

Usporedba WordPressa s ručnim kodiranjem mrežne stranice

Butigan, Jelena

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:131:281326>

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

Download date / Datum preuzimanja: **2024-07-28**



Sveučilište u Zagrebu
Filozofski fakultet
University of Zagreb
Faculty of Humanities
and Social Sciences

Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2019./ 2020.

Jelena Butigan

**USPOREDBA WORDPRESSA S RUČNIM KODIRANJEM MREŽNE
STRANICE**

Završni rad

Mentor: dr.sc. Kristina Kocijan, izv. prof.

Zagreb 2020.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

(potpis)

Sadržaj

1. Uvod	3
2. Izrada mrežnog sadržaja.....	5
3. Ručno kodiranje	7
3.1. HyperText Markup Language –HTML.....	9
3.2. Cascading Style Sheets –CSS	11
3.3. JavaScript.....	13
4. Web 2.0 – nova era World Wide Web-a	15
5. Sustavi za upravljanje Web sadržajem-WCM.....	17
6. WordPress.....	19
6.1. Primjena WordPress-a.....	19
6.2. Razvoj WordPressa	20
6.3. Varijante WordPress-a	21
7. Praktična usporedba	22
7.1. Izrada Web stranice koristeći WordPress.....	23
7.2. Izrada Web stranice ručnim kodiranjem.....	25
7.2.1. Kôd HTML-a za izradu izbornika	25
7.2.2. Dodavanje CSS-a i JavaScripta	26
7.3. Rezultati usporedbe	28
8. Zaključak.....	30
9. Literatura.....	32
Sažetak.....	34

1. Uvod

Izrada Web sadržaja (engl. *Web development*) je termin koji se koristi za proces razvoja mrežnih stranica za Internet (World Wide Web) ili intranet (privatnu mrežu)¹ („Eyerys.com“, 2020). Razvoj tog fenomena u ovom radu razdvojit će se na dva dijela. Prvi dio razvoja jest izrada mrežnog sadržaja devedesetih godina kojeg je obilježio razvoj HTML-a (engl. *HyperText Markup Language*). Drugi dio razvoja bile bi rane 2000-te kada dolazi do pojave aplikacija za izradu Web sadržaja bez kodiranja. Objedinit će se i predstaviti te dvije strane izrade Web stranica stavljajući naglasak na HTML, CSS i JavaScript u okviru ručnog kodiranja i na WordPress kao predstavnik sustava za upravljanje Web sadržajem.

Za početak valja naglasiti da Web stranice imaju različite funkcije s obzirom na sadržaj koji predstavljaju te se i različito izrađuju. Ovaj rad bavit će se mrežnim identitetom obrazovnih ustanova te će praktičnim načinom doći do zaključka koji model je bolje koristiti za izradu sadržaja za škole – ručno kodiranje ili WordPress. Cilj usporedbe jest prikazati prednosti i mane ručnog kodiranja i WordPressa te na temelju toga donijeti zaključak kada se bolje služiti kodiranjem, a kada WordPressom te odgovoriti na pitanje: „Mogu li CMS alati zamijeniti ručno kodiranje Web stranica?“.

U prvom narednom poglavlju objasnit će se što je izrada Web sadržaja u kontekstu današnjeg doba te će se dati pregled industrije Web razvoja. Drugo poglavlje bavi se samim počecima izrade mrežnog sadržaja pritom objašnjavajući koje su se tehnologije razvijale u tom periodu te kako je objavljivanje sadržaja funkcioniralo. Također, u drugom poglavlju definirat će se HTML, CSS i JavaScript. Treće poglavlje prikazat će Web 2.0 kao novu eru svjetske mreže (engl. *World Wide Web*) te promjene koje su se pojavile

¹ Slobodni prijevod autora ovog rada koji u originalu glasi: “*Web development is a term for the work involved in developing a website for the internet (World Wide Web) or an intranet (private network).* (Eyerys.com. *Web Development Industry In The Modern World*, Eyerys, 2020. Preuzeto s <https://www.eyerys.com/articles/news/web-development-industry-modern-world> (posjećeno 28.6.2020.))

ranih 2000-ih. Jedna od novina bila je pojava sustava za upravljanje sadržajem (CMS). Četrto poglavlje se nadovezuje na sustave za upravljanje sadržajem te detaljnije ulazi u sustave za upravljanje Web sadržajem gdje se prikazuje statistička važnost WordPressa. Peto poglavlje definira WordPress te prikazuje njegov razvoj. Nakon što je prikazan teoretski dio, šesto poglavlje bavi se usporedbom ručnog kodiranja i WordPressa na primjeru mrežnog sučelja za obrazovnu ustanovu. Prvo potpoglavljje će prikazati kako izgleda izrada sadržaja za WordPress, a drugo potpoglavljje bavit će se kodiranjem služeći se HTML-om, CSS-om i JavaScriptom. U zaključku će se komentirati rezultati usporedbe na temelju kojih će se pokušati predvidjeti budući trendovi u Web razvoju.

2. Izrada mrežnog sadržaja

Jedan od glavnih pojmova koji će se ponavljati u cijelom radu jest Web razvoj (engl. *Web development*) odnosno izrada mrežnog sadržaja. Stoga će prvo poglavlje objasniti što taj pojam predstavlja.

Izrada Web sadržaja obuhvaća Web dizajn, razvoj Web sadržaja, skriptiranje na strani klijenta/poslužitelja, konfiguraciju sigurnosti Web poslužitelja i mreže te razvoj e-trgovine („Eyerys.com“, 2020). Međutim, među stručnjacima, „Web razvoj“ obično se odnosi na glavne aspekte dizajniranja Web stranica: pisanje, markiranje i kodiranje. Izrada mrežnog sadržaja obuhvaća raspon od najjednostavnije stranice običnog teksta do najsloženijih internetskih aplikacija utemeljenih na internetu, primjerice elektroničke tvrtke (e-trgovine) ili društvene mreže.

Rastu Web razvoja kao industrije ponajviše pomažu velike tvrtke kojima je cilj prodavati proizvode i usluge svojim kupcima i automatizirati proces rada. Uz to, troškovi izrade Web stranica drastično su pali u proteklih dvadesetak godina. Umjesto za desetke tisuća dolara, kao što je to bio slučaj za rane Web stranice, sada se jednostavno Web mjesto može razviti za manje od tisuću dolara (ovisno o složenosti i količini sadržaja). Poslovanja za razvoj Web stranica sada čine Web dizajn dostupnim manjim tvrtkama i pojedincima, dodatno potičući rast industrije za razvoj Web stranica. Što se tiče alata i platformi za razvoj Web stranica, javnosti je na raspolaganju mnogo sustava za pomoć u kreiranju sadržaja. Ova se činjenica sama očitovala kod mnogih ljudi širom svijeta koji svakodnevno postavljaju nove Web stranice i tako doprinose povećanju popularnosti Web razvoja (Sfetcu, 2014).

Osim što društvena okolina utječe na evoluciju industrije Web razvoja vrijedi i obratno. Idealni primjer tomu su Web trgovine. Mrežne stranice za aukcije, poput eBay, promijenile su način na koji potrošači konzumiraju i kupuju proizvode i usluge.

Internetski preprodavači poput Amazona (među brojnim drugima) preobrazili su trgovinu i kupnju učinili povoljnim iskustvom za mnoge potrošače.

Često vrijedi pravilo da se manje Web stranice mogu napisati samo pomoću HTML-a i CSS-a. Veće stranice, posebice one koje se redovito ažuriraju, koriste sustav za upravljanje sadržajem (engl. *Content Management System - CMS*) (Duckett, 2014). CMS je softverski paket koji pruža određenu razinu automatizacije za zadatke potrebne za učinkovito upravljanje sadržajem (Barker, 2015). Prema istraživanjima „*W3techs.com*“ najpopularniji sustav za upravljanje sadržajem je WordPress koji su 2003. razvili američki bloger Matt Mullenweg i britanski bloger Mike Little. WordPress se najčešće koristi za stvaranje blogova, ali program je dovoljno fleksibilan da se može koristiti za stvaranje i dizajn bilo koje vrste Web stranica (Olinik & Jackson Armitage, 2011). Ta dva modela izrade, kodiranje i WordPress, bit će glavna tema rada, te će se u narednim poglavljima postaviti teorijska podloga tih načina izrade Web sadržaja.

3. Ručno kodiranje

Prethodno poglavlje dalo je uvid u izradu internetskog sadržaja u kontekstu današnjeg doba. Osim što je jedna od najmlađih i najprofitabilnijih, Web razvoj je jedna od najbrže rastućih industrija na svijetu od 1990-ih („Eyers.com“, 2020). Drugo poglavlje prikazat će same početke izrade Web sadržaja i potom definirati tri najvažnija jezika za kodiranje Web sadržaja: HTML, CSS i JavaScript.

Devedesete se s razlogom smatraju ključnim godinama za razvoj internetskog sučelja. Rođenjem Web razvoja može se smatrati datum 6. kolovoza 1991. godine kada je Tim Berners-Lee izradio prvu Web stranicu (Sammartino McPherson, 2010). Na slici 1 vidljiva je očuvana verzija stranice iz 1992. godine.

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#).)

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Slika 1. Prva Web stranica (verzija iz 1992.)

Preuzeto s: <http://info.cern.ch/hypertext/WWW/TheProject.html> (Posjećeno 4.4.2020.)

Tog datuma omogućen je softver World Wide Web za NeXT računalne urednike, preglednik i osnovni poslužitelj koji se može koristiti na svim računalima. Prije ovog datuma, samo su rijetki znali za Web. Uz tri stavke koje je Tim objavio, svatko s NeXT računalom i pristupom internetu mogao je početi s istraživanjem mreže. Datum 6. kolovoza 1991. BBC je prozvao danom kada je Web postao internacionalan. (Sammartino McPherson, 2010).

Tim Berners-Lee je dakle osim prve Web stranice razvio tri tehnologije za olakšavanje komunikacije putem interneta, koje su postale središnje u načinu na koji danas koristimo Web, to su: HTML, URI (ono što većina naziva URL-ovima)² i HTTP. Berners-Leejev koncept za svjetsku mrežu bio je softverski program, Web preglednik, koji bi korisnicima omogućio lakši pristup i kretanje sadržajem Interneta. Prva inovacija, HTML bio je uobičajeni jezik za izradu Web stranica koje su koristile unaprijed definirane 'oznake' za strukturiranje sadržaja na Web stranici. Korištenjem zajedničkog jezika za programiranje za izradu mrežnih stranica, autori su mogli osigurati da će čitatelji moći pregledati stranicu kako je namijenjeno, stvarajući konzistentnu strukturu za definiranje aspekata Web stranica kao što su slike, naslovi i odlomci. URL-ovi ili jedinstveni lokatori resursa omogućili su da sadržaj na Webu ima dosljedne „adrese“ koje se mogu identificirati i dijeliti. Dakle, sustav imenovanja „<http://www.primjer.com/podstranica>“ omogućio je korisnicima interneta širom svijeta da lako pronađu Web stranicu i njezine različite podstranice putem interneta koristeći konvencije o imenovanju ili „adrese“.

Naposljetku, HTTP (engl. *Hyper Text Transfer Protocol*) je bio specifični protokol koji je definirao kako slati i primiti Web stranice putem interneta, gdje korisnik traži datoteke s poslužitelja; olakšava komunikaciju Web preglednika s poslužiteljima što omogućava bolje upravljanje resursima (Ferguson, 2019). Berners-Lee omogućio je velikom broju ljudi pristup Web serveru. Rezultat toga bilo je rađanje novih mrežnih stranica svakodnevno. Taj fenomen on je nazvao sporom eksplozijom (Sammartino McPherson, 2010).

U tom periodu, da bi imali Web mjesto, bilo je potrebno dobro razumijevanje kôda za sastavljanje Web sadržaja. Podrazumijevalo se da morate biti stručnjak koji je sposoban spojiti sekvence HyperText Markup Language-a (HTML), što može rezultirati Web mjestom koje privlači puno ljudi³ (Brazell, 2010).

² URI i URL se često poistovjećuju, ali nisu isti, URI (engl. *Uniform Resource Identifier*) je identifikator resursa kao što su stranice, knjige ili dokumenti.

URL (engl. *Uniform Resource Locator*) je specifična vrsta URI-a koji pruža način pristupa (HTTPS, FTP, itd.) (Meissler, 2005).

³ Napomena: za to vrijeme *puno ljudi* značilo je oko 500 čitatelja.

HTML se nastavlja s razvojem pa tako 1995. dobiva verziju HTML 2.0 koja podržava obrasce, tablice, grafike i brojne nove oznake, poput `<!DOCTYPE>`, `<head>`, `<body>`, `<form>`, ``, `<input>`, `<select>`, `` i dr.. Iste te godine Brendan Eich iz Netscapea dizajnirao je prvu verziju objektno-orijentiranog jezika JavaScript koji se široko koristio za izradu interaktivnih Web stranica. 1996. godine je W3C izdao službene preporuke za jezične specifikacije za CSS1. Kaskadni stilovi (engl. *Cascading Style Sheets*) opisuju kako se elementi prikazuju u dokumentima pisanim na HTML, XHTML i XML jezicima za označavanje („Webdesignmuseum.org”, 2020). To su bili počeci velikih jezika koji će se definirati u idućem poglavlju.

Evidentno je da je Web iznimno napredovao od devedesetih. Standardi kôda su se promijenili, neke su se tehnologije prestale koristiti, a druge su se razvile. Međutim tih godina su se sagradili čvrsti temelji kôda koje i danas većina Web stranica koristi. Trendovi dolaze i odlaze, ali još uvijek je za uspješnog Web dizajnera potrebno poznavati HTML, CSS i JavaScript kao i prije 20 godina.

3.1. HyperText Markup Language –HTML

HTML je prethodno već definiran kao jezik za izradu Web stranica koji koristi unaprijed definirane oznake za strukturiranje sadržaja na Web stranici. Tim Berners-Lee, nekoliko mjeseci nakon izrade već spomenute prve Web stranice 1991., objavio je dokument pod nazivom HTML oznake (engl. *HTML Tags*). Dokument je sadržavao opis prvih 18 HTML oznaka (`<title>`, `<nextid>`, `<a>`, `<isindex>`, `<plaintext>`, `<popis>`, `<p>`, `<h1>... <h6>`, `<adresa >`, `<hp1>`, `<hp2>...`, `<dl>`, `<dt>`, `<dd>`, ``, ``, `<menu>` i `<dir>`) od kojih se većina koristi i danas. Temelj dizajna te prve verzije jezika HTML-a bio je univerzalni jezik za označivanje SGML („Webdesignmuseum.org”, 2020).

„*Britannica.com*“ SGML (engl. *Standard Generalized Markup Language*) definira kao međunarodni računalni standard za definiranje jezika za označivanje, dakle kao metajezik koji je postavio pravila kojih se HTML pridržava. Može se primijetiti da HTML oznake započinju kutnom zagradom koja je usmjerena lijevo (`<`) i završava se kutnom

zagrdom desno (>). Između zagrada nalaze se ključne riječi koje označavaju vrstu oznake. Početne oznake mogu uključivati sve parametre potrebne za oznaku, a završne oznake sadrže samo ključnu riječ s kosom crtom (/). Na primjer, ako u dokumentu jedna riječ treba biti podebljana, podeblja se oznakama **** i **** na način vidljiv u kodnom bloku 1.

```
<b>Ova rečenica je podebljana.</b>
```

Kodni blok 1. Podebljavanje u HTML-u

Kad se Web stranica preuzme u preglednik, preglednik čita i interpretira HTML oznake za prikaz web stranice s organiziranim i formatiranim tekstom, slikama i vezama (Campbell, 2018).

HTML dokument je tekstualna datoteka s ekstenzijama .htm ili .html koja u sebi sadrži HTML oznake. Svaki takav dokument započinje oznakom **<html>** i završava oznakom **</html>** (Morley& Parker, 2015).

Unatoč toj definiciji koja kaže da je oznaka **<html>** na početku dokumenta, Duckett (2014) naglašava da se na početak treba postaviti deklaracija *DOCTYPE* koja pregledniku definira točnu inačicu standarda koja se koristi za izradu HTML dokumenta. Primjerice za HTML verziju 5 dovoljno je postaviti **<!DOCTYPE html>** na sami početak datoteke da bi preglednik ispravno prikazao stranicu. Također je važno postaviti jezik stranice na početku dokumenta atributom „*lang*“. Tako bi se za hrvatski jezik napisala jezična oznaka 'hr' kao što je prikazano u kodnom bloku 2.

```
<html lang="hr">
```

Kodni blok 2. Označavanje jezika u HTML-u

HTML dokumenti imaju zaglavlje **<head>** i glavni dio **<body>**. Zaglavlje služi za definiranje metapodataka o stranici. Također se u zaglavlje dodaje stil koji može biti ugrađen (engl. *embedded*) ili poveznica na vanjsku CSS datoteku. Unutar zaglavlja se često definiraju i skripte kreirane u JavaScript jeziku. Unutar **<body>** oznaka je glavni

sadržaj Web stranice koji će se prikazati u glavnom prozoru preglednika, za razliku od sadržaja koji se nalazi u zaglavlju stranice koji se ne prikazuje u pregledniku.

3.2. Cascading Style Sheets –CSS

HTML se koristi za opisivanje sadržaja na Web stranici, a CSS opisuje kako taj sadržaj treba izgledati. U poslovanju Web dizajna, način na koji stranica izgleda naziva se prezentacija. Fontovi, boje, pozadinske slike, razmaci linija, izgled stranice i dr. kontrolirani su CSS-om (Robbins, 2012). „W3schools.com“ definira CSS kao jezik koji određuje stil HTML dokumenta te opisuje kako bi se HTML elementi trebali prikazati. Koristi se za definiranje stilova Web stranica, uključujući dizajn, izgled i varijacije prikaza za različite uređaje i veličine ekrana.

CSS omogućuje Web autorima i programerima precizno podešavanje elemenata za objavljivanje na mreži i na nekoliko različitih vrsta medija. Stilovi su elektronički izdavački izum za dinamičko kodiranje teksta i drugih elemenata dokumenta s oblikovanjem. Na primjer, stil nazvan „Naslov“ bio bi vezan uz svaki naslov u dokumentu. Definicija stila sadržavala bi informacije o tome kako naslovi trebaju biti oblikovani. Na primjer, uporaba većeg i podebljanog fonta. Prednost stilova je u tome što možete jednom promijeniti definiciju i promjena utječe na svaki element uređen pomoću tog stila. Dakle, stilovi pružaju jednostavan način za ažuriranje formatiranja dokumenata i održavanje dosljednosti na Web mjestu.

Stilovi se mogu na nekoliko različitih načina priložiti dokumentu. Jedna od metoda za definiranje stilova je dodavanje bloka stila u zaglavlje HTML dokumenta (kodni blok 3).

```
<head>
  <style type="text/css">
    /* CSS kôd */
  </style>
</head>
```

Kodni blok 3. Dodavanje CSS-a u zaglavlje

Najpoželjnije je stil kreirati u zasebnoj datoteci i povezati s dokumentom pomoću veze (<link>) kao što je vidljivo u kodnom bloku 4.

```
<head>
  <link rel="stylesheet" type="text/css" href="stil.css" />
</head>
```

Kodni blok 4. Dodavanje CSS-a u zasebnoj datoteci

Ideja kreiranja stila u zasebnoj datoteci je da se na njega može povezati više HTML dokumenata što će rezultirati da sve stranice imaju isti stil odnosno zajedničku temu. Kad se definicije stila u vanjskoj datoteci promijene, svi povezani dokumenti odražavaju promjenu. Povezivanje vanjskog CSS-a putem veze trebao bi biti preferirani način primjene stilova na dokument, jer omogućuje najviše skalabilnu uporabu stilova - morate promijeniti samo jednu vanjsku listu stilova kako biste utjecali na mnoge dokumente (Schafer, 2005).

CSS je jezik zasnovan na pravilima – definiraju se pravila koja određuju grupe stilova koje treba primijeniti na određene elemente ili grupe elemenata na vašoj Web stranici. Primjerice, "želim da glavni naslov na mojoj stranici bude prikazan kao veliki crveni tekst". Kôd koji slijedi (kodni blok 5) prikazuje vrlo jednostavno CSS pravilo kojim bi se postigao gore opisani stil.

```
h1 {
  color: red;
  font-size: large;
}
```

Kodni blok 5. Stiliziranje naslova

Pravilo se otvara selektorom kojim se odabire HTML element koji ćemo oblikovati. U ovom slučaju oblikujemo naslove prve razine (<h1>).

Zatim imamo skup vitičastih zagrada {}. Unutar njih bit će jedna ili više deklaracija koje će imati oblik parova svojstava i vrijednosti. Svaki par određuje svojstvo elemenata koje odabiremo, zatim vrijednost kojoj želimo dati svojstvo.

Prije dvotočke imamo svojstvo, a nakon dvotočke vrijednost. CSS svojstva imaju različite dopuštene vrijednosti, ovisno o tome koje je svojstvo određeno. U primjeru imamo svojstvo boje koje može poprimiti različite vrijednosti boja. Također imamo svojstvo veličine fonta. Ovo svojstvo može uzeti vrijednosti različitih veličina.

Dokument s CSS stilovima (engl. *CSS stylesheet*) sadržavat će mnoštvo takvih pravila, napisanih jedno za drugim (MDN, 2019).

3.3. JavaScript

Prethodna potpoglavlja predstavila su osnove HTML-a i CSS-a. Mnogi bi se složili da su ti jezici lakši za korištenje od JavaScripta jer je JavaScript programski jezik koji omogućuje implementaciju složenih stvari na Web stranicama (MDN, 2019). Svaki put kada mrežna stranica obuhvaća prikazivanje pravovremenih ažuriranja sadržaja, interaktivnih mapa, animirane 2D/3D grafike, pomicanje video prozora ili više – u pozadini stoji JavaScript.

JavaScript se koristi za dodavanje interaktivnih značajki na Web mjesto kao što su igre, stvari koje se događaju kad se klikne na određeni element ili unesu podaci u obrasce, dinamični efekti, animacije i još mnogo toga. JavaScript je također skriptni jezik. *Britannica.com* skriptni jezik definira kao računalni jezik namijenjen rješavanju relativno malih problema s programiranjem koji ne zahtijevaju prekomjerne deklaracije podataka i druge značajke potrebne za upravljanje velikim programima.

JavaScript se koristi za manipuliranje elementima na Web stranici te stilovima koji se na njih primjenjuju. Postoje i drugi jezici skriptiranja (primjerice VBScript, PHP, Perl, Python, Ruby, itd.), ali JavaScript (koji je nastao iz ECMAScripta⁴) smatra se standardnim (Robbins, 2012).

JavaScript je interpretativni (engl. *interpreted*) programski jezik. U interpretativnim jezicima kôd se pokreće odozgo prema dolje i rezultat pokretanja kôda odmah se vraća.

⁴ ECMAScript (engl. *European Computer Manufacturer's Association Script*) je standard za skriptne jezike u koje spada i JavaScript.

Web preglednik prima JavaScript kôd u izvornom tekstualnom obliku i iz njega pokreće skriptu (MDN, 2019).

JavaScript se primjenjuje na HTML stranicu na sličan način kao i CSS u zaglavlje dokumenta. Dok CSS koristi `<link>` elemente za primjenu vanjskih stilova, a `<style>` elemente za primjenu internih stilova na HTML, JavaScript u svijetu HTML-a treba element `<script>`. Unutarnji (engl. *internal*) kôd izgledao bi poput ovog u kodnom bloku 6.

```
<script>
// JavaScript
</script>
```

Kodni blok 6. Implementiranje JavaScripta

Vanjski (engl. *external*) kôd, uzimajući u obzir da je „skripta.js“ datoteka u istom direktoriju kao originalna HTML datoteka, prikazan je u kodnom bloku 7.

```
<script src="skripta.js" defer> </script>
```

Kodni blok 7. Dodavanje JavaScripta iz vanjske datoteke⁵

JavaScript ima mogućnosti programskih jezika primjerice pohranjivanje vrijednosti u varijable, izvršavanje operacija nad tekstom i brojne druge, što omogućava korisniku da kreira interaktivni sadržaj.

Sumirano, HTML predstavlja strukturu stranice i na njega se može gledati kao temelj na koji se mogu implementirati drugi jezici, CSS određuje kako se sadržaj HTML dokumenta prezentira klijentima, a JavaScript određuje ponašanje odnosno omogućuje interaktivnost. Današnje Web stranice se ne koriste samo spomenutim jezicima, ali budući da se smatraju trima najvažnijima spominjat će se u radu u usporedbi s WordPressom pod zajedničkim nazivom ručno kodiranje.

⁵ Atribut *“defer”* upućuje preglednik da prvo učita cijelu stranicu, potom skriptu u pozadini

4. Web 2.0 – nova era World Wide Web-a

Prethodno poglavlje prikazalo je kako je Web okruženje funkcioniralo devedesetih te koje su se tehnologije razvile u tom periodu, a ovo poglavlje predstaviti će promjene koje su se odvile ranih 2000-ih godina. Ulaskom u novo tisućljeće Internet je napredovao do točke kad je povezanost postala dostupna gotovo svugdje, a Web preglednik postao je široko dostupan i jednostavan alat za istraživanje Web-a (Frydenberg & Shelly, 2010). Rezultat tih evolucija bila je nova generacija World Wide Web-a.

„2004. godine shvatili smo da smo nadomak nove ere Web-a, ere koja će napokon osloboditi moć efekata mreže koji će pokrenuti lavinu inovacija i mogućnosti. Kako bi požurili novu eru, O'Reilly Media⁶ i CMP Technologies pokrenuli su konferenciju na kojoj su predstavljeni inovatori koji su je vodili. Kada je Dale Dougherty⁷ iz O'Reilly-ja osmislio termin „Web 2.0“ tijekom predstavljanja ideja, znali smo da imamo ime konferencije. Ono što nismo znali je da će industrija prigrliti pojam Web 2.0 i da će upravo taj pojam predstavljati novi Web“ (Musser & O'Reilly, 2006:3).

Web 2.0 danas označava pojam koji obuhvaća označavanje mrežnih stranica koje sadrže jaku socijalnu komponentu, te potiču korisnički sadržaj u obliku teksta, videa i objava fotografija zajedno s komentarima, oznakama i ocjenama (Cormode & Krishnamurthy, 2008). Nova era WWW-a omogućila je korisnicima da sudjeluju u doprinosu, organiziranju i stvaranju svog sadržaja. Time je želja za suradnjom i svježim, relevantnim sadržajem rasla, a potreba za upravljanjem sadržajem postajala je nužna. Web stranice trebalo je svakodnevno ažurirati, a sve veći broj ljudi želio je dodavati i uređivati sadržaj. Uloga sustava za upravljanje sadržajem (CMS) bila je pružanje

⁶ O'Reilly Media američka je tvrtka za učenje koja izdaje knjige, proizvodi tehnološke konferencije i pruža internetsku platformu za učenje.

⁷ Dale Dougherty je uz Tim O'Reillyja glavni suosnivač tvrtke O'Reilly Media

mogućnosti većem broju korisnika s različitim razinama dozvola za upravljanje sadržajem za Web mjesto ili dio sadržaja (Heslop, 2018).

Sustav za upravljanje sadržajem softverski je paket koji pruža određenu razinu automatizacije za zadatke potrebne za učinkovito upravljanje sadržajem. CMS je obično višekorisnički softver temeljen na poslužitelju koji komunicira sa sadržajem pohranjenim u spremištu. CMS omogućuje stvaranje novog sadržaja, uređivanje postojećeg sadržaja, izvođenje uredničkih postupaka te omogućuje dostupnost sadržaja drugim korisnicima (Barker, 2015). Treba naglasiti da je već spomenuti sustav za upravljanje Web sadržajem jedan od četiri tipa sustava upravljanja sadržajem.

Barker (2015) identificira 4 tipa upravljanja sadržajem kao:

- upravljanje organizacijskim sadržajem (engl. *Enterprise Content Management-ECM*),
- upravljanje digitalnom imovinom (engl. *Digital Asset Management-DAM*),
- upravljanje dokumentima (engl. *Records Management-RM*),
- upravljanje Web sadržajem (engl. *Web Content Management-WCM/ WCMS*).

Upravljanje sadržajem neke organizacije, nije nužno namijenjeno masovnoj isporuci ili potrošnji (npr. životopis zaposlenika, izvješća o incidentima, dopisi i sl.). Taj je podtip tradicionalnije bio poznat kao „upravljanje dokumentima“, ali oznaka se generalizirala tijekom godina. ECM se ističe u suradnji, kontroli pristupa i upravljanju datotekama.

DAM se odnosi na upravljanje i manipulaciju bogatim digitalnim sadržajima kao što su slike, audio i video za upotrebu u drugim medijima. DAM ima izvrsne rezultate u metapodacima i reprodukciji.

RM je upravljanje transakcijskim informacijama i drugim zapisima koji se stvaraju kao nusprodukt poslovnih aktivnosti (npr. evidencija prodaje, evidencija pristupa, ugovori itd.).

WCM je upravljanje sadržajem prvenstveno namijenjeno masovnoj isporuci putem Web stranice. WordPress se smatra pripadnikom ove podvrste CMS-a stoga će iduće poglavlje detaljnije objasniti što su sustavi za upravljanje mrežnim sadržajem.

5. Sustavi za upravljanje Web sadržajem-WCM

Sustavi za upravljanje sadržajem imaju široku primjenu za stvaranje sadržaja na Webu. Mnogi od ovih sustava nude mogućnost objavljivanja sadržaja klasificiranog kao Web 2.0 sadržaj. Sustavi za upravljanje Web sadržajem omogućuju korisnicima s malim ili nikakvim iskustvom Web dizajna ili programiranja da kreiraju i ažuriraju sadržaj na Internetu. Oni nude razinu sigurnosti za omogućavanje pristupa ažuriranju ili brisanju sadržaja. Sustavi za upravljanje sadržajem mogu korisnicima ponuditi tijekom rada za upravljanje koracima unosa, odobravanja i objavljivanja kako bi korisnik mogao unositi novi sadržaj (Itahriouan et al., 2012). Počevši od 2003. godine, CMS Web mjesta koja se lako koriste su nudila gotove predloške za ljude koji nemaju iskustva s kodiranjem, poput WordPressa (2003), SquareSpacea (2003), Weeblyja (2006), Wixa (2006) itd. Navedene platforme pružale su put do stvaranja malog i jeftinog mrežnog mjesta koje nije zahtijevalo znanje o HTML-u, CSS-u i kodiranju (Heslop, 2018).

	2019 1 Mar	2019 1 Apr	2019 1 May	2019 1 Jun	2019 1 Jul	2019 1 Aug	2019 1 Sep	2019 1 Oct	2019 1 Nov	2019 1 Dec	2020 1 Jan	2020 1 Feb	2020 1 Mar	2020 29 Mar
None	44.7%	44.6%	44.2%	44.1%	43.9%	43.7%	43.6%	43.6%	43.4%	43.2%	43.1%	42.8%	42.7%	42.9%
WordPress	33.3%	33.5%	33.8%	34.0%	34.1%	34.3%	34.5%	34.6%	34.8%	35.1%	35.4%	35.7%	35.9%	36.1%
Joomla	2.9%	2.9%	2.9%	2.8%	2.8%	2.8%	2.8%	2.8%	2.7%	2.7%	2.6%	2.6%	2.5%	2.5%
Shopify	1.5%	1.5%	1.5%	1.6%	1.6%	1.6%	1.7%	1.7%	1.7%	1.8%	1.9%	2.0%	2.1%	2.2%
Drupal	1.9%	1.9%	1.9%	1.9%	1.8%	1.8%	1.8%	1.7%	1.7%	1.7%	1.7%	1.7%	1.6%	1.6%
Squarespace	1.5%	1.5%	1.5%	1.5%	1.6%	1.6%	1.6%	1.6%	1.6%	1.6%	1.5%	1.5%	1.5%	1.5%
Wix	1.0%	1.0%	1.1%	1.1%	1.1%	1.1%	1.2%	1.2%	1.3%	1.3%	1.3%	1.3%	1.3%	1.3%
Bitrix	0.6%	0.6%	0.6%	0.6%	0.6%	0.7%	0.7%	0.7%	0.8%	0.8%	0.9%	0.9%	0.9%	1.0%

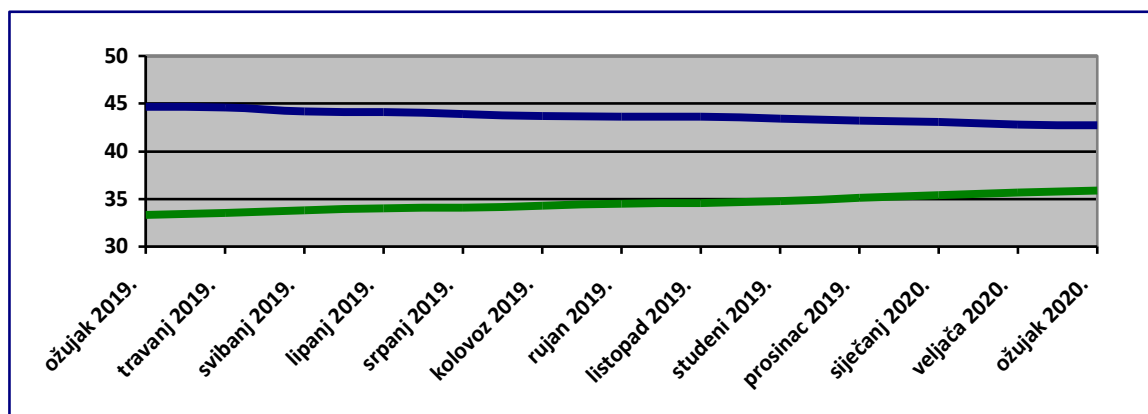
Tablica 1. Trendovi u korištenju sustava za upravljanjem Web sadržajem od ožujka 2019. Do ožujka 2020. Preuzeto s:

https://w3techs.com/technologies/history_overview/content_management/all (Posjećeno 29.3.2020.)

Razlog zašto se uzima WordPress kao predstavnik WCM-a odnosno CMS-a općenito u ovom radu prikazuje Tablica 1 na kojoj su prikazani trendovi u korištenju CMS sustava za izradu Web sadržaja od ožujka 2019. do ožujka 2020. Može se jasno iščitati kolika je nadmoć WordPressa nad drugim sustavima u postotcima za svaki mjesec protekle

godine. Postotak stranica koje se koriste WordPressom je jedini koji dolazi blizu stranicama koje su izrađene bez CMS sustava, odnosno stranice izrađene ručnim kodiranjem.

Tako dolazimo do druge bitne stavke koja se da iščitati iz tablice, a to je odnos postotka WordPressa i ručnog kodiranja kroz istih 12 mjeseci koji nam prikazuje graf na slici 2. Graf je izrađen pomoću podataka iz Tablice 1 za potrebe ovog završnog rada. U grafu plava linija predstavlja popularnost ručnog kodiranja, a zelena linija popularnost WordPressa u postotcima. Da se primijetiti da popularnost WordPressa raste, a suprotno vrijedi za ručno kodiranje. Pretpostavka je da će u bliskoj budućnosti broj stranica izrađenih u WordPressu biti jednak broju stranica izrađenih čistim kodiranjem. Glavni razlog zašto se WordPressom sve više koristi jest jednostavnost korištenja i drugi faktori koji će se prikazati u idućem poglavlju.



Slika 2. Odnos postotka stranica na internetu izrađenih ručnim kodiranjem i WordPressom od ožujka 2019. do ožujka 2020.

6. WordPress

U prethodnom poglavlju prikazan je WordPress kao najvažniji sustav za upravljanje mrežnim sadržajem. Cilj ovog poglavlja je predstaviti zašto WordPress bilježi toliku popularnost danas. „*Technopedia.com*“ (2017) definira WordPress kao besplatnu aplikaciju za Web objavljivanje odnosno kao alat kojim se mogu izraditi Web stranice koje se jednostavno ažuriraju. WordPress omogućio je da korisnici kreiraju Web mjesta bez napornog programiranja odabirom postojeće teme i dodavanjem sadržaja. Također, iskusniji korisnici s programskim vještinama i znanjem mogu kreirati platforme ili uređivati PHP predloške, stilove CSS-a i JavaScript funkcije kako bi proširili funkcionalnost ili stvorili vlastite teme i / ili dodatke (Norrie et al., 2014).

6.1. Primjena WordPress-a

Uz sve alate dostupne putem WordPressa, vrste Web lokacija koje možete stvoriti gotovo su beskrajne. Caldwell (2019) navodi 7 tipova stranica za koje se WordPress najčešće koristi, a koje ćemo ukratko opisati u nastavku.

1. Osobne stranice

WordPress nudi alate koji pomažu obogatiti osobnu stranicu koja se koristi u informativne svrhe, primjerice za elektroničke životopise.

2. Blogovi

WordPress je u početku bio namijenjen isključivo blogovima. Bilo da se radi o javnom ili privatnom blogu, puno je tema i dodataka koje bi Web mjesto bloga učinilo popularnim. WordPress omogućava platformu za kreiranje i isporuku sadržaja.

3. Online trgovine

WordPress nudi brojne mogućnosti za postavljanje internetskih trgovina.

4. Poslovne stranice

Osim online trgovina WordPress se može koristiti za mnoge druge vrste poslovanja na internetu. Tvrtke kao što su New York Post, TED, USA Today, CNN, TIME.com i TechCrunch sve koriste WordPress za upravljanje sadržajem. Lako je brzo izgraditi i pokrenuti Web lokaciju koja će rasti s proširivanjem poslovanja.

5. Neprofitne organizacije

Neke tvrtke, poput neprofitnih organizacija i vjerskih skupina, ne postoje u kapacitetu za ostvarivanje profita. Pristup besplatnim uslugama kad god je to moguće od presudne je važnosti za ove skupine. WordPress ima mnogo besplatnih i otvorenih rješenja koja i dalje omogućuju tvrtkama koje ne ostvaruju prihod da izrade profesionalne Web stranice.

6. Direktoriji

Direktorij je kataloška struktura sustava datoteka koja sadrži reference prema ostalim računalnim datotekama i prema drugim direktorijima. WordPress omogućava automatsko povezivanje podstranica što čini izradu bilo kakvih direktorija jednostavnim.

7. Multikulturalne stranice

Višejezična podrška ogroman je faktor za programere koji možda ne govore isti jezik kao njihovi klijenti ili ciljana publika na Web mjestu.

6.2. Razvoj WordPressa

Michel Valdrini 2001. pokrenuo je projekt b2/cafelog, open-source sustav za upravljanje sadržajem pomoću PHP1 i koristeći MySQL2 kao svoju bazu podataka. 2003. godine Matt Mullenweg i Mike Little stvaraju ogranak projekta u kojem su iskoristili legalnu kopiju softvera za stvaranje izrazito novog proizvoda, s drugačijim svrhom i smjerom. Softver stvoren kao rezultat postao je WordPress. Mullenweg i Little nastojali su razviti sustav za stvaranje blogova (*engl. blogging*) koji je bio više fokusiran na korisničko iskustvo i Web standarde, što u to vrijeme još nije bilo popularno.

Vremenom su glavne značajke (uključujući dodatke, teme, vrste objava i prilagođenu navigaciju) dodane i poboljšane, što je rezultiralo izuzetno robusnim sustavom internetskog objavljivanja koji se i dalje razvija. Sam WordPress aktivno razvija i podržava nekoliko glavnih programera - uključujući Mullenwega, Littlea i Valdrinija i volonterski tim od oko 100 ključnih suradnika koji marljivo rade na tome da WordPress učine što kvalitetnijim softverom. Također, programeri širom svijeta nastavljaju s izgradnjom i podrškom novih dodataka i tema koji su korisni za niz svrha, a prevoditelji pomažu dobrovoljnim tumačenjem dodataka, tema i samog WordPressa na dodatne jezike na kojima postoji potreba, dodatno povećavajući opseg i domet projekta (Olinik, Jackson Armitage, 2011).

6.3. Varijante WordPress-a

Danas je WordPress dostupan u dvije varijante WordPress.com i WordPress.org. **WordPress.org** je besplatni program otvorenog kôda koji se instalira na vlastito računalo koji omogućuje kreiranje vlastitih predložaka.

WordPress.com je inačica WordPress.org-a koja se ne instalira već je dostupna online. Jednostavnija je za korištenje, ali manje fleksibilna od instalirane verzije WordPressa.

Pronalaženje, bavljenje i održavanje odnosa s pružateljem usluga hostinga najveći su potencijalni faktori gnjavaže u korištenju WordPress.org i zajedno bi mogli biti najveći pojedinačni problem zbog kojeg se blogeri okreću ka WordPress.com. Također troškovi su u pravilu veći za WordPress.org. S druge strane, WordPress.com je strogo limitiran predlošcima i nadogradnje nisu moguće. WordPress.org nije limitiran što znači da će se uložiti više vremena u izradu Web stranice, ali će se dobiti bolji rezultati (Smith & McCallister 2011). Iduće će se poglavlje baviti praktičnom primjenom WordPressa.

7. Praktična usporedba

Prethodno se u radu postavila teorijska podloga koja je predstavila što je ručno kodiranje i koji su najvažniji jezici u izradi Web sadržaja, također je definiran najvažniji sustav za upravljanje Web sadržajem (WCM) – WordPress. Naglasak je na podatku iz četvrtog poglavlja da je WordPress sve popularniji u izradi Web sadržaja te da se postotak stranica izrađenih korištenjem WordPressa približava postotku stranica izrađenih kodiranjem. Ta činjenica postavlja pitanje što je bolje koristiti za izradu Web sadržaja.

Ovaj dio rada bavit će se praktičnom izradom Web sadržaja koristeći se WordPressom i ručnim kodiranjem. Učinkovitost će se promatrati kroz tri faktora: utrošeno vrijeme, jednostavnost izrade sadržaja i fleksibilnost.

Spomenuta dva modela izrade Web sadržaja promatrat će se na mrežnom sjedištu obrazovne ustanove. Primarna svrha Web stranica takvih ustanova jest pregledno prikazati važne informacije za učenike, njihove roditelje te djelatnike. Također je važno da se sadržaj na takvim stranicama često ažurira, dok vizualni identitet najčešće nije od velike važnosti. Cilj ove usporedbe jest uvidjeti glavne prednosti i mane dvaju načina izrade te odgovoriti na pitanje koji model je praktičnije koristiti za izradu sučelja jedne obrazovne ustanove.

Za temu stranice izabrana je naslovna stranica Osnovne škole Opuzen čije je trenutno mrežno sjedište prikazano na slici 3. Za takvu stranicu važna je preglednost te dostupnost važnih informacija, stoga se naglasak usporedbe stavlja na izradu elementa glavnog izbornika, jer je to jedna od najvažnijih stavki stranica.



Slika 3. Mrežno sučelje Osnovne škole Opuzen. Preuzeto s: <http://os-opuzen.skole.hr/> (Posjećeno 4.7.2020.)

7.1. Izrada Web stranice koristeći WordPress

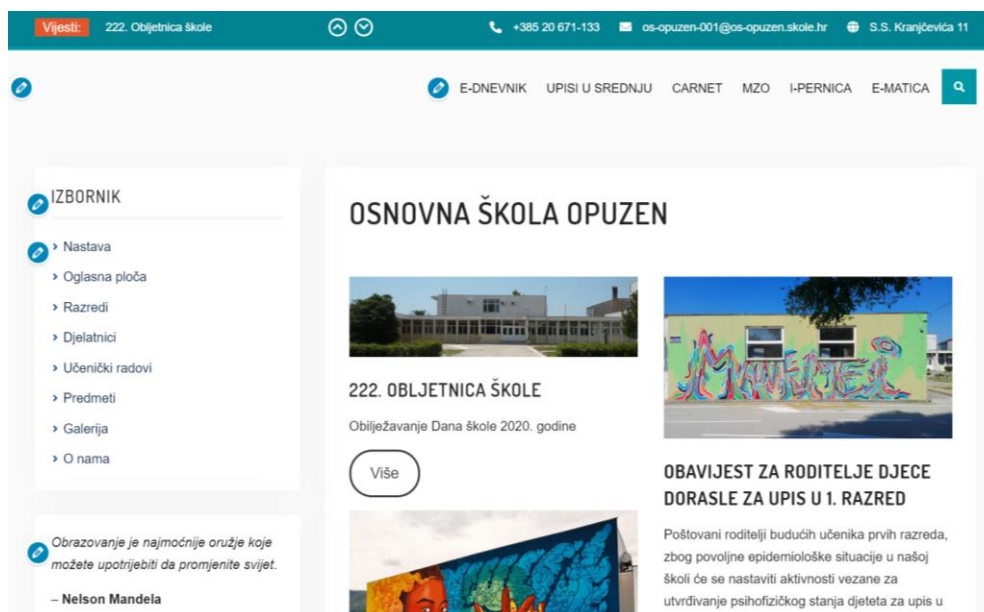
Nakon instalacije WordPressa otvara se preglednik s jednostavnim predloškom Web stranice. Prvi korak je biranje jedne od oko 4000 predložaka/tema koje WordPress nudi. Osim što se teme razlikuju vizualno, svaka ima posebne opcije uređivanja, primjerice neke teme podržavaju samo jednu lokaciju izbornika, dok druge mogu podržavati i do pet različitih lokacija na kojima je moguća pozicija izbornika. Za ovu stranicu je odabrana tema „Kids Education Soul“. Nakon što se izabere tema, uređuje se identitet stranice u kojem se postavljaju logo, ime stranice, slogan i ikona stranice.

Izabrana tema podržava više lokacija za izbornike od kojih je izabran izbornik u uzglavlju s važnim poveznicama te padajući izbornik koji je postavljen s lijeve strane glavne stranice (vidljivo na slici 4). Kada se u WordPressu kreiraju izbornici, točnije stavke izbornika, automatski se kreiraju podstranice koje se otvaraju klikom na te stavke. Tako automatski kreirane stranice imaju istu temu kao i naslovna strana. Taj automatizam štedi mnogo vremena prilikom izrade.

Nakon što je uređen izgled stranice, kreće se s uređivanjem sadržaja naslovnice. U ovom slučaju izabrano je da će na naslovnoj stranici biti prikazane objave. WordPress funkcionira po principu blokova, što znači da se za dodavanje sadržaja mora prethodno specificirati o kakvom se sadržaju radi. Za dodavanje novog bloka odredi se hoće li blok biti paragraf, naslov, slika, zvuk, lista, itd. Problematika takvih blokova je fleksibilnost. Korisnik nije u mogućnosti potpuno upravljati pozicijom sadržaja što vrlo lako može rezultirati vizualno neatraktivnom stranicom.

Na posljertku slijedi dodavanje objava, što je ujedno i najlakši dio izrade sadržaja koristeći WordPress. U glavnom meniju se otvori opcija *Dodaj objavu* (engl. *Add Blog post*) i otvara se čista stranica u koju se dodaje naslov objave, a potom i ostatak sadržaja koji se dodaje u blokovima. Prednost WordPressa za izradu stranice osnovne škole bi bio da svi nastavnici jednostavno mogu dodavati svoje objave bez prethodnog znanja o izradi Web sadržaja.

Rezultat je sučelje vidljivo na slici 4 koje je izrađeno za potrebe ovog završnog rada. U vrlo kratkom vremenu uspjeli smo dobiti pregledno i funkcionalno sučelje. Na slici se može primijetiti dosta neiskorištenog prostora iznad glavnog dijela stranice što se nije moglo popraviti.



Slika 4. Mrežno sučelje izrađeno WordPressom

7.2. Izrada Web stranice ručnim kodiranjem

Za razliku od WordPressa, kodiranje stranice korisniku omogućava potpunu slobodu prilikom dodavanja sadržaja. To znači da nema nikakvih predložaka i sličnih automatizama. S kreiranjem sadržaja se kreće od potpuno prazne stranice, stoga se podrazumijeva da korisnik ima prethodno znanje iz područja HTML-a, CSS-a i JavaScripta.

Za kreiranje potpunog Web sjedišta jedne osnovne škole, potrebno je na desetke HTML datoteka koje će kreirati naslovnicu s pridruženim podstranicama. Uz HTML datoteke se pridruži jedna CSS datoteka u kojoj su sadržani svi elementi koji će se stilizirati, a kod JavaScripta će biti sadržan u HTML datotekama. Idealni primjer elementa na kojem se može prikazati implementiranje sva tri jezika jest gornji izbornik stranice koji sadrži poveznice: E-dnevnik, upisi u srednju, Carnet, MZO, i-Pernica, e-Matica. Cilj je dobiti funkcionalni izbornik sa poveznicama koje vode na podstranice koristeći HTML, zatim implementirati CSS kojim će se izbornik prilagoditi željenom izgledu stranice te dodati kôd JavaScripta koji će moći prikazati i skrivati izbornik ovisno o korisniku.

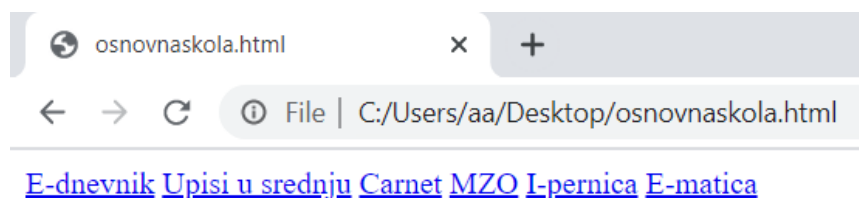
7.2.1. Kôd HTML-a za izradu izbornika

HTML je temelj za izradu svakog elementa Web stranice. Zato se kreće od definiranja samog izbornika (kodni blok 8).

```
<div id="linkovi">
  <a href="ednevnik.html">E-DNEVNIK</a>
  <a href="upisi.html ">UPISI U SREDNJU</a>
  <a href="carnet.html ">CARNET</a>
  <a href="mzo.html ">MZO</a>
  <a href="ipernica.html ">I-PERNICA</a>
  <a href="ematica.html ">E-MATICA</a>
</div>
```

Kodni blok 8. HTML kôd za izradu izbornika

Rezultat tog kôda jest slika 5 koja prikazuje prvu verziju izbornika prije uređivanja stila. Oznaka <a> kojoj je pridružen atribut href označava poveznicu na drugu stranicu. Prije dodavanja stilova kreira se 6 HTML datoteka na koje će poveznice voditi. Postignut je funkcionalni izbornik koji omogućava navigaciju na ostale stranice klikom miša. Može se primijetiti da su klasične plave poveznice identične onima sa slike 1 na kojoj je prva Web stranica. Danas se izgled čistog HTML-a svakako smatra zastarjelim i neprivlačnim. Na korisnika takva stranica može ostaviti dojam da je stranica nepouzdana stoga je važno urediti stil.



Slika 5. Izbornik izrađen HTML-om

7.2.2. Dodavanje CSS-a i JavaScripta

Nakon što je izbornik izrađen, potrebno ga je stilizirati implementacijom CSS-a (kodni blok 9).

```
#linkovi {
  background-color: #018391;
  position: fixed;
  top: 0;
  width: 100%;
  transition: top 1s;
}

#linkovi a {
  float: left;
  display: block;
  color: white;
  text-align: center;
  padding: 15px;
  text-decoration: none;
}
```

Kodni blok 9. CSS kôd za izradu izbornika

Dodavanjem CSS-a dobije se izbornik kao na slici 6. Iako je HTML temelj, CSS svakako ima veliku važnost za svaku Web stranicu, jer prvo što će korisnik primijetiti jest izgled stranice. CSS omogućava web dizajneru potpunu slobodu i kreativnost prilikom postavljanja sadržaja. Tako se ne može dogoditi da ostane neiskorišten prostor kao kod WordPressa.



Slika 6. Izgled izbornika nakon dodavanja CSS-a

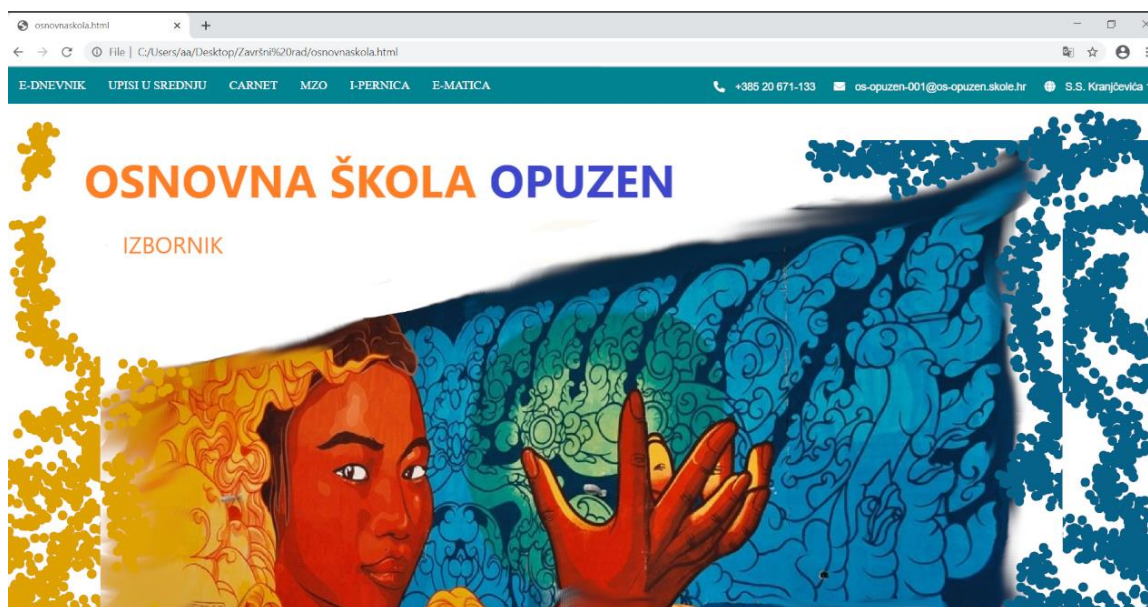
Dodavanjem JavaScripta ne mijenja se izgled izbornika već njegovo „ponašanje“. JavaScripta će nam omogućiti da prilikom pomicanja stranice prema dnu, izbornik nestaje, a prilikom pomicanja prema vrhu stranice se izbornik ponovno pojavljuje. U ovom slučaju JavaScript se dodao u zaglavlje HTML dokumenta unutar elementa <script> (kodni blok 10).

```
var prevScrollpos = window.pageYOffset;
window.onscroll = function() {
  var currentScrollPos = window.pageYOffset;
  if (prevScrollpos > currentScrollPos) {
    document.getElementById("linkovi").style.top = "0";
  } else {
    document.getElementById("linkovi").style.top = "-50px";
  }
  prevScrollpos = currentScrollPos;
}
```

Kodni blok 10. JavaScript kôd za izradu izbornika

Jasno se zaključuje da izrada sadržaja kodiranjem zahtjeva više vremena i znanja od izrade sadržaja služeći se WordPressom. Međutim, sadržaj izrađen kodiranjem potpuno je jedinstven i njime se može manipulirati. Ručno kodiranje bilo bi rješenje za privatne škole iz dva glavna razloga: privatnim školama važan je mrežni identitet jer im je u interesu zainteresirati posjetitelje, te takve škole mogu priuštiti informatičku službu koja bi održavala mrežno sjedište.

Za stranicu Osnovne škole Opuzen za potrebe završnog rada izrađeno je sučelje vidljivo na slici 7. Za glavnu temu naslovnice je iskorišten simbol škole (mural koji je oslikan na samoj ustanovi) čije su boje također važan čimbenik vizualnog identiteta. Time je postignuto da se dobije sučelje vizualno zanimljivo djeci te da im se prikaže poznati prizor za razliku od WordPress naslovnice koja bi se vjerovatno učenicima činila dosadnom.



Slika 7. Mrežno sučelje Osnovne škole Opuzen izrađeno ručnim kodiranjem

7.3. Rezultati usporedbe

Izrada Web sadržaja WordPressom i ručnim kodiranjem dva su potpuno različita postupka koja daju sličan produkt. Glavna razlika je svakako činjenica da je WordPress alat koji je sličniji Microsoft Office PowerPointu nego bilo kakvom kodiranju. Glavni nedostatak WordPressa je fleksibilnost, sadržaj se ne može pozicionirati po želji zato se događa da ostaje previše ili premalo bjelina između blokova. S druge strane prednost je, osim jednostavnosti korištenja, jednostavno objavljivanje novog sadržaja. WordPress bi bio rješenje za Web stranice koje često imaju nove objave od korisnika koji nužno ne poznaju kodiranje.

S druge strane, ručno kodiranje kreatoru pruža potpunu slobodu da razvije jedinstveni sadržaj. Korisnici koji su vješti u kodiranju kao rezultat rada mogu dobiti vizualno atraktivnu stranicu koja će privući velik broj posjetitelja. Za uspješnu stranicu ulaže se vrijeme u učenje kodiranja te u samu izradu stranice, što se može smatrati nedostatkom ručnog kodiranja. Kodiranje se najčešće koristi za stranice za koje je važno prikazati neki brend u najboljem svjetlu te privući što više posjetitelja. Za stranicu Osnovne škole Opuzen bi bolje bilo koristiti WordPress jer za stranicu škole vizualni identitet nije od presudne važnosti, već preglednost i dostupnost sadržaja. WordPress bi omogućio svim djelatnicima škole da učitavaju svoje materijale i objave koji bi bili dostupni učenicima na jedinstvenoj platformi.

8. Zaključak

Kako tehnologija napreduje tako postaje sve jednostavnija za korištenje, te samim time, sve dostupnija široj javnosti. Korisnici su uvijek u potrazi za lakšim i bržim načinima za obavljanje nekog posla. Danas tako imamo slučaj da je sve veći broj internetskih stranica izrađen služeći se WordPressom zbog jednostavnosti korištenja. WordPress je omogućio da svatko tko je informatički pismen može izraditi mrežnu stranicu potpuno besplatno. Time se nameće pitanje hoće li WordPress nadomjestiti ručno kodiranje kao broj jedan u izradi Web stranica.

Ovaj rad ponudio je teoretski i praktični prikaz izrade Web sadržaja WordPressom te ručnim kodiranjem da bi se raspravilo kako funkcioniraju oba načina izrade. Prvo je predstavljen temelj rada koji povezuje WordPress i ručno kodiranje – izrada Web sadržaja. Potom je prikazan period devedesetih u kojem su se razvili HTML, CSS i JavaScript. Iduća poglavlja predstavila su Web početkom 2000-ih u kojem se razvio i WordPress. Praktična usporedba je dala uvid kako se izrađuje sadržaj kodiranjem i WordPressom na primjeru Web stranice osnovne škole. Rezultati usporedbe su naglasili da je Web stranicu brže i jednostavnije izraditi koristeći WordPress, ali kodiranjem se dobije stranica koja omogućava više radnji te je vizualno privlačnija.

Može se pretpostaviti da će popularnost WordPressa nastaviti s rastom te da će uskoro na Webu više stranica biti izrađeno koristeći se WordPressom nego čistim kodiranjem. Međutim, to nužno ne znači da ručno kodiranje izumire. Kodiranjem se dobivaju profesionalnije Web stranice i teško je zamisliti bilo kakav sustav koji bi mogao zamijeniti sve mogućnosti koje kodiranje nudi. Na kraju krajeva, valja spomenuti da WordPress ne bi postojao bez kodiranja i da ne vrijedi obratno. Kodiranje se konstantno razvija te se dodaju nove inačice postojećih jezika, nove mogućnosti, pravila i novi jezici za izradu Web stranica. Razlog zašto je WordPress sve korišteniji jest da razvojem

WordPress on postaje jednostavniji za korištenje, a razvojem ručnog kodiranja kodiranje postaje kompliciranije.

9. Literatura

1. Barker, Deane. *Web content management: Systems, features, and best practices*. Sebastopol: O'Reilly Media, 2016.
2. Brazell, Aaron. *Wordpress bible*. Indianapolis: John Wiley & Sons, 2010.
3. Caldwell, Ronald. *The History of WordPress: From B2/Cafelog to Managed WordPress Hosting*. Nexcess, 2019. Preuzeto s: <https://blog.nexcess.net/wordpress-history/>. (Posjećeno 28.6.2020.)
4. Cormode, Graham & Balachander Krishnamurthy. *Key differences between Web 1.0 and Web 2.0*. Chicago: First Monday, 2008.
5. Duckett, Jon. *HTML & CSS Design and Build Websites*. Indianapolis: John Wiley & Sons, Inc., 2011.
6. Eyerys.com. *Web Development Industry In The Modern World*. Eyerys, 2020. Preuzeto s: <https://www.eyerys.com/articles/news/web-development-industry-modern-world> (posjećeno 28.6.2020.)
7. Ferguson, Kevin L. "Game Changers." In *Pop Goes the Decade: The Nineties*. Santa Barbara: ABC-CLIO, 2019.
8. Heslop, Brent. "History of Content Management Systems and Rise of Headless CMS." Contentstack, 2018. Preuzeto s: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms>. (Posjećeno 28.6.2020.)
9. Itahriouan, Zakaria, et al. "Validated CMS: Towards new generation of web content management systems on Web 2.0." *International Journal of Information Technology and Computer Science (IJITCS)* 4 (2012): 12-40.
10. MDN. "What Is CSS?," 2019. Preuzeto s: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS (Posjećeno 28.6.2020.)

11. Miessler, Daniel. "What's the Difference Between a URI and a URL?"
danielmiessler.com, 2005. Preuzeto s: <https://danielmiessler.com/study/difference-between-uri-url/> (Posjećeno 4.7.2020.)
12. Morley, Deborah, and Charles S. Parker. *Understanding computers: Today and tomorrow, comprehensive*. Stamford: Cengage Learning, 2014.
13. Musser, John, and Tim O'Reilly. *Web 2.0: Principles and best practices*. Sebastopol: O'Reilly Media, 2007.
14. Norrie, Moira C., et al. "The forgotten many? a survey of modern web development practices." *International Conference on Web Engineering*. New York City: Springer, 2014.
15. Olinik, Mick, and Raena Jackson Armitage. *The WordPress Anthology*. SitePoint, 2011.
16. Robbins, Jennifer Niederst. *Learning Web design: A beginner's guide to HTML, CSS, JavaScript, and Web graphics*. Sebastopol: O'Reilly Media, Inc., 2012.
17. Sammartino McPherson, Stephanie. *Tim Berners-Lee: Inventor of the World Wide Web*. Minneapolis: Twenty-First Century Books, 2009.
18. Schafer, Steven M. *Web standards programmer's reference: HTML, CSS, JavaScript, Perl, Python, and PHP*. Indianapolis: John Wiley & Sons, 2007.
19. Sfetcu, Nicolae. *Web design & development*. Izdavač nepoznat, 2014.
20. Shelly, Gary, and Mark Frydenberg. *Web 2.0: Concepts and applications*. Boston: Cengage Learning, 2010.
21. Smith, Bud E., and Michael McCallister. *Wordpress in depth*. Que Publishing, 2011.
22. Web design Museum. "Web Design History Timeline," 2020. Preuzeto s: www.webdesignmuseum.org/web-design-history (Posjećeno 28.6.2020.)

Sažetak

U ovom radu usporedit će se dva načina izrade Web sadržaja: najpoznatiji alat za upravljanje sadržajem –WordPress i ručno kodiranje. Prikazat će se općenito što su alati za upravljanje sadržajem (CMS) te koja je važnost WordPressa za Web razvoj. Također će se objasniti počeci Weba te zašto su HTML, CSS i JavaScript od iznimne važnosti za izradu Web sadržaja općenito. Drugi dio rada je praktični dio koji se bavi usporedbom kreiranja jednostavne Web stranice koristeći WordPress i ručno kodiranje. Pritom će se prikazati glavne prednosti, mane i mogućnosti oba načina izrade Web sadržaja. Cilj usporedbe je odgovoriti na pitanje: „Mogu li CMS alati zamijeniti ručno kodiranje Web stranica?“ Zaključno će se raspraviti moguća budućnost za izradu Web sadržaja.

Ključne riječi: *Web sadržaj, Web razvoj, WordPress, HTML, CSS, JavaScript*

Comparison of WordPress and manual coding

Abstract

This paper will compare two ways of creating Web content: the most well-known content management tool - WordPress and manual coding. It will be shown what Content Management Systems (CMS) are and what is the importance of WordPress for Web development. It will also explain the beginnings of the Web and why HTML, CSS and JavaScript are extremely important for creating Web content in general. The second part of the paper is a practical part that deals with the comparison of creating a simple Web site using WordPress and manual coding. The main advantages, disadvantages and possibilities of both ways of creating Web content will be presented. The aim of the comparison is to answer the question "Can CMS tools replace manual Website coding?" In conclusion, the possible future for the creation of Web content will be discussed.

Key words: *Web content, Web development, WordPress, HTML, CSS, JavaScript*