

# Prilagodba procesa razvoja Wordpress teme

---

Jakabfi, Sven

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:364234>

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

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



**SVEUČILIŠTE U ZAGREBU**

**GRAFIČKI FAKULTET**

**SVEN JAKABFI**

**PRILAGODBA PROCESA RAZVOJA  
WORDPRESS TEME**

**DIPLOMSKI RAD**

Zagreb, 2015.



Sveučilište u Zagrebu  
Grafički fakultet

**SVEN JAKABFI**

**PRILAGODBA PROCESA RAZVOJA  
WORDPRESS TEME**

**DIPLOMSKI RAD**

Mentor:

Prof. dr. sc., Lidija Mandić

Student:

Sven Jakabfi

Zagreb, 2015.

## **Zahvala:**

***Zahvaljujem se cjelokupnom predavačkom  
osoblju Grafičkog fakulteta u Zagrebu,  
posebice mentorici, prof. dr. sc. Lidiji  
Mandić, na utrošenom vremenu i  
prenesenom znanju i iskustvu.***

# SAŽETAK

U svijetu u kojem se broj uređaja sa pristupom internetu eksponencijalno širi dolazi do sve veće potražnje za rješenjima dostupnim za relativno pristupačnu cijenu i brzim razvojem koje bi zadovoljilo potrebe malih i srednjih tvrtki kao i potrebe ljudi koji žele imati osobnu web stranicu.

Kao jedno od vodećih rješenja nametnula se Wordpress platforma. Od svojih početaka kao blogerska platforma do danas izrasta u jedan od najmoćnijih alata za izradu dinamičnih web stranica.

Uz nagli porast korisnika *Wordpressa* gotovo jednakom brzinom raste i developerska zajednica koja na Internet izdaje razne, lako dostupne, dodatke i teme.

Iako Wordpress velikim dijelom svoju popularnost i prilagodljivost duguje dodacima razvijenima od treće strane, zbog performansi i sigurnosti, najboljim temama se smatraju one koje zahtijevaju što manje potrebe za instalacijom dodatka.

## KLJUČNE RIJEČI

Wordpress, web, responzivni web, *wordpress* teme, optimizacija downloada

## **ABSTRACT**

In a world where the number of Internet accessible device is increasing exponentially, the need for relatively affordable solutions with quick development process which would satisfy necessities of small and medium enterprises, as well as people who wish to own a personal web site, is increasing.

As one of the leading solutions which imposed itself is the Wordpress platform. Since its beginnings as a blogging platform until this day it has come to be one of the most powerful tools used in building dynamic web sites.

As the number of Wordpress users grow, so did, at almost the same rate, the Internet community which launches all sorts of easily accessible plugins and themes.

Although Wordpress owes its popularity and adaptability largely to appendages developed by third party, due to performances and safety issues, the best theme are deemed those which require the least amount of installed plugins.

## **KEYWORDS**

Wordpress, web, responsive web, Wordpress themes, download optimization

# Sadržaj

<b>1.</b>	<b>UVOD .....</b>	<b>1</b>
<b>2.</b>	<b>TEHNOLOGIJE I PROGRAMSKI JEZICI.....</b>	<b>3</b>
2.1.	HTML.....	3
2.2.	CSS.....	5
2.3.	JAVASCRIPT .....	8
2.4.	PHP.....	10
2.5.	WORDPRESS .....	12
<b>3.</b>	<b>PROBLEM PRILAGODBE UREĐAJIMA S RAZLIČITIM VELIČINAMA EKRANA .....</b>	<b>14</b>
3.1.	RESPONZIVNA TEMA .....	16
<b>4.</b>	<b>PROBLEM BRZINE UČITAVANJA.....</b>	<b>19</b>
4.1.	MINIMIZIRANA TEMA.....	20
<b>5.</b>	<b>EKSPERIMENTALNI DIO RADA .....</b>	<b>22</b>
5.1.	KONTROLNA TEMA .....	23
<b>6.</b>	<b>REZULTATI I DISKUSIJA .....</b>	<b>27</b>
<b>7.</b>	<b>ZAKLJUČAK.....</b>	<b>34</b>
<b>8.</b>	<b>LITERATURA .....</b>	<b>36</b>
<b>9.</b>	<b>POPIS SLIKA.....</b>	<b>37</b>

# 1. Uvod

Wordpress je jedan od najpoznatijih *Content Management* sustava (CMS) danas. Razvijen je 2003. godine, a do danas je izrastao u najvećeg samostalnog blog *hosta* na svijetu kojeg koriste milijuni web stranica i kojeg vide deseci milijuna posjetitelja svakog dana. Wordpress je počeo kao blog servis koji je svojim korisnicima omogućavao jednostavno objavljivanje članaka i ažuriranje blogova. Danas se Wordpress koristi u različite svrhe, od samog blog sustava do platforme za dinamične stranice i internetske trgovine, prvenstveno iz razloga što je razvio cijeli sustav predložaka, ali i dodataka koji su relativno jednostavni za korištenje jednom prosječnom korisniku ovog CMS-a.

Implementiran je PHP programskim jezikom te mu je potpora MySQL sustav za upravljanje bazom podataka. Korištenje Wordpress CMS-a regulirano je GNU licencom (*General Public Licence 2.0*) koja definira pravila korištenja, reprodukcije i distribucije izvornog koda sustava. Zbog svoje jednostavnosti korištenja i dostupnosti velikog broja modula i dodataka za vlastitu prilagodbu, Wordpress je danas jedan od najčešće korištenih sustava za upravljanje sadržajem (CMS).

Wordpress je moguće primijeniti na različitim hosting okruženjima. Korisnik ima mogućnost preuzeti zadnju inačicu *Wordpressa* sa stranice Wordpress.org, te nakon toga izvršiti instalaciju preuzetog paketa preko sustava za upravljanje poznatijeg kao *Cpanel*. Svakom korisniku *WordPress* CMS-a omogućen je odabir određenog broj temi, odnosno predložaka. Osim instalacije gotovih predložaka, moguće je kreirati vlastiti predložak korištenjem Wordpress alata te tehnologija HTML i CSS, *JavaScript* i PHP.

U ovom radu korištene su tri različite prilagođene Wordpress teme. Kontrolna tema koja nema značajke *responzivnog* dizajna weba te dvije optimizirane teme. Optimizacije brzine skidanja stranice i prilagodljivosti svim uređajima su odabrane



zbog rastuće potrebe prikaza na uređajima različitih veličina ekrana . Jedna od tema optimizirana je po principu responzivnog dizajna prilagodbe za učitavanje na različitim veličinama ekrana. Ovim načinom prilagodbe dobiva se na preglednosti stranice bilo da je učitana na prijenosnom računalu, tabletu ili mobilnom uređaju bilo koje rezolucije. Osim preglednosti istražiti će se dolazi li do smanjivanja vremena učitavanja u odnosu na kontrolnu temu na uređajima manje rezolucije ukoliko se učitava stranica s automatski prilagođenom širinom i visinom sveukupnog sadržaja. U drugoj prilagođenoj temi koristiti će se minimiziranje datoteka te testirati brzina učitavanja stranice u odnosu na učitavanje kontrolne stranice. Minimiziranjem datoteka brišu se svi razmaci i prazni redovi, a cijeli kod se generira unutar jednog retka. Ovaj proces moguće je provesti putem mnoštva dostupnih besplatnih programa na internetu.

Potrebno je analizirati problem brzine učitavanja kontrolne teme te pronaći rješenja prilagodbom i optimiziranjem ostale dvije teme. Mjerenjima će biti prikazane razlike između optimiziranih i kontrolne teme pomoću broja akcija na ekranu te vremena potrebnog da se izvrši određena akcija.

## 2. Tehnologije i programski jezici

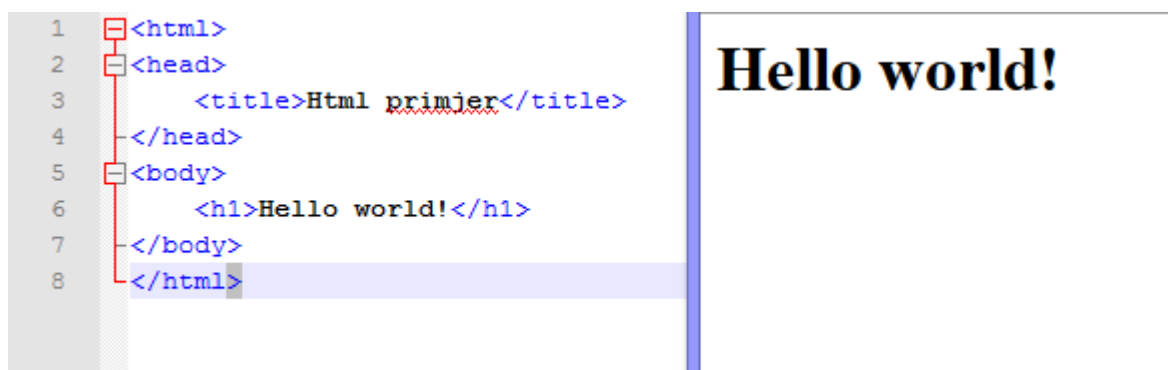
### 2.1. HTML

**HTML (*HyperText Markup Language*)** je računalni *markup* (označivački) jezik koji je standard za web dokumente odnosno on je jezik u kojem je pisana struktura web stranica.

Razvijen je od strane fizičara, i osnivača Tima Berners-Leeja u CERN-u početkom 90ih godina 20-og stoljeća. Prva verzija HTML-a razvijena je radi lakše razmjene dokumenata među znanstvenicima u CERNU i sadržavao je 18 elemenata. Jedanaest od tih elemenata se koriste i u modernom webu.

HTML se sastoji od elemenata zvanih *tagovi*. Imena *tagova* se pišu unutar zagrada „`<element>`„. Svaki *tag* (izuzev nekolicine: `<br/>`, `<hr/>`...) ima svoj otvarajući oblik `<element>` i zatvarajući element `</element>`. Iako zatvaranje elemenata nije obavezno te će većina pretraživača uspješno interpretirati željeni *tag*, zatvaranje svakog *taga* smatra se dobrom praksom i smanjuje mogućnost preklapanje s drugim *tagovima* i neželjene poteškoće prilikom kodiranja kompleksnijih web dokumenata.[1] Između otvarajućeg i zatvarajućeg elementa *taga* upisuju se vrijednosti koje pretraživač treba interpretirati ili se ugnježđuju drugi *tagovi* (slika 1). Svakom HTML elementu moguće je pridružiti određen set atributa kao što su „*class*“, „*id*“, „*value*“. Atributi se dodaju unutar otvarajućeg elementa *taga* te služe kao dodatne informacije koje pretraživač koristi prilikom njihovog interpretiranja.

```
1 <html>
2 <head>
3   <title>Html primjer</title>
4 </head>
5 <body>
6   <h1>Hello world!</h1>
7 </body>
8 </html>
```



Slika 1. Primjer jednostavnog HTML koda

Sam HTML dokument se sastoji od nekoliko dijelova: deklaracija, zaglavlje (*head*) i tijelo stranice unutar kojeg je sadržan sav sadržaj (*body*). Unutar deklaracije upisuju se osnovni podaci o dokumentu koji su potrebni pretraživaču da ga ispravno interpretira (koje standarde koristi, o kojoj se inačici HTML- a radi...). Unutar zaglavlja, odnosno zaglavlja upisuju se meta podaci i poveznice na vanjske datoteke koje će se pozivati unutar samog dokumenta. Unutar zaglavlja se definira tip kodiranja znakova, ključne riječi za web tražilice i pozivanje vanjskih CSS i *JavaScript* dokumenata. *Body* ili tijelo dokumenta je dio koji se prikazuje krajnjim korisnicima, odnosno unutar njega se prikazuje cjelokupan sadržaj dokumenta.

Napretkom tehnologije javljaju se i inačice HTML-a od kojih je HTML5 najnovija, a s njom je došao i skup novih elemenata poput „<video>„, „<audio>„ ili primjerice „<nav>„ koji u sebi ima predefinirane attribute navigacije pa je samo rukovanje kodom i elementima na stranici prilično jednostavnije. HTML5 je zapravo novi standard prihvaćen od strane W3C (*World Wide Web Consortium*) koji pruža nove mogućnosti, ali i ispravke dosadašnjih nedostataka i smatra se evolucijom HTML-a 4.1.[2] Osim novosti u samoj sintaksi, HTML5 donio je sa sobom i promjene koje oslobađaju od uporabe plug-inova za pretraživače poput *QuickTime*-ma, *Real Playera* ili *Adobe Flasha*.

Iako se HTML smatra otvorenim jezikom te njegova sintaksa nije striktna u odnosu na neke programske jezike, pridržavanje svih pravila smatra se dobrom praksom.[2]

## 2.2. CSS

**CSS (Cascading Style Sheets)** komplementarni jezik HTML-u pomoću kojeg se uređuje izgled i stiliziranje HTML dokumenta. [1]

Sredinom 90-ih godina 20. stoljeća, rastom popularnosti interneta i uporabe HTML-a, pojavila se potreba za jezikom koji bi omogućavao jednostavnije uređivanje dokumenata i konzistentniji izgled stranica u vidu prikazivanja na sve većem broju pretraživača koji su se pojavljivali na tržištu (problem koji je i danas aktualan u web dizajnu). Godine 1994. W3C konzorciju je predstavljeno 9 jezika od kojih dva, CHSS i SSP, će kasnije predstavljati bazu na kojoj je razvijen CSS danas u upotrebi.

Prema načinu uporabe CSS se može podijeliti u tri grupe: vanjski, unutarnji i *inline*. Unutarnji CSS se piše u zaglavlju samog HTML dokumenta unutar taga `<style>` kao što je prikazano na slici 2.

Primjer uporabe:

```
<style type=„text/css“>
    h1{
        color:red;
    }
</style>
```

*Inline* pisanje CSS-a se odnosi pisanje CSS koda unutar samoj taga koji se želi stilizirati, i to kao vrijednost atributa „style“.

Primjer uporabe:

```
<h1 style=„color:red;“>Hello world!</h1>
```

Vanjski CSS se definira u zaglavlju HTML dokumenta kao tag `<link>` s atributima „href“ koji označava gdje se referencirana CSS datoteka nalazi, te atributima „rel“ i „type“ koji pružaju pretraživaču podatke o datoteci potrebne za

interpretaciju. U praksi se preferira korištenje vanjskog CSS-a te se unutarnje i *inline* pisanje smatraju lošom praksom. [1]



```
1 <html>
2 <head>
3   <title>Html primjer</title>
4   <link rel="stylesheet" href="style.css" />
5 </head>
6 <body>
7   <h1>Hello world!</h1>
8 </body>
9 </html>

1 h1 {
2   color: red;
3 }
```

**Hello world!**

Slika 2. Primjer uporabe vanjskog CSS-a

Princip rada i sintaksa CSS-a su jednostavni. Unutar datoteke prvo se referencira element HTML-a te se unutar vitičastih zagrada upisuju atributi njihove vrijednosti koji će se pridodati. Referenciranje na pojedini element moguće je na jedan od četiri načina: referenciranjem na „*id*“ atribut elementa, referenciranjem na „*class*“ atribut elementa, upisivanjem hijerarhijskog poretka roditelja tog elementa ili referenciranjem na sam element.

Također, prednost CSS-a je nasljednost atributa. Atributi i njihove vrijednosti pridodani nekom elementu se prosljeđuju njihovoj djeci, što je vrlo važna značajka koja pruža više mogućnosti uz manje koda, što je bitno za brzinu otvaranja stranica, te je vrlo bitna za prilagodbu stranica uređajima različitih veličina ekrana.

Iako se sam CSS smatra tehnologijom za razdvajanje dizajna stranice od sadržaja, danas CSS ima glavnu ulogu u razvoju *responzivnog* dizajna, odnosno dizajna stranice prilagođenog za sve veličine ekrana.[1]

Također jedna od većih prednosti i glavnih uloga u današnjem razvoju Weba je činjenica da su HTML i CSS su u savršenom odnosu razina težine i kontrole za

interakciju dizajnera. Ispravno učenje koda omogućuje dizajneru direktno sudjelovanje u implementaciji programskog rješenja. [3]

## 2.3. JavaScript

*JavaScript* je dinamički objektno orijentiran skriptni programski jezik koji se koristi u web tehnologijama. Njegova primarna funkcija je dodavanje dinamičnosti i interaktivnosti web stranicama. [4]

Razvijen je od strane Netscape-a (jedan od vodećih ljudi u razvojnom timu nekoliko godina kasnije postaje suosnivač Mozille) kao laganija i kompatibilnija verzija Jave. Prva stabilna verzija puštena je 1995. godine, pod imenom *LiveScript* te ju je podržavao *Netscape Navigator 2.0*. Od svog puštanja u javnost *JavaScript* je postao standardni programski jezik na web stranicama za izvođenje skripti sa strane klijenata.

Prema svojoj sintaksi *JavaScript* često podsjeća na programske jezike poput C-a, C++ i Jave s konstrukcijom koda kao što su „if“ uvjeti, „while“ i „for“ petlje te „&&“ operatori. *JavaScript* također nije specifičan programski jezik već je definiran kao skriptni jezik zbog njegove relativne jednostavnosti korištenja, odnosno omogućuje korištenje i razumijevanje većem broju korisnika, ne striktno programerima. [4]

Slično CSS-u, *JavaScript* se može pisati unutar samog HTML dokumenta unutar zaglavlja ili se može referencirati na vanjski *JavaScript* file. I unutarnji *JavaScript* i vanjski se koriste pomoću `<script>` taga.

Vrlo bitan aspekt *JavaScripta* je činjenica da je *event driven* jezik. *JavaScript* funkcije se mogu pisati na način da ih pokreće određen event, odnosno događaj na web stranici, najčešće korisnička interakcija s nekim elementom. [5]

Rastom popularnosti *JavaScripta* počeli su se pojavljivati korisnički *library* odnosno skupine funkcija i prototipova pisanim u *JavaScriptu* koji omogućuju jednostavniji zapis koda. Najpoznatiji i najpopularniji *JavaScript library* je *Jquery*.

Prva stabilna inačica je izdana 2006. godine pod krilaticom „*Write less, do more*“ (Manje piši, radi više) *JQuery* se koristi na više od 80% od 10 000 najpopularnijih web stranica. Prednost *JQuerija* je izrazito intuitivna sintaksa koja je u potpunosti kompatibilna sa standardnim *JavaScript-om* koja omogućava jednostavno dohvaćanje i manipulaciju elementima HTML dokumenta. Od svog izdavanja *JQuery* je doživio nekoliko nadopuna od kojih su najvažnije *JQuery UI*, koja pruža dodatne opcije korisničke interakcije te *JQuery Mobile* koji pojednostavljuje interakciju korisnika s web stranicom na mobilnim uređajima.



## 2.4. PHP

**PHP (*PHP Hypertext Preprocessor*)** je najpopularniji programski jezik koji se koristi prilikom razvoja web stranica. Za razliku od ostalih ovdje opisanih tehnologija PHP se izvodi na serveru umjesto unutar pretraživača.

Razvoj PHP-a je započeo 1994. godine nezavisni programer Rasmus Lerdorf. U svojim počecima PHP nije bio zamišljen kao novi programski jezik već kao niz skripti pisanih u C-u čija je sintaksa nalikovala programskom jeziku Perl. Prva verzija je izdana 1995. godine pod nazivom PHP/FI (*Personal Home Page/Form Interpreter*). Kako se PHP razvijao sve se više odvajao od svojih početaka vezanih uz C te je 1997. godine tim programera razvio novi *parser* za PHP čime je postao neovisni programski jezik. [6]

Najveća prednost PHP-a nad ostalim programskim jezicima koji se izvode na serveru je njegova kompatibilnost sa HTML-om. Web stranice koje se dinamično generiraju pišu se unutar dokumenta sa .php nastavkom. U njima se HTML piše kao i u svojem nativnom formatu, s tim da je u njima moguće pisati PHP linije koda. PHP kod se piše unutar svog *taga* koji se otvara pomoću znakova `<?php` te se zatvara znakovima `?>`. [6]

Primjer uporabe:

```
<!DOCTYPE html>

<html>

  <head>

    <title>PHP Test</title>

  </head>

  <body>
```

```
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

PHP se najčešće koristi u kombinaciji sa *MySQL* bazama podataka odakle, pomoću upita, dohvaća podatke te generira stranicu ovisno o korisničkom zahtjevu. Orijentira se po C i *Pearl* sintaksi, a koristi se isključivo za programiranje dinamičnih web stranica. PHP je slobodan softver i kao takav se koristi pod PHP licenciranim uvjetima. [6]

Danas se PHP koristi u svom originalnom obliku za razvijanje dinamičnih web stranica ili pomoću CMS platformi kao što su *Wordpress*, *Drupal* ili *Moodle*.

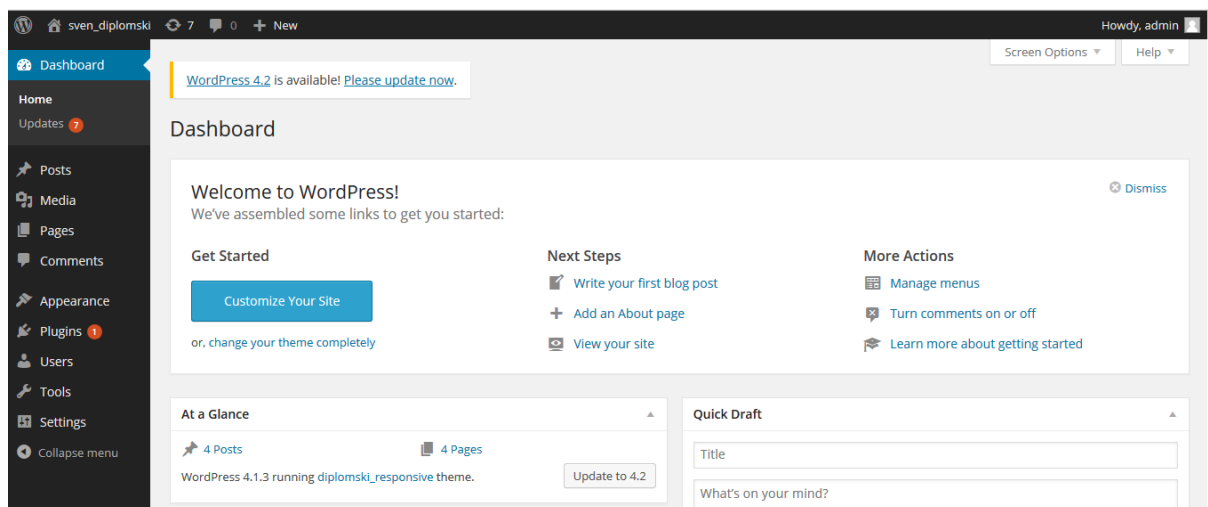
2013. godine više od 240 milijuna web stranica i 2.1 milijun web servera bilo je podržano od strane PHP programskog jezika.

## 2.5. Wordpress

Wordpress je jedno od vodećih CMS rješenja današnjice. Razlozi njegove popularnosti su jednostavnost implementacije i instalacije, velika razina prilagodljivosti vlastitim potrebama te iznimno velika zajednica *developer*a koji neovisno razvijaju teme i dodatke. [7]

Upravo zajednica korisnika, koje je jedna od najvećih prednosti, je također i uzrok nekih od glavnih nedostataka. Svaka razvijena tema odobrena od strane *Wordpress foundation*a nije na istoj razini s današnjim standardima profesionalnog razvoja web stranica.

Sve što je potrebno za korištenje Wordpress CMS-a je instalirati posljednju inačicu na već odabrani hosting. Nakon toga moguće je pristupiti administrativnom sučelju stranice na način da se iza poveznice dodaje ekstenzija „*/wp-admin*“ što u ovom projektu izgleda „*sven.com.hr/wp-admin*“ (slika 3) te se otvara u jednom od internet pretraživača. Unutar administrativnog sučelja stranice izvršava se kontrola svog sadržaja koji se vidi na stranici. Moguće je odabrati gotovu temu, dodatke i pisati blog članke na gotovim formama.



Slika 3. Izgled početne stranice WP *admin* sučelja

Kao što je već i prije spomenuto, glavna prednost *WordPress* sustava za upravljanje sadržajem je ta da je razvijen za širi krug korisnika, što dovodi do toga da za kreiranje i objavu web-a putem *Wordpressa* nije potrebno puno predznanja tehnologija kao što su HTML, CSS, *JavaScript* ili PHP. Kao posljedica toga, na internetu nalazimo mnoštvo stranica koje su izrađene nekvalitetno, sporo se učitavaju jer su pune sadržaja koji se učitava odjednom kad i stranica i na kraju krajeva prikazuju se gotovo nečitke na manjim uređaju kao što su smartphone ili tablet.

U ovom poglavlju rada analizirat će se dva problema Wordpress tema: problem prilagodbe uređajima s različitim veličinama ekrana te problem brzine učitavanja. Nakon analize metodama dedukcije i istraživanja ponuditi će se moguća rješenja čija će učinkovitost biti testirana unutar eksperimentalnog dijela rada.

Mnoge stranice danas su građene na blog platformama kao što je Wordpress jer je sučelje razvijeno da i oni koji nisu tehnološki potkovani postanu profesionalni u kreiranju članaka i stranica.[8]

### 3. Problem prilagodbe uređajima s različitim veličinama ekrana

Sredinom 2007. godine tvrtka Apple je pustila u prodaju svoj prvi pametni telefon iPhone. Slijedećih nekoliko godina tržište je postalo preplavljeno pametnim telefonima različitih proizvođača, a nedugo potom i tablet računalima. Pojavom velikog broja uređaja sa različitim veličinama ekrana nastao je problem pregleda web stranica na svakom od tih uređaja.[9] S obzirom da su stranice prvobitno bile namijenjene za pregled na desktop računima, pregled tih istih stranica na manjim ekranima bio je nepregledan i trajao duže. Iz tog razloga nastaje potreba za prilagođavanjem stranica na više veličina ekrana, a standardi su obično tri veličine:

- Desktop (1680 x 1050)
- Tablet (1024 x 768)
- Smartphone (480 x 320)

Početno rješenje je bila izrada mobilnih stranica. Stranice su u sebi imale skriptu, izvedivu unutar pretraživača, koja je prepoznavala veličinu ekrana te, po potrebi, preusmjeravala korisnika na mobilnu stranicu. S vremenom to rješenje se pokazalo neučinkovito iz više razloga.

Prvo prilikom učitavanja elemenata i tipova datoteka kompleksnijih stranica *JavaScript* se posljednji učitava i izvodi.[4] Direktna posljedica toga je da se prije preusmjeravanja prvo učitala cijela stranice zatim je korisnik preusmjeren na mobilnu stranicu koja se ponovo učitava. Takav postupak je bio neučinkovit ne samo zbog dvostruko većeg vremena učitavanja stranica nego i zbog povećanog prometa. Drugi razlog neučinkovitosti sustava mobilnih stranica je raspon veličine ekrana mobilnih uređaja. Veličina ekrana mobilnih uređaja može varirati od 3 do 10 inča. U tom rasponu gotovo je dizajnerski nemoguće ili vrlo ograničavajuće napraviti pregledan i estetski zadovoljavajući raspored elemenata. [1]

U praktičnom radu optimizirana je Wordpress tema „*wp-theme-responsive*“ po principu *responzivnog* dizajna za tri gore navedene veličine (desktop, tablet i smartphone).

### 3.1. Responzivna tema

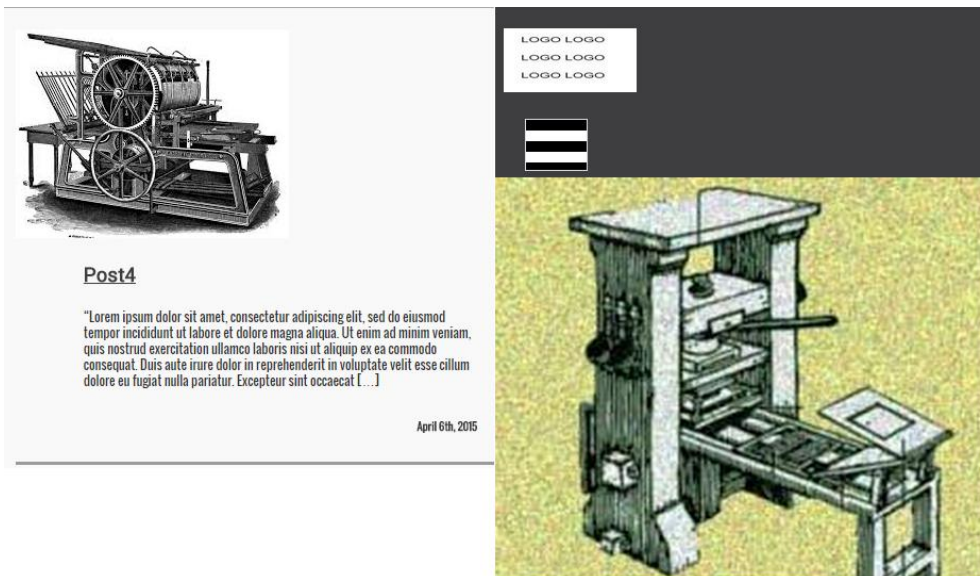
Responzivna tema „*wp-theme-responsive*“ sastoji se od identičnih elemenata kao i kontrolna tema:

- *Header*
- *Slider*
- Lista članaka
- Kalendar i popis komentara s desne strane
- *Footer*

Za razliku od kontrolne teme, responzivna tema prilagođena je za ekrane 1920px i manje (slika 4), što znači da je isti sadržaj moguće vidjeti na mobilnom uređaju širine 480px (slika 5) te na HD ekranu širine 1920px. Na manjim ekranima poput mobilnih, osim što se širina prilagođava odnosno smanjuje, responzivnosti doprinosi i promjena rasporeda sadržaja na stranici. To znači da ukoliko na velikoj rezoluciji imamo sliku pa pored slike tekst članka, smanjivanjem će u jednom trenutku doći do prevelikog lomljenja teksta što će donijeti još više nepreglednosti. U tom slučaju potrebno je preraspodijeliti sadržaj na način da se na određenoj veličina ekrana tekst članka prelama ispod slike te na taj način stranica zadržava preglednost i konzistentnost. [9]



Slika 4. Pregled stranice na ekranu širine 1366px



Slika 5. Pregled stranice na ekranu širine 480px

U responzivnoj verziji teme širina se definira na drugačiji način od kontrolne teme (slika 6). Za razliku od kontrolne teme, u responzivnoj se definira širina *headera* u postocima (100%) te se na taj način na svakoj veličini ekrana prikazuje cjelokupna širina. Unutar *header-a* nalazi se prostor „*container*“ koji ima zadanu maksimalnu moguću širinu u *pixelima*. Unutar koda to izgleda kako je prikazano na slici 6.

```

50 header {
51     width:100%;
52     min-height: 160px;
53     background-color:#3e3e3f !important;
54 }
55
56 .kontejner {
57     max-width:1200px;
58     position:relative;
59     margin-left:auto;
60     margin-right:auto;
61     margin-bottom:20px;
62     overflow:hidden;
63 }
64
65 .kontejner > img {
66     display:block;
67     padding-top:20px;
68     padding-left:1.666666666666667%;
69     position:static;
70     width:auto;
71     height:auto;
72 }
73

```

Slika 6. Primjer koda responzivne stranice



*Header* je definiran na 100% kako bi se na svakom uređaju obavezno prikazala cijela širina dok je *container* unutar *headera* definiran s „max-width:1200px“ što znači da će i na najvećim ekranima širina biti 1200px, ali će se na svim ekranima manje širine od 1200px širina „kontejnera“ smanjiti proporcionalno s visinom. I jedan i drugi od ovih načina definiranja su ispravni kod izrade *responzivnog* web-a. Razlika je u tome o kakvom se sadržaju u pojedinom „box-u“ radi, odnosno je li poželjno da se sadržaj uvijek prilagodi veličini ekrana ili ukoliko se primjerice radi o slici (*slideru*) je potrebno definirati maksimalnu moguću širinu kako bi na svim većim ekranima izbjegli „rastezanje“ i *pikseliziranje* slike, fotografije.

Za kreiranje *responzivnog* web-a koristi se također *media query*. Definiranjem *media queryja* unutar CSS koda zapravo kreiramo određeno pravilo kako se pojedini sadržaj mora prikazivati na različitim vrstama ekrana. *Media query* se definira na način da se zadaje primjerice maksimalna širina za koju se žele primijeniti određene postavke, a unutar vitičastih zagrada se pozivaju klase unutar kojih se definiraju pojedini atributi.

Za veličine ekrana do 670px kreiran je *media query* koji klasičnu navigaciju na stranici pretvara u vertikalnu navigaciju koja se otvara klikom na ikonu. Primjer ovakvog *media queryja* je prikazan na slici 7.

```
@media screen and (max-width: 670px) {
  #opener {
    display: inline-block;
    width: 25px;
    height: 25px;
    background: repeating-linear-gradient(
      90deg, #000, #000 10px, #fff 10px, #fff 20px);
    margin-left: 1.666667%;
    margin-top: 25px;
  }
}
```

Slika 7. *Media query*

## 4. Problem brzine učitavanja

Brzina učitavanja bilo kojeg tipa internet stranice iznimno je važna zbog posjećenosti kao i zbog optimizacije i rangiranja na tražilicama. Stranice sa povećanom brzinom učitavanja češće su posjećenije i donose više klijenata. Također, Google je u svoj algoritam rangiranja uključio i brzinu učitavanja stranice što utječe na SEO (*Search Engine Optimization*), odnosno optimizaciju web stranice za tražilice. To znači da ukoliko se stranica sporo učitava, osim što posjećenost pada, gubi se i mogućnost dobre pozicije na tražilicama što će u konačnici opet rezultirati sa slabom posjećenosti. [10]

Postoji nekoliko metoda ubrzavanja učitavanja web stranica izgrađenih na Wordpress sustavu. Neki od njih su odabir teme koja nije prebogata medijskim sadržajem, korištenje *Cache* dodatka te raznih dodataka za postepeno učitavanje sadržaja kao što je primjerice „*Lazy Load*“ kojim se sve slike na stranici učitavaju tek onda kada su u fokusu na ekranu. Također, jedna od metoda koja je korištena u ovom radu je metoda minimiziranja izvornih datoteka.

Svaka Wordpress tema sastoji se od brojnih PHP, CSS i *Javascript* datoteka. Kako bi bar jednim dijelom riješili problem brzine učitavanja stranice potrebno je pobrinuti se da kod unutar svake od datoteka bude što čišći. Pisanjem i formatiranjem koda da bude pregledan ljudskom oku, dobiva se velik broj praznog prostora i komentara koji stvaraju preglednost i čitkost ljudskom oku, a računalu nisu potrebni kako bi se pokrenula aplikacija. Kao rješenje ovog problema koristi se minimiziranje datoteka (PHP, CSS, *Javascript*).

## 4.1. Minimizirana tema

Minimiziranje datoteka u osnovi znači komprimiranje datoteke koda na način da se brišu svi nepotrebni znakovi koje računalo ne čita kod izvršavanja. Ovim procesom unutar koda se brišu:

- Razmaci
- Novi redovi (cijeli kod se prebacuje u jedan red)
- Komentari
- Granice blokova koda

Kad govorimo o minimiziranju datoteka, potrebno je spomenuti da se ova metoda može izvesti ručno i uz pomoć softvera. Neki od poznatijih softvera:

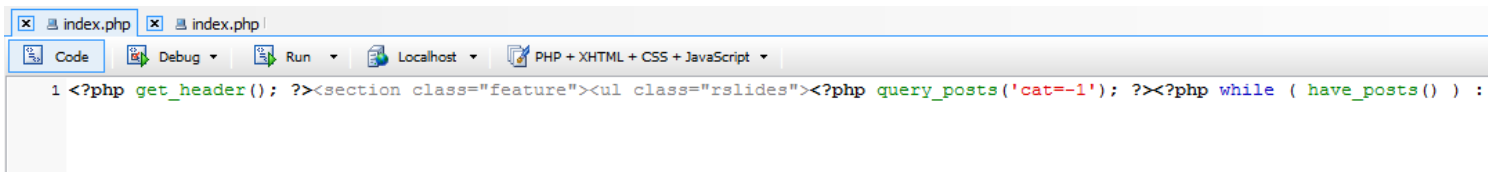
- *Closure Linter*
- *Gzip*
- *CSS Minifier*
- *JavaScript Minify Tool*
- *Smaller*[10]

Jedan od poznatijih softvera je i *Gzip* koji smanjuje kod za otprilike 70% i podržan je na oko 90% internet pretraživača danas. Isto je moguće napraviti i ručno, odnosno brisanjem svih komentara i razmaka kao što je prikazano na slici 9 (osobito je bitno paziti da se ne brišu razmaci između vrijednosti parametara unutar CSS koda jer su dio sintakse) te sažimanjem koda u jedan red.

Osim navedenog, minimiziranje unutar Wordpress okruženja moguće je izvesti i putem „*plugin-ova*“ (dodataka) koji se instaliraju direktno na temu unutar Wordpress administrativnog sučelja. Jedan od primjera je WP *Super Minify* koji radi na vrlo jednostavnom principu odabira željene datoteke za kompresiju.

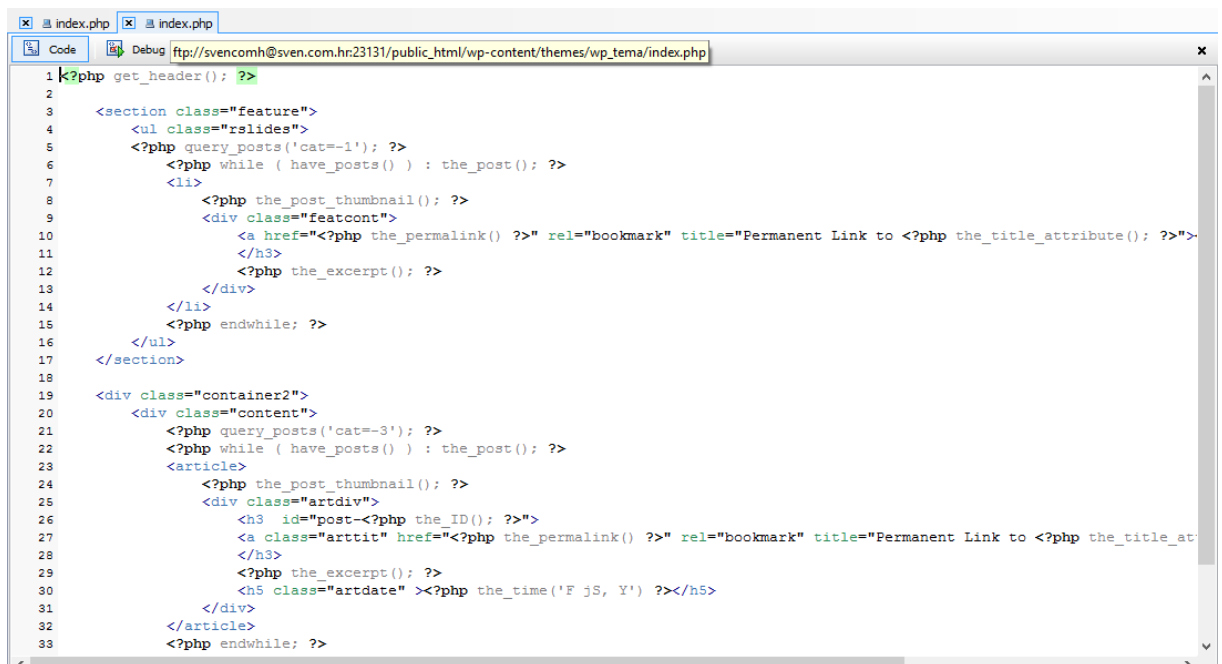
Na kraju, potrebno je spomenuti i neke nedostatke i opasnosti ovakvog načina optimiziranja stranice. Jedan od njih je i taj da je vrlo jednostavno deaktivirati dio

koda ukoliko se primjerice izbriše razmak koji je potreban u sintaksi. S druge strane, kod koji se dobije minimiziranjem vrlo je teško čitati (slika 8).



```
1 <?php get_header(); ?><section class="feature"><ul class="rslides"><?php query_posts('cat=-1'); ?><?php while ( have_posts() ) :
```

Slika 8. Primjer minimiziranog koda



```
1 <?php get_header(); ?>
2
3 <section class="feature">
4 <ul class="rslides">
5 <?php query_posts('cat=-1'); ?>
6 <?php while ( have_posts() ) : the_post(); ?>
7 <li>
8 <?php the_post_thumbnail(); ?>
9 <div class="featcont">
10 <a href="<?php the_permalink() ?>" rel="bookmark" title="Permanent Link to <?php the_title_attribute(); ?>">
11 </h3>
12 <?php the_excerpt(); ?>
13 </div>
14 </li>
15 <?php endwhile; ?>
16 </ul>
17 </section>
18
19 <div class="container2">
20 <div class="content">
21 <?php query_posts('cat=-3'); ?>
22 <?php while ( have_posts() ) : the_post(); ?>
23 <article>
24 <?php the_post_thumbnail(); ?>
25 <div class="artdiv">
26 <h3 id="post-<?php the_ID(); ?>">
27 <a class="arttit" href="<?php the_permalink() ?>" rel="bookmark" title="Permanent Link to <?php the_title_at
28 </h3>
29 <?php the_excerpt(); ?>
30 <h5 class="artdate" ><?php the_time('F jS, Y') ?></h5>
31 </div>
32 </article>
33 <?php endwhile; ?>
```

Slika 9. Primjer regularnog koda (prije minimiziranja)

## 5. Eksperimentalni dio rada

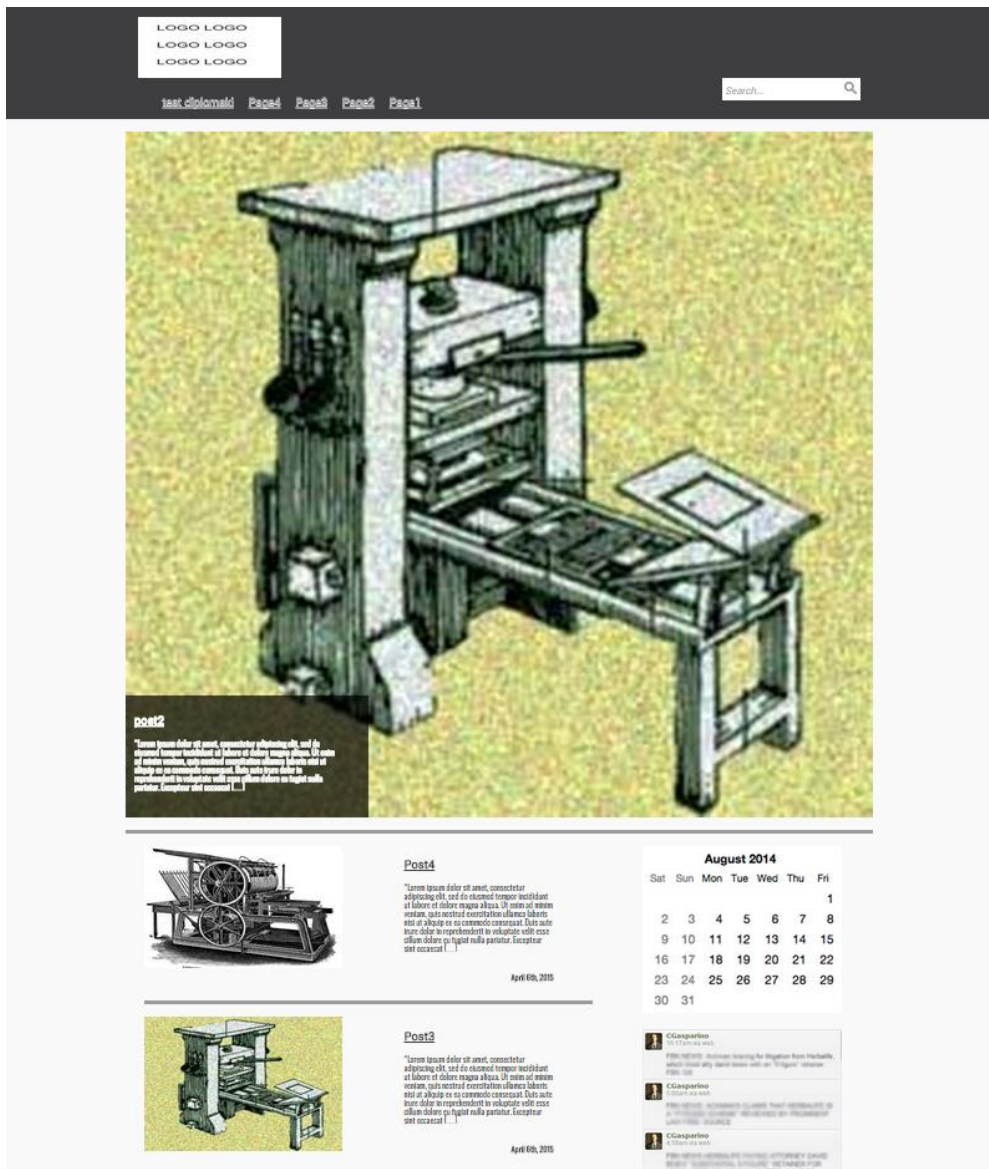
Kao prijedlozi rješenja analiziranih problema navedeni su:

- Prilagodba stranice uređajima s različitim veličinama ekrana
- Smanjenje brzine učitavanja minimiziranjem izvornih datoteka

Za svaki od ovih prijedloga napravljena su mjerenja koja pokazuju koliko je određeno rješenje učinkovito, odnosno dobiva li se na vremenu učitavanja i koliko je akcija na ekranu potrebno da bi se postigla određena radnja. Za sva mjerenja kontrolne teme i obje optimizirane teme korišten je alat *Google Developers*.

## 5.1. Kontrolna tema

Kontrolna tema je tema čiji rezultati će se koristiti kao kontrolni uzorak za provjeru učinkovitosti predloženih rješenja.(slika 10)



LOGO LOGO  
LOGO LOGO  
LOGO LOGO

test.diplomski Page2 Page3 Page4 Page1

Search...

**Post2**

"Larum ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat..."

**Post4**

"Larum ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat..."

April 09, 2015

**Post3**

"Larum ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat..."

April 09, 2015

**August 2014**

Sat	Sun	Mon	Tue	Wed	Thu	Fri
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

**CCZero**  
© 2014 by CCZero

**CCZero**  
© 2014 by CCZero

**CCZero**  
© 2014 by CCZero

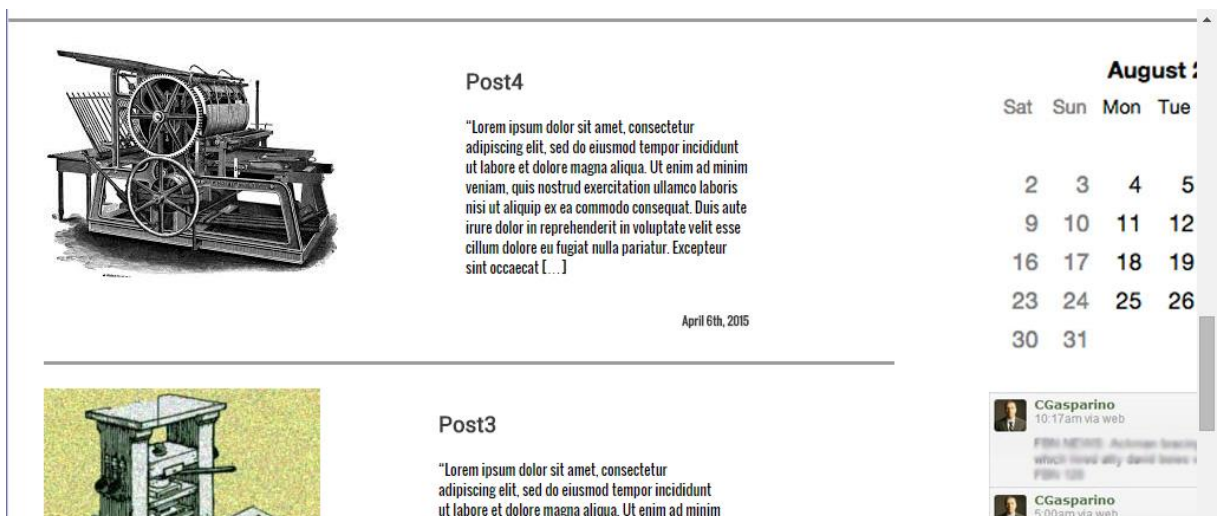
**CCZero**  
© 2014 by CCZero

Slika 10. Kontrolna tema

Kontrolna tema „wp-theme“ sastoji se od sljedećih elemenata:

- *Header* s logotipom, navigacijom i pretraživačem
- *Slider* unutar kojeg se slike mijenjaju vremenski, svaka slika je u originalnoj veličini, no unutar svakog *slide-a* nalazi se apsolutno pozicioniran članak
- Lista članaka
- Kalendar i popis komentara s desne strane
- *Footer*

Tema je prilagođena samo za rezolucije širine 1920px dok *container* unutar zaglavlja sadrži 1200px (slika 11). Dakle, na svim manjim ekranima vidi se samo dio cjelokupne stranice, iako je se učitava cijela, što automatski donosi neke nedostatke. Učitavanje velikog sadržaja traje duže, a ukoliko se učitava sadržaj koji se ne vidi u prirodnoj veličini ili se ne vidi uopće potrebno je ostatak ili ne učitati uopće ili prilagoditi veličini svakog ekrana. Nadalje, jedan od problema je također i bespotrebno trošenje internet prometa (MB) ukoliko se učitava „višak“ sadržaja.



Slika 11. Stranica na ekranu širine 1024px (tablet)

Ovakva neprilagođenost stranice za sve veličine ekrana osim 1920px postiže se podešavanjem fiksne širine elemenata unutar CSS koda. Dakle, potrebno je unutar CSS-a dohvatiti željeni element ili skupinu elemenata i odrediti im širinu na točno određen broj pixela (slika 12). U ovom slučaju je to „*width: 1200px*“ za klase „*container*“, „*container2*“ te „*feature*“ ili „*width: 1920px*“ za *header*.

U prijevodu, neovisno o veličini ekrana na kojem se stranica učitava, širina zaglavlja (prostor u kojem se nalazi navigacija i logotip) uvijek je 1920px, dok je širina „kontejnera“ uvijek 1200px (slika 13).

```
50 header {
51     width:1920px;
52     min-height: 160px;
53     background-color:#3e3e3f;
54 }
55
56 .kontejner {
57     width:1200px;
58     position:relative;
59     margin-left:auto;
60     margin-right:auto;
61     margin-bottom:20px;
62     overflow:hidden;
63 }
```

Slika 12. CSS kod – fiksna širina



```

120 .feature {
121     width:1200px;
122     position:relative;
123     margin-left:auto;
124     margin-right:auto;
125 }
154 .container2 {
155     width:1200px;
156     position:relative;
157     margin-left:auto;
158     margin-right:auto;
159     margin-top:20px;
160     border-top-style: solid;
161     border-top-color: #9b9b9d;
162     overflow:hidden;
163 }
164
165 .content{
166     margin-left:30px;
167     width:720px;
168     position:relative;
169     float:left;

```

Slika 13. CSS kod – fiksna širina unutar tijela stranice

Ukoliko se stranica učitava na mobilnom uređaju (480px širine), na ekranu se vidi otprilike polovica cjelokupnog sadržaja dok je za pregled ostatka sadržaja potrebno „*scrollati*“ u desno. Dakle, osim što je sadržaj prepolovljen i na prvi pogled poprilično nepregledan, na mobilnoj verziji (ili svim ekranima manjim od definiranog) učitava se pozamašna količina sadržaja koji se zapravo ne vidi ukoliko se ne koristi „*scroll*“ ili pomoću *zoom-a* smanjuje veličina stranice.

Također sam kod kontrolne teme pisan je uredno i čitko te u skladu sa standardima pisanja koda. Iako je takav način zapisa poželjan te čak i obavezan prilikom razvoja pruža lošije performanse prilikom downloada stranice i interpretiranja iste od strane web pretraživača.

## 6. Rezultati i diskusija

U prvom mjerenju korištena je optimizirana tema „wp\_tema\_responsive“ i kontrolna „wp\_tema“. Mjeren je broj akcija na ekranu sa zadatkom ulaska u kategoriju i odabira određenog članka te pročitati članak. Iako samo čitanje članka nije povezano snalaženjem na stranici, postavljeno je kao dio zadatka kako bi se povećala praktična relevantnost rezultata s obzirom na čitkost sadržaja

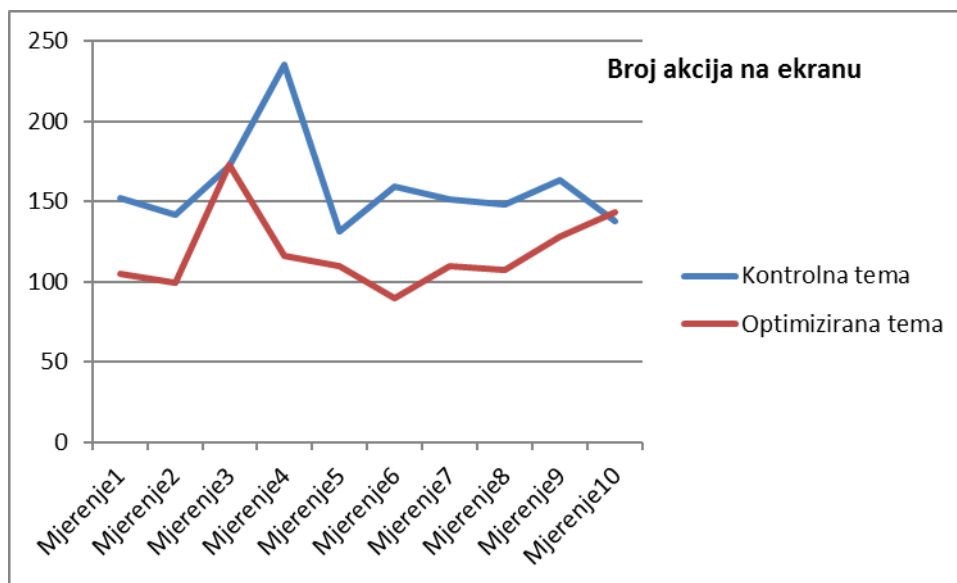
Mjerenje akcija po ekranu sa zadatkom ulaska u kategoriju i odabira određenog članka te čitanja teksta	
Prilagođena tema	
broj akcija	napomena
	5
	5
	6 veći broj od prosjeka zbog polaganijeg scrollanja
	4
	6 veći broj od prosjeka zbog polaganijeg scrollanja
	5
	8 povećan broj akcija na ekranu jer je korisnik pokušao povećati ekran
	6 veći broj od prosjeka zbog polaganijeg scrollanja
	5
	5
prosjek	5.5
kontrolna tema	
broj akcija	napomena
	10 između svakog odabira na stranici korisnik je morao koristiti "pinch to zoom" povećanje ekrana
	6
	9 korisnik je kliknuo na krivi link
	9 korisnik je kliknuo na dva linka istovremeno te je morao ponovno odabrati željeni link u povećanom
	11 između svakog odabira na stranici korisnik je morao koristiti "pinch to zoom" povećanje ekrana
	8 korisnik je morao morao koristiti "pinch to zoom" povećanje ekrana za linkove manjih dimenzija
	8 korisnik je morao morao koristiti "pinch to zoom" povećanje ekrana za linkove manjih dimenzija
	10 između svakog odabira na stranici korisnik je morao koristiti "pinch to zoom" povećanje ekrana
	11 između svakog odabira na stranici korisnik je morao koristiti "pinch to zoom" povećanje ekrana
prosjek	9.11111111

Slika 14. Mjerenje akcija na ekranu respozivne i kontrolne teme

Iz analize mjerenja vidljivo je da je za isti zadatak na respozivnoj i kontrolnoj temi potreban različit broj akcija. Na manjim ekranima s kontrolnom temom prosječni broj akcija za odabir kategorije i otvaranje željenog članka iz kategorije iznosi 9.1, dok je na optimiziranoj temi (prilagođenoj za sve veličine ekrana) prosječan broj akcija istog zadatka 5.5 (slika 14). Na prilagođenoj temi u nekoliko mjerenja vidi se

broj akcija veći od prosjeka zbog polaganijeg pomicanja stranice ili povećavanja ekrana što je slučaj na jednom od mjerenja.

Na kontrolnoj temi, s druge strane, vidljiv je povećan broj akcija na svakom mjerenju (slika 15). Na mjerenjima s najvećim brojem akcija (11) korisnik je između svake akcije morao koristiti povećavanje ekrana metodom „*pinch to zoom*“ na malom ekranu iz razloga što tema nije prilagođena manjim uređajima i sav sadržaj koji je pregledan na velikom ekranu zbijen je na nekoliko puta manji ekran. Mjerenja koja pokazuju nešto manji broj akcija (8) dobila su ovakve rezultate jer je korisnik koristio povećanje ekrana na poveznicama manjih dimenzija koje nije moguće odabrati na manjem ekranu. Također, broj akcija se u nekoliko navrata povećao iz razloga što je korisnik kliknuo na krivu poveznicu upravo zbog manjih dimenzija i nepreglednosti na malom ekranu, isto kao što je kliknuo na dvije poveznice odjednom te se ekran automatski povećao za ponovni odabir.



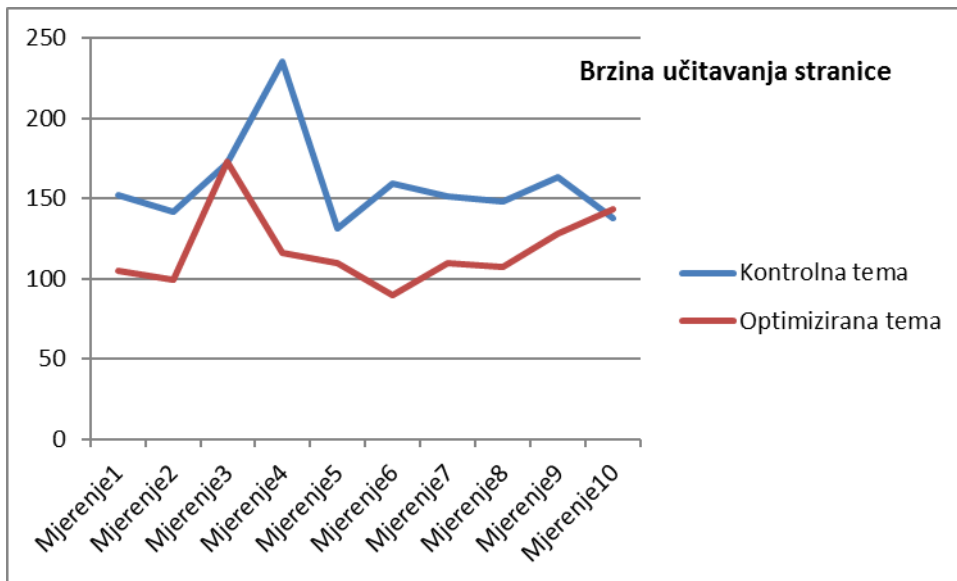
Slika 15. Prikaz broja akcija na ekranu unutar 10 mjerenja

Rješenja problema brzine učitavanja mjereno je na kontrolnoj temi „wp\_tema“ i optimiziranoj temi „wp\_tema\_download“. U mjeranju je prikazana brzina učitavanja jedne i druge teme u nekoliko navrata te prosječno vrijeme učitavanja u milisekundama (slika 16).

Vrijeme downloada/ms		
kontrolna tema	optimizirana tema	
	152	105
	142	99
	172	173
	235	116
	131	110
	159	90
	151	110
	148	107
	163	128
	138	143
<b>Prosjek</b>		
	159.1	118.1
<b>Median</b>		
	151.5	110

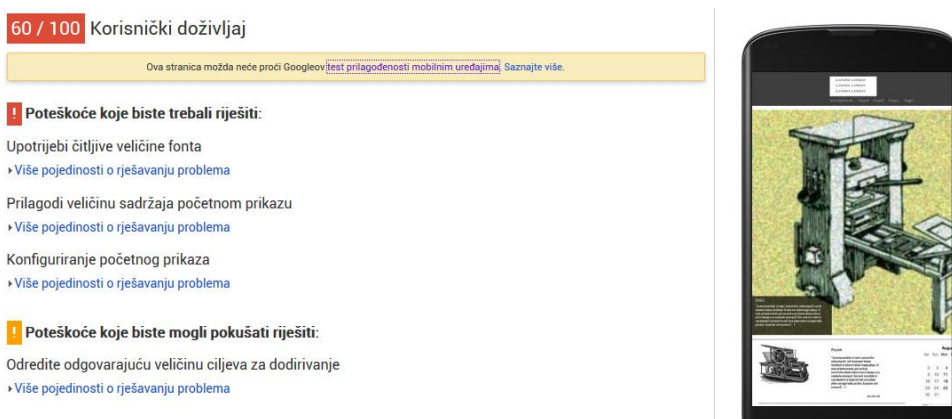
Slika 16. Mjerenje brzine učitavanja

Razlika brzine učitavanja kontrolne i optimizirane teme vidljiva je u gotovo svakom od mjerenja. U prosjeku kontrolnoj temi potrebno je 159.1 milisekundi za učitavanje dok se optimizirana tema učita za 118.1 milisekundi. U 2/10 mjerenja optimiziranoj temi bilo je potrebno više vremena za učitavanje od kontrolne teme (slika 17). Razlog ovakvog rezultata može biti i prijenos podataka, odnosno slabija internetska veza u tom trenutku, zbog čega je u ovakvom tipu mjerenja potrebno uzeti više scenarija kako bi se dobio što vjerodostojniji podatak.



Slika 17. Prikaz brzine učitavanja stranice kroz 10 mjerenja

Koristeći *Google Developer PageSpeed Insights* alat, prikazane su ocjene Google-a za korisnički doživljaj koristeći kontrolnu i optimizirane teme. Kontrolna tema očekivano je prikazala najlošije rezultate. Ocjena za korisnički doživljaj kontrolne teme je 60/100 što je stavlja u najlošiji (crveni) rang prema Google-u (slika 18).



**60 / 100** Korisnički doživljaj

Ova stranica možda neće proći Googleov [test prilagodivosti mobilnim uređajima](#). Saznajte više.

**! Poteškoće koje biste trebali riješiti:**

- Upotrijebi čitljive veličine fonta
  - » [Više pojedinosti o rješavanju problema](#)
- Prilagodi veličinu sadržaja početnom prikazu
  - » [Više pojedinosti o rješavanju problema](#)
- Konfiguriranje početnog prikaza
  - » [Više pojedinosti o rješavanju problema](#)

**! Poteškoće koje biste mogli pokušati riješiti:**

- Odredite odgovarajuću veličinu ciljeva za dodirivanje
  - » [Više pojedinosti o rješavanju problema](#)

The image also shows a mobile phone displaying a technical diagram of a machine, which is the content of the page being analyzed.

Slika 18. Ocjena kontrolne teme

Također, alat je prepoznao da stranica nije prilagođena za sve ekrane (slika 19).

**Stranica nije prilagođena mobilnim uređajima**

**Čini se da stranica nije prilagođena mobilnim uređajima**

- ✗ Tekst je presitan za čitanje
- ✗ Sadržaj je širi od zaslona
- ✗ Nije postavljen vidljivi dio za mobilne uređaje
- ✗ Veze su preblizu jedna drugoj

Pojedinosti o tome na koje dijelove stranice utječu poteškoće u vezi s upotrebljivošću potražite u [Pagespeed Insightsu](#).

**Kako Googlebot vidi ovu stranicu**



Slika 19. Neprilagođena tema

Navedene su značajke ne optimizirane teme koje je potrebno izmijeniti kako bi korisnički doživljaj bio bolji. Može se primijetiti da je svaka od ovih značajki prilikom mjera akcija na ekranu uzela nekoliko više akcija koje je bilo potrebno izvesti za isti zadatak. Tekst koji je presitan za čitanje doveo je do akcije povećanja ekrana na svakom koraku, sadržaj koji je širi od ekrana doveo je do pomicanja ekrana u lijevo i desno kako bi se dobio pregled čitave stranice, a poveznice koje su preblizu jedna drugoj dovele su do slučajno krivog odabira ili u jednom slučaju do odabira dvije poveznice odjednom (slika 2.3).

Otklanjanje navedenih poteškoća izvedeno je u optimiziranoj responzivnoj temi „wp\_tema\_responsive“. Rezultati analize prikazuju ocjenu 100/100 za korisnički doživljaj stranice (slika 20).

100 / 100 Korisnički doživljaj

✓ Izvrsno! Poteškoće nisu pronađene.

#### Izbjegavanje dodataka

Čini se da vaša stranica ne upotrebljava dodatke koji bi spriječili mogućnost upotrebe sadržaja na mnogim platformama. Saznajte više o važnosti [izbjegavanja dodataka](#).

#### Konfiguriranje početnog prikaza

Vaša stranica ima naveden početni prikaz koji odgovara veličini uređaja, što stranici omogućuje ispravno generiranje na svim uređajima. Saznajte više o [konfiguriranju početnih prikaza](#).

#### Odredite odgovarajuću veličinu ciljeva za dodirivanje

Sve veze i svi gumbi vaše web-stranice dovoljno su veliki da ih korisnik može lako dodirnuti na dodirnom zaslonu. Saznajte više o [određivanju odgovarajuće veličine ciljeva za dodirivanje](#).

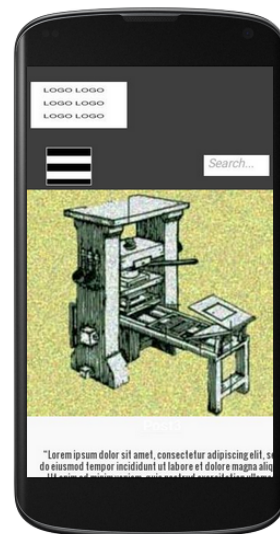
#### Prilagodi veličinu sadržaja početnom prikazu

Sadržaj vaše stranice odgovara veličini početnog prikaza. Saznajte više o [prilagodavanju veličine sadržaja početnom prikazu](#).

#### Upotrijebi čitljive veličine fonta

Tekst na vašoj stranici čitljiv je. Saznajte više o [upotrebi čitljivih veličina fonta](#).

\*Rezultati se predmemoriraju na 30 s. Ako ste unijeli promjene na svojoj stranici, pričekajte 30 s prije ponavljanja testa.



Slika 20. Responzivna tema

Tema koja je optimizirana minimiziranjem datoteka pokazala je iste rezultate kao i kontrolna tema što se tiče korisničkog doživljaja (60/100). Iako se brzina učitavanja smanjila, korisnički doživljaj ovakve stranice ostao je nizak iz razloga što ovakav način optimizacije ne podržava responzivnosti teme. CSS datoteke koje su minimizirane ne prepoznaju značajke responzivnosti kao što su širina izražena u postotku ili parametri klase kao što su „*min-width*“ i „*min-height*“ (slika 21).

```
Code Debug Run Localhost CSS
1 /*
2 Theme Name: diplomski-download
3 Theme URI: http://www.sven.com.hr
4 Author: Sven Jakabfi
5 Author URI: http://www.sven.com.hr
6 Description: Tema za diplomski
7 License: GNU General Public License v2 or later
8 License URI: http://www.gnu.org/licenses/gpl-2.0.html
9 Tags: black, brown, orange, tan, white, yellow, light, one-column, two-columns, right-sidebar, flexible-width, custom-header, cu
10 Text Domain: diplomski
11 */
12 .rslides{position:relative;list-style:none;overflow:hidden;width:100%;padding:0;margin:0}.rslides li{-webkit-backface-visibility
13
14
15
16
```

Slika 21. „wp\_tema\_download“ – minimizirana CSS datoteka sa značajkom responzivnosti (width:100%)



## 7. Zaključak

Obzirom na rezultate oba mjerenja koja su provedena na optimiziranim temama vidljivo je da se pomoću responzivnosti dobiva znatno manji broj akcija na ekranu za određeni zadatak što dovodi do lakšeg i bržeg korištenja stranice. Rezultati pokazuju gotovo dvostruko veću učinkovitost responzivne teme u odnosu na kontrolnu što pokazuje i ocjena *Google Insights* alata za korisnički doživljaj koja iznosi maksimalnu vrijednost 100/100.

Tema „*wp\_tema\_download*“ kojoj je minimiziranjem datoteka smanjeno vrijeme učitavanja pokazuje nekoliko mana. Prvenstveno jedna od mana je nedostatak responzivnosti zbog kojeg unatoč bržem učitavanju stranice postoji problem povećanog broja akcija na ekranu prilikom izvođenja nekog zadatka. Povećan broj akcija produljuje vrijeme „korištenja“ same stranice pa je stoga upitan učinak povećanja brzine učitavanja putem minimiziranih datoteka.

Primjenjivanjem svakog od predloženih rješenja vidljivo je poboljšanje performansi web stranice. Poboljšanja utječu na korisnikov doživljaj (UX – *user experience*) stranice kao i na poziciju stranice na pretraživačima. Korisnički doživljaj i rangiranje na pretraživačima su, uz sadržaj, najvažniji čimbenici koji odlučuju o posječenosti stranice.

U današnje doba sa sve naglijim rastom mobilnih tehnologija može se reći da se Internet polako seli na mobilne uređaje. Prema nekim istraživanjima u svijetu već postoji više pametnih mobilnih uređaja nego desktop računala.

Takvim razvojem tehnologije postaje sve važnije prilagoditi dizajn i prezentaciju sadržaja uređajima koji preuzimaju tržište.

U jednom pogledu prilagodba se mora odvijati na vizualnoj razini. *Dizajneri* moraju na umu imati raspored elemenata koji se lako prilagođava različitim veličinama ekrana. S druge strane *developer* moraju vladati tehnologijama i biti u

mogućnosti pretočiti *dizajnerovu* viziju u funkcionalan softver koji će se s lakoćom prilagođavati svi uređajima.

Također osim vizualne prilagodbe, *developer* koji rade na modernom webu moraju imati na umu i praktičnu stranu. Mobilni Internet je često sporiji i skuplji nego fiksni, stoga se ulaže veliki trud u smanjivanje podatkovne veličine web stranica.

Tehnološkom razvoju se prilagođavaju i velike IT tvrtke poput Google-a, koji je počeo neprilagođene stranice smještati na lošija mjesta od onih koje su optimizirane za sve uređaje.

Stoga pojedincima preostaje držati korak s napretkom tehnologije kroz stalnu edukaciju profesionalno usavršavanje, što je i jedna od uzbudljivijih prednosti rada u informatičkom sektoru.

## 8. Literatura

- [1] Ben Frain: Responsive web design with HTML & CSS, Packt Publishing Ltd, April 2012.
- [2] Ante Marsic: HTML5 URL: <http://imef.me/html-5/> (20.4.2015.)
- [3] Peter Lavin: Object – oriented PHP, Second Edition, No Starch Press, 2006.
- [4] David Flanagan: Javascript, the definitive guide, Fourth edition, O'Reilly & Associates, June 2002.
- [5] Jake Archibald: JavaScript Promises URL:  
<http://www.html5rocks.com/en/tutorials/es6/promises/> (27.4.2015.)
- [6] Peter Lavin: Object – oriented PHP, Second Edition, No Starch Press, 2006.
- [7] Jeff Starr, Chris Coyier :Digging into WordPress(svibanj, 2011.)
- [8] Jessica Neuman Beck: Wordpress Second Edition (listopad, .2011.)
- [9] Ethan Marcotte: On responsive web design URL:  
<http://www.html5rocks.com/en/tutorials/es6/promises/> (27.04.2015.)
- [10] Brenda Barron: How to minify your websites CSS, HTML & JavaScript URL:  
<http://www.elegantthemes.com/blog/tips-tricks/how-to-minify-your-websites-css-html-javascript> (29.04.2015.)

## 9. Popis slika

SLIKA 1.	PRIMJER JEDNOSTAVNOG HTML KODA.....	4
SLIKA 2.	PRIMJER UPORABE VANJSKOG CSS-A.....	6
SLIKA 3.	IZGLED POČETNE STRANICE WP <i>ADMIN</i> SUČELJA .....	13
SLIKA 4.	PREGLED STRANICE NA EKRANU ŠIRINE 1366PX .....	16
SLIKA 5.	PREGLED STRANICE NA EKRANU ŠIRINE 480PX .....	17
SLIKA 6.	PRIMJER KODA RESPONZIVNE STRANICE .....	17
SLIKA 7.	<i>MEDIA QUERY</i> .....	18
SLIKA 8.	PRIMJER MINIMIZIRANOG KODA .....	21
SLIKA 9.	PRIMJER REGULARNOG KODA (PRIJE MINIMIZIRANJA).....	21
SLIKA 10.	KONTROLNA TEMA.....	23
SLIKA 11.	STRANICA NA EKRANU ŠIRINE 1024PX (TABLET).....	24
SLIKA 12.	CSS KOD – FIKSNA ŠIRINA .....	25
SLIKA 13.	CSS KOD – FIKSNA ŠIRINA UNUTAR TIJELA STRANICE .....	26
SLIKA 14.	MJERENJE AKCIJA NA EKRANU RESPONZIVNE I KONTROLNE TEME .....	27
SLIKA 15.	PRIKAZ BROJA AKCIJA NA EKRANU UNUTAR 10 MJERENJA .....	28
SLIKA 16.	MJERENJE BRZINE UČITAVANJA.....	29
SLIKA 17.	PRIKAZ BRZINE UČITAVANJA STRANICE KROZ 10 MJERENJA.....	30
SLIKA 18.	OCJENA KONTROLNE TEME .....	31
SLIKA 19.	NEPRILAGOĐENA TEMA .....	32
SLIKA 20.	RESPONZIVNA TEMA .....	33
SLIKA 21.	„ <i>WP_TEMA_DOWNLOAD</i> “ – MINIMIZIRANA CSS DATOTEKA SA ZNAČAJKOM RESPONZIVNOSTI (WIDTH:100%).....	33