Grafička obrada i dizajn Web stranica

Ana, Markić

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:341145

Rights / Prava: Attribution-NonCommercial-NoDerivs 3.0 Unported/Imenovanje-Nekomercijalno-Bez prerada 3.0

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ana Markić

Grafička obrada i dizajn Web stranica

ZAVRŠNI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ana Markić

Matični broj: Z-43499/14-1

Studij: Primjena informacijske tehnologije u poslovanju

Grafička obrada i dizajn Web stranica

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Dragutin Kermek

Varaždin, rujan 2019.

Ana Markić

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovim radom obraditi ću izradu vizualnog identiteta web stranice, obradu i prilagodbu slika za web te predstaviti tehnologije koje možemo koristiti kako bi svoju zamisao pretvorili u djelo. Poseban naglasak ide na HTML i CSS koji su osnova izgradnje web stranica. S obzirom da se za pristup internetu sve više koriste prijenosni uređaji postalo je pravilo da web stranice moraju biti prilagođene njima. Obraditi ću temu responzivnog/odzivnog dizajna, te predstaviti alate i načine kojima dobivamo vizualno lijepu stranicu na različitim uređajima i rezolucijama ekrana. Za kraj ću predstaviti CMS sustave i objasniti njihovu primjenu i prednosti. Posebno ću obraditi Wordpress, s obzirom da je najkorišteniji CMS sustav. Uz dodatak završnom radu priložiti ću i web stranicu koju ću sama izraditi za potrebe završnog rada. Web stranica će mi ujedno biti i prezentacija prilikom obrane završnog rada.

Ključne riječi: Web, dizajn, HTML, CSS, CMS, odzivni/responzivni dizajn

SADRŽAJ

1. UVOD …….. 1

2. UVOD U HTML i CSS 2

2.1. Povijest HTML-a i CSS-a.....3

3. PSIHOLOGIJA DIZAJNA.....5

3.1. Gestalt psihologija 5

3.2. Von Restorff efekt.....6

3.3. Psihologija boja 6

3.4. Maslowijeva hijerarhija potreba 7

3.5. Hickov zakon 8

3.6. Ostale teorije i zakoni 9

4. OBRADA I PRIPREMA SLIKA ZA WEB 10

5. HTML …….....12

5.1. Osnovni termini HTML-a.....13

5.1.1. Elementi 13

5.1.2. Atributi 14

5.2. Pravila pisanja koda i koncepti izrade 15

5.2.1. Velika i mala slova pri pisanju koda 15

5.2.2. Zatvaranje elemenata 15

5.2.3. Vrijednosti atributa unutar navodnika 15

5.2.4. Element unutar elementa.....16

5.2.5. Uređivači teksta ili editori 16

5.3. Struktura dokumenta 17

5.4. Kategorizacija elemenata 18

5.5. Blok elementi.....19

5.6. Elementi reda 19

5.7. Meta elementi.....20

5.8. Obrada teksta.....21

5.8.1. Komentari 21

5.8.2. Odlomci 21

5.8.3. Naslovi.....22

5.8.4. Novi red 22

5.8.5. Oblikovanje teksta 22

5.9. Liste 23

5.10. Rad sa grafičkim datotekama 26

5.11. Tablice 27
5.12. Obrasci ili forme 28
5.13. Struktura sadržaja 32
5.13.1. Ne-semantički strukturni elementi 32
5.13.2. Semantički strukturni elementi 33
5.14. Linkovi ili poveznice 34
6. CSS …….. 36
6.1. Sintaksa CSS-a 36
6.2. Načini pisanja CSS-a 37
6.3. Selektori 38
6.4. Svojstva 39
7. RESPONZIVNI WEB DIZAJN 40
7.1. Medijski upiti 40
8. Sustavi za upravljanje sadržajem - CMS 43
8.1. WordPress 44
8.1.1. Kako se razvijao Wordpress? 44
8.2. Preduvjeti za instalaciju i korištenje 45
8.3. Instalacija Wordpress-a 46
8.4. Izgled sustava 47
9. PRAKTIČNI DIO ZAVRŠNOG RADA 56
10. ZAKLJUČAK 56
POPIS LITERATURE 57
POPIS SLIKA 58
POPIS TABLICA 59

1. UVOD

Kada korisnik pretražuje web stranice sve što vidi je vizualna cjelina koja se sastoji od hrpe teksta i raznih multimedijalnih sadržaja kao što su slike, audio i video zapisi, animacije i slično. Da bi korisnik sve to doživio kao jednu cjelinu netko treba sve to povezati i posložiti tako da sve to bude razumljivo i lako čitljivo i najobičnijim korisnicima. Većina korisnika se vrlo vjerojatno ponekada zapita: „tko je sve ovo izmislio?“, „kako je sve ovo posloženo?“ ili „kakva je tehnologija korištena da bi se sve ovo dobilo?“. Cilj ovog rada je predstaviti i pojasniti osnove tehnologija koje se kriju iza web stranica, objasniti kada i kako se koriste i načine na koje možemo dobiti ono što smo sebi u glavi vizualizirali.

Da bi nečija ideja o web stranici postala stvarnost netko mora sve to spojiti u jednu cjelinu. U većini slučajeva web dizajneri su ti koji imaju „viziju“, a ujedno i tu viziju provedu u stvarnost tako da napišu kod i naprave funkcionalnu web stranicu. U pozadini svake web stranice, u onome što korisnici ne vide, nalazi se hrpa koda pisanog u nekom jeziku za razvoj web-a koji omogućava da stranica ima izgled i funkcionalnost – tako da je svatko može koristiti. Trenutno postoje stotine jezika kojima bi neku ideju mogli pretvoriti u stvarnost, no oni najbitniji, o kojima ovaj rad govori su HTML i CSS. To su jezici koje koriste gotovo sve web stranice i bez njih Internet kakvog ga mi poznajemo ne bi bio moguć.

Iako je ovaj rad više usmjeren na sam web dizajn, odnosno na HTML i CSS, da bi bili dobar web dizajner morate znati barem osnove grafičke obrade i prilagodbe slika za web, te osnove CMS-a (Content Management System[1]). Stoga je jedan dio rada posvećen upravo i tome, obradi slika pomoću alata za obradu, te Wordpress-u – najpopularnijem CMS-u. Rad će se osvrnuti i na psihološke aspekte dizajna, te neke od osnovnih zakona i psiholoških teorija bitnih web dizajnerima za što bolje zadovoljstvo korisnika.

[1] Prijevod s engleskog: Sustav za upravljanje sadržajem

2. UVOD U HTML i CSS

HTML (skraćeno sa eng. Hypertext Markup Language) i CSS (skraćeno sa eng. Cascading Style Sheets) su jezici koji koriste kod čitljiv Web pretraživačima (kao što su IE, Chrome, Mozilla itd.), te koji web pretraživači pretvaraju u grafički prikaz kakvog krajnji korisnik vidi. Taj kod je običnim korisnicima nečitljiv, te iz njega ništa ne bi mogli vidjeti, no ljudima koji se bave web dizajnom i razvojem web stranica to je alat koji svakodnevno koriste i koji im omogućava da izvršavaju svoje zadatke.

HTML je osnova svake web stranice, te je nemoguće napraviti web stranicu bez znanja i korištenja HTML-a. Iako se na prvu čini kompliciran i težak za naučiti, u stvari je jako jednostavan i logičan jezik koji svatko može naučiti uz odgovarajuću literaturu ili neki od tečajeva koji su danas dostupni posvuda. Ovaj rad će posebnu pažnju dati tome što je zapravo HTML, čemu služi, kako se piše, upoznavanju s njegovim tehnologijama i načinima pisanja.

Stranice pisane u čistom HTML-u imaju svoju strukturu i svoj sadržaj. No takve stranice su jednostavne – crni tekst definiranog fonta na bijeloj pozadini, uz mogućnost dodavanja slika ili nekog multimedijskog sadržaja. S vremenom je to postalo dosadno i proizvođači preglednika su htjeli tomu dodati nekakav dizajn i prilagoditi izgled tako da korisnicima bude dopadljivije i zanimljivije. Počeli su takvim web stranicama dodavati elemente i atribute, no s obzirom da ništa nije bilo standardizirano nakon nekog vremena razvojni programeri su počeli nailaziti na razne poteškoće i odlučeno je pokrenuti novi jezik kojim će se standardizirati izgled i dizajn stranica, te će se sam sadržaj odvojiti od prezentacije, odnosno izgleda. Razvijen je CSS koji definira kako će pojedini elementi iz HTML-a biti prikazani u web pregledniku i kako će sama stranica biti dizajnirana.

CSS-om je definirano sve vezano uz dizajn stranice kao što je boja, veličina ili font teksta, dimenzije pojedinih elemenata, razmake između elemenata pozicije teksta ili elemenata i slično. Važno je istaknuti da se u CSS-u na jedan element može djelovati iz više stilova tako da se svi oni stope u jednom elementu. Dolazak CSS-a je prekretnica u web dizajnu. On je web dizajnerima dao mogućnost da se bolje izražavaju i svoje vizije uvjerljivije prikažu krajnjim korisnicima, što je uvelike pomoglo i marketingu, oglašavanju i prezentaciji proizvoda ili usluga. Kao i HTML, CSS je također nezaobilazan i koriste ga sve današnje web stranice. Osoba koja se želi baviti kojom granom razvoja web-a mora znati barem osnove CSS-a.

2.1. Povijest HTML-a i CSS-a

[1] Tim Berners-Lee, tada zaposlenik CERN-a [2] u Švicarskoj, je 1991. godine napisao dokument „*HTML tags"* koji je bio prvi javno dostupan opis HTML-a. Dokument je sadržavao početni HTML, odnosno 20 elemenata jednostavnog HTML-a kojima su neke stvari prvi put bile standardizirane. Berners-Lee se također smatra i izumiteljem WEB-a (World Wide Web), a uz to je i 1994. godine pokrenuo World Wide Web Consortium (poznatije kao W3C) kojem je osnovna zadaća standardizacija tehnologija koje se koriste na web-u. Od tada se konstantno radilo na HTML-u i izlaskom svake sljedeće verzije dodane su nove mogućnosti te su se neke stvari pojednostavnile.

- Prva službena verzija HTML-a je izbačena 1993. godine. Cilj prve verzije je bio jednostavnije strukturiranje podataka, odnosno teksta. Tada se nije pridavalo puno značaja samoj prezentaciji sadržaja, no svaka nova verzija je donosila neka poboljšanja i novosti:
- Verzija 2.0 je bila prva imenovana verzija, no ni ona nije bila dovoljna da bi se proglasila standardnom
- 1995. godine W3C objavljuje prvu radnu verziju nazvanu HTML 3.0 kojoj je glavna novost i značajka bila mogućnost definiranja tablica
- 1997. godine sa nadopunama se izbacuje HTML3.2 koji postaje prva službena verzija
- U prosincu iste godine je predstavljen i HTML4, da bi 2 godine kasnije bila izbačena i konačna verzija ovog jezika – HTML4.01
- 2000. godine predstavljen je XHTML koji je zapravo kombinacija XML[3] i HTML. XHTML je skup pravila XML-a primijenjen na HTML-u
- 8 godina kasnije, odnosno 2008. godine W3C objavljuje i aktualnu verziju HTML5 – nastao je u suradnji W3C-a i HWATWG-a[4]

[2] Međunarodna institucija za nuklearna istraživanja kojoj su svrha fundamentalna istraživanja (bez izravnih tehnoloških ili komercijalnih ciljeva) na području fizike elementarnih čestica, osnovana 1954.
[3] Extensible Markup Language – meta jezik koji je jednostavno sredstvo za stvaranje drugih jezika
[4] Web Hypertext Application Technology Working Group – prije ujedinjenja sa W3C radili su sa web formama i aplikacijama

- 2014. godine kada je razvoj dovršen HTML5 je postao i službeni HTML standard. Neke od najvažnijih značajki koje je donio su:
    - Semantički elementi – detaljnije definiranje sadržaja
    - Bolje povezivanje s poslužiteljem
    - Omogućena pohrana podataka lokalno na strani klijenta – bolji izvan mrežni rad
    - Prikaz multimedijalnog sadržaja bez dodataka
    - 3D grafički elementi
    - Optimiziranije korištenje hardware-a i bolje performanse
    - Pristup različitim I/O uređajima
    - Izrada sofisticiranijih vizualnih tema

Kako je već ranije spomenuto da je HTML-u nedostajala stavka dizajna i izgleda, odnosno prilagodljivosti za krajnje korisnike, odlučeno je da se taj zadatak treba dati nekom drugom jeziku. Danas se za to koristi isključivo CSS koji je nastao kombinacijom CHSS-a[5] i SSP-a[6]. [2] Prije CSS-a postojali su mnogi nestandardizirani stilski jezici, no ova dva su se pokazala kao najučinkovitiji i njihovom kombinacijom i razvojem nastao je CSS koji se i danas koristi. 1993. godine je W3C preuzeo zadatak razvoja CSS-a da bi konačno 1996. bila predstavljena prva standardizirana verzija nazvana CSS1.

Vrlo brzo, samo nekoliko mjeseci kasnije, 1997. je izbačena i radna skica CSS2, a kao službeni standard se počeo primjenjivati 1998. godine. S verzijom 2 CSS je uveo dodatne mogućnosti i prvi put je ušao u širu primjenu diljem svijeta. Već godinu dana kasnije pojavljuje se i prva radna skica CSS3 standarda. Razlika od prijašnjih verzija je to što je CSS3 podijeljen na više odvojenih modula. 2012. godine je bilo izdano više od 50 različitih modula od kojih je svaki zadužen za određene funkcionalnosti. Četiri modula su dobila službenu preporuku za korištenje: Medijski upiti, imenski prostori, selektor razine 3 i boja, dok je određeni broj ostalih modula dobio status preporuke kandidata (eng. Candidate Recommendation status), što razvojnim programerima govori da ih se smatra umjereno stabilnima pri korištenju. Iako je CSS3 aktualni standard i dalje se radi na razvoju i novim mogućnostima.

[5] Cascading HTML Style Sheets
[6] Stream-based Style Sheet Proposal

3. PSIHOLOGIJA DIZAJNA

Kako bi se netko mogao uspješno baviti dizajnom (ne samo web, nego bilo kakvom vrstom dizajna) potrebno je jednim dijelom biti i umjetnik. Dizajnirati nešto nije samo posložiti tekst i elemente tako da budu ugodni oku, da ih je lijepo vidjeti. U dizajnu, posebice web dizajnu, pažnju je potrebno obratiti na puno toga i postaviti neka pitanja:

- Kakve je tematike web stranica koja se izrađuje?
- Kojeg profila je ciljana skupina kojoj web stranica mora biti prezentirana?
- Kakve osjećaje treba izazvati web stranica kod ciljane skupine?
- Kojeg profila je klijent za kojeg izrađujemo projekt?
- Zna li klijent što želi, ima li nekakvu viziju? Ako ima svoju viziju, je li ona u skladu sa standardima i mišljenjima struke?
- Kako klijent razmišlja, odnosno koji je cilj koji klijent želi postići, zašto treba dizajnera?

Da bi dizajner mogao dobiti odgovore na navedena pitanja, a i neka koja će se postaviti u procesu izrade, mora biti upoznat s psihologijom čovjeka. Kako je psihologija čovjeka vrlo široko područje, možda čak i najšire znanstveno područje, s vremenom i razvojem web-a stvorile su se razne teorije, principi i standardi koje bi svaki dizajner trebao znati prije nego krene raditi sa klijentima na stvarnim projektima.

3.1. Gestalt psihologija

Ljudski mozak konstantno stvara poveznice između informacija. Tijekom godina istraživanja tim njemačkih psihologa je 1920. napisao tekst o načinu na koji ljudski mozak stvara relacije među informacijama. U tekstu navode kako na primjer za elemente koji su slični i fizički u blizini mozak odmah stvara pretpostavku da su na neki način povezani, iako to ne moraju nužno i biti. Također ako su neki elementi slični ljudski mozak ih odmah povezuje, dok ako su različiti ljudski mozak pretpostavlja da nemaju nikakvih poveznica. Pošto je ljudski mozak sklon stvaranju

značenja na temelju pretpostavki treba biti jako oprezan pri raspoređivanju elemenata kako bi se postigao željeni efekt kod krajnjeg korisnika. [3]

3.2. Von Restorff efekt

Što se neki element više razlikuje od drugih to će se više istaknuti i krajnji korisnici će ga više pamtiti. Psihologinja Hedwig von Restorff je 1993. provodila testiranja kako bi utvrdila ovu činjenicu. Izvodila je to na način da bi ispitaniku dala listu podataka, od kojih bi neke podatke istaknula na način da ih podcrta, zaokruži ili na bilo koji način istakne među ostalima. Svi takvi podatci su kod svih ispitanika bili zapamćeni u većoj mjeri. Ovaj efekt se počeo primjenjivati u dizajnu, posebice kod oglašivača, tako da korisnici više pamte i šire informacije o pojedinim elementima. Ovo je također jako bitno i kod digitalnog marketinga, na taj način se neki proizvodi ili usluge ističu da bi lakše došli do korisnika. Uzmimo za primjer *Google ads*[7] gdje su pojedini pojmovi unutar teksta podcrtani, ispisani drugom bojom ili slično samo kako bi korisnici kliknuli na link. [4]

3.3. Psihologija boja

[5] Boje su oduvijek moćan alat kada pričano o ljudskoj psihologiji. Ljudski mozak različito reagira na pojedinu boju, tako da se boje mogu koristiti i kao alat u marketingu. Određena boja kod ljudi može izazvati određene emocije, promijeniti raspoloženje, potaknuti da naprave nešto ili povezati s nekim proizvodom, događajem ili robnom markom. Na primjer ako nekoga pitate da bez razmišljanja navede kompaniju koju povezuje sa žutom i crvenom bojom, velika većina ljudi će reći McDonalds. Ako pitamo nekoga s kojom bojom bi povezao Facebook, odgovor će uvijek biti plava.

Odabir boje je najbitniji kod izrade loga i vizualnog identiteta. S bojama treba biti i oprezan. Nikada ne bi trebali koristiti previše boja. Dvije ili tri boje su sasvim dovoljne da se uspješno predstavi neki proizvod ili usluga, tako da ni web stranica ne bi trebala biti šarena, dovoljno je koristiti dvije ili tri boje.

[7] Google platforma pomoću koje pružatelji usluga plaćaju Google-u da njihove proizvode ili usluge istakne unutar svoje tražilice

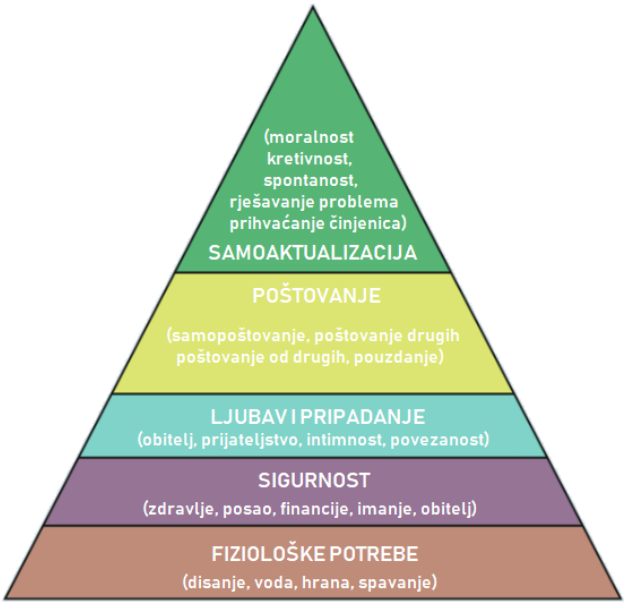
Različite boje predstavljaju i različite karakteristike. Adobe je radio na istraživanju psihologije boja i na temelju rezultata su predstavili listu[8] doživljaja boja koja se koristi u većini marketinških i dizajnerskih kampanja:

- Crna – sofisticiranost, moć
- Bijela – čistoća, sofisticiranost, čednost, nježnost
- Crvena – moć, snaga, opasnost, hrabrost, simulira apetit
- Plava – smirenost, mir, povjerenje, sigurnost
- Žuta – optimizam, sreća
- Zelena – balans, rast, prirodno, svježe, organsko
- Ljubičasta – kreativnost, maštovitost, luksuz, kraljevstvo, nostalgija
- Narančasta – prijateljstvo, ugoda, hrana
- Ružičasta – ženstvenost, romantika, spokoj

3.4. Maslowijeva hijerarhija potreba

Nakon što je psiholog Abraham Maslow objavio svoje viđenje ljudskih potreba i objavio piramidu psiholoških potreba ljudskog bića ona se široko počela koristiti i postala je jedan od temelja moderne psihologije. Piramida govori o tome da bi čovjek mogao postići zadovoljstvo u nekom segmentu prvo mora dostići samo aktualizaciju u prethodnim stupnjevima. Na primjer da bi osoba osjetila pravu ljubav prije mora imati ispunjene sve fiziološke potrebe, te osjećati se sigurnom itd.

[8] https://blogs.adobe.com/creativecloud/the-psychology-and-emotion-behind-color-in-web-design/



Slika 1: Maslowijeva hijerarhija [9]

Tom hijerarhijom mogu se izazvati i određeni osjećaje i emocije koji će na kraju rezultirati zadovoljnijim krajnjim korisnicima. [6]

3.5. Hickov zakon

Ukoliko osoba ima puno izbora automatski će joj trebati i više vremena da donese neku odluku. To nekada može dovesti i do toga da na kraju odluku i ne donese zbog viška informacija. Kada se radi dizajn sučelja stranice potrebno je pobrinuti se da korisnik dobije što jednostavnije iskustvo. Potrebno je dizajnirati stranicu tako da se do određene informacije može doći u što manje koraka i da se ne gubi puno vremena za stvari koje se mogu pojednostavniti. Ako je potrebno korisnika navesti da nešto napravi to bi trebalo biti u ne više od dvije radnje, tako da mu se na jasan način da do znanja što se želi postići s onime što se traži od njega. [7]

[9] Izvor slike: https://www.psychologytoday.com/us/blog/hide-and-seek/201205/our-hierarchy-needs

3.6. Ostale teorije i zakoni

[4] Psihologija je jako široko područje i moglo bi se pisati stotine stranica, a na kraju opet ne reći sve. Osim gore navedenih dobro je znati i još neke metode i zakone:

- Prepoznavanje lica – ljudski mozak je sklon prepoznavanju lica, odnosno emocija koje to lice prikazuje. Tako se mogu koristiti snimke, ilustracije ili fotografije lica kako bi se korisnicima prenio određeni osjećaj ili emocija.
- Teorija dvostrukog kodiranja – ljudi bolje pamte članke i tekstove gdje osim samog teksta postoji i vizualni prikaz željenog sadržaja. Gdje god se može koristiti dobro je ubaciti i nekakve vizualne elemente kako bi potaknuli memoriju kod korisnika.
- Fittov zakon – ukratko govori da bi u web dizajnu trebalo korisniku biti jednostavno i brzo kliknuti na bilo koji gumb ili link kako bi imao dobro iskustvo korištenja web stranice. Na primjer ponuda za pretplatu na određenu uslugu bi trebala biti jako blizu i samog gumba za pretplatu jer povećava šanse da će se korisnik odlučiti kliknuti na njega. Ako je već kliknuo i same šanse za pretplatom su znatno veće.
- Occam's razor – najjednostavnije rješenje je u većini slučajeva i najbolje. Nije potrebno zatrpavati stranicu nebitnim i pretjeranim sadržajem.

4. OBRADA I PRIPREMA SLIKA ZA WEB

Slike za web se ne razlikuju ni po čemu od slika za bilo koji drugi medij ili oblik. Dolaze u različitim formatima, veličinama ili rezolucijama, jedino na što je potrebno znati su nekoliko trikova pomoću kojih će te slike biti optimizirane za prikaz na webu, da se što bolje i brže učitavaju, te da su prilagođene zaslonima različitih uređaja. Za obradu slika kao početnik nije potrebno puno znanja i vještina, a sam proces je brz i jednostavan. Profesionalci koji se bave isključivo ovom vrstom posla uglavnom koriste Photoshop. To je moćan alat koji daje svakakve mogućnosti obrade i prilagodbe slika, no za osnovne stvari koje su potrebne u web dizajnu dovoljan je i neki od besplatnih alata kao što su Google Picasa, GIMP ili PhotoPlus. Svi oni nude mogućnost promjene veličine slike, konvertiranje slike u drugi format, dodavanje efekata ili rezanje dijelova slike. Sve nabrojeno je sasvim dovoljno za početak obrade slika i prilikom prilagodbe gotovih slika za web drugi alati nisu niti potrebni. Web dizajneri često surađuju sa grafičkim dizajnerima koji ovaj posao obave umjesto njih, ali svaki web dizajner mora znati barem osnove. U nekim situacijama kada su potrebne sitne prilagodbe nema smisla zamarati grafičkog dizajnera, a i brže je da sami obavite dio posla.

U digitalnom svijetu postoji mnoštvo formata slika, no u webu se najčešće susrećemo s JPEG, GIF ili PNG formatima. GIF slike su u zadnje vrijeme sve manje popularne i preuzimaju ih slike u PNG formatu. Najveći razlog tome je što GIF format koristi samo 256 boja, pa stoga ne može ni prikazati sve detalje neke bolje fotografije. JPEG format koristi Lossy kompresiju podataka kojom se gubi na kvaliteti slike, no za korisnika će u većini slučajeva taj gubitak kvalitete biti neprimjetan. PNG je noviji format koji koristi kompresiju bez gubitaka, a može pohraniti više boja od JPEG-a pa je iz tog razloga i sama slika ljepša i izgleda kvalitetnije. JPEG je i dalje više zastupljen na webu zato što je kompaktniji, zauzima manje prostora na računalu, a kvaliteta je dovoljna u većini slučajeva. Ponekada se može susresti i sa TIFF formatom koji je zamišljen da se koristi u novinama, časopisima te medijima sa kombinacijom teksta i slika. TIFF ima mogućnost koristiti obje kompresije, lossy (s gubitcima) i lossless (bez gubitaka), a kvaliteta slika u ovom formatu je vrlo visoka.

Bitnije od formata slike je rezolucija, posebno ako se rade odzivne web stranice. Ovisno o vrsti uređaja, odnosno veličini zaslona na kojem će se prikazivati slika treba prilagoditi i rezoluciju te slike. Rezolucija je broj pixela koji se može prikazati na zaslonu, a definirana je širinom i visinom u pixelima. Danas su ekrani većinom u HD (High Definition) rezolucijama, pa se i web stranice,

odnosno slike na njima, prilagođavaju njihovim rezolucijama. Neke od poznatijih standardnih rezolucija su HD (1280 x 720), Full HD (1920 x 1080) i Ultra HD (4k, 3840 x 2160). Kod odabira rezolucije slike na web stranici treba pripaziti da slika ne bude niti mala niti prevelika s obzirom na standardne veličine ekrana. Trenutno su najzastupljeniji Full HD ekrani, no sve više uređaja dolazi sa 4k ekranima, pa će s vremenom zauzimati sve veći postotak na tržištu. Prije odabira rezolucije i prilagodbe fotografije dobro je napraviti kratko istraživanje kakvi su ekrani u tom trenutku najzastupljeniji na tržištu.

Na kraju postoje još i razni automatizirani alati unutar alata za obradu slika koji će pomoći da slika izgleda bolje i atraktivnije, te da je korisnik doživi na najbolji mogući način. Ako se koriste filteri za mijenjanje boja, izoštravanje slika, ili pojačavanje svjetla, treba paziti da slika nije previše različita od originalne, jer samim uređivanjem slika može izgubiti na kvaliteti, no za to nema posebnog pravila nego je sliku potrebno urediti po instinktu i osobnoj procjeni.

5. HTML

Kao što je ranije u radu spomenuto HTML je skraćenica za Hypertext Markup Language. Za lakše shvaćanje samog naziva slijedi pojašnjenje njegovih elemenata.

Hypertext bi u računalnom svijetu bio tekst koji se sastoji od više povezanih skupina informacije prikazanih na računalu ili nekom drugom elektroničkom uređaju. Hypertext se od klasičnog teksta razlikuje u tome što se ne čita od reda (s lijeva na desno, od gore prema dolje) nego redoslijed izlistavanja, odnosno čitanja informacija krajnji korisnik određuje dinamički za vrijeme čitanja. Po tome bi Hypertext bio ne sekvencijalan, dok se obični tekst može nazvati sekvencijalnim.

Markup Language (prijevod: jezik za označavanje) je također jasno samo po sebi –HTML je jezik u kojemu se može označiti dio teksta tako da ih internetski preglednici mogu tumačiti na određeni način.

U konačnici HTML se tumači kao računalni jezik koji govori internetskim preglednicima kako i na koji način da prikažu određeni HTML dokument koji je dizajner ili razvojni programer napisao.

5.1. Osnovni termini HTML-a

5.1.1. Elementi

Su blokovi od kojih se sastoji HTML kod, odnosno to su osnovni građevni blokovi HTML dokumenta. Oni određuju na koji će način internetski preglednici prikazivati određene dijelove sadržaja.

HTML elementi se prepoznaju po tome što imaju početnu i završnu oznaku između kojih se nalazi sadržaj elementa. Oznake ili tagovi[10] uvijek se nalaze između znakova < i >, iz to tako da završna oznaka sadrži i znak /. Primjer:

```
<> Sadržaj </>
```

Početna oznaka u primjeru iznad je < > dok je završna </ > . Cijeli dio koda je jedan HTML element.

Kada bi oznake bile prazne internetski preglednici ne bi imali dovoljno informacija i ne bi znali što moraju napraviti s tim elementom. Iz tog razloga svaka početna i završna oznaka mora sadržavati i identifikator HTML elementa kako bi mu bila dodijeljena određena uloga. Identifikator u početnoj oznaci se upisuje u formatu: <identifikator>, dok se u završnoj oznaci upisuje nakon kose crte: </identifikator> i mora biti jednak kao i u početnoj oznaci. Za različite vrste informacija koje se trebaju prikazivati postoje i različiti identifikatori, stoga postoje identifikatori za odlomak teksta, naslov, poveznicu ili link, sliku ili slično. Na primjer za odlomak teksta koristi se identifikator *p* (eng. Paragraf), za glavni naslov *h1* (eng. Header 1) , za podnaslov *h2* (eng. Header 2). Primjeri:

```
<p> Tekst koji se ispisuje </p>
   <h1> Glavni naslov </h1>
     <h2> Podnaslov </h2>
```

Osim elemenata sa sadržajem koji su navedeni do sada, postoje i elementi bez sadržaja. O njima, kao i o svim ostalim identifikatorima elemenata će biti govora u ostatku rada.

[10] Oznaka – engleski prijevod je tag

5.1.2. Atributi

Neki elementi imaju mogućnost prenijeti dodatne atribute koji internetskim preglednicima daju dodatne informacije o samom elementu. Atributi su uglavnom neobavezni i koriste se samo onda kada postoji potreba za njima, ali postoje i oni obavezni koji se moraju pisati kako bi kod bio funkcionalan.

Atributi se uvijek pišu unutar početne oznake nakon identifikatora i sastoje se od imena atributa i vrijednosti atributa napisane unutar navodnika. Ime atributa i vrijednost su odvojeni znakom jednako „=“. Primjer:

```
<identifikator atribut="vrijednost"> Sadržaj </identifikator>
```

Ako je potrebno pisati više atributa pišu se jedan za drugim, obavezno odvojeni razmakom.

Za lakše shvaćanje može se navesti primjer, kako bi se u kod ubacio link na *www.google.com* , uz to da link ne pokazuje sam URL[11], nego ispisuje tekst „Ovo je link na Google“ koji će korisnika ako klikne na njega odvesti na lokaciju.

```
< a href="https://www.google.com"> Ovo je link na Google </a>
```

U primjeru iznad identifikator <a> se koristi za hiperlinkove, dok je *href* atribut kojemu je dodijeljen URL dokumenta ili stranice na koji se želi odvesti korisnika.

[11] Uniform Resource Locator. URL je putanja do određenog sadržaja na Internetu te se obično naziva poveznica, ponekad i mrežna adresa. – wikipedia.hr

5.2. Pravila pisanja koda i koncepti izrade

Do dolaska HTML5 pravila pisanja dokumenata su bila dosta stroga, no HTML5 je omogućio da web dizajneri budu fleksibilniji i dopušta da se neke stvari pišu na više načina. Iako je fleksibilan svaki razvojni programer ili dizajner će preporučiti da se zbog estetike koriste neka pravila pisanja, tako da je dosta pravila i dalje zadržano iz prijašnjih verzija, osobito iz XHTML-a. Da bi kod bio čitljiviji i razumljiviji, posebice ako na kodu radi više od jedne osobe, navesti ću neka preporučena pravila kojih bi se trebali pridržavati:

5.2.1. Velika i mala slova pri pisanju koda

Iako HTML5 dopušta da se pri pisanju koda kombiniraju velika i mala slova, preporučeno je da se sve piše malim slovima. Kod napisan velikim slovima izgleda neuredno i zauzima više prostora na zaslonu ekrana.

5.2.2. Zatvaranje elemenata

Završne oznake elemenata u HTML5 također nisu obavezne, dok su u prijašnjim verzijama bile. Iako to iziskuje više tipkanja puno lakše je pratiti i snaći se u kodu sa zatvorenim završnim oznakama. Također ukoliko web dizajner radi na velikom kodu može se desiti da zbog nedostatka završnih oznaka kod bude pogrešno interpretiran, te će biti puno teže uočiti pogreške i ispraviti ih.

5.2.3. Vrijednosti atributa unutar navodnika

Također kao i u prethodnim primjerima, HTML5 nam za razliku od prijašnjih verzija dopušta da se vrijednosti atributa pišu odmah iza znaka jednakosti, bez obveze otvaranja i zatvaranja dvostrukih navodnika.

5.2.4. Element unutar elementa

U HTML-u jedan element može biti sadržaj drugog elementa i to je nekada neophodno prilikom pisanja koda. Na engleski se ovaj postupak naziva *nesting* , a prijevod na hrvatski bi bio ugnježđivanje. Prilikom ugnježđivanja elemenata najbitnija stvar na koju je potrebno obratiti pažnju je redoslijed otvaranja, odnosno zatvaranja pojedinog elementa. Elementi moraju biti zatvoreni onim redoslijedom kojim su i otvoreni, po principu 1,2,3 – 3,2,1. Primjer:

```
<p><strong><em> Nekakav sadržaj </em></strong></p>
```

U gornjem primjeru paragraf „Nekakav sadržaj“ je podebljan (identifikator *strong*) i nakošen (identifikator *em*).

Postoje i 3 osnovna koncepta izrade web mjesta:

- Odvojena struktura HTML dokumenta od prezentacije i stila – za prezentaciju i stil preporučuje se koristiti liste stilova, to je u današnjici CSS
- Pristupačnost svim korisnicima – ovdje se mora uzeti u obzir različita govorna područja na koja se cilja ili uređaje s kojih će korisnici pristupiti web stranici. Jedan od primjera pristupačnosti je i responzivni dizajn koji omogućuje da se web stranica prilagodi za razne rezolucije ekrana, tako da se stranica ispravno prikazuje bez obzira s kojeg uređaja se otvara
- Omogućiti što brže otvaranje stranica u web pretraživačima – pisati kod tako da je što čitljiviji pretraživačima i da ga što brže i sa što manje resursa otvaraju. HTML5 je tu doveo novost sa semantičkim elementima koji definiraju strukturu sadržaja stranice, što se nekada radilo sa tablicama.

5.2.5. Uređivači teksta ili editori

Pošto je HTML samo hrpa koda u tekstualnom obliku, koji pretraživači pretvaraju u ono što korisnik vidi, za pisanje koda dovoljan je i najjednostavniji uređivač teksta ili editor koji omogućava unos teksta bez formatiranja. Jedan od najpoznatijih takvih editora dolazi već pred instaliran u Windowsima i to je Notepad. No s obzirom da Notepad nudi samo osnovne funkcije unosa teksta za ugodniji i brži rad koriste se drugi većinom besplatni editori. Editori poput Sublime, Vim ili Notepad++ imaju ugrađene funkcije koje pomažu prilikom pisanja koda, kao što su automatska nadopuna, ispis broja retka u kojem se nalazi pokazivač, automatsko strukturiranje koda i slično. Osim čisto tekstualnih uređivača postoje i WYSIWYG uređivači. Razlika od

tekstualnih uređivača je u tome što ovdje korisnik vidi krajnji rezultat odmah, umjesto da prvo mora pisati kod čiji učinak će se vidjeti tek kasnije. WYSIWYG je skraćenica od engleskog izraza „What you see is what you get“, što bi u prijevodu značilo „ono što vidiš, to ćeš i dobiti”. Najpoznatiji takvi uređivači HTML koda su Microsoft-ovi proizvodi Expression Web i Visual Web designer, te Adobe-ov Dreamweaver.

5.3. Struktura dokumenta

Prije početka interpretacije HTML koda potrebno je pretraživačima dati do znanja da se uopće radi o HTML-u. Za sami početak potrebno je ispuniti 2 stvari:

1. Da je kreiran HTML dokument – u editoru otvoriti novu datoteku i spremiti je s ekstenzijom .html. Početna stranica uglavnom ima naziv index.html i preporuka je da početna stranica uvijek ima takvo ime, sve ostale se mogu nazvati po potrebama i željama klijenta ili korisnika

2. Svaki HTML dokument mora sadržavati određene elemente koji samom pregledniku govore da dokument interpretira kao HTML. Osnovna struktura dokumenta izgleda ovako:

```
<!doctype html>
     <html>
             <head>
             <title> </title>
             </head>
             <body> </body>
     <html>
```

Elementi iz primjera moraju se nalaziti u svakom HTML dokumentu, jer bez njih preglednici ne znaju da se radi o HTML-u. Detaljnije pojašnjenje po elementu slijedi:

<!doctype html>

Ovo je jedini element iz osnovne strukture koji nema završne oznake i dokument mora početi uvijek s njim. U starijim verzijama HTML-a doctype se pisao nešto drugačije. Doctype govori pregledniku o kojoj verziji HTML-a se radi, i napisano kao u primjeru internetski preglednici će shvatiti da je riječ o HTML5 i tako i izvršavati kod.

<html>

Govori pregledniku da sve ono što se nalazi unutar elementa je HTML kod. Na kraju dokumenta dolazi završna oznaka kao što se može vidjeti u primjeru.

<head>

Ovaj element se ne prikazuje u pregledniku, nego samo koristi pretraživačima i tražilicama kao izvor dodatnih informacija o samom dokumentu. Unutar elementa head mogu se pisati razni drugi elementi, ovisno o potrebama, no jedini obavezni element je title.

<title>

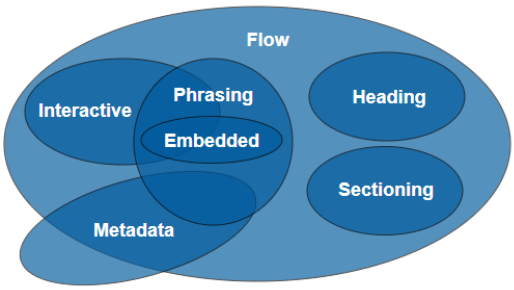
Unutar ovog elementa definiran je naziv HTML dokumenta koji se prikazuje u kartici web preglednika.

<body>

Unutar elementa body nalazi se sve ono što se prikazuje u pregledniku. Doslovno prevedeno s engleskog, sve unutar body elementa je tijelo web stranice.

5.4. Kategorizacija elemenata

Unutar HTML5 postoje kategorije u koje su posloženi elementi sličnih karakteristika. Tako su elementi podijeljeni po kategorijama *Flow, Metadata, Sectioning, Heading, Phrasing, Embedded* i *Interactive.* Neki elementi mogu istovremeno pripadati u više kategorija. Elementi svih kategorija osim *Metadata* ujedno pripadaju i kategoriji *Flow*. Jedino kategorija *Metadata* sadrži neke elemente koji nisu dio kategorije *Flow*.



Slika 2: Kategorizacija elemenata [12]

[12] Izvor: https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html

S obzirom da je ova kategorizacija dosta kompliciranа za shvatiti, posebno za početnike, jednostavnije je slijediti kategorizaciju u samo dvije osnovne skupine, a to su elementi bloka i elementi reda.

5.5. Blok elementi

Ovi elementi nisu nužno vidljivi u pregledniku, no najbitnija karakteristika kod njih je da zauzimaju 100% širine elementa u kojem se nalaze (roditeljskog elementa). Blok elementi se uglavnom koriste za naslove, odlomke, zaglavlja ili slično. Jedan od primjera je element <p>. Ako je kod napisan u formatu:

```
Ovo je prvi dio teksta <p> dok je ovo drugi dio </p> a ovo treći
```

Internetski preglednik će to prikazati ovako:

Ovo je prvi dio teksta

Dok je ovo drugi dio

A ovo treći

Zašto je u kodu sve pisano u jednom redu, a preglednik je prikazao u više redova? Upravo zbog toga što je <p> element ujedno i blok element, te zauzima 100% širine roditeljskog elementa i ne dopušta drugom elementu da se prikazuje pored njega.

5.6. Elementi reda

To su elementi kojima je širina određena širinom sadržaja koji se nalazi u njima. Oni će dopustiti prikazivanje sadržaja iz drugih elemenata pored sebe. Elementi reda mogu se pisati unutar elemenata bloka, dok obrnuto ne. Primjer takvog elementa je <u> koji se koristi za podcrtavanje teksta (skraćeno od eng. underline). S njim se može podcrtati neki dio teksta i ne bi imalo smisla da se za neki dio istaknutog teksta koristi nova linija unutar preglednika.

5.7. Meta elementi

Meta elementi se uvijek pišu u zaglavlju stranice, a kao što je spomenuto ranije sve u zaglavlju stranice ostaje „skriveno“ odnosno ne prikazuje se u pretraživaču, tako ni meta elementi nisu vidljivi korisnicima. Meta elementi su definirani elementom <meta> koji nema sadržaja niti mu je potrebna završna oznaka da bi radio. Oni se koriste da bi detaljnije opisali dokument, a najčešće da pretraživačima i tražilicama damo informacije o načinu kodiranja stranice (da bi stranica prikazivala pojedine znakove iz određenih pisama i jezika), opisu stranice, ključnim riječima za pronalazak stranice u tražilicama, podatcima o autoru ili preusmjeravanju i automatskom osvježavanju web stranice. Meta element se piše tako da mu se doda atribut ovisno o vrsti informacije koju web dizajner želi poslati pretraživaču.

Dolaskom Unicode[13] standarda za prikaz slova više nema potrebe za starijim standardima, tako da se sada koristi standard UTF-8 koji će prikazati sva slova sa dijakritičkim znakovima onako kako ih korisnici trebaju vidjeti.

Dakle, kako bi se svi dijakritički znakovi ispravno prikazivali neovisno o jeziku, potrebno je u zaglavlje koda dodati meta element sa atributom *charset* i dodijeliti mu vrijednost UTF-8. U kodu to izgleda ovako:

```
<head>
        <meta charset="utf-8">
        <title>Web stranica</title>
</head>
```

Zatim se meta elementom može detaljnije opisati stranicu, i to tako da se dodaju sljedeći atributi:

```
<head>
        <meta name="description" content="opis sadrzaja">
        <title>Web stranica</title>
</head>
```

Postoje razni meta elementi koji se mogu koristiti, a može se pisati i više meta elemenata zajedno.

[13] Detalje o Unicode standardu možete naći na www.unicode.org

5.8. Obrada teksta

Kako bi sadržaj stranica u web pregledniku bio prikazan na pravilan način, odnosno na način na koji je dizajner zamislio da bude prikazan u HTML-u se oblikuje pomoću određenih elemenata. Osnovna kategorizacija prikaza sadržaja dijeli se na tekstualni, grafički (slike i fotografije), multimedijalni (audio i video zapisi) i poveznice na druge stranice. Veliku većinu sadržaja ipak sačinjava tekstualni sadržaj. Kako je već ranije spomenuto, sav sadržaj web stranice koji će se prikazati u web pregledniku mora se nalaziti unutar <body> elementa. Da bi tekst bio prikazan na web stranici dovoljno ga je samo napisati unutar <body> elementa, no da bi bio formatiran i ispisan na način na koji dizajner želi potrebno ga je oblikovati adekvatnim HTML elementima, koji će detaljnije biti obrađeni u daljnjem tekstu.

5.8.1. Komentari

Komentari služe uglavnom kao dodatna pojašnjenja napisanog koda za druge osobe koje će raditi na kodu, ili za samog web dizajnera ili programera, kako bi mu kod kasnije bio jasniji. Oni nisu vidljivi u web pregledniku, nego ostaju skriveni unutar HTML koda i moguće ih je čitati samo proučavanjem koda. U programiranju se uvijek mogu sresti komentari jer se time olakšava rad sa kodom, posebno ako je uključeno više ljudi. U HTML-u se komentari pišu unutar oznaka <!-- i --> kao na primjer:

```
<body>
        <!-- Tekst komentara -->
</body>
```

5.8.2. Odlomci

Pod blok elementima spomenut je element <p>. Iako tamo nije naglašeno, on se koristi za definiranje odlomaka (eng. Paragraph). Odlomci se koriste kako bi se dobio ljepše oblikovan i pregledniji tekst. Odlomci su u pregledniku prikazani kao odvojene cjeline teksta.

5.8.3.Naslovi

Uređivanje teksta je u HTML-u slično kao i bilo gdje drugdje, pa tako i u klasičnim programima za obradu teksta kao što je MS Word ili slično. Naslovi služe da bi se lakše orijentirali u nekom tekstu, a kao i svuda ostalo i u HTML-u postoji hijerarhijska podjela naslova, i to u 6 razina. Za naslove se koriste oznake <h1> , <h2>, <h3>, <h4>, <h5> i <h6> uz to da je <h1> najviša razina naslova, a <h6> najniža razina. U dokumentu bi svaka cjelina sadržaja trebala imati naslov prve razine, dok se ostale razine koriste po potrebi, ovisno o tipu teksta i sadržaja. Pošto su naslovi dio grupe blok elemenata oni moraju obavezno biti zatvoreni završnom oznakom:

```
<h1> Glavni naslov </h1>
<h2> Podnaslov </h2>
```

5.8.4.Novi red

Element <br> se koristi bez sadržaja i bez završne oznake, a on web pregledniku govori da sav kod iza njega ispisuje u novi red. U kodu se dodaje gdje god preglednik treba sadržaj prikazati u novom redu.

5.8.5.Oblikovanje teksta

Također kao i kod klasičnog uređivanja teksta i u HTML-u se slova mogu prikazati zadebljana, ukošena, podcrtana ili slično. Za prikaz specifičan prikaz slova u pregledniku postoje elementi koji se pišu u HTML kodu i omogućavaju prikaz teksta kako je dizajner to zamislio. Elementi za ovakvo uređivanje su elementi reda, stoga moraju sadržavati poletnu i završnu oznaku. Završna oznaka je kao i do sada slična početnoj, razlikuje se jedino po znaku / iza znaka < (npr. : je početna, dok je završna oznaka).

**** **- (eng. Bold) koristi se za podebljani tekst.**

**** **- koristi se za važan tekst. U pregledniku je prikazan kao podebljan.**

<i> *- (eng. Italic) koristi se za ukošeni tekst (drugi naziv je kurziv).*

<mark> - istaknuti tekst. Obilježen drugom bojom pozadine

<small> - koristi se za umanjeni tekst, fusnote ili slično

<del> - ~~koristi se kao izbačeni sadržaj. U pregledniku prikazan kao precrtan.~~

<ins> - koristi se kao ubačeni tekst. U pregledniku će biti prikazan kao podcrtan

5.9. Liste

Četiri osnovna tipa listi su:

- Liste bez rednih brojeva
- Liste sa rednim brojevima
- Liste za definiranje
- Ugniježđene liste

Liste bez rednog broja sadrže početnu i završnu oznaku <ul> koja označava početak i kraj liste, a unutar toga elementa se pišu početne i završne oznake <li> koje označavaju pojedini element liste:

```
<ul>
        <li> Element 1 </li>
        <li> Element 2 </li>
        <li> Element 3 </li>
</ul>
```

Lista će u pregledniku biti prikazana ovako:

- Element 1
- Element 2
- Element 3

Liste sa rednim brojem se pišu istim principom, samo što se za obilježavanje početka, odnosno kraja koristi element <ol>:

```
<ol>
        <li> Element 1 </li>
        <li> Element 2 </li>
        <li> Element 3 </li>
</ol>
```

Lista će u pregledniku biti prikazana ovako:

1. Element 1
2. Element 2
3. Element 3

Liste za definiranje se razlikuju po tome što umjesto grafičkog elementa ili broja u pregledniku za svaki element liste ispisuju ono što im je dodijeljeno, kao pojam koji će opisati element liste. U ovim listama se početak i završetak liste označava oznakom <dl>, pojedini element sa oznakom <dt>, a objašnjenje odnosno opis elementa oznakom <dd>, kao u primjeru:

```
<dl>
        <dt> Ovaj rad obrađuje temu: </dt>
        <dd> Web design </dd>
        <br>
        <dt> Neka od poglavlja rada su: </dt>
        <dd> HTML </dd>
        <dd> CSS </dd>
        <dd> Wordpress </dd>
</dl>
```

Lista će u pregledniku biti prikazana ovako:

Ovaj rad obrađuje temu:

Web design

Neka od poglavlja rada su:

HTML

CSS

Wordpress

Ugniježđene liste ili laički rečeno „lista u listi“. To su liste koje u sebi sadrže više manjih listi. Pišu se kao više listi bez rednog broja unutar jednog dodatnog <ul> elementa, kao u primjeru:

```
<ul>
     <li> PRVA RAZINA
             <ul>
             <li> Element 1.1 </li>
             <li> Element 1.2 </li>
             <li> Element 1.3 </li>
             </ul>
     </li>
     <li> DRUGA RAZINA
             <ul>
             <li> Element 2.1 </li>
             <li> Element 2.2 </li>
             <li> Element 2.3 </li>
             </ul>
     </li>
</ul>
```

Lista će u pregledniku biti prikazana ovako:

- PRVA RAZINA
- Element 1.1
- Element 1.2
- Element 1.3
- DRUGA RAZINA
- Element 2.1
- Element 2.2
- Element 2.3

5.10. Rad sa grafičkim datotekama

U poglavlju 4 ovog rada objašnjeno je na koje načine se slike i grafički sadržaji mogu i trebaju pripremiti za web stranice. Kada su slike spremne za objavu potrebno je unutar HTML koda iskoristiti određene alate i trikove pomoću kojih će preglednik znati da se radi o slici, da je treba prikazati u određenoj veličini i na određenom mjestu te stranice. Sve se to definira u HTML kodu, što će biti detaljnije pojašnjeno u nastavku. HTML kao kod ne može samo tako prikazati sliku, ona se mora nalaziti negdje na web-u, na nekom poslužitelju, a u HTML kodu se pomoću elementa <img> prikazuje ta slika, odnosno kod će pregledniku reći da prikaže sliku koja se nalazi na definiranoj lokaciji.

Element <img>

Ovaj element se piše bez završne oznake i spada u elemente reda. Obavezan atribut kod elementa <img> je *src* (eng. Source) u čiju vrijednost se dodaje adresa, odnosno lokacija slike koja se treba prikazati u pregledniku. Adresa slike može biti URL ili relativna.

URL adrese su bilo kakvi vanjski linkovi, odnosno adrese koje nisu na poslužitelju na kojem je web stranica na kojoj se radi. Za primjer ću uzeti da je URL slike https://www.stranica.hr/slika.png.

Da bi se slika uvezla u HTML kodu i prikazala u pregledniku, element <img> s piše na sljedeći način:

```
<body>
        <img src="https://www.stranica.hr/slika.png">
</body>
```

Ukoliko se slika nalazi na istom poslužitelju i u istoj mapi kao i sam HTML dokument, onda se atributu *src* dodaje ime slike sa ekstenzijom:

```
<body>
        <img src="slika.png">
</body>
```

No ako je slika u nekoj podmapi, onda vrijednost atributa *src* mora počinjati sa „/“ i nakon toga se upisuje cijela putanja do slike. Uzmimo za primjer da se na lokaciji s HTML dokumentom nalazi podmapa „slike“ u kojoj je spremljena grafička datoteka slika.png. U tom slučaju kod se piše na sljedeći način:

```
<body>
        <img src="/slike/slika.png">
</body>
```

U elementu <img>[14] jedini obvezan atribut je *src*, no u praksi je poželjno koristiti i atribut *alt*. Njemu se kao vrijednost dodaje neki tekst koji će biti prikazan u slučaju da se slika iz nekog razloga ne uspije učitati. U kodu se piše odmah iza atributa *src*.

5.11. Tablice

Ukoliko se nešto želi prikazati strukturirano i organizirano, o kakvom god dokumentu ili mediju se radilo, koriste se tablice. U HTML-u su tablice sastavljene od više elemenata koji čine redove, odnosno stupce. Kao i svaka druga tablica i u HTML-u tablice imaju stupce i redove. Struktura tablice se izrađuje unutar elementa <table>, unutar kojeg su definirani redovi pomoću elementa <tr> (table row - eng. red tablice). Ovisno o broju ćelija u jednom redu definirano je koliko će tablica imati stupaca, što znači da se broj stupaca ne mora definirati posebno. Ćelije su definirane elementom <td> (table data - eng. podatci tablice) ili elementom <th> (table header – eng. zaglavlje tablice) ako se ćeli istaknuti prvi red tablice.

Tablice definirane elementima <table>, <tr> i <td> u pregledniku neće biti vidljive same po sebi, nego će strukturirano biti prikazani podatci u ćelijama. Ako se želi prikazati i rub tablice u element <table> se dodaje HTML atribut „border“ (eng. rub) te mu se dodjeljuje vrijednost u pixelima koja definira širinu ruba. Npr.: Ako se želi prikazati tablica s okvirom od 1 pixel piše se <table border=“1“>.

Primjer HTML koda tablice koja ima rub od 1 pixel, te prvi red istaknut:

```
<table border="1">
<tr>
            <th> KONTINENT </th>
            <th> DRŽAVA </th>
            <th> GRAD </th>
            <th> DATUM PUTOVANJA </th>
```

[14] Svi atributi koje je moguće koristiti izlistani su na https://www.w3schools.com/tags/tag_img.asp

```
</tr>
<tr>
            <td> Afrika </td>
            <td> Tanzanija </td>
            <td> Stone Town </td>
            <td> 22.8.2019. </td>
</tr>
<tr>
            <td> Azija </td>
            <td> Kina </td>
            <td> Peking </td>
            <td> 22.4.2020. </td>
</tr>
</table>
```

Prethodno napisani kod web pretraživač će korisniku prikazati ovako:

KONTINENT	DRŽAVA	GRAD	DATUM PUTOVANJA
Afrika	Tanzanija	Stone Town	22.8.2019.
Azija	Kina	Peking	22.4.2020.

Slika 3: Primjer prikaza tablice u web pretraživaču

5.12. Obrasci ili forme

HTML i CSS su osnovni jezici za korisničko sučelje bilo koje web stranice. Kao što je na početku već spomenuto, kod pisan u HTML-u internetski preglednici pretvaraju vizualno u ono što korisnik vidi. No što ako korisnik hoće „komunicirati“ sa web stranicom? Na primjer korisnik želi pretražiti nešto u web trgovini, mora imati polje u koje će upisati pojam koji traži, te mora imati gumb kojim će potvrditi pretragu. Podatci koje korisnik očekuje su spremljeni na nekom poslužitelju, te se procesuiraju na poslužiteljskoj strani pomoću nekog skriptnog jezika, kao što je PHP kojeg u radu ne obrađujemo u detalje jer to već izlazi iz okvira HTML-a i CSS-a, odnosno web dizajna. HTML ima mogućnost da korisniku ponudi unos informacija, te da prikupljene informacije šalje poslužitelju na obradu, a to su forme ili obrasci.

Obrasci se mogu sastojati od različitih elemenata, polja za unos teksta, polja za odabir, gumba itd. U dokumentu su definirani elementom <form> unutar kojeg se pišu različiti elementi obrasca ovisno o potrebama. Elementu <form> se obavezno moraju dodati atributi action i method, no postoje i drugi koji se koriste po potrebi:

Action – upisuje se URL na koji se podatci šalju

Autocomplete – uključuje automatsku dopunu sadržaja. Dodjeljuju mu se vrijednosti „on“ ili „off“

Enctype – način kodiranja podataka pri slanju na poslužitelja

Method – metoda slanja podataka na server. Vrijednosti mu mogu biti „get“ ili „post“

Name – naziv obrasca

Target – gdje će biti prikazan rezultat nakon obrade obrasca

Elementi obrasca se definiraju pomoću različitih HTML elemenata unutar <form> elements. Osnovni elementi za kreiranje funkcionalnog obrasca su <input>, <textarea>, <select>, <option> i <label>.

Element <input> (eng. unos)

Mora sadržavati atribut type. Vrijednosti atributa govore o tome o kakvom unosu podataka se radi:

text – jednolinijsko polje za unos teksta

password – polje sa skrivenim tekstom, za lozinke ili pinove

radio – polje za odabir vrijednosti

file – odabir datoteke s računala

reset – poništavanje unosa u obrazac

submit – gumb za slanje podataka iz obrasca na poslužitelja

checkbox – polje za potvrdu

Element <input> može sadržavati i neke druge atribute:

checked – koristi se ako je vrijednost atributa type radio ili checkbox i unaprijed odabire element obrasca

disabled – onemogućava korištenje elementa obrasca

maxlenght – najveći broj znakova koji se može unijeti u polje

name – ime elementa obrasca

readonly – elementi samo za čitanje

required – određuje obvezne elemente obrasca

size – širina elementa obrasca, određena brojem znakova

value – vrijednost elementa koja će biti proslijeđena poslužitelju

Element <textarea>

Element se obavezno zatvara završnom oznakom. Pomoću njega se kreira polje za unos više redova teksta. Uz njega se mogu koristiti atributi ovisno o potrebama. Neki su jednaki kao i kod elementa <input>: disabled, maxlenght, name, readonly i required, no postoje i oni koji se ne koriste u <input>, kao npr.:

cols – broj stupaca u polju

rows – broj redova u polju

Elementi <select> i <option>

Koriste se za padajuće izbornike. Elementom <select> se definira da se radi o padajućem izborniku, dok se s <option> definira pojedini element padajućeg izbornika koji će korisnik moći odabrati. Element <select> može sadržavati atribute disabled, name, required i size, ali može mu se dodijeliti i atribut multiple, koji omogućava odabir više opcija. Elementu <option> se dodjeljuju atributi disabled i value, kao i u prijašnjim elementima, no također ima i neke svoje atribute:

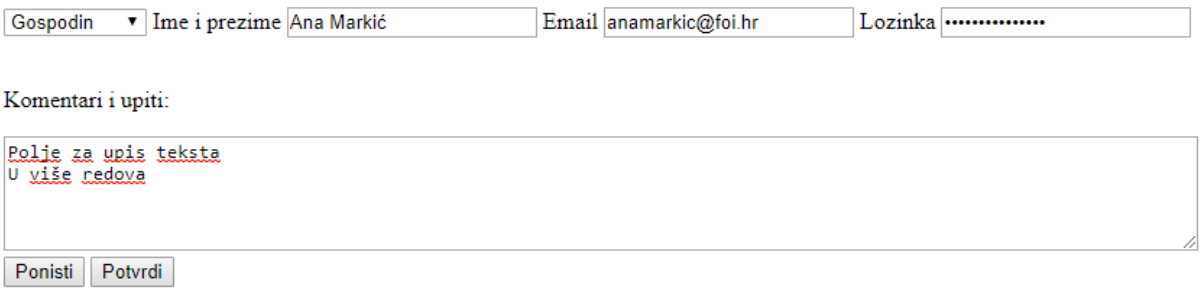
label – specificirana skraćena oznaka opcije

selected – unaprijed određena odabrana opcija

Element <label>

Ovo je zapravo oznaka elementa obrasca, kako bi korisnici znali što se u kojem polju traži. Njemu se obavezno pridružuje atribut for sa vrijednosti koja je jednaka imenu elementa obrasca na koji se oznaka odnosi.

Kao što je već rečeno, ovisno o tipu obrasca koji se zahtjeva na web stranici, mogu se kombinirati svi navedeni elementi, te tako napraviti obrazac kakav je potreban. Kada se sve to spoji u jednu cjelinu, obrazac može izgledati ovako:



Slika 4: Primjer prikaza obrasca u web pretraživaču

U pozadini ovoga što je web pretraživač prikazao, odnosno u HTML dokumentu, nalazi se sljedeći kod:

```
<form name="Obrazac za prijavu" action="" method="post">
            <select>
                    <option>Gospodin</option>
                    <option>Gospođa/ica</option>
            </select>

<label for="Ime i prezime">Ime i prezime</label>
            <input type="text" name="Ime i prezime">

<label for="email">Email</label>
            <input type="text" name="email">
```

```
<label for="lozinka">Lozinka</label>
            <input type="password" name="lozinka">

<br>
<br>

<p>Komentari i upiti:</p>

<textarea cols="115" rows="5"></textarea>

<br>

<input type="reset" value="Ponisti" name="reset">
            <input type="submit" value="Potvrdi" name="submit">

</form>
```

5.13. Struktura sadržaja

Na samom početku poglavlja o HTML-u objašnjena je struktura dokumenta, a kao jedan od dijelova dokumenta naveden je i element <body> u kojem se nalazi sadržaj dokumenta. Do sada je u radu sav sadržan pisan kao dio <body> elementa, no neke od sadržaja se ne mora isključivo i nužno pisati unutar <body> elementa, nego u HTML5 postoje drugi načini da bi se dokument organizirao i strukturirao, te kao takav bio lakši za obrađivati CSS-om. HTML5 je kao novitet doveo nove semantičke strukturne elemente koji se trenutno koriste strukturiranje cijelog sadržaja web stranice.

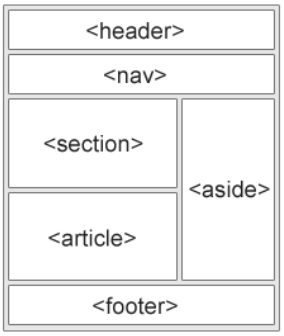
5.13.1. Ne-semantički strukturni elementi

Postoje dva elementa koji nemaju semantičko značenje, odnosno koriste se kao generički kontejneri za grupiranje sadržaja u dokumentu. Dolaskom HTML5 standarda ne-semantički elementi su izgubili svoju prijašnju primjenu te se sada koriste ograničeno.

Element ** ** je jedan od njih, te se koristi isključivo za grupiranje dijela sadržaja za naknadno stiliziranje CSS-om. Drugi element je **** i on je sličan elementu , jedina razlika je što se on koristi na nivou reda, odnosno ako samo jednu riječ ili mali dio želimo istaknuti i vizualno prikazati na drugačiji način od ostatka sadržaja.

5.13.2. Semantički strukturni elementi

HTML5 je svojim dolaskom definirao semantičku strukturu web stranice, te se od tada koriste i semantički elementi koji olakšavaju prikaz i organizaciju cijelog sadržaja. U preglednicima neće biti prikazani na neki poseban način, odnosno neće se vizualno primijetiti da su korišteni u kodu, nego će samo biti drugačije vizualno prikazani pomoću CSS-a. Na slici pokraj je prikazano koji element bi trebao prikazati koji dio web stranice.



Slika 5: Semantička struktura (Izvor: www.w3schools.com)

<header> - kao što i sam naziv govori, ovo je zaglavlje dokumenta (prijevod s engleskog). Tu se stavljaju naslovi, logo, kratak opis ili uvod i slično.

<nav> - skraćeno od navigacija – za glavnu navigaciju ili sadržaj web stranice, u koji se stavljaju linkovi koji će voditi na druge eksterne ili interne web stranice.

<section> - koristi se za različite sekcije na web stranici. S njime se odvajaju tematske cjeline sadržaja, uglavnom započete nekim naslovom.

<article> - koristi se za neovisnu cjelinu sadržaja, odnosno „članak“. Npr. svaki blog post na stranici će biti u jednom <article> elementu.

<aside> - nekakav sadržaj koji nije dio glavnog sadržaja. Uglavnom se prikazuje s lijeve ili desne strane glavnog sadržaja (kao prikazano na slici) te se u njega mogu staviti reklame, dodatne informacije o nečemu ili bilo što što nije izravno bitno i povezano za glavni sadržaj.

<footer> - podnožje sadržaja. Ovdje se stavlja sve ono što i inače ide tamo, kao autor, informacije o stranici, kontakt podatci i slično.

5.14. Linkovi ili poveznice

Da bi na kraju izrađene stranice dobile smisao, te bile povezane u jednu cjelinu, potrebno je u HTML-u napisati kod koji će preglednicima reći koja poveznica će kamo odvesti korisnika. Za izradu takve poveznice u HTML-u se koriste element <a> (eng. anchor – sidro). Da bi preglednik znao kamo odvesti korisnika kada prepozna element <a> potrebno mu je dodati atribut *href* kojemu se kao vrijednost dodjeljuje adresu dokumenta na koji će poveznica voditi. Primjer:

```
<a href="https://www.google.com"> Google </a>
```

Preglednik će riječ Google prikazati kao link, i klikom na nju odvesti će korisnica na adresu Google-a.

Poveznice na druge stranice, odnosno dokumente mogu biti interne ili eksterne, mogu biti poveznice na određeni dio sadržaja trenutne stranice, te mogu biti poveznice prema email adresama. Kod internih poveznica atributu *href* se dodaje relativna adresa do dokumenta, odnosno adresa na trenutnom web sjedištu, dok za eksterne se dodaje potpuni URL do dokumenta ili stranice. Kao primjer eksterne poveznice se može uzeti gornji kod koji poveznicu vodi na Google, a primjer interne poveznice može biti sljedeći:

```
<a href="fotografije.html"> Fotografije </a>
```

Ovaj primjer govori da na istom web poslužitelju, u istoj mapi, postoji dokument fotografije.html, te klikom na link otvara se navedeni dokument.

Kao što je navedeno, poveznica može voditi i na određeni element unutar samog dokumenta. Da bi preglednik znao kuda treba odvesti korisnika klikom na link, web dizajner ili programer mora u HTML kodu definirati gdje će voditi ta poveznica. Da bi to bilo moguće potrebno je elementu na koji se želi da poveznica vodi dodati atribut *id* kojem kao vrijednost dodjeljujemo jedinstveno ime, a zatim atributu *href* unutar poveznice kao vrijednost dodjeljujemo to isto ime, no uz znak # ispred imena. Primjer:

```
<p>Sve ponude putovanja u Zanzibar možete vidjeti ovdje:
<a href="#tanzanija"> Tanzanija </a></p>
.
.
.
```

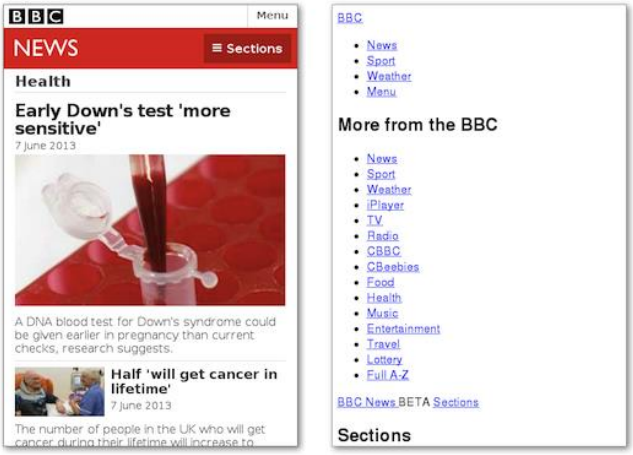
```
<h1 id="tanzanija"> Putovanja u Tanzaniju / Zanzibar </h1>
.
.
.
```

I na kraju postoje poveznice prema email adresama, klikom na njih otvara se klijent za elektroničku poštu, te se kao adresa primatelja automatski dodaje adresa dodijeljena kao vrijednost atributa *href* uz „mailto:“ ispred adrese. Primjer:

```
<a href="mailto:putovanja@agencija.com"> Pišite nam! </a>
```

6. CSS

Nakon što web dizajner u HTML-u dobije ono što želi i napuni stranicu sadržajem, potrebno je cijeli taj sadržaj stilizirati. Već iz samog uvoda, a i kasnije u radu da se zaključiti da je za vizualni izgled web stranice zadužen CSS. CSS je kao i HTML jezik nečitljiv običnim korisnicima, no web dizajneri ga poznaju i pišu, a web pretraživači napisani kod pretvaraju i interpretiraju u ono što korisnik vidi i doživljava. Bez CSS-a web stranice bi bile samo hrpa crnog teksta, jednakog fonta, bez lijepo izgleda.



Slika 6: Web stranica BBC-a sa i bez CSS-a[15]

6.1. Sintaksa CSS-a

CSS ima pravila pisanja kao i svaki drugi kod, te ima sintaksu, odnosno način na koji se određeni element piše u kodu. Pravilo pisanja u CSS-u je da svaki blok koda mora sadržavati selektor, svojstvo i vrijednost. Selektor je glavni dio pravila, on pretraživaču govori na koji HTML element se odnosi napisano CSS pravilo, svojstvo govori koje stilsko sredstvo će biti primijenjeno, a vrijednost je dodijeljena stilskom sredstvu te mu govori što da se primjeni (npr. koje boje će neki HTML element biti). Jedan selektor može u istom pravilu imati više svojstava i vrijednosti. U primjeru ispod se može vidjeti CSS pravilo koje definira da se naslovima razine 2 u HTML-u (h2

[15] Izvor: https://ariya.io/2013/06/capturing-web-page-without-stylesheets

naslov - selektor) dodjeljuju svojstva color (boja) i to da vrijednost bude blue (plava), font-size (veličina fonta) od 12 pixela, te font-family (porodica fonta) Arial.

Slika 7: Sintaksa CSS pravila

6.2. Načini pisanja CSS-a

Ovisno o vrsti i veličini web stranice CSS se može pisati na 3 načina: unutarnji, linijski i vanjski. Linijski i unutarnji način pisanja slični su po tome što se pišu zajedno sa HTML-om, u istom dokumentu, dok se vanjski CSS piše u odvojenom dokumentu koji se sprema sa dodatkom .css.

Linijski CSS

U praksi se koristi najrjeđe iz razloga što se tako napisano CSS pravilo odnosi isključivo na element u HTML-u uz koji je napisano. Piše se na način da se određenom HTML elementu kao atribut doda *style*, a kao vrijednosti mu se dodjeljuje svojstvo i vrijednost svojstva po sintaksi CSS-a. U ovom slučaju selektor se ne piše jer je definiran HTML elementom u kojem je pravilo napisano.

Unutarnji CSS

Piše se po standardnoj CSS sintaksi, unutar HTML dokumenta, uz to da se u zaglavlju (head) unutar oznake <style> (mora imati i završnu oznaku) pišu CSS pravila. Nedostatak ovog načina pisanja je što se pravila mogu koristiti samo u tom HTML dokumentu u kojem su napisana. Ovaj način se koristi samo ako određeno CSS pravilo treba primijeniti na samo taj jedan dokument.

Vanjski CSS

Najpraktičniji način pisanja CSS pravila iz razloga što se sva napisana pravila mogu primijeniti u bilo kojem HTML dokumentu, nije ih potrebno pisati više puta, posebno ako će sve stranice imati jednak vizualni identitet. Na ovaj način se pravila pišu u bilo kojem uređivaču teksta, preporuka struke je da se svaki selektor, odnosno svojstvo piše u novi red radi lakšeg iščitavana koda. Nakon

napisanih pravila datoteka se treba pohraniti u isti direktorij kao i HTML dokument, te spremiti sa ekstenzijom .css, što će web pretraživačima reći da se radi o CSS datoteci.

Dodatno je potrebno u zaglavlju svakog HTML dokumenta u kojem će se pravila iz CSS dokumenta koristiti, povezati CSS i HTML datoteku. Povezivanje se izvodi elementom <link> kojemu se dodjeljuju atributi *href* kojemu je vrijednost naziv/adresa CSS dokumenta, i atributom *rel* kojemu se dodjeljuje fiksna vrijednost *stylesheet*.

```
<head>
        <title> Putnička agencija </tytle>
        <link href="style.css" rel="stylesheet">
</head>
```

6.3. Selektori

Ranije spomenuti selektori su najjednostavniji oblik selektora. To su **selektori HTML elemenata** i oni se pišu jednako kao i nazivi HTML elemenata. Pravila pisana s takvim selektorima se odnose na sve HTML elemente za koje su pravila namijenjena i svi takvi elementi će poprimiti dizajn koji je određen CSS pravilom.

Ako web dizajner želi napraviti CSS pravilo koje će moći dodijeliti određenim HTML elementima, bez da svi elementi tog tipa poprime izgled definiran CSS-om kao u prethodnom primjeru, za to može koristit klase. **Klase** se dodjeljuju određenom HTML elementu tako da mu se dodijeli atribut *class* koji ima vrijednost naziva klase, a koji je definiran u CSS-u. Pravilo u CSS-u za klase se piše po CSS sintaksi uz to da klase ispred imena imaju točku (.).

HTML dokument:

```
<p class="dizajn"> Molimo sve korisnike da se pridržavaju uputa. </p>
```

CSS dokument:

```
.dizajn {font-size: 20px; color:red;}
```

U primjeru iznad je paragrafu dodijeljena klasa s imenom *dizajn*, te u CSS-u definirano pravilo za klasu. Da se radi o klasi može se prepoznati po točki ispred imena klase.

ID selektori su slični klasama, uz razliku da se pravila pišu sa znakom „#“ ispred imena (umjesto točke), te su ti selektori primjenjivi na samo jedan element.

Selektore je moguće i grupirati, tako da se pišu jedan za drugim odvojeni zarezom. U slučaju grupiranja selektora oni ne moraju biti niti istog tipa, ovo se radi samo ako se želi za više selektora dodijeliti jednaka pravila stiliziranja. Postoji još mnoštvo različitih selektora u CSS3, no oni zahtijevaju višu razinu poznavanja CSS-a te je stoga dovoljno spomenuti samo ove osnovne s kojima se može napraviti većina posla u CSS-u.

6.4. Svojstva

U CSS pravilima svojstva su ono što daje drugo ruho web stranicama i omogućuje promjenu izgleda teksta, pozadina, tablica ili slično. Svojstvima se definira što se mijenja u izgledu, te se dodjeljuje vrijednost na koju se određeni element želi promijeniti. O svojstvima se nema što puno pričati niti učiti, s vremenom i iskustvom se lako zapamte nazivi svih svojstava te vrijednosti koje im je moguće dodijeliti. Svojstava u CSS3 postoji na tisuće i vjerojatno niti jedan web dizajner ne može zapamtiti sva svojstva i vrijednosti koje im se dodjeljuju, niti je potrebno sve imati u glavi. Kod web dizajna, ili bilo kakvog programiranja ili razvoja aplikacija, najbitnije je znati kako doći do informacije koja se traži, a popisa CSS svojstava i njihovih pojašnjenja postoji svuda na internetu. Najpouzdanije mjesto za pronalazak svih mogućih svojstava je ranije spomenuti W3C koji radi na regulaciji i standardizaciji CSS-a. Sva dostupna svojstva pa tako i ona nova koja se uvode novim verzijama CSS-a redovno se ažuriraju na stranicama W3C-a[16].

CSS svojstva nisu isključivo vezana za prikaz teksta, njima se može definirati i boja odnosno slika pozadine, kursor miša korisnika, margine i obrubi tablica i slično. Bitno je napomenuti da se sva svojstva pišu istom sintaksom, samo njihova funkcija ovisi o samoj namjeni određenog svojstva.

[16] Web stranica sa aktualnim svojstvima: https://www.w3.org/Style/CSS/all-properties

7. RESPONZIVNI WEB DIZAJN

Prije početka masovne upotrebe pametnih mobitela, tableta pa i većih zaslona kao pametnih televizora i slično, web stranice su se otvarale isključivo na računalima. Većina zaslona računala su imali standardne rezolucije i veličine tako da se web dizajneri nisu morali puno brinuti o načinu prikaza web stranice ovisno o vrsti zaslona. No sve veća upotreba gore navedenih uređaja pokazala je potrebu da se i web stranice prilagode tim uređajima. Gotovo nemoguće bi bilo prilagoditi web stranicu svakom uređaju, te za svaki tip zaslona raditi poseban dizajn. Iz tog razloga razvijen je način da se unutar samog CSS-a definiraju drugačija stilska pravila ovisno s kakvog uređaja korisnik pristupa web stranici, te se ta pravila primjenjuju automatski za svaki uređaj.

Slika 8: Prikaz responzivne web stranice na različitim uređajima[17]

7.1. Medijski upiti

Prije CSS3 verzije za responzivnost web stranica koristili su se medijski tipovi, no dolaskom treće verzije CSS-a stvari su postale bolje i jednostavnije, odnosno lakše prilagodljive uređajima različitih veličina i rezolucija zaslona. Na starijim verzijama CSS-a bilo je potrebno definirati na kakvom tipu uređaja se koristi web stranica, tako bi za svaki tip uređaja bilo potrebno pisati posebne linije koda da bi se web stranica otvarala ispravno i prilagođeno. CSS3 je omogućio da se izbjegne definiranje svakog tipa posebno, nego se definiraju svojstva uređaja, odnosno svojstva zaslona na kojemu će se web stranica prikazivati. Ovo je pojednostavnilo stvari jer više

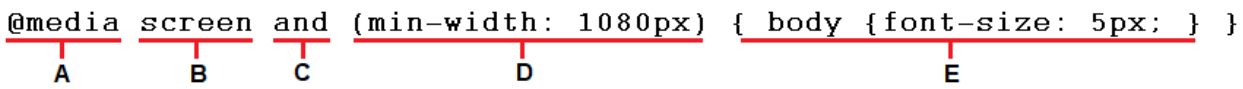
[17] Izvor: https://www.goweb99.com/website-design

uređaja različitog tipa može imati jednaka svojstva, tako da nema smisla pisati odvojen kod kada se za sve uređaje stilska pravila mogu definirati skupa.

Prikaz web stranice ovisi uglavnom o rezoluciji zaslona, tako da je najčešće definirano svojstvo u medijskim upitima upravo rezolucija. Osim rezolucije često se kao svojstvo koristi i orijentacija uređaja (portret ili pejzaž).

Sintaksa medijskih upita je slična kao i običnih CSS pravila, a također i piše se skupa sa ostatkom CSS-a. Preporuka struke je da se za medijske upite koriste vanjski CSS, iako nema razlike u funkcionalnosti ako se pišu unutar HTML dokumenta, zbog bolje preglednosti i jednostavnijeg snalaženja u slučaju da više ljudi radi na projektu bolje ih je pisati u vanjskoj CSS datoteci.

Za lakše shvaćanje sintakse medijskih upita može se reći da je to svojstvo koje će u sebi sadržavati jedno ili više drugih svojstava. Sintaksa počinje s pozivanjem svojstva @media, te definiranjem tipa medija na kojeg će se svojstvo odnositi. Na primjeru ispod biti će detaljnije objašnjeno kako se pišu, te što se sve može definirati sa medijskim upitima:



Slika 9: Primjer medijskog upita

A – ključna riječ kojom se poziva medijski upit

B – tip medija na koji se svojstvo odnosi. Može biti:

all – za sve tipove medija

print – za ispis na pisačima

screen – za zaslon uređaja (najčešće korišteni)

speech – za čitače sadržaja (koriste slijepe osobe)

C – ključna riječ koja povezuje tip medija sa svojstvom koji treba ispuniti (and – logički operator „i“ – može se koristiti i zarez „ , “ koji služi kao logički operator „ili“)

D – svojstvo koje tip medija treba ispuniti da bi se stilsko svojstvo primijenilo na njega. U ovom primjeru minimalna širina zaslona (screen) mora biti 1080px.

E – stilsko svojstvo koje će napraviti promjenu u izgledu web stranice ukoliko je prethodno svojstvo ispunjeno. Dio unutar vitičastih zagrada se ne razlikuje od standardnih CSS pravila, to zapravo i jesu CSS pravila koja se ispunjavaju ako su prethodni uvjeti ispunjeni. U ovom slučaju ako je zaslon minimalne širine 1080px, veličina teksta u *body* elementu unutar HTML-a biti će postavljena na 5 pixela.

Korištenjem medijskih upita web stranica se savršeno može prilagoditi bilo kojem tipu uređaja. Postoji mnogo svojstava koji se mogu primijeniti, a ovo su najviše korištena svojstva te kratak opis njihove namjene:

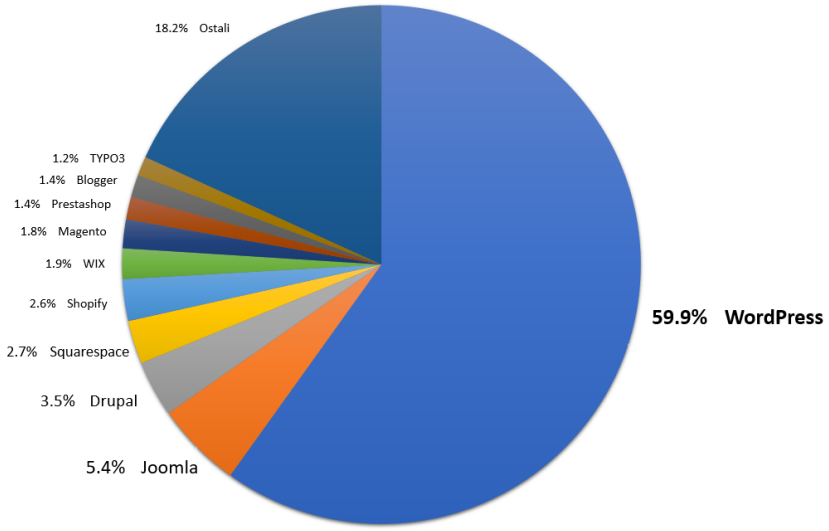
Tablica 1: Svojstva medijskih upita

Svojstvo	Opis
aspect-ratio	Omjer visine i širine zaslona. Npr. omjer 16 naprama 3 pišemo u formatu "16/3"
height	Visina zaslona na kojem se prikazuje web stranica
max-aspect-ratio	Maksimalan omjer visine i širine zaslona
max-color-index	Maksimalan broj boja zaslona
max-height	Maksimalna visina zaslona
max-resolution	Maksimalna rezolucija zaslona
max-width	Maksimalna širina zaslona
min-aspect-ratio	Minimalan omjer visine i širine zaslona.
min-color-index	Minimalan broj boja zaslona
min-height	Maksimalna visina zaslona
min-resolution	Minimalna rezolucija zaslona
min-width	Maksimalna širina zaslona
orientation	Orijentacija uređaja (može biti "landscape" za pejzaž ili "portrait" za portret)
resolution	Rezolucija zaslona
width	Širina zaslona

8. Sustavi za upravljanje sadržajem - CMS

Do sada su u radu obrađene osnove svih alata i tehnologija koje su potrebne za napraviti web stranicu. No sve to do sada ima jednu manu, a to je da su stranice napravljene pomoću svega navedenom statične. Što to znači? Recimo da web dizajner napravi web stranicu koristeći HTML i CSS, da je stranica responzivna, da je uređen sadržaj stranice i da sve to skupa izgleda kako treba. Takva stranica bi bila dovoljna ako na nju nitko neće dodavati nikakav novi sadržaj, odnosno ako će duže vrijeme biti dostupna takva kakva je. No što ako korisnik na svoju web stranicu želi konstantno objavljivati nove stvari? Kao na primjer ako se korisnik bavi fotografijom, te bi svaki dan htio svoju stranicu osvježiti tako da će na nju staviti po jednu novu fotografiju. Ili ako korisnik želi stranicu sa novostima. Sa statičnom stranicom, ako se želi dodati novi tekst, za svaki takav tekst potrebno je uređivati izvorni kod stranice. U HTML kodu bi bilo potrebno dodavati linije i linije koda, te sve to ponovno ažurirati na poslužitelju samo kako bi se dodala nova fotografija ili članak. Da bi se izbjegao konstantan rad s kodovima, te da bi web stranice koje web dizajner napravi mogli koristiti i korisnici koji nisu upoznati sa HTML-om i CSS-om koriste se CMS sustavi. CMS je skraćenica od Content Management System, odnosno Sustav za upravljanje sadržajem. Kao što i samo ime govori, CMS omogućava da na lak i jednostavan način web stranice postanu dinamične, te da se jednostavnim postupkom može dodavati novi sadržaj kao što su tekstovi, slike, video i audio zapisi i slično. Osim mogućnosti ažuriranja sadržaja, najzastupljeniji CMS-ovi danas omogućavaju i dodavanje raznih dodataka koji donose još više dinamičnosti web stranicama i otvaraju brojna druga vrata.

Neki od najpoznatijih i najraširenijih CMS-ova su Wordpress, Drupal, Joomla!, Blogger i drugi. Svaki od njih ima svoju najbolju primjenu i tip stranica za koje su prvotno bili namijenjeni. Tako je Wordpress ispočetka bio platforma za upravljanje blogovima, dok je danas postao toliko zastupljen i popularan da se mogu naći dodatci pomoću kojih se može napraviti bilo kakva web stranica. Da bi bolje shvatili zastupljenost Wordpress-a, što je razlog zbog kojeg se o njemu nešto više govori u daljnjem radu, pomoći će grafički prikaz trenutne zastupljenosti CMS-ova na internetu:



Slika 10: Zastupljenost sustava za uređivanje sadržaja (CMS)
(Izvor podataka: https://websitesetup.org/popular-cms)

8.1. WordPress

Wordpress je u cijelosti napisan u programskom jeziku PHP i njegov kod je u potpunosti otvoren (open source). Iako mu se početci bilježe još 2001. godine s radom je službeno krenuo 2003. godine, i to ispočetka isključivo kao platforma za web blog. Kao takav nije niti bio zamišljen kao CMS sustav, nego kao nadogradnja već gotovim web stranicama. Wordpress je drugačiji od ostalih sustava za upravljanje sadržajem po tome što nudi mogućnost naprednijeg dizajna, tisuće nadogradnji i aplikacija koje se mogu integrirati u web stranice i ogromna podrška korisnika i razvojnih programera. Od svih mogućnosti koje Wordpress nudi najviše se ističu dvije – mogućnost dodavanja mnogobrojnih proširenja te mogućnost izrade predložaka web stranice.

8.1.1. Kako se razvijao Wordpress?

2001. godine – predstavljen je b2/cafelog, prva verzija blog platforme koja kasnije postaje Wordpress

2003. godine – Matt Mullenweg i Mike Little pokreću Wordpress temeljen na b2/cafelog

2004. godine – Wordpress verzija 1.2 prvi put dobiva mogućnost dodavanja nadogradnji

2005. godine – Wordpress verzija 1.5 dobiva sustav tema i statičkih stranica

2008. godine – Tvrtka Happy Cog dizajnirala je novo administratorsko sučelje

2010. godine – Dodana opcija za upravljanje izbornika i pozadine. Uvedeni su prilagođeni post tipovi.

2011. godine – Verzija 3.2 znatno ubrzala Wordpress, unaprijedila minimalne specifikacije za PHP i MySQL. Ovom verzijom je Wordpress postao jednostavniji za korištenje laicima i onima koji nisu imali prilike ranije raditi sa CMS sustavima.

2013. godine – Verzija 3.6 donosi ugrađenu audio i video podršku, dodana opcija za automatsko ažuriranje što je poboljšalo sigurnost.

2016. godine – Verzija 4.6 ponovno znatno ubrzava sustav i uvodi mogućnost oporavka sadržaja.

Svaka od ovih verzija kroz povijest donosila je i puno više dodataka i mogućnosti, ovo su neke najznačajnije. U prosjeku svake godine izađe 2-3 verzije Wordpress-a tako da i sam sustav jako brzo raste i napreduje.

8.2. Preduvjeti za instalaciju i korištenje

S obzirom da je Wordpress pisan u PHP-u da bi ga uopće mogli koristiti, odnosno instalirati, potreban je web hosting koji podržava PHP. Sav sadržaj i sve ono s čime se upravlja Wordpress sprema u bazu podataka koja se mora kreirati, iz tog razloga potreban je i MySQL na poslužitelju. Programeri koji rade na daljnjem razvoju Wordpress-a preporučuju da se koriste Apache internetski poslužitelji, PHP minimalne verzije 5.6 te MySQL 5.5. S obzirom da se u praksi stranice prije nego se postave na poslužitelj razvijaju lokalno na računalima, postoje alati koje će ove preduvjete ispuniti i na običnom korisničkom računalu. Jedan od takvih alata je i XAMPP.

XAMPP

Je besplatna aplikacija, odnosno skup programskih paketa koja služi za razvoj i testiranje web stranica. XAMPP omogućava da se na običnom računalu, prije podizanja stranica na poslužitelje, može razvijati stranica te pokretati PHP kod i MySQL baze koje se inače ne mogu pokrenuti u web pregledniku kao na primjer HTML, CSS ili Javascript. Laički rečeno XAMPP simulira server. XAMPP u sebi sadrži više programa i alata, a dizajnerima su najbitniji Apache i MySQL koji se pojavljuju prvi na XAMPP-ovoj kontrolnoj ploči.

Apache

Je najkorišteniji besplatni web poslužitelj te trenutno pokreće oko 70% web poslužitelja na cijelom internetu. On koristi za to da bi primio korisnikov zahtjev za nekom stranicom, te uz nekoliko sigurnosnih provjera korisnika vodi na željenu web stranicu. Apache je jako brz sustav i možemo biti sigurni da je sigurnosno zaštićen. Apache osigurava različita radna okruženja i potrebe koristeći dodatke, pa tako ima i podršku Wordpress-a.

MySQL

Program za upravljanje bazama podataka koji je razvio Oracle. Kao i Wordpress, MySQL je potpuno otvorenog koda. MySQL koristi baze relacijskog tipa koji se smatra najboljim i najbržim tipom skladištenja podataka. U MySQL bazu Wordpress sprema sav sadržaj koji korisnici objave na stranicama kao još mnoge druge informacije vezane uz stranicu. Da bi mogli kreirati bazu, upravljati njome i slično dizajneri i programeri koriste alat phpMyAdmin koji je zapravo web sučelje koje služi za interakciju između korisnika i baze podataka.

8.3. Instalacija Wordpress-a

Na službenoj stranici www.wordpress.org mogu se naći sve upute za instalaciju, kao i sam instalacijski paket i besplatno preuzeti. Cijeli instalacijski proces je vrlo jednostavan i detaljno je opisan na službenim stranicama Wordpressa. Datoteka koja se preuzme prilikom početka instalacije je u zip formatu, te ju takvu zapakiranu potrebno postaviti na poslužitelja. Većina današnjih poslužitelja ima integriran neki upravitelj podataka (File Manager), pa tako se i preko web pretraživača može prijaviti na administratorsku površinu poslužitelja, i tamo postaviti sve potrebne datoteke. Nekada ako se radi o većoj količini podataka bolje je koristiti i FTP[18] protokol i uz pomoć FTP klijentskih programa se spojiti na poslužitelja i tako mu poslati sve datoteke. Nakon što je zip datoteka Wordpress-a prebačena na poslužitelja može se otpakirati u direktorij *public_html* iz kojeg će se prikazivati početna stranica (index.php).

Ovime su postavljene i pripremljene sve datoteke potrebne za instalaciju Wordpress-a, no potrebno mu još podesiti bazu podataka koju će koristiti. Ovaj dio se odrađuje u MySQL-u, ali je potrebno i Wordpress-u dati podatke o imenu baze, korisničkom imenu, lozinci i slično. U phpMyAdmin se pokrene čarobnjak za upravljanje bazama i kreira nova baza. Prvo je potrebno

[18] FTP – File Transfer Protocol

dodijeliti ime baze koje se uglavnom sastoji od imena projekta, korisnika ili nečega što nas asocira na projekt na kojem se radi te se doda dodatak „_wp“ koji koristi većina web razvojnih programera ili dizajnera ako se radi o Wordpress bazi. Zatim je potrebno odrediti korisnika koji će imati sva upravljačka prava nad bazom i njegovo ime je uglavnom „admin“. Lozinka koja se postavlja za korisnika bi trebala biti što sigurnija. Po naputku CERN-a[19] sigurna lozinka bi trebala biti minimalno 8 znakova duga, da sadrži barem 3 različita tipa podatka (veliko slovo, malo slovo, broj i simbol) te da se ne spominje nigdje u tekstu stranice. Na kraju se određuje koja prava će taj korisnik imati, a pošto je riječ o administratorskom korisniku dodjeljujemo mu sva prava. Ovime je postavljeno sve što je potrebno za početak instalacije.

Za nastavak potrebno je preko pretraživača otići na web mjesto na kojem su postavljene instalacijske datoteke. Ako se radi lokalno preko XAMPP-a to je uglavnom */localhost* , a ako je na poslužitelju to je onda adresa samog poslužitelja. Kada je otvorena lokacija u pretraživaču pojaviti će se početna stranica instalacije gdje se odabire jezik sučelja, a ostatak je vrlo jednostavan proces koji traje svega nekoliko minuta uz detaljno objašnjene korake. Zatim je potrebno unijeti podatke o bazi podataka koji su postavljeni u mySQL, podatke o samoj web stranici i podatke o administratoru web stranice za koje se savjetuje da budu različiti od podataka administratora baze podataka. Nakon procesa instalacije Wordpress omogućava i instalaciju jezičnih paketa (standardni je engleski).

8.4. Izgled sustava

Osnova CMS-a, a tako i Wordpress-a je upravljačka površina kojoj se pristupa preko web pretraživača, s obzirom da je Wordpress instaliran na poslužitelju. Upravljačka ploča se nalazi unutar Wordpress instalacijskog direktorija pod nazivom „wp-admin“, a pristupa joj se tako da se u web pretraživaču ode na adresu web mjesta na kojoj je instaliran Wordpress (npr. za lokalnu inačicu preko XAMPP-a potrebno je u adresnu traku upisati adresu /localhost/wp-admin). Sustav će zahtijevati prijavu, a kao podatke za prijavu administratora se koriste oni koji su postavljeni prilikom instalacije Wordpress-a. Nakon prijave otvoriti će se početna stranica sa upravljačkom površinom, na kojoj se nalaze sve opcije i mogućnosti Wordpress-a, neke od najvažnijih opcija i značajki su:

[19] Sigurnost lozinke po CERN-u - https://security.web.cern.ch/security/recommendations/en/passwords.shtml

Nadzorna ploča ili dashboard

Ovo je početni i prvi prozor koji se otvara. On služi za detaljniji uvid u statističke podatke web stranice, omogućava brzi unos novog posta na web stranicu i razne izbornike za opcije koje Wordpress nudi.

Članci ili posts

Kao što i samo ime kaže, u ovom dijelu upravlja se postovima koje korisnik objavljuje, uređuje tekst, ima na uvid tablicu svih tekstova koji su objavljeni po kategorijama i oznakama. Članci se uglavnom pojavljuju na naslovnici, te ih je moguće kategorizirati.

Medijski zapisi ili media

Sve medijske datoteke koje se koriste ili će se koristiti web stranici nalaze se ovdje. Moguće ih je grupirati i organizirati po željama i potrebama. Ovdje korisnik može naći sve svoje slike, audio ili video zapise.

Stranice ili pages

U Wordpress-u su stranice po sadržaju slična vrsta kao i članci. Dok se članci prikazuju na naslovnici, stranice su odvojene od nje i na njima se uglavnom ne prikazuje nikakav aktivan sadržaj, odnosno na njima se uglavnom prikazuje statičan sadržaj kao što su neke servisne informacije, kontakt podatci, podatci o djelatnosti i slično.

Komentari ili comments

Služi za uređivanje i upravljanje komentara posjetitelja.

Izgled ili appearance

Pod opcijom izgled nalazi se mjesto s kojeg se upravlja izgledom web stranice, odnosno gdje se može promijeniti tema web stranice, dodati i instalirati nove teme, upravljati izbornicima, pozadinom i slično. Wordpress ima mogućnost dodavanja dodatnih programa (eng. Widget) koji stranici daju dodatne funkcionalnosti, ukoliko to tema koja je trenutno odabrana i koja se koristi dopušta, opcija za dodatke će također biti prikazana ovdje. U Worpress-u je također moguće kreirati svoje teme, no to je već opširnija tema koja zahtjeva poznavanje PHP-a.

Dodatci ili plugins

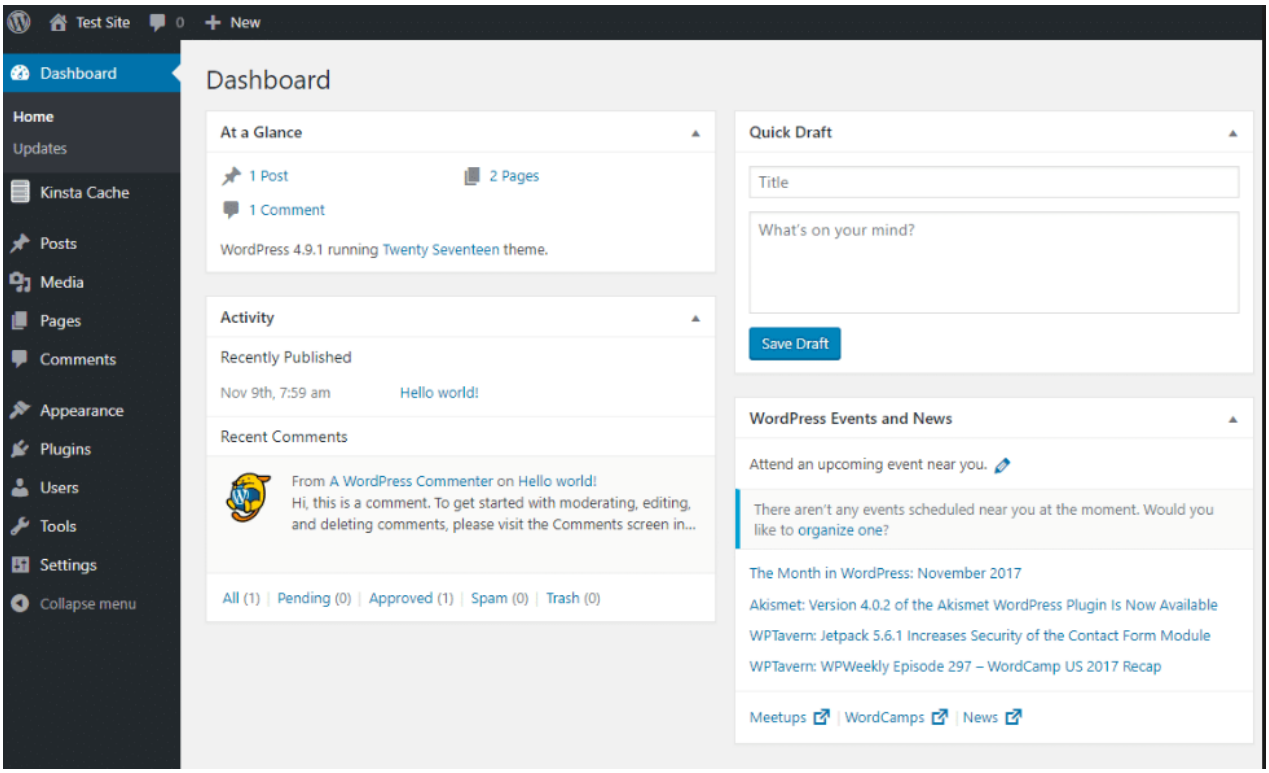
Bit Wordpressa i je da mu se mogu dodati razne funkcionalnosti koje su razvijene za pojedine teme. Dodatke je vrlo jednostavno pretraživati i dodati, nakon klika na *Dodaj novi* čarobnjak sam navodi na potrebne korake.

Korisnici ili users

Za pretpostaviti je čemu služi ova opcija. Ovdje se definiraju svi korisnici web stranice te im se definira rola odnosno uloga (npr. Administrator, urednik, autor itd.)

Postavke ili settings

Postoji više kategorija postavki, no svima se upravlja s ovoga mjesta. U ovom dijelu se nalaze kategorije Općenite postavke, gdje se nalaze postavke vezane za ime stranice, slogan, ključne riječi, adresa instalacije, datum i vrijeme. Osim općenitih postavki stranice ovdje se podešavaju i postavke za pisanje, za prikaz (čitanje) i permalinkove (stalne veze).



Slika 11: Upravljačka površina Wordpress-a

9. PRAKTIČNI DIO ZAVRŠNOG RADA

Primjenom nabrojenih tehnologija, alata i načina pisanja koda mogu se napraviti web stranice koje će zadovoljiti potrebe većine korisnika. Iako danas postoje mnogi moćniji alati s kojima je moguće napraviti puno više toga na web stranicama, ovo što je opisano u radu je sasvim dovoljno za početak, a i bez toga teško bi bilo naučiti koristiti naprednije i bolje tehnologije. Praktični dio je napravljen za lakše shvaćanje prilikom prezentacije i čitanja rada, a sadržava primjere i dijelove cjelina vezanih za pisanje koda u ovom radu.

Prvi dio rada posvećen je osnovama HTML-a i CSS-a, te su početna stranica i sve pod stranice napisane korištenjem isključivo ta 2 jezika i u potpunosti su statične. S obzirom da je sve pisano u čistom HTML-u i CSS-u, dodavanje sadržaja ili izmjena stranica mora se raditi isključivo u samom kodu.

Drugi, odnosno treći dio sastoji od 2 primjera Wordpress stranica sa različitom namjenom i različitim elementima koje Wordpress omogućuje, te je uz korištenje kontrolne ploče moguće jednostavno dodavanje ili uređivanje sadržaja.

Bitno je napomenuti da je cijeli projekt responzivan, odnosno prikaz stranica prilagođava se veličini zaslona na kojem se stranice otvaraju.

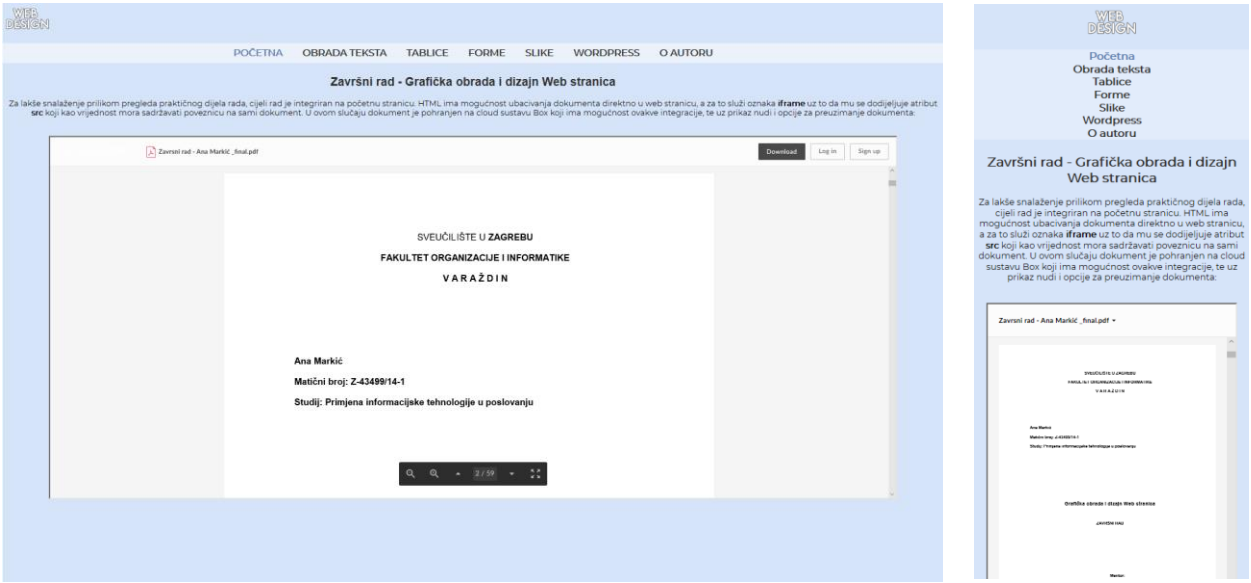
Cijeli projekt je dostupan na Internet adresi www.anamarkic.eu/foi/zavrsni_rad.

9.1. Statična web stranica

Na početnoj stranici projekta iskorištena je mogućnost HTML5 koja nije ranije spomenuta u radu. Važno je odmah istaknuti da postoji još mnoštvo mogućnosti i funkcija HTML-a koje nisu opisane, no uglavnom se sve piše na isti način i uz pomoć dostupne dokumentacija, ukoliko se znaju osnove sadržane u ovom radu, nije teško shvatiti kako ih koristiti. Ideja je bila dodati sam završni rad u PDF obliku na web stranicu, tako da dok se pregledava projektni dio, zbog lakšeg snalaženja korisnik ima odmah dostupan i cijeli tekst rada. Da bi se to izvelo postoji oznaka *iframe* koja integrira neki dio druge web stranice u zaseban okvir i takav se prikazuje na stranici. Mnogi servisi na internetu danas omogućuju automatsko generiranje koda koji se može ubaciti u web stranicu i odmah imati integriran okvir sa željenim sadržajem. Tako je i u ovom primjeru PDF dokument završnog rada spremljen na cloud sustav, koji koristi *iframe* funkciju i preko njega je

generiran kod ispod, koji kao argument ima poveznicu koja vodi do dokumenta, te visinu i širinu okvira u kojem će se prikazivati dokument:

```
<iframe src="https://poveznica.na.dokument.com" width="90%" height="58%">
</iframe>
```

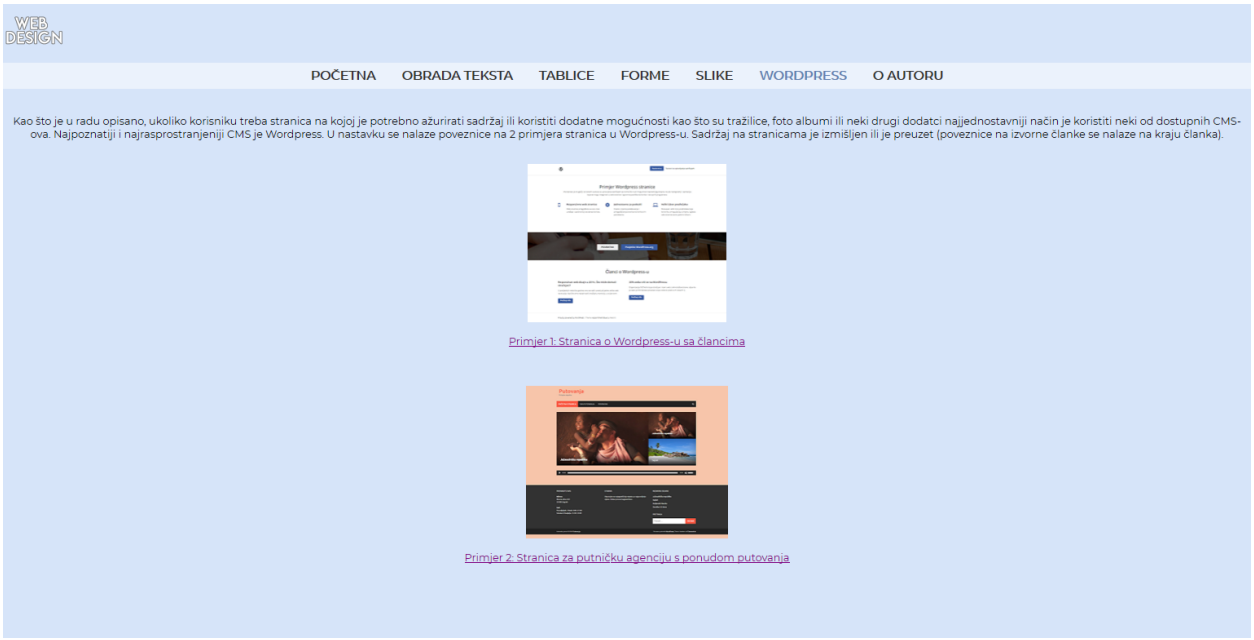


Slika 12: Početna stranica završnog rada (lijevo u desktop, desno u mobilnom prikazu)

Nadalje stranica sadrži navigacijsku traku odnosno izbornik. Svaka pod stranica sadrži dijelove koda koji su korišteni u završnom radu kako bi se prikazalo kako to izgleda u stvarnom okruženju, odnosno u web pregledniku. Također je kao referenca naveden broj stranice završnog rada na kojoj je opisan pojedini dio.

9.2. Dinamička web stranica - Wordpress

Pod stranica Wordpress se ne razlikuje puno od ostalih, no ona zapravo sadrži poveznice na 2 druge stranice, primjere Wordpress stranica. Sama pod stranica je statična kao i ostale, no klikom na jedan od primjera web preglednik će korisnika usmjeriti na stranicu koja koristi CMS za uređivanje sadržaja.



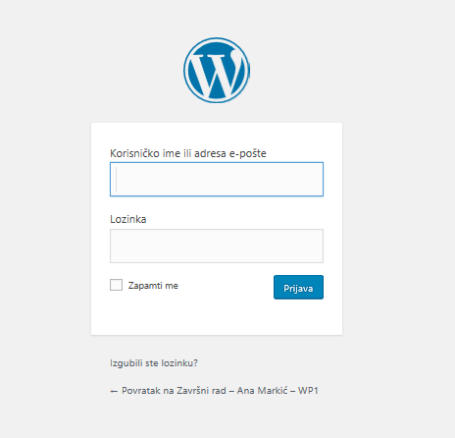
Slika 13: Pod stranica s poveznicama na primjere Wordpress stranica

Oba primjera Worpdress stranica se nalaze na istom poslužitelju kao i glavni dio, odnosno statička stranica, samo su odvojene u zaseban direktorij. Kao što je u cjelini 8.3. opisano, sama instalacija Wordpress-a je preuzeta sa službenih stranica te je podignuta na poslužitelja i provedena instalacija. Većina poslužitelja danas uz sam prostor nudi i baze podataka, pa tako i poslužitelj koji je korišten za potrebe praktičnog dijela. Prije početka instalacije bilo je potrebno kreirati bazu podataka, postaviti korisnika i lozinku, te sve to unijeti prilikom instalacije Wordpress-a kako bi znao gdje da sprema podatke koji će se unositi odnosno objavljivati na stranicama. Sam postupak je poprilično jednostavan i većina poslužitelja će korisnika voditi korak po korak do samog kreiranja baze i svega potrebnog.

Nakon što je instalacija Wordpress-a na poslužitelju završena, sve od postavljanja teme, sadržaja i ostalog obavlja se preko Wordpress kontrolne ploče. Pošto je sama instalacija Wordpress-a

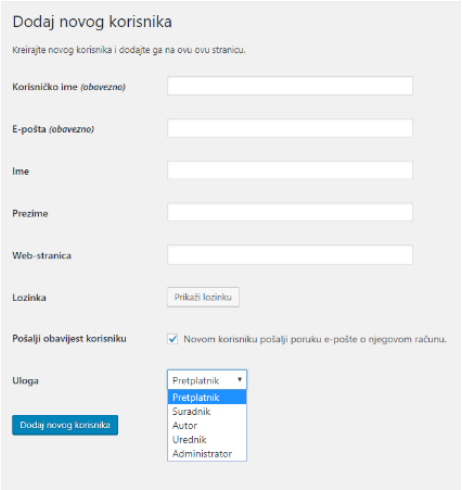
svuda jednaka, tako se i kontrolnoj ploči svuda pristupa na sličan način. Poveznica koja vodi do stranica Wordpress-a je http://anamarkic.eu/foi/wp1/ za prvi primjer, odnosno http://anamarkic.eu/foi/wp2/ za drugi primjer. Nadzorna ploča Wordpress-a je ništa drugo nego PHP skripta koja se otvara u web pregledniku, a pristupiti joj se može tako da se na kraju poveznice doda *wp-admin* , što je ujedno i naziv skripte. Stoga je poveznica za kontrolnu ploču prvog primjera http://anamarkic.eu/foi/wp1/wp-admin , a drugog primjera http://anamarkic.eu/foi/wp2/wp-admin.

Klikom na gore navedene poveznice u pregledniku se otvara sljedeći obrazac u koji je potrebno unijeti podatke korisnika koji će uređivati stranicu. Podatke glavnog korisnika odnosno administratora potrebno je podesiti prilikom instalacije. Administrator ima sva prava i mogućnosti da kreira druge korisnike i dodjeljuje njima prava koja trebaju imati. Nakon unosa podataka i prijave otvara se nadzorna ploča koja je prikazana ranije u radu (Slika 11).



Slika 14: Prijava na kontrolnu ploču

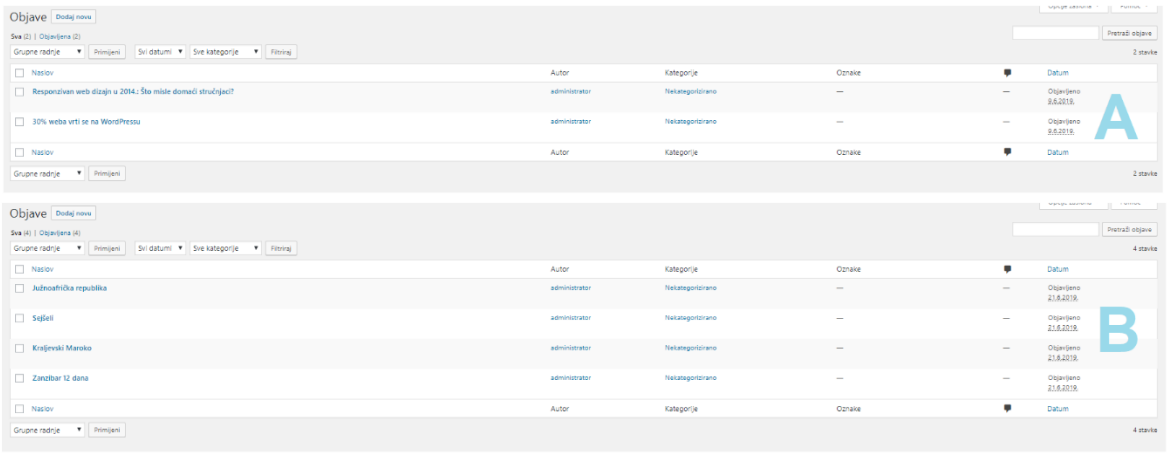
Na nadzornoj ploči se podešavaju sve postavke, upravlja se sa novim člancima i stranicama, uređuju medijski zapisi te upravlja komentarima posjetitelja stranice. Pri prvom postavljanju stranice dobro je odmah kreirati korisnike koji će upravljati stranicom. Pod stavkom *Korisnici* nalazi se opcija *Dodaj novi* za kreiranje korisnika. Kao što se vidi na slici 15, sam postupak je poprilično osnovan, unesu se podatci, odredi se uloga korisnika, te se potvrdom na gumb podatci korisnika spremaju u bazu podataka, te omogućava prijava korisniku s podatcima koji su za njega određeni.



Slika 15: Forma za kreiranje korisnika na Wordpress-u

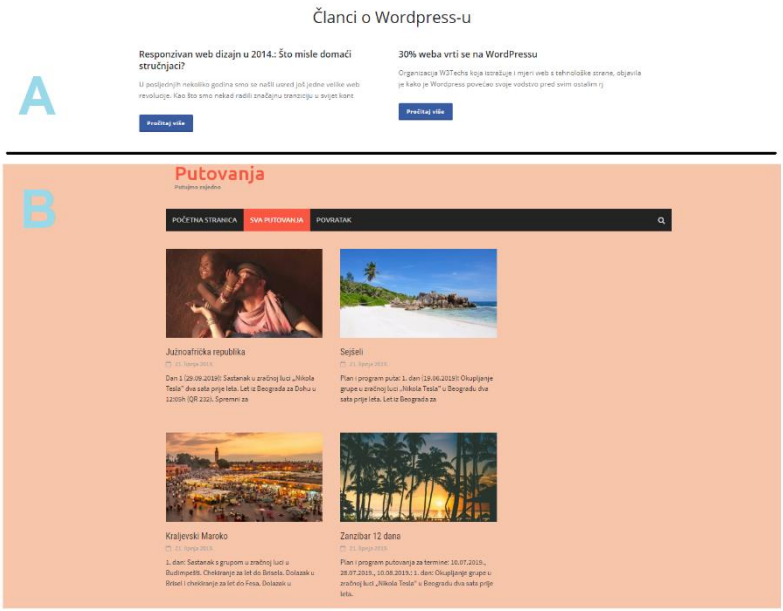
Pod opcijom *Objave* se uređuju sve objave koje će biti dostupne posjetiteljima stranice. Ovdje je moguće kreirati nove i brisati stare objave, izmjenjivati objave, skrivati ih sa stranice i slično. Objave mogu imati različitu namjenu, kao u slučaju ovog

praktičnog rada. U prvom primjeru Wordpress stranice objave su zapravo članci i novosti o Wordpress-u, dok su iste u drugom primjeru ponude putovanja.



Slika 16: Prikaz objava na upravljačkoj ploči (A – primjer 1, B – primjer 2)

Ovisno o temi stranice, objave se prikazuju na različite načine. U upravljačkoj ploči će one izgledati jednako, dok je ono što posjetitelji vide različito. Slike 16 i 17 prikazuju objave na upravljačkoj ploči te na samoj web stranici kakve ih posjetitelji vide. Može se primijetiti da na slici 16 objave izgledaju jednako, samo je sadržaj drugačiji, dok je prikaz za posjetitelje potpuno drugačiji (Slika 17).



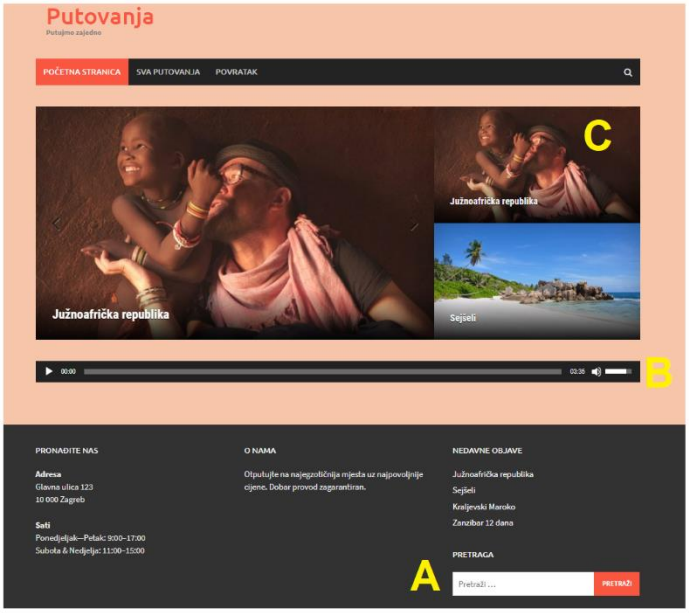
Slika 17: Prikaz objava na stranici (A - primjer 1, B - primjer 2)

Ostale opcije i mogućnosti Wordpress-a su ukratko opisane u poglavlju 8.4. Izgled sustava, no Wordpress sam po sebi ima jako dobro razvijenu podršku na jako puno jezika (uključujući i hrvatski), te su sve opcije objašnjenje na upravljačkoj površini prilikom podešavanja. Cilj Wordpress-a i jest da se i najosnovniji korisnici mogu lako snaći s njim, te bez potrebe za

plaćanjem web dizajnera ili upravitelja sadržajem (eng. Content Manager) mogu sami uređivati svoje stranice.

U prvom primjeru Wordpress-a korištene su samo osnovne opcije bez nekih dodataka. Nekim korisnicima će to biti sasvim dovoljno. No mogućnosti Wordpress-a su jako velike, pa tako i primjer 2 pokazuje neke od dodataka koji su mogući. Bitno za napomenuti je da tih dodataka ima mnoštvo, teško je zamisliti nekakav dodatak za web stranicu a da ga se ne može naći na Wordpress-u. Na slici 18 se može vidjeti 3 primjera dodataka koji Wordpress stranicu čine bogatijom. S obzirom da je za primjer napravljena stranica za putovanja, korisno je na njoj imati pretraživač koji će posjetitelju olakšati potragu za određenom destinacijom. S toga je na stranicu dodana opcija pretrage (A). Zbog ljepšeg izgleda i pristupačnosti stranice (a i toga što odabrana tema podržava mogućnost) dodan je i prečac na tražilicu u navigacijskoj traci desno.

Zatim, zbog ostavljanja boljeg dojma na posjetitelja, dodan je izvođač MP3 datoteke, odnosno pjesme (B).



Slika 18: Dodatci na Wordpress stranici

I na kraju dodatak za automatsko listanje objava u ovom primjeru stranice za putovanja može poslužiti kao mali marketinški alat, da bi putovanja koja posjetitelji ne planiraju ipak bila i njima vidljiva, pa možda se netko odluči i za to (C).

Korištenjem dodataka na Wordpress-u može se lakše postići cilj i opravdati postojanje stranice, samo je bitno prije početka stvaranja stranice dobro proučiti i razraditi samu svrhu i namjenu stranice, razmisliti što treba sadržavati ta stranica i poslužiti se trikovima iz poglavlja 3: Psihologija dizajna. Ako se pridoda puno truda i rada, web stranica će jako brzo biti ono za što je namijenjena.

10. ZAKLJUČAK

Otkako je čovjek otkrio čari interneta ne prestaje potreba za stalnim usavršavanjem, prilagođavanjem i razvijanjem novih opcija na njemu. Isprva je Internet, a i ono što bi se trebalo nalaziti na njemu, bio zamišljen za potrebe vojske ili vladinih službi, kako bi mogli bolje i brže razmjenjivati informacije i međusobno komunicirati. Ubrzo su ljudi shvatili da se s njim može puno više od samo toga, te da se ne treba ograničavati na samo jedan dio korisnika, tako da je danas Internet postao mjesto gdje s može naći bilo što, bilo kakva web stranica, bilo kakav sadržaj. Od stranica za zabavu, do ozbiljnih stranica za razmjenu znanja, edukacije, kupovinu i slično. Kroz ovaj rad može se vidjeti kako se i sadržaj na internetu mijenjao. Ispočetka smo imali samo tekst, pa su dodane slike i kojekakvi simboli, nakon toga je došla potreba da sve to izgleda malo bolje, pa kada se razvila mogućnost uređivanja sadržaja korisnici su poželjeli imati razne animacije, video zapise i slično. Do toga da više nije bilo dovoljno imati samo statičke web stranice, nego svakodnevne izmjene sadržaja su natjerale razvojne programere da naprave i sustave za upravljanje sadržajem. I na kraju danas imamo i tehnike i načine za prilagoditi te web stranice i sve njihove funkcionalnosti na uređaje različitih namjena i dimenzija. Iako su u radu obrađene samo osnove, i s tim se može jako puno toga napraviti.

Internet tehnologije napreduju iz dana u dan i gotovo je nemoguće biti u toku sa svima njima. Većina poslova i zanimanja se polako prebacuju u on-line svijet i sve više i više toga će u skorijoj budućnosti se obavljati samo na internetu. Iako se mnogima čini kompliciranо, naučiti neku od tehnologija za razvoj i dizajn na internetu nije toliko teško. Potrebno je samo malo truda, vremena i volje, a ostalo sve s vremenom dođe na mjesto. Bitno je napraviti prvi korak i krenuti u taj „novi“ svijet, jer kako stvari trenutno stoje, internet i tehnologije za razvoj i dizajn na internetu će samo nastaviti rasti, a tako i potreba za novim web dizajnerima.

POPIS LITERATURE

[1] Web Design Museum, org. (n.d.). Web Design History Timeline 1990-2019. Preuzeto 22.3.2019. s https://www.webdesignmuseum.org/web-design-history

[2] Bos, B. (2016., December 17). A brief history of CSS until 2016. Preuzeto 2.1.2019. s https://www.w3.org/Style/CSS20/history.html

[3] Perls, F. S., Hefferline, R. F., & Goodman, P. (2013). *Gestalt Therapy: Excitement and Growth in the Human Personality*. London: Souvenir Press.

[4] Weinschenk, S. (2011). *100 things every designer needs to know about people*. Berkeley, CA: New Riders.

[5] ZWICK, C. (2003). *Digital colour for the internet and other media*. Lausanne i Worthing UK: Ava Publishing.

[6] MASLOW, A. H. (2018). *THEORY OF HUMAN MOTIVATION*. SAD: WILDER PUBLICATIONS.

[7] Soegaard, M. (2019., February). Hick's Law: Making the choice easier for users. Preuzeto 12.3.2019. s https://www.interaction-design.org/literature/article/hick-s-law-making-the-choice-easier-for-users

[8] Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. Indianapolis, IN: John Wiley & Sons.

[9] MEYER, E. A. (2017). *CSS: The definitive guide*. SAD: OREILLY MEDIA.

[10] Peterson, C. (2014). *Learning Responsive Web Design: A beginners guide*. Sebastopol: OReilly.

[11] Wordpress org. (n.d.). How to install Wordpress. Preuzeto 10.2.2019. s 1https://wordpress.org/support/article/how-to-install-wordpress/

[12] WordPress. (n.d.). Preuzeto 18.2.2019. s https://en.wikipedia.org/wiki/WordPress

POPIS SLIKA

Slika 1: Maslowijeva hijerarhija 8
Slika 2: Kategorizacija elemenata 18
Slika 3: Primjer prikaza tablice u web pretraživaču 28
Slika 4: Primjer prikaza obrasca u web pretraživaču 31
Slika 5: Semantička struktura 33
Slika 6: Web stranica BBC-a sa i bez CSS-a 36
Slika 7: Sintaksa CSS pravila 37
Slika 8: Prikaz responzivne web stranice na različitim uređajima 40
Slika 9: Primjer medijskog upita 41
Slika 10: Zastupljenost sustava za uređivanje sadržaja (CMS) 44
Slika 11: Upravljačka površina Wordpress-a 49
Slika 12: Početna stranica završnog rada 51
Slika 13: Pod stranica s poveznicama na primjere Wordpress stranica 52
Slika 14: Prijava na kontrolnu ploču 53
Slika 15: Forma za kreiranje korisnika na Wordpress-u 53
Slika 16: Prikaz objava na upravljačkoj ploči (A – primjer 1, B – primjer 2) 54
Slika 17: Prikaz objava na stranici (A - primjer 1, B - primjer 2) 54
Slika 18: Dodatci na Wordpress stranici 55

POPIS TABLICA

Tablica 1: Svojstva medijskih upita.....42