% tamniju nijansu sive boje u odnosu na ostale elemente

koji posjeduju pozadinsku boju s vrijednosti „\$gray“. [12] Takav primjer često se koristi kod pseudo-klase „:hover“ kod linkova kako bi se prilikom prijelaza mišem preko linka navedeni element potamnio za određeni postotak.

3.2.2. SASS grananje

Grananje (eng. nesting) je mogućnost koju pružaju preprocesorski jezici. Njime se omogućava jednostavnije i lakše „pogađanje“ pojedinih elemenata HTML koda. Dobilo je naziv zato što se kod prilikom grananja piše unutar blokova vanjskih elemenata. Ovaj postupak omogućava znatno brže i preglednije pisanje koda, te se sami elementi znatno lakše „pogađaju“ u odnosu na pisanje klasičnog CSS koda. Tako je stiliziranje navigacije jedan od najčešćih primjera stiliziranja elemenata pomoću grananja, iako se na isti način često stiliziraju tablice ili pak se na taj način može stilizirati bilo koji element HTML koda koji posjeduje nekoliko razina elemenata unutar glavnog elementa ili takozvanog kontejnera. [12]

```
.mainNav {
  ul { margin: 0; padding: 0; @include clearfix; @include span-
    columns(12); text-align: right; padding: 0; margin: 0;}
  li { margin: 0; margin-right: 2%; display: inline-block; padding: 0;
    a { @include span-columns(12); text-align: right; font-family:
      $default; font-weight: 300; text-decoration: none; color: $brand;
      margin: 0; padding: 0; line-height: 30px; padding-top: 10px; position:
      relative; z-index: 1;

      &:after { content: ''; z-index: -1; position: absolute; left: 0px; bottom:
        0px; height: 2px;
        background: $brand; width: 0%; transition: 200ms;
      }
      &:hover { color: $brand;
      &:after { left: 0px; bottom: 0px; width: 80%; background: $brand;
        height: 2px; }
      }}
  }
  &__Button { display: none; }
  .current-menu-item a { color: $brand; position: relative; z-index: 1;
  &:after { left: 0px; bottom: 0px; width: 80%; background: $brand;
    height: 2px; }
  }
}
```

Slika 3.4 SASS primjer grananja blokova elemenata

U primjeru izvornog koda „3.4“ nalazi se dio izvornog koda putem kojeg se vrši stilizacija pomoću grananja. Pomoću klase „mainNav“ blok se povezuje s elementom istog naziva klase u HTML kodu. Unutar bloka „mainNav“ definira se cjelokupna stilizacija navigacije na stranici. Tako se unutar navedenog bloka nalaze kontekstni selektori kojima se definiraju elementi unutar

kontejnera i ta se definicija odnosi isključivo na njih. Ukoliko se „ul“ selektor napiše izvan navedenog bloka, njegova će se stilizacija odnositi na sve „ul“ elemente na stranici, ili se isti selektor definira drugačije iza spomenute definicije i element navigacije će poprimiti posljednju definiciju. Kako je već spomenuto, kontekstni selektori unutar bloka „mainNav“ „pogoditi“ će isključivo određene elemente tako da u HTML dokumentu nije potrebno dodavanje pojedinih klasa. Na primjeru se osim kontekstnih selektora putem kojih se definira stilizacija elementa nalaze i oznake kao što je „&“. Spomenuta oznaka je podselektor kojim se na nadređeni element, odnosno element višeg stupnja prioriteta, primjenjuje dodatno svojstvo. Tako je link definiran pomoću selektora „a“ koji se nalazi unutar bloka „li“ selektora, dok se dodatna stilizacija vrši pomoću podselektora s dodatkom pseudo-klase „:hover“ čime će se promijeniti boja fonta ukoliko posjetitelj mišem prijeđe preko linka, dok će pomoću spomenutog selektora s dodatkom pseudo-klasa „before, after“ biti linku dodani pojedini elementi stilizacije.

Pomoću grananja omogućava se značajno smanjenje potrebnog koda kako bi se pojedini elementi stilizirali što utječe na veličinu datoteke što na posljeticu utječe na vrijeme učitavanja web stranice. Isto tako značajno smanjuje broj klasa koje bi bilo potrebno definirati kako stilizacija ne bi utjecala na druge elemente istog tipa.

3.2.3. Mixins

Kako bi se korisnik jednostavnije koristio pojedinom web stranicom, te kako bi ista pratila pojedine dizajnerske standarde potrebno je prilikom izrade zadržati konzistentnost, što znači da ukoliko se isti element pojavljuje na više različitih mjesta da svugdje mora sadržavati iste dimenzije, boje, fontove, veličine i slično. Jedna od opcija preprocesorskih jezika koja omogućava postizanje potpune konzistentnosti je takozvani mixins. Mixins je jedan od predložaka koji sadrži sve definicije vezane za pojedini element. Tako se u njemu mogu definirati tablice, inputi, gumbi (eng. buttons) i slični elementi kako bi na svakom predlošku web stranice izgledali jednako.

Ukoliko se po grafičkom predlošku navedeni elementi razlikuju po nekom od svojstava koji su prethodno definirani, mogu se izmijeniti u datoteci u kojoj se upisuje odnosno definira preostali dio SASS CSS-a gdje se stiliziraju ostali elementi koji se nalaze na stranici.

Osim navedenih mogućnosti koje pruža opcija mixins, ista omogućava primjenu petlji, argumenata i kombinacija na pojedine elemente. Obzirom da SASS nije opisni već skriptni jezik omogućena je primjena „for, each i while“ petlji kako bi se mogli definirati pojedini elementi. Tako će se znatno smanjiti potreban prostor za pisanje koda ukoliko se na web stranici nalazi više elemenata iste definicije kojima će primjerice širina biti varijabilna, odnosno različita.

Pomoću „for“ petlje može se utjecati tako da svaki sljedeći pod element elementa višeg prioriteta ima za određeni postotak veću ili manju dimenziju širine, visine, pozadinske boje, veličine fonta i slično.

Argument je također jedna od opcija koje spadaju pod opciju mixins, a utječe na smanjenje broja linija koda potrebnih za dodatnu stilizaciju pojedinih elemenata. Globalno definirani argument može se jednostavno primijeniti na pojedini element, kao npr. povećanjem veličine pojedine margine, a osigurava smanjenje broja linija koda i preglednost koda na način da se globalno definirani argument pomoću jedne linije koda pozove i primjeni na pojedini element, te će se kasnije ukoliko i dođe do potrebnih promjena širine margine promjena odraditi na jednom mjestu i biti će primijenjena na sve elemente u kojima se isti element poziva i primjenjuje.

3.3. Usporedba CSS-a i SASS-a

SASS služi kao jezik kojem je funkcija i uloga stiliziranje elemenata koji se nalaze u HTML datoteci kao što je to slučaj i kod opisnog jezika CSS, iako SASS koristi ista svojstva i vrijednosti kao i CSS postoje velike razlike između ova dva jezika. Kao što je već spomenuto za razliku od CSS-a, SASS je skriptni jezik koji omogućava uporabu funkcija, razna povezivanja predložaka te uporabe varijabli kao i mnoge druge mogućnosti koje ubrzavaju i olakšavaju stiliziranje elemenata. Krajnji rezultat je uvijek datoteka s ekstenzijom „.css“ pa sve veći broj web programera koristi upravo jedan od preprocesorskih jezika.

Osnovna razlika između ova dva jezika jest znatno smanjenje linija koda, preglednost koda i brzina izrade stilizacije elemenata. Smanjenjem broja linija koda smanjit će se i veličina datoteke što utječe na brzinu reprodukcije web stranice, preglednost koda omogućiti će web programeru znatno lakše snalaženje u tuđem ili vlastitom kodu ukoliko dođe do potrebe za izmjenom pojedinih dijelova koda. Mogućnost korištenja varijabli pomaže da se potrebna izmjena dogodi na samo jednom mjestu, a nije potrebno pregledati cijelu datoteku kako bi se na svim mjestima izvršila potrebna izmjena. Konzistentnost je osigurana primjenom raznih mixins-a koji će isto utjecati na svaki element na koji ih se pozove.

Kako bi se u potpunosti mogle iskoristiti sve opcije i funkcionalnosti koje preprocesorski jezici pružaju neophodno je dobro poznavanje standardne CSS sintakse, svojstava i vrijednosti koje omogućavaju stiliziranje elemenata HTML koda.

```

.message
border: 1px solid #ccc
padding: 10px
color: #333

.success
@extend .message
border-color: green

.error
@extend .message
border-color: red

.warning
@extend .message
border-color: yellow

```

Slika 3.5 Primjer pisanja SASS koda [12]

```

.message, .success, .error, .warning {
border: 1px solid #cccccc;
padding: 10px;
color: #333;
}

.success {
border-color: green;
}

.error {
border-color: red;
}

.warning {
border-color: yellow;
}

```

Slika 3.6 Primjer pisanja CSS koda [12]

Na slici 3.1 nalazi se primjer pisanja sintakse SASS preprocesorskog jezika, dok se na slici 3.2 nalazi primjer pisanja CSS opisnog jezika, kako bi se поближе objasnila uporaba navedenog

preprocesorskog jezika i usporedila se sa standardnom sintaksom CSS opisnog jezika. Na slici 3.1 se nalazi primjer deklaracija pojedinih blokova unutar kojih se nalaze svojstva i vrijednosti kao što je to slučaj i pri pisanju standardnog CSS opisnog jezika. Osnovna razlika pri definiciji jest ta da se pri uporabi SASS-a u jednom bloku mogu definirati svojstva i vrijednosti koja će se primjenjivati i na ostalim blokovima bez potrebe za definiranjem istih u ostalim blokovima. Pri uporabi CSS-a se u svakom bloku svojstva i vrijednosti moraju nanovo definirati ili kao što je to u primjeru sa slike 3.2 nazivi blokova se moraju nabrajati kako bi se na njima primijenile iste definicije. Razlog navedene razlike je što SASS omogućava, takozvano, nasljeđivanje pojedinog bloka uporabom oznake „@extend“ kojim se pojedini blok „uvozi“ u sljedeći blok koji poprima njegova svojstva. Tako će u gornjoj slici svi blokovi unutar kojih se nalazi takozvani importer „@extend“ poprimiti svojstva bloka „message“ unutar kojeg su ta svojstva definirana.

U standardnoj sintaksi CSS-a, ne postoje ovakve mogućnosti tako da je, kao što je već spomenuto potrebno u svakom bloku na novo definirati ista svojstva što zauzima mnogo prostora, odnosno zahtjeva korištenje mnogo linija koda čime se ujedno dovodi i do nepreglednosti koda. Drugi način je da se nazivi blokova koji će nositi isto definirana svojstva i vrijednosti, nabrajaju te se kasnije svaki dodatno oblikuje sa svojstvima koja im nisu zajednička. Ovakva situacija dovodi do problema ukoliko se kasnije mijenja neka od vrijednosti koja je u početku definirana kao zajednička za sve blokove ili prilikom izrade responzivne verzije web stranice.

4. Sustav za upravljanje sadržajem

Sustav za upravljanje sadržajem web mjesta (eng. Content Management System) ili skraćeno CMS je sustav koji omogućava korisniku upravljanje sadržajem u smislu manipulacija, dodavanja, brisanja, izmjena sadržaja koji se nalazi na njegovoj web stranici. Ovaj sustav omogućava znatno brže izvršavanje upravljanja sadržajem u odnosu na promjene koje se izvršavaju putem izmjena u kodu. Osim jednostavnosti omogućava svakome tko posjeduje web stranicu izmjene sadržaja zato što bi u suprotnom svaku promjenu morao izvršavati web programer koji je izradio web stranicu. Tako se sav sadržaj koji se nalazi na web stranici, a moguće ga je mijenjati putem CMS-a, može izmijeniti sam vlasnik web stranice u bilo kojem trenutku. [15,16]

Razvoj i primjena CMS-a započela je 1990. godine, dok danas postoji veliki broj raznih vrsta CMS sustava. CMS sustavi se u osnovni dijele na besplatne, odnosno „open-source“ CMS sustave i sustave koji su izrađeni za vlastite potrebe. Obzirom da se radi o veoma kompleksnom sustavu čija izrada zahtjeva dobro poznavanje područja programiranja, mnogo vremena provedenog na planiranje i izradu, ali kasnije i na održavanje sustava, vrlo često se koriste gotovi CMS sustavi. Neki od najčešće korištenih sustava su WordPress, Drupal i Joomla!. Posljednja istraživanja pokazala su da gotovo 40% web stranica koje su objavljene na webu rade putem WordPress CMS sustava. Razlog tome je to što je WordPress besplatan CMS sustav, vrlo dobro razvijen, s mnogo dodatka (eng. plugin), kvalitetnom i stalnom podrškom korisnicima te je gotovo sve vezano za WordPress moguće naći na webu. [15]

Za razliku od statičnih web stranica koje se izrađuju pomoću jezika HTML, CSS te uz eventualnu primjenu programskih jezika JavaScript ili jQuery, za takozvano spajanje web stranice na CMS sustav, odnosno za njenu integraciju potrebno je poznavanje PHP programskog jezika te poznavanje baze podataka. Nakon integracije web stranica koja je prethodno izrađena u statičnom obliku postaje dinamična. Dinamičnost web stranice prepoznaje se po mogućoj povratnoj vezi od strane klijenta, odnosno po komunikaciji klijenta i servera, odnosno osobe koja uređuje web sadržaj s web serverom.

PHP skriptni jezik punog naziva Hypertext Preprocessor je jezik koji se izvršava na strani servera i omogućava komunikaciju web stranice koja se nalazi na web serveru i osobe koja upravlja web sadržajem. Danas se tek u rijetkim slučajevima web stranice izrađuju direktno pomoću PHP jezika što ne isključuje izradu web stranice po ovom principu, već je mnogo češći slučaj da se prvo izradi statična web stranica nakon čega slijedi takozvana integracija.

Integracija je proces u kojem se pojedini dijelovi HTML koda mijenjaju s dijelovima PHP koda kako bi se određenom dijelu sadržaja omogućile izmjene bez ponovnog pristupanja

izvornom kodu. Upravo pomoću ovog procesa vrši se povezivanje web stranica s WordPress CMS sustavom o čemu će biti više u sljedećem poglavlju.

4.1. WordPress CMS sustav

Prva inačica WordPress CMS sustava objavljena je 27. svibnja 2013. godine. Osim što služi kao besplatan servis za upravljanje sadržajem web stranica, često se koristi i kao alat za izradu blogova. Temeljne tehnologije od kojih se WordPress sastoji su PHP i MySQL baza podataka. Obzirom da je besplatan i vrlo često korišten sustav svatko ga može besplatno preuzeti i instalirati na svoje računalo preko službenih stranica. [15,16]

Obzirom da računalo ne može samostalno pokrenuti tj. pročitati datoteke koje se sastoje od PHP koda jer je to jezik koji se izvršava na strani servera, prethodno je potrebno preuzeti i instalirati neke od programa koji računalo „pretvaraju“ u lokalni server kako bi se spomenuti jezici mogli pokretati. Neki od često korištenih programa su XAMPP, MAMP, WAMP i LAMP. Pomoću navedenih programa računalo ostvaruje mogućnost prepoznavanja i pokretanja MySQL baze podataka i PHP jezika. Obzirom da klijent upravlja podacima, odnosno sadržajem koji se nalazi na serveru nije obavezan preuzimati navedene programe.

4.2. Datoteke u WordPress CMS sustavu

WordPress sustav zahtjeva poseban sustav datoteka i mapa, te kodova spremljenog u istima. Tako su primjerice obavezne datoteke pod nazivima `index.php`, `footer.php`, `header.php` i `style.css`. Navedene datoteke moraju postojati u mapi projekta zato što sustav ima specifičan način „čitanja“ pojedinih datoteka. Tako će se u datoteci `header.php` nalaziti sav kod koji prikazuje zaglavlje stranica, unutar datoteke `index.php` nalazi se sav kod tijela web stranice ili dijelovi koda koji će poslužiti kao lijepilo pomoću kojeg se različiti predlošci povezuju, u datoteci `footer.php` nalazi se sav kod koji ispisuje podnožje web stranice, dok se u datoteci `style.css` nalazi sav kod potreban za stiliziranje web stranice ili pak se u njemu nalaze kodovi koji implementiraju razne predloške CSS predložaka.

Prilikom izrade integracije u spomenute se datoteke spremaju pojedini dijelovi koda koji se nalazi unutar HTML datoteka te se povezuju s PHP kodom kako bi se ostvarila dinamičnost dijelova web stranice. Ukoliko se web stranica sastoji i od takozvanih unutarnjih stranica stvaraju se novi predlošci unutar kojih se pojedina unutarnja stranica povezuje s PHP kodom. Osim što PHP jezik služi kao komunikacijski kanal s web serverom isti se kod koristi kako bi se pojedini predložak povezoao s WordPress sučeljem.[16]

4.2.1. Header.php datoteka

Sve veze sa stilovima koji se izvršavaju na web stranici kao i zaglavlja, odnosno takozvani header elementi stranica nalaze se spremljeni u header.php datoteci. U nju se najčešće kopira dio HTML koda koji se sastoji od svih tagova uključujući i deklaraciju HTML5 datoteke do početka tijela web stranice odnosno do oznake „<body>“. Obzirom da je najčešće slučaj da se na svim stranicama prikazuje isto zaglavlje integracija zaglavlja zasebno će svaki puta pozvati spomenuti predložak te će se web stranica prikazati kao jedinstveni HTML dokument.

Budući da se kako je već spomenuto, gotovo uvijek koristi isti predložak za zaglavlje web stranice, spomenuta datoteka se može iskoristiti kako bi se ostvarila veza s predlošcima u kojima se definira stiliziranje elemenata. To omogućava smanjenje opterećenja poslužitelja koji samo jednom učitava datoteke CSS-a i definicije prikazuje jednom umjesto da za svaki predložak mora izvršavati novo čitanje i izvršavanje stiliziranje elemenata pojedinog predloška.

Obzirom da je navedeni predložak ujedno i zasebna datoteka, on se poziva unutar drugih datoteka pomoću funkcije gdje WordPress poziva header.php datoteku „<?php get_header(); ?>“. Ova funkcija mora biti definirana na početku svake datoteke u kojoj se nalazi tijelo pojedine stranice neovisno da li se radi o vanjskoj ili unutarnjoj stranici. Funkcija ima zadaću „obavijestiti“ server da se upravo na ovo mjesto poziva predložak pod nazivom datoteke header.php. Ukoliko neki od predložaka u kojima se nalazi tijelo stranice ne posjeduje navedenu funkciju ispisat će se stranica bez zaglavlja i stilizacije, odnosno bez svih definicija koje se nalaze u datoteci header.php. [15,16,17]

4.2.2. Index.php datoteka

Kako je već spomenuto, ukoliko se radi o web stranici koja se sastoji i od unutarnjih stranica, tada spomenuta datoteka ima funkciju „veziva“ odnosno povezuje sve predloške unutarnjih stranica te ih smješta između predložaka header i footer.php. Ukoliko se radi o web stranici sa samo jednim predloškom, odnosno o web stranici koja nema unutarnje stranice, tada se tijelo stranice može nalaziti unutar spomenute datoteke.

Kao i ostale unutarnje stranice i ova datoteka mora posjedovati funkciju za pozivanje header i footer.php datoteke. Iste se pozivaju na već spomenuti način, jedina je razlika što se datoteka, odnosno predložak footer.php poziva na načina da se umjesto oznake „header“ upiše oznaka „footer“. [15,16,17]

4.2.3. Footer.php datoteak

Kao što je datoteka header.php tako je i datoteka footer.php svojevrsan i zaseban predložak koji se prikazuje na svim stranicama. Jedina razlika je što footer.php nosi podatke o podnožju web stranice, ne posjeduje veze za stiliziranje, obzirom da se one uvijek nalaze u zaglavlju dokumenta, međutim datoteka footer.php nosi sve veze za pozivanja funkcija JavaScript i jQuery koda. Ove se veze nalaze u podnožju zato što usporavaju vrijeme učitavanja web stranice. Tako će se web stranica prvo učitati u potpunosti što se tiče vizualnog identiteta, a nakon učitavanja biti će učitane i sve funkcije koje se nalaze na web stranici. Ovakav sistem se primjenjuje kako bi se poboljšalo korisničko iskustvo, odnosno kako posjetitelj web stranice ne bi morao čekati da se prvo učitaju sve funkcije te zatim dobio ispis na ekranu. [15,16,17]

4.2.4. Style.css datoteka

CSS datoteke mogu posjedovati bilo koji proizvoljan naziv, međutim, WordPress CMS sustav zahtjeva postojanje upravo ove datoteke sa spomenutim nazivom zato što sustav automatski pretražuje mapu projekta i isporučuje datoteku sa stilski definiranim stavkama pod nazivom style. [15,16,17]

U samoj datoteci može se nalaziti kompletan CSS kod ili pak se mogu nalaziti oznake kojima se uvažava CSS kod iz eksternih predložaka. U sljedećem primjeru nalazi se izvorni CSS kod web stranice koji posjeduje poveznice s vanjskim datotekama CSS koda.

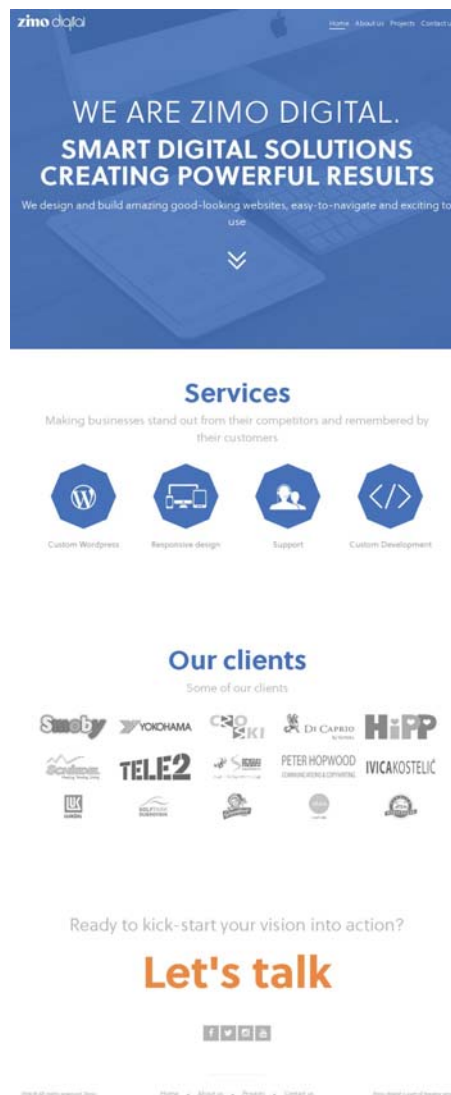
```
/*
Theme Name: ZIMO digital
Theme URI: http://zimodigital.com
Author: the ZIMO digital team
Author URI: http://zimodigital.com
Description: Wordpress theme hancrafted for the ZIMO digital agency
website.
Version: 1.0
Text Domain: zimo-digital
*/

/* Stylesheet files located in /css */
```

Slika 4.1 Sadržaj style.css datoteke u WordPress projektu

5. Dizajn web stranice

Dizajniranje web stranice odrađeno je kroz nekoliko koraka na temelju ranije definiranih skica. Kako bi se osigurao što manji broj naknadnih preinaka u kodu sve stranice izrađene su u alatu Photoshop, te se tijekom samog postupka dizajniranja prolazilo kroz određen broj izmjena kako bi se došlo do konačnog rješenja.



Slika 5.1 Prijedlog dizajnerskog rješenja

Kroz postupak dizajniranja većina elemenata je nadograđivana. Sa naslovne stranice izbačeni su elementi poput ispisa posljednjih radova, dok je dodano nekoliko novih elemenata kao što je poziv na akciju u podnožju, ispis poveznica na društvene mreže te dodatni izbornik kako bi se lakše dolazilo do željenih informacija na samoj stranici. Također, na konačnoj verziji izbačeni su gradijenti boja, te su postavljene jednostavne ikone.

5.1. Dizajn naslovne stranice

Kroz komunikaciju s tvrtkom postavljen je zadatak da se na naslovnoj stranici moraju istaknuti glavne djelatnosti i klijenti. Tako su na samom vrhu, odmah ispod glavne slike i poruke postavljene četiri ikone koje predstavljaju četiri glavne djelatnosti kojima se bavi tvrtka, dok odmah ispod njih dolazi popis klijenata.

5.2. Dizajn standardnih unutarnjih stranica

Na unutarnjim stranicama bilo je važno naglasiti trenutnu poziciju korisnika, što je učinjeno s velikim naslovom ispod kojih dolazi sam sadržaj unutarnje stranice.

About us

So what makes us tick? At Zimo digital it's simple. We're a close-knit team of passionate, go-getting digital experts who share excitement for beautiful and functional web design. Who understand every business needs a solid, professional and functional web presence. Who believe in making businesses stand out from their competitors and remembered by their customers.

And who know customers should be given truly exceptional digital experiences.



Goran Mudri
CEO/Project manager

Project management and organisation is my passion.

Vedran Mihalić
Developer

Appreciate humor, quality code and a friendly work atmosphere.

Peter Hopwood
Copywriter

You need a copywriter who uses words people want to read.

Ready to kick-start your vision into action?

Let's talk



Slika 5.2 Dizajn unutarnjih stranica

5.3. Dizajn kontakt stranice

Kroz više navrata naglašeno je da je komunikacija tvrtke s klijentom najvažnija u novom dizajnu. Za kontakt stranicu izrađen je poseban dizajn kroz kojeg se na prvo mjesto stavlja klijent tvrtke. Na ovom dijelu stranice klijent može direktno kontaktirati tvrtku kroz formu, ili dobiti bilo koju od važnih informacija o tvrtki.

6. Prijelom statične web stranice

U procesu prijeloma statične web stranice govorit će se o pravilnom pisanju strukture dokumenta, kao i o metodama stiliziranja web stranice. Obraditi će se prijelaz iz grafičkog rješenja u kôdno rješenje. Također definirati će se stilovi koji su bili potrebni da stranica vizualno izgleda kao i grafičko rješenje.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width = device-width,initial-scale = 1.0">
  <title>Zimo Digital</title>
  <link rel="stylesheet" type="text/css" href="css/style.css" media="all">
</head>
<body>
  <!-- Sav sadržaj ide ovdje -->
</body>
</html>
```

Slika 6.1 - Osnovna struktura index.html datoteke projekta

6.1. Pisanje HTML kôda

Početni korak za kodiranje naslovne stranice je izrada okvirne strukture HTML dokumenta. Ona se sastoji od HTML oznaka, naslova, paragrafa i sekcijskih elemenata kao što su article, nav, section i sl.

Prvi korak je definiranje sekcija unutar <body> elementa. Kroz dizajn definirano je da na naslovnoj stranici budu četiri sekcije, no kroz razvijanje kôd-a izbačena je jedna sekcija. Unutar svake sekcije postoji barem jedna slika, naslov i podnaslov.

Nakon definiranja sekcija unutar <body> elementa potrebno je definirati zaglavlje elementa gdje se nalaze logotip i navigacija. Zaglavlje je zamišljeno tako da prilikom pomicanja prema dolje na stranici ono ostaje pri samom vrhu i prati korisnika na bilo kojem djelu stranice. To je važno da bi svaki korisnik u bilo kojem trenutku mogao promijeniti lokaciju i poziciju na stranici.

```

<header class="siteHead siteHead--transparent">
  <div class="center">
    <div class="topLeft">
      <h1 class="logo">
        <a href="http://zimodigital.com"></a>
      </h1>
    </div>
    <div class="topRight">
      <nav class="mainNav" role="navigation">
        <ul>
          <li><a href="">Home</a></li>
          <li><a href="">About us</a></li>
          <li><a href="">Projects</a></li>
          <li><a href="">Contact us</a></li>
        </ul>
      </nav>
    </div>
  </div>
</header>

```

Slika 6.2 Struktura zaglavlja

Prvi <div> element unutar zaglavlja, koristi klasu „center“ u kojoj su definirane margine i pozicije da se sadržaj centrira na dizajniranu širinu od 1160px. Ova klasa koristi se svugdje gdje je željeno postići isti efekt poravnanja. Logotip se najčešće nalazi unutar H1 naslov elementa, koji unutar sebe sadrži poveznicu na naslovnu stranicu.

Navigacija stranice postavljena je unutar neodređene liste jer ne postoji točno definiran slijed prema kojem će se sam korisnik kretati po stranici. Stilskim jezikom zadani stilovi neodređene liste se uklanjaju, tako da sama lista izgleda poput popisa elementa, no postavljena je u ravninu, odnosno u jedan red.

zimo digital

Home [About us](#) Projects Contact us

Slika 6.3 Izgled zaglavlja s ubačenim logotipom i navigacijom

Nakon zaglavlja potrebno je kreirati i podnožje, u kojem je prema dizajnu određeno da se u njemu nalazi veliki poziv na akciju i komunikaciju s klijentom, izbornik i popis društvenih mreža. Poziv na akciju nalazi se unutar <a> elementa koji je stilskim oblikovanjem povećan i stiliziran da dolazi do izražaja, dok se popis društvenih mreža i izbornik nalaze unutar stilski obrađene neodređene liste u horizontalnom smjeru.

```

<section class="standardFooter">
  <div class="standardFooter_Inner">
    <h2>Ready to kick-start your vision into action?</h2>
    <a href="contact.html" class="fancybox letsTalk">Let's talk</a>
    <ul class="socialNetworks">
      <li><a href="#" target="_blank"><i class="fa fa-
facebook"></i></a></li>
      <li><a href="#" target="_blank"><i class="fa fa-
instagram"></i></a></li>
    </ul>
    <div class="standardFooter_Wrap">
      <div class="standardFooter_Left">
        <span>2016. &copy; All rights reserved. Zimo digital</span>
      </div>
      <div class="standardFooter_Nav">
        <ul>
          <li><a href="">Home</a></li>
          <li><a href="">About us</a></li>
          <li><a href="">Projects</a></li>
          <li><a href="">Contact us</a></li>
        </ul>
      </div>
      <div class="standardFooter_Right">
        <span>Zimo digital is part of Kreator group</span>
      </div>
    </div>
  </div>
</section>
<script type="text/javascript" src="jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="jquery-migrate-
1.2.1.min.js"></script>
<script type="text/javascript" src="/slick/slick.min.js"></script>
<script src="/js/scripts.js"></script>
</body>
</html>

```

Slika 6.4 Struktura podnožja

Važno je napomenuti da je prilikom integracije ikona društvenih mreža korišten web font, kako bi se što manje opterećivalo server i samu internet stranicu sa učitavanjem dodatnih ikona u nekom od standradnih oblika (.png, .jpg).

Prilikom izrade stranice je obraćena pozornost na učitavanje iste, stoga se slike gdje je to bilo moguće nisu koristile, već se koristilo oblikovanje CSS-om ili web fontovi.

Ready to kick-start your vision into action?

Let's talk



2016 © All rights reserved. Zimo digital

[Home](#) • [About us](#) • [Projects](#) • [Contact us](#)

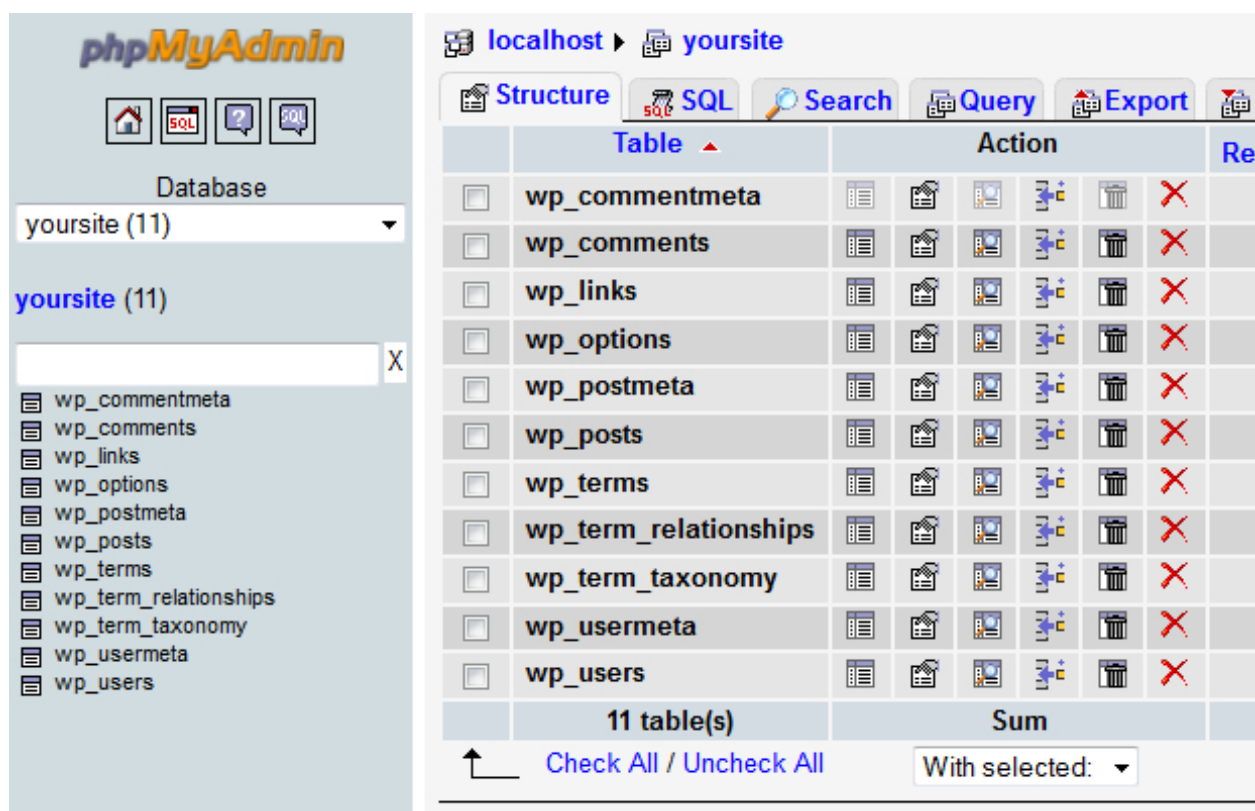
Zimo digital is part of Kreator group

Slika 6.5 Izgled programiranog podnožja

7. Integracija statične web stranice na WordPress CMS sustav

Spajanje na WordPress CMS sustav omogućit će korisniku dinamičnost i mogućnost izmjena informacija na samoj stranici bez potrebe za izmjenama u statičnom dijelu web stranice.

Za spajanje statične web stranice na WordPress CMS sustav biti će potreban lokalni web server koji omogućava učitavanje PHP kôda i MySQL bazu podataka u kojoj će biti zapisane sve informacije koje je kasnije moguće izmijeniti.



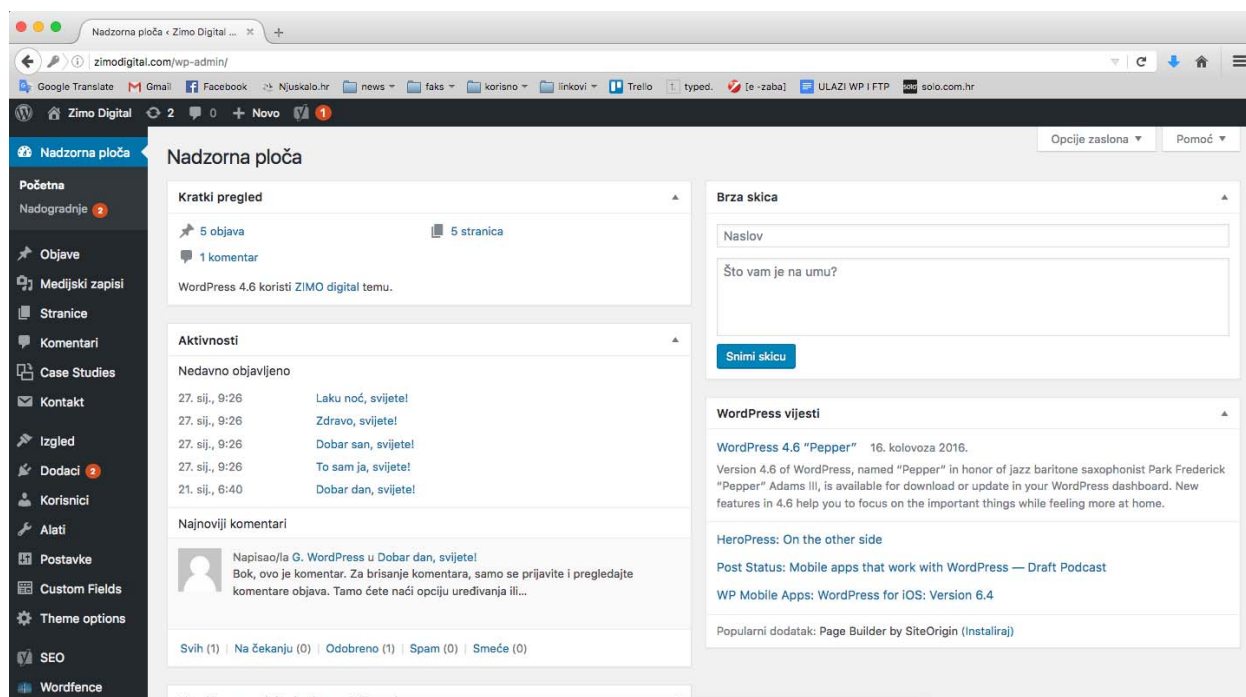
Slika 7.1 Osnovna struktura baze podataka s WordPress sustavom

Za instalaciju WordPress CMS sustava potrebno je instalacijsku datoteku smjestiti u zadanu datoteku prethodno instaliranog programa za lokalni web server. Kroz web preglednik upisuje se adresa <http://localhost/zimodigital> i pokreće se instalacija WordPress CMS sustava.

7.1. WordPress CMS sučelje

Korisničko WordPress CMS sučelje jednostavno je za korištenje. S lijeve strane nalazi se glavni izbornik gdje se manipulira web stranicom, dok se u sredini nalazi glavni prozor u kojem se vrše izmjene sadržaja na web stranici.

Također sučelje sadrži gornju traku gdje se nalazi izbornik za pristup pregledu web stranice ili pristup za nadogradnju sustava i njegovih dodataka.



Slika 7.2 Početna stranica WordPress CMS sučelja

7.2. Datoteka header.php

U datoteku header.php kopira se kôd iz index.html datoteke, od prve linije koda do otvaranja <body> elementa. Ova datoteka omogućava korištenje zaglavlja na svim kreiranim stranicama, bez da se sam kôd kopira na svaku od kreiranih stranica. Važno je provjeriti putanju CSS datoteke jer WordPress CMS sustav ima zasebne datoteke za stiliziranje, te bi putanja trebala izgledati kao na slici ispod.

```
<link rel="stylesheet" href="<?php bloginfo('template_directory'); ?>/public/css/style.css">
```

Slika 7.3 Poveznica na stilsku datoteku

U header.php datoteku moguće je umetnuti i vanjske poveznice za oblikovanje koje se koriste pri oblikovanju stranice. Također, prije samog zatvaranja </head> elemenata ukoliko se on koristi, u datoteku se ubacuje kod za analitiku i praćenje posjećenosti. Isto tako, u header.php datoteci mora postojati <?php wp_head(); ?> linija kôda prije zatvaranja </head> elementa kako bi WordPress CMS sustav u isprogramiran kôd dodao svoje stilove potrebne za rad sustava.

```

<html>
  <head>
    <title><?php bloginfo('name'); ?></title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-
awesome.min.css" rel="stylesheet">
    <link rel="stylesheet" href="<?php bloginfo('template_directory');
?>/css/style.css">
    <link rel="stylesheet" type="text/css" href="<?php
bloginfo('template_directory'); ?>/slick/slick.css"/>
    <link rel="stylesheet" href="<?php bloginfo('template_directory');
?>/css/lightbox.css">
    <script src="<?php bloginfo('template_directory');
?>/js/SmoothScroll.js"></script>
    <script src="https://use.typekit.net/yfu3bfw.js"></script>
    <script>try{Typekit.load({ async: true });}catch(e){}</script>
    <script>
      (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
        (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,
m)
      })(window,document,'script','https://www.google-
analytics.com/analytics.js','ga');

      ga('create', 'UA-35468101-1', 'auto');
      ga('send', 'pageview');

    </script>
    <?php wp_head(); ?>
  </head>
</body>

```

Slika 7.4 Izvorni kôd header.php datoteke projekt

7.3. Datoteka footer.php

Na isti način kao i kod header.php datoteke kôd se kopira iz index.html datoteke. U footer.php datoteci nalazi se kôd koji je napisan u podnožju index.html datoteke, a jednak je na svim stranicama. U datoteci footer.php najčešće se stavljaju vanjske JavaScript datoteke, prije zatvaranja </body> elementa.

Također, kao i kod header.php datoteke, u footer.php datoteci važno je staviti <?php wp_footer(); ?> liniju kôda i to prije samog zatvaranja </body> elementa kako bi sustav dodao potrebne JavaScript datoteke nužne za funkcioniranje sustava.

```
</main>
<section class="standardFooter">
  <div class="standardFooter__Inner">
    <h2><?php _e('Ready to kick-start your vision into action?', 'zimo-
digital'); ?></h2>
    <a href="#contact_form_pop" class="fancybox letsTalk"><?php _e("Let's
talk", 'zimo-digital'); ?></a>
    <?php if( have_rows('social_networks','option') ): ?>
    <div class="center">
      <ul class="socialNetworks">
        <?php while( have_rows('social_networks','option') ): the_row();
        // vars
        $icon = get_sub_field('social_icon','option');
        $link = get_sub_field('social_link','option');
        ?>
        <li><a href="<?php echo $link; ?>" target="_blank" class="h-<?php
echo $icon; ?>"><i class="fa <?php echo $icon; ?>"></i></a></li>
        <?php endwhile; ?>
      </ul>
    </div>
    <?php endif; ?>
  </div>
</section>
<script type="text/javascript" src="//code.jquery.com/jquery-
1.11.0.min.js"></script>
<script type="text/javascript" src="//code.jquery.com/jquery-migrate-
1.2.1.min.js"></script>
<script type="text/javascript" src="<?php bloginfo('template_directory');
?>/slick/slick.min.js"></script>
<script src="<?php echo get_template_directory_uri();
?>/js/scripts.js"></script>
<script src="<?php bloginfo('template_directory');
?>/js/lightbox.js"></script>
<?php wp_footer(); ?>
</body>
</html>
```

Slika 7.5 Izvorni kôd footer.php datoteke projekta

7.4. Predložci stranica

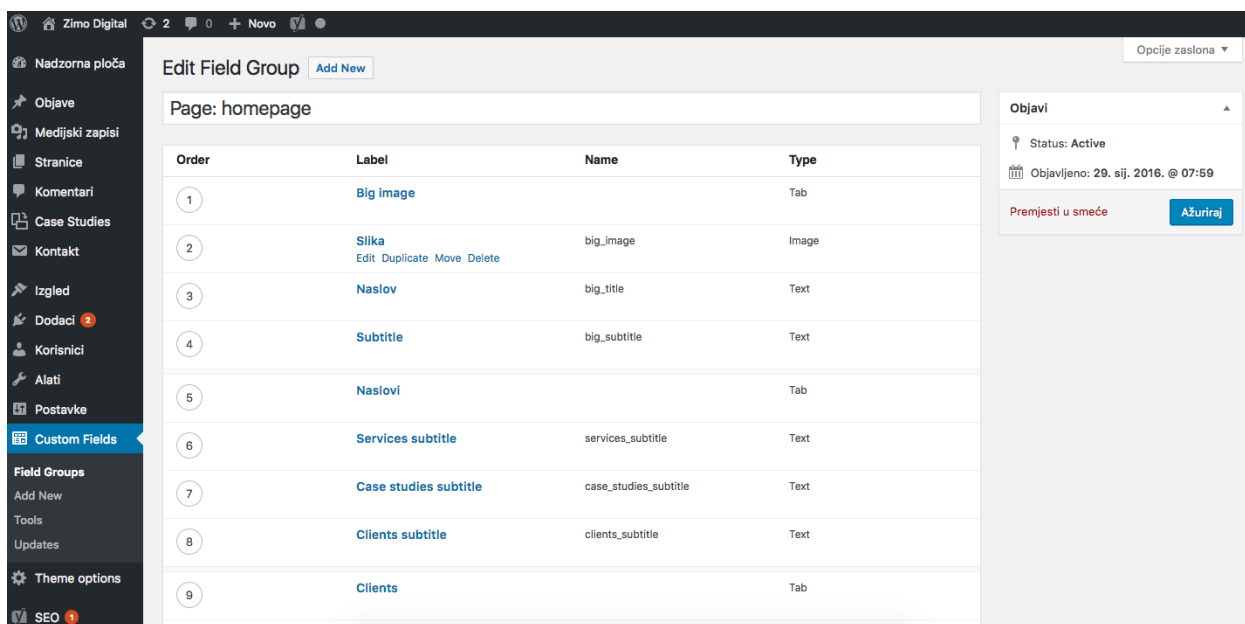
Za izradu stranica korišteni su predložci, kako bi se omogućilo personaliziranje svake od kreiranih stranica. Kreiran je predložak „Home page template“ za naslovnu stranicu, na kojoj se nalazi velika slika, popis usluga i klijenata. Potpuna manipulacija sadržajem omogućena je dodatkom Advanced custom fields kako bi se dodala dodatna polja za upis željenog sadržaja, izmjenu slika, naslova i sl.

Prilikom kodiranja, na samom vrhu .php datoteke važno je naznačiti da se radi o predlošku za stranice kako bi sustav znao o kakvoj se datoteci radi, te kako bi se na toj stranici mogla ubacivati dodatna polja vezana isključivo za taj tip predloška.

```
<?php get_header(); ?>
<?php
/*
Template Name: Front page
*/
?>
```

Slika 7.6 Početak datoteke predloška

Nakon definiranog predloška, potrebno je upisati dodatna polja pomoću Advanced custom fields dodatka.



The screenshot shows the WordPress Custom Fields interface. The main content area displays a table of fields for the 'homepage' page. The table has four columns: Order, Label, Name, and Type. The fields are as follows:

Order	Label	Name	Type
1	Big image		Tab
2	Slika <small>Edit Duplicate Move Delete</small>	big_image	Image
3	Naslov	big_title	Text
4	Subtitle	big_subtitle	Text
5	Naslovi		Tab
6	Services subtitle	services_subtitle	Text
7	Case studies subtitle	case_studies_subtitle	Text
8	Clients subtitle	clients_subtitle	Text
9	Clients		Tab

On the right side, there is a 'Objavi' (Publish) section with the following information:

- Status: Active
- Objavljeno: 29. sij. 2016. @ 07:59
- Buttons: Premjesti u smeće, Ažuriraj

Slika 7.7 Dodavanje polja za predložak naslovnice

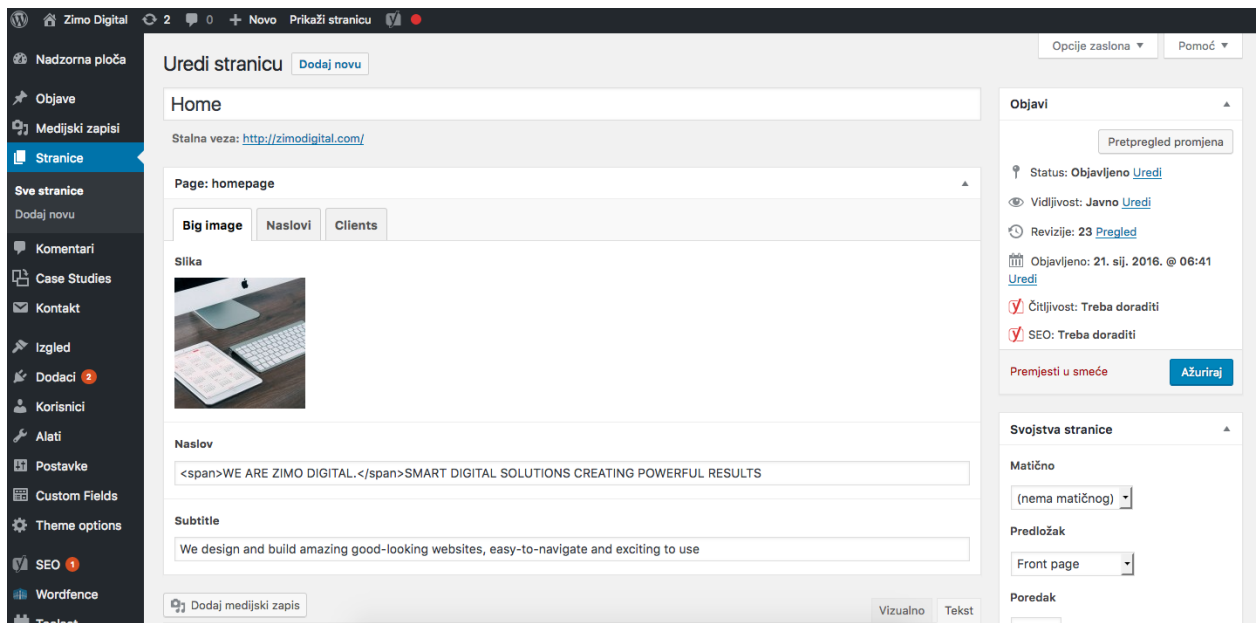
Svako definirano polje dobiva svoj „slug“ koji se upisuje u PHP datoteku kako bi sustav ispisivao polja prilikom pregleda stranice.

```
<section class="bigImage" style="background: url('<?php
the_field('big_image');?>')">
  <div class="center">
    <div class="bigImage_Inner">
      <h2><?php the_field('big_title');?></h2>
      <h3><?php the_field('big_subtitle');?></h3>
      <div class="button">
        <a href="#services" class="cmn-t-pulse"></a>
      </div>
    </div>
  </div>
  <div class="bigImage_Overlay"></div>
</section>
```

Slika 7.8 Primjer upisivanja PHP kôda za dodatna polja

Potrebno je napomenuti da se polja koja se pozivaju preko kôda opisanog u primjeru Izvornog koda 7.4 prikazuju samo ukoliko su ispunjena kroz sustav, te na zadanom predlošku stranice. Prikaz polja na drugom predlošku ili stranici je moguć uz određene izmjene u samom kôdu koji tada poprima dodatnu stavku, odnosno „ID“ stranice.

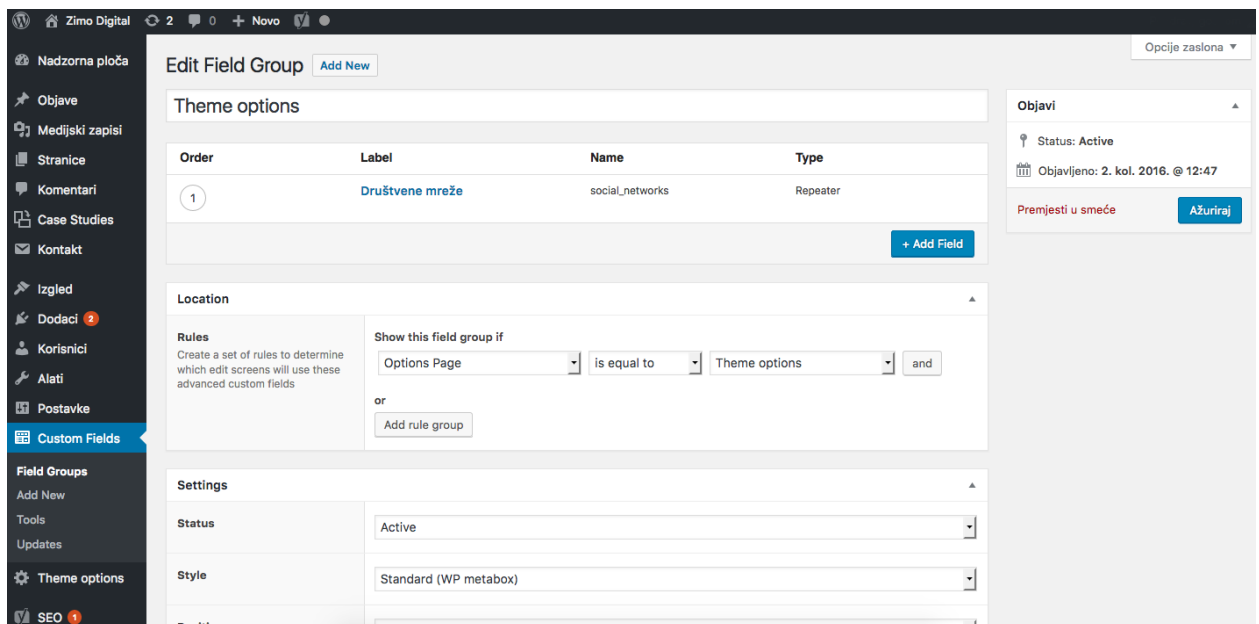
Punjenje stranice odvija se kroz sučelje sustava. U izborniku se odabiru stranice, kreira se nova stranica, te joj se odabire predložak. Nakon odabranog predloška stranice pojavljuju se polja koja su prethodno definirana kroz Advanced custom fields dodatak.



Slika 7.9 Izgled popunjene naslovnice s odabranim predloškom

Ostale stranice popunjene su na jednak način, prvo su definirana dodatna polja, a zatim su kreirane stranice kroz WordPress sustav.

Dodatna polja koja se prikazuju na svim djelovima stranice definirana su pod zasebnim djelom dodatka koji je nazvan „Theme options“ kako bi korisnik dobio mogućnost da stvari koje se nalaze na više mjesta ispunjava samo jednom, kao što su npr. poveznice na društvene mreže.



Slika 7.10 Dodatna polja za stavke koje se prikazuju na više mjesta

Nakon definiranja polja, važno ih je pravilno upisati s preinakom u samom pozivanju kôda. U ovom slučaju dodatno polje prikazuje se u podnožju stranice, odnosno u datoteci footer.php. Dodatna polja korištena su za prikazivanje poveznica na društvene mreže.

```

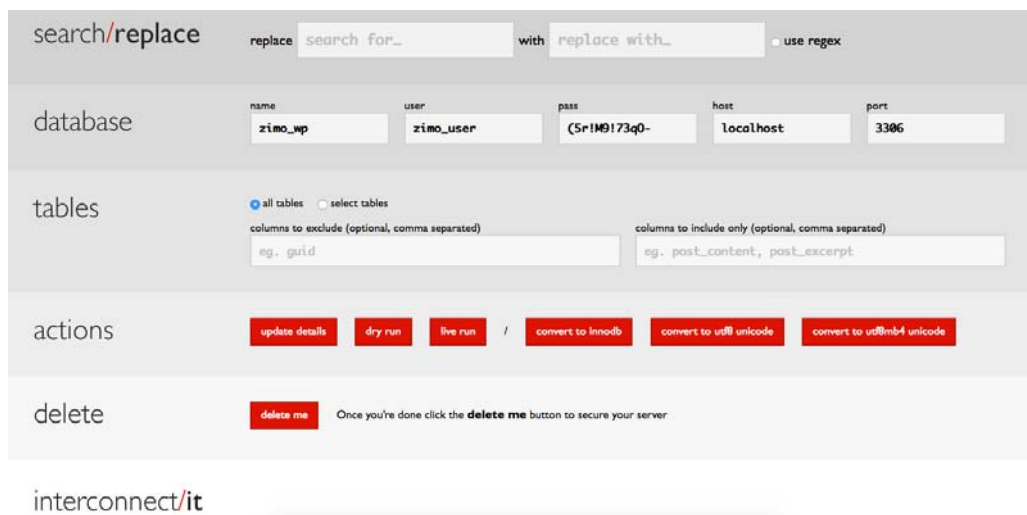
<?php if( have_rows('social_networks','option') ): ?>
<div class="center">
  <ul class="socialNetworks">
    <?php while( have_rows('social_networks','option') ): the_row();
      $icon = get_sub_field('social_icon','option');
      $link = get_sub_field('social_link','option');
    ?>
    <li><a href="<?php echo $link; ?>" target="_blank" class="h-<?php echo
$icon; ?>"><i class="fa <?php echo $icon; ?>"></i></a></li>
    <?php endwhile; ?>
  </ul>
</div>
<?php endif; ?>

```

Slika 7.11 Izvorni kôd za dodatna polja koja se prikazuju na više mjesta

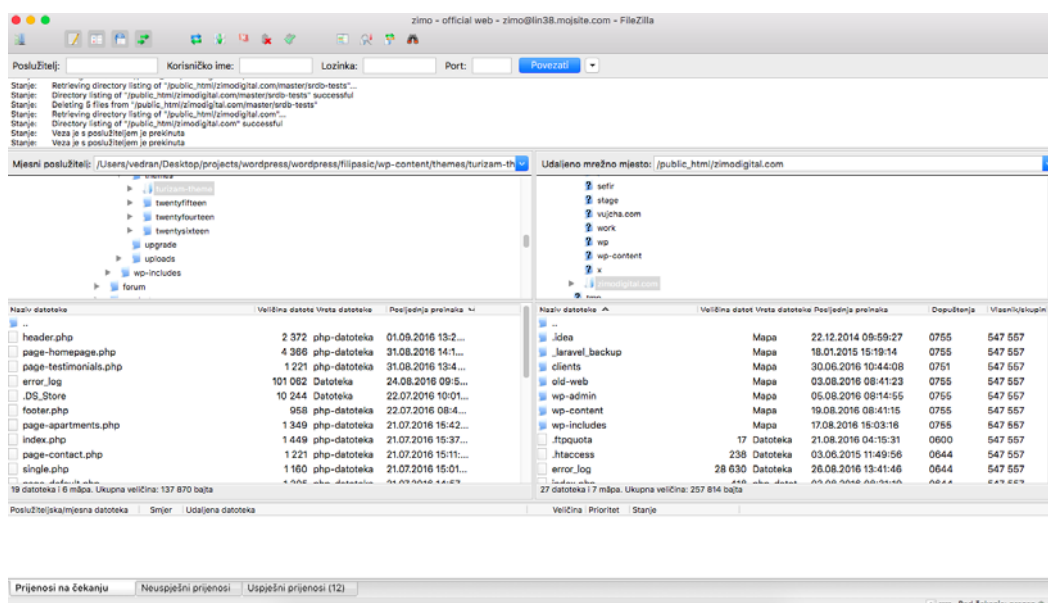
8. Objava web stranice

Sadržaj web stranice punio se na testnom serveru, odnosno na adresi <http://demo.zimodigital.com> kako se stara stranica ne bi ugasila i kako se klijentu u nekom trenutku ne bi uskratile željene informacije. Sve se radilo na istom serveru, te nije bila potrebna izrada nove baze podataka već se iskoristila postojeća baza, te su se dodatkom za editiranje baza podataka «Interconnect» izmijenile stavke zapisane u tablicama baze.



Slika 8.1 Prikaz izmjene podataka u tablicama baze

Nakon napravljenih izmjena u tablicama baze potrebno je staru web stranicu premjestiti u pod datoteku, a novu stranicu postaviti u glavnu datoteku. Seljenja je potrebno napraviti kada internet stranica ima najmanje posjetitelja.



Slika 8.2 Prikaz FTP klijenta i seljenja podataka

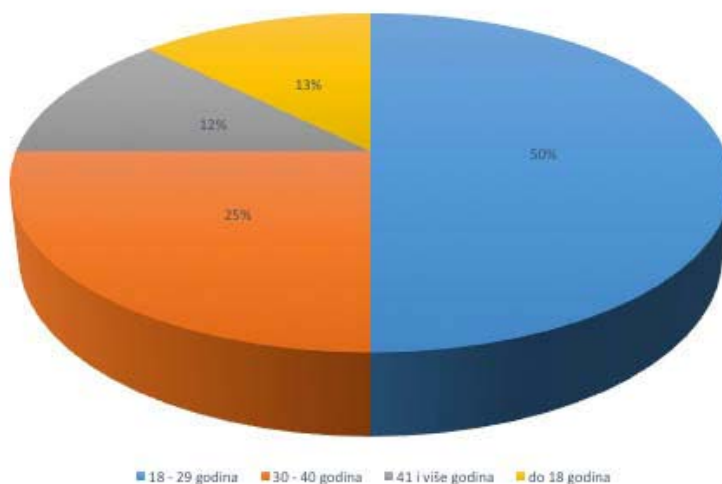
9. Istraživanje učinkovitosti redizajna

Istraživanje o učinkovitosti redizajna sprovedeno je anonimnom anketom na internetu. Ona se sastoji od tri dijela.

- U prvom dijelu prikupljali su se demografski podaci ispitanika: dob, spol, obrazovanje, zaposlenost i prebivalište.
- U drugom dijelu ispitanici su odgovarali na pitanja o njihovom znanju o izradi web stranica.
- Treći dio ankete odnosi se na opći dojam nove stranice, o pronalaženju informacija na njoj i zadržavanju krajnjeg korisnika na stranici.

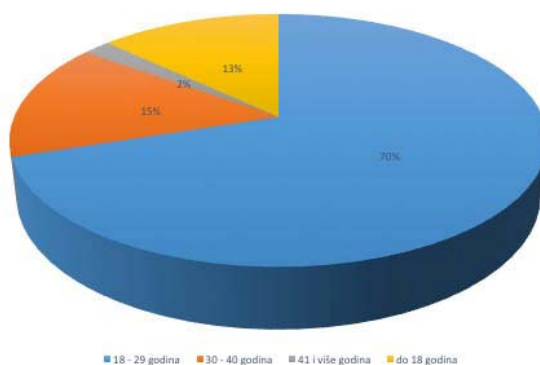
Cilj ove ankete bio je istražiti osnovne stavove ljudi prema izradi internet stranica, korištenju informacija, dobivenog dojma o novim stranicama tvrtke Zimo digital i pronalaženju željenih informacija na njoj.

Polovina ispitanika koja posjećuju internet stranice starosti su između 18 i 29 godina, te imaju minimalno srednjoškolsko obrazovanje i prebivalište u gradu. Odmah nakon njih su korisnici između 30 i 40 godina s 25% od ukupnog broja ispitanika.



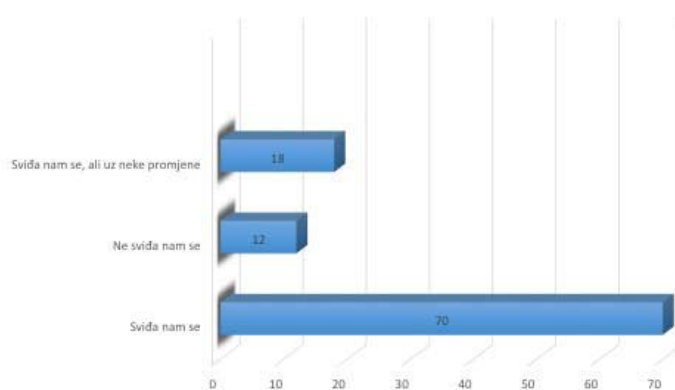
Slika 9.1 Starost ispitanika koji posjećuju web stranice

Najviše znanja o izradi internet stranica imaju ispitanici između 18 i 29 godina, dok su odmah iza njih ispitanici između 30 i 40 godina i ispitanici do 18 godina.



Slika 9.2 Znanje ispitanika o izradi i korištenju internet stranica

U trećem djelu obrađen je opći vizualni dojam o novoj internet stranici i o mogućnosti pronalaska informacija. Većina ispitanika je izrazila pozitivno mišljenje, dok odmah nakon njih slijede ispitanici kojima se stranica sviđa, no napravili bi na njoj neke izmjene.



Slika 9.3 Opći dojam ispitanika o izgledu novih stranica

10. Zaključak

Redizajn web stranice je kompleksan posao, gdje klijent i tim za izradu stranica moraju surađivati i vjerovati jedno drugome. Kvalitetnom web stranicom može se uvelike unaprijediti poslovanje tvrtke, jer ako je ona zastarjela ili loše dizajnirana često je to prvi najvažniji razlog koji odbija potencijalne klijente.

Svakodnevni razvoj i napredak tehnologija postupno je doveo do stvaranja novog modela pristupa informacijama na internetu. Sam internet više nije striktno mreža klasičnih stolnih računala, već se njemu pristupa s uređaja koji se razlikuju prema obliku, performansama i primjeni.

Upravo zbog razvoja web tehnologija stranica je prilagođena pametnih telefonima i tabletima, jer prema istraživanjima trend korištenja mobilnog weba je u velikom porastu te korisnici sve više pretražuju informacije putem pametnih telefona. Upravo iz tog razloga, veoma je važno razviti web stranicu koja je prilagođena svim uređajima. Razvojem web dizajna, nije potrebno izrađivati zasebnu mobilnu aplikaciju, već se postojeća web stranica, prilagođena za stolna računala, prilagođava veličini uređaja, odnosno veličini ekrana samog uređaja.

Osim samog vizualnog izgleda, posložene su funkcionalnosti koje koriste u rangiranju internet stranica na internet tražilicama, kako bi što bolje kotirali prilikom pretraga na istim.

Nakon završenog redizajna, dodavanja funkcionalnosti i prebacivanja, odnosno ubacivanja starog i novog sadržaja web stranica je postavljena na servere tvrtke Zimo digital, te se sada koristi kao službena web stranica tvrtke.



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, Vedran Mihalić (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Redizajn web stranica tvrtke Zimo digital (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Mihalić Vedran

(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, Vedran Mihalić (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Redizajn web stranica tvrtke Zimo digital (upisati naslov) čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

Mihalić Vedran

(vlastoručni potpis)

11. Literatura

- [1] Tim Berners-Lee, Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web, New York, 2011.
- [2] HTML and CSS: Design and Build Websites B.Jon Duckett, New Jersey, 2011.
- [3] <https://www.w3.org/> , dostupno 31.07.2016.
- [4] Erik Wilde, Wilde's WWW: Technical Foundations of the World Wide Web, New York, 1999.
- [5] <https://wordpress.org/about/>, dostupno 31.07.2016.
- [6] Jesse Friedman, Web Designer's Guide to WordPress: Plan, Theme, Build, Launch, Berkeley, 2013.
- [7] Terry Morris, Terry Felke-Morris, Basics+ of Web Design: HTML5 and CSS, Boston, 2014
- [8] Brett McLaughlin: What Is HTML5?, Sebastopol, 2011.
- [9] Eric A. Meyer: CSS and Documents, Sebastopol, 2012.
- [10] David Sawyer McFarland: CSS: The Missing Manual, Sebastopol, 2013.
- [11] Aravind Shenoy: Thinking in CSS, Birmingham, 2014.
- [12] <http://sass-lang.com/>, dostupno 02.08.2016.
- [13] <http://lesscss.org/>, dostupno 26.05.2016.
- [14] <http://koala-app.com/>, dostupno 26.05.2016.
- [15] <https://wordpress.org/about/>, dostupno 07.08.2016.
- [16] Davies Jack, Wiley Sarah: WordPress Made Super Simple How Anyone Can Build A Professional Looking, Oklahoma City, 2013.
- [17] Brad Williams, David Damstra, Hal Stern: Professional WordPress: Design and Development, New Jersey, 2013.
- [18] Caimin Jones: WordPress Websites Step-by-Step - The Complete Beginner's Guide to Creating a Website or Blog With WordPress, London, 2014.
- [19] P. Vivoda: Primjena genetskih algoritama u rješavanju PTP, Diplomski rad, FER, Zagreb, 1997.

Popis slika

Slika 2.1 osnovna struktura HTML prezentacijskog jezika.....	10
Slika 2.2 primjer strukture HTML prezentacijskog jezika.....	11
Slika 2.3 struktura naslova HTML prezentacijskog jezika.....	12
Slika 2.4 HTML unutarlinijska stilizacija	13
Slika 2.5 HTML audio i video.....	14
Slika 3.1 Css	16
Slika 3.2 SASS implementacija modula u jednu datoteku	18
Slika 3.3 SASS primjer inicijalizacije varijeble i njene primjene	19
Slika 3.4 SASS primjer grananja blokova elemenata.....	20
Slika 3.5 Primjer pisanja SASS koda [12].....	23
Slika 3.6 Primjer pisanja CSS koda [12]	23
Slika 4.1 Sadržaj style.css datoteke u WordPress projektu	28
Slika 5.1 Prijedlog dizajnerskog rješenja	29
Slika 5.2 Dizajn unutarnjih stranica	31
Slika 6.1 - Osnovna struktura index.html datoteke projekta	32
Slika 6.2 Struktura zaglavlja.....	33
Slika 6.3 Izgled zaglavlja s ubačenim logotipom i navigacijom	33
Slika 6.4 Struktura podnožja	34
Slika 6.5 Izgled programiranog podnožja.....	35
Slika 7.1 Osnovna struktura baze podataka s WordPress sustavom.....	36
Slika 7.2 Početna stranica WordPress CMS sučelja.....	37
Slika 7.3 Poveznica na stilsku datoteku.....	37
Slika 7.4 Izvorni kôd header.php datoteke projekt.....	38
Slika 7.5 Izvorni kôd footer.php datoteke projekta	39
Slika 7.6 Početak datoteke predloška	40
Slika 7.7 Dodavanje polja za predložak naslovnice	40
Slika 7.8 Primjer upisivanja PHP kôda za dodatna polja	41
Slika 7.9 Izgled popunjene naslovnice s odabranim predloškom.....	42
Slika 7.10 Dodatna polja za stavke koje se prikazuju na više mjesta.....	42
Slika 7.11 Izvorni kôd za dodatna polja koja se prikazuju na više mjesta	43
Slika 8.1 Prikaz izmjene podataka u tablicama baze.....	44
Slika 8.2 Prikaz FTP klijenta i seljenja podataka	44
Slika 9.1 Starost ispitanika koji posjećuju web stranice.....	45

Slika 9.2 Znanje ispitanika o izradi i korištenju internet stranica	46
Slika 9.3 Opći dojam ispitanika o izgledu novih stranica	46