

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matjaž Lovše

# **Odzivno spletno oblikovanje**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matjaž Kukar

Ljubljana, 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\LaTeX$ .*





Št. naloge: 00379 / 2013  
Datum: 1.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MATJAŽ LOVŠE**

Naslov: **ODZIVNO SPLETNO OBLIKOVANJE  
RESPONSIVE WEB DESIGN**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V zadnjih letih se je zelo povečal delež mobilnih odjemalcev spletnih vsebin, ki imajo v primerjavi z namiznimi odjemalci bistveno manjše vidne površine in manjše zmogljivosti procesiranja, uporabljajo pa počasnejše in dražje omrežne povezave. Priprava in oblikovanje različnih spletnih vsebin, prilagojenih za različne odjemalce, sta vedno nezaželeni, včasih pa tudi nesprejemljiva. Zato se je na podlagi sodobnih spletnih standardov pojavil pristop odzivnega spletnega oblikovanja, ki na podlagi prejetih informacij o odjemalcu prikaz spletnih vsebin avtomatsko prilagaja njegovim omejitvam. Kandidat naj podrobno razišče področje odzivnega spletnega oblikovanja in sorodne pristope, ter pregleda tehnologije in orodja, ki se v ta namen uporabljajo. Opredeli naj glavne težave in omejitve odjemalcev, ter predlaga tehnologije in lokacije (odjemalec ali strežnik) za njihovo implementacijo. Pridobljeno znanje naj kandidat predstavi na primeru praktične implementacije.

Mentor:

doc. dr. Matjaž Kukar

Dekan:

prof. dr. Nikolaj Zimic





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matjaž Lovše, z vpisno številko **63080266**, sem avtor diplomskega dela z naslovom:

### *Odzivno spletno oblikovanje*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matjaža Kukarja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. oktobra 2013

Podpis avtorja:



*Najlepša hvala vsem, ki so kakorkoli pripomogli k nastanku diplomskega dela. Še posebej se zahvaljujem mentorju doc. dr. Matjažu Kukarju za strokovno vodenje, usmerjanje in dajanje koristnih nasvetov.*

*Hvala podjetju Pip d.o.o. za možnost uporabe njihovega spletnega strežnika ter podjetju Vini d.o.o. za sodelovanje pri izdelavi odzivne spletne trgovine. Zahvala gre tudi vsem urednikom spletnih strani delo.si, slovenskenovice.si, dolenskilist.si, prevoz.org ter enaa.com, ki so bili pripravljeni za namene te diplomske naloge deliti svoja analitična poročila.*

*Iskreno se zahvaljujem tudi vsem sošolcem in prijateljem, brez katerih bi bil študij nedvomno težji ter staršem, ki so me moralno in finančno podpirali na moji študijski poti. Hvala tudi Mancu za vse spodbudne besede, ki so mi pomagale, ko sem jih najbolj potreboval.*



# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Pregled uporab. naprav na slo. spletnih straneh . . . . .	3
<b>2</b>	<b>Značilnosti in omejitve spleta</b>	<b>5</b>
2.1	Naprave za dostop do spleta . . . . .	5
2.2	Ločljivost zaslona . . . . .	7
2.3	Gostota slikovnih točk . . . . .	8
2.4	Barvna globina . . . . .	10
2.5	Spletni brskalniki . . . . .	11
2.6	Internetna povezava . . . . .	13
<b>3</b>	<b>Uporabljene tehnologije</b>	<b>15</b>
3.1	Označevalni jezik HTML5 . . . . .	15
3.2	Slogovni jezik CSS3 . . . . .	16
3.3	Skriptni jezik JavaScript . . . . .	17
3.4	Programski jezik PHP . . . . .	17
<b>4</b>	<b>Odzivna spletna stran</b>	<b>19</b>
4.1	Predhodne rešitve . . . . .	19
4.2	Odzivno spletno oblikovanje . . . . .	23
4.2.1	Tekoča postavitev elementov . . . . .	24
4.2.2	Prilagodljive slike in druge medijske datoteke . . . . .	26
4.2.3	Medijske poizvedbe znotraj slogovne kode CSS3 . . . . .	27
4.3	Optimizacijske tehnike . . . . .	28

4.3.1	Prilagodljive slike . . . . .	29
4.3.2	Zakasnjeno nalaganje slik . . . . .	30
4.3.3	Predpomnilnik brskalnika . . . . .	33
4.3.4	Stiskanje vsebin z metodo Gzip . . . . .	34
4.3.5	Orodje Modernizr . . . . .	36
4.3.6	Vektorske slike SVG . . . . .	39
4.3.7	Sistem dostave s strežniki CDN . . . . .	40
4.3.8	Vrstni red vključevanja virov . . . . .	41
4.3.9	Hitro odzivanje povezav pri mobilnih napravah . . . . .	42
4.3.10	Zmanjšanje velikosti datotek CSS in JS . . . . .	43
<b>5</b>	<b>Implementacija odzivne spletne trgovine</b>	<b>45</b>
5.1	Delovni tok oblikovanja . . . . .	45
5.1.1	Popis elementov spletne strani . . . . .	46
5.1.2	Oblikovanje besedil . . . . .	47
5.1.3	Struktura tekoče postavitve elementov . . . . .	47
5.1.4	Oblikovanje v spletnem brskalniku . . . . .	49
5.1.5	Oblikovanje pri različnih prelomnih točkah . . . . .	50
5.1.6	Preizkus na različnih napravah in brskalnikih . . . . .	57
5.2	Rezultati testiranja . . . . .	60
<b>6</b>	<b>Sklepne ugotovitve</b>	<b>63</b>
	<b>Literatura</b>	<b>65</b>

# Seznam uporabljenih kratic

<b>BPP</b>	(angl.) Bit per Pixel
<b>CDN</b>	(angl.) Content Delivery Network
<b>CSS</b>	(angl.) Cascading Style Sheets
<b>HTML</b>	(angl.) Hyper Text Markup Language
<b>HTTP</b>	(angl.) Hyper Text Transfer Protocol
<b>JS</b>	(angl.) JavaScript
<b>MIME</b>	(angl.) Multipurpose Internet Mail Extensions
<b>PPI</b>	(angl.) Pixels Per Inch
<b>RESS</b>	(angl.) Responsive Design with Server Side Components
<b>RFC</b>	(angl.) Request For Comments
<b>RGB</b>	(angl.) Red Green Blue color model
<b>RTT</b>	(angl.) Round Trip Time
<b>RWD</b>	(angl.) Responsive Web Design
<b>SEO</b>	(angl.) Search Engine Optimization
<b>SVG</b>	(angl.) Scalable Vector Graphics
<b>TCP</b>	(angl.) Transmission Control Protocol
<b>URL</b>	(angl.) Uniform Resource Locator
<b>WAP</b>	(angl.) Wireless Application Protocol
<b>XML</b>	(angl.) Extensible Markup Language



# Povzetek

Cilj diplomskega dela je podrobno raziskati področje odzivnega spletnega oblikovanja in njegovih optimizacijskih tehnik ter izdelati praktičen primer, ki ponuja optimalno izkušnjo uporabnikom na najrazličnejših napravah. Ob poplavi pametnih telefonov, tabličnih in prenosnih računalnikov se pojavlja vedno več vprašanj o prilagoditvi spletnih strani za posamezne tipe naprav. Po pridobitvi analitičnih poročil nekaterih priljubljenih slovenskih spletnih strani smo uporabljene vrste naprav primerjali s svetovnimi trendi ter ocenili prihodnjo rast mobilnih naprav v Sloveniji. S spoznavanjem sodobnih tehnologij smo se opredelili za pristop k razvoju odzivnih spletnih strani in poudarili razlike od klasičnega oblikovanja. Pri razvoju smo bili pozorni na značilnosti in omejitve spleta ter naprav. Predvsem smo se osredotočili na optimizacije, ki jih lahko izvedemo že na spletnem strežniku, da naprav s slabšo zmogljivostjo ne obremenjujemo po nepotrebnem. Kot primer smo na izdelani odzivni spletni trgovini implementirali in testirali nekaj najučinkovitejših tehnik optimizacij odzivnega spletnega oblikovanja, s katerimi smo celovito izboljšali uporabniško izkušnjo ne glede na uporabljeno napravo in njene omejitve.

## Ključne besede

Odzivno spletno oblikovanje, medijske poizvedbe CSS3, optimizacijski pristopi, RESS.



# Abstract

The aim of the following thesis is to explore the scope of responsive web design with its optimization techniques in details and create a practical example that offers an optimal user experience on various devices. Due to the increased popularity of smart phones, tablets, and laptops, a lot of questions arise on how to adapt websites to a range of different devices. After obtaining analytical reports from popular Slovenian websites with regards to user devices, we compared this data with global trends in order to predict estimations for the future use of mobile devices in Slovenia. By learning modern technologies, we are able to identify the best approach to the development of responsive web pages and highlight the advantage of this method over the classical way of design. During research and development we focused our attention on the characteristics and limitations of web and devices. Our primary study was on server side optimization, in order to reduce any unnecessary burden on devices with lower performance. As an example, we designed a responsive online store and equipped it with some of the most effective optimization techniques of responsive web design, improving the overall user experience regardless of the device and its limitations.

## Keywords

Responsive Web Design, CSS3 Media Queries, optimization approaches, RESS.



# Poglavje 1

## Uvod

Spletni oblikovalci so dolga leta oblikovali spletne strani predvsem za namizne in prenosne računalnike. Skozi razvoj tehnologij in naprav so se pojavljali pomisleki glede prikaza spletnih strani na mobilnih napravah, kjer je bila uporabniška izkušnja med brskanjem po neprilagojenih spletnih straneh zelo neprijetna.

Ob analiziranju leta 2012 opazimo veliko spremembo, saj bila prodaja osebnih računalnikov zelo nenavadna. Prvič po enajstih letih je prodaja osebnih računalnikov v primerjavi s predhodnim letom padla. V letu 2012 je bilo prodanih 348 milijonov osebnih računalnikov, medtem ko leto poprej 352 milijonov, kar pomeni skoraj 1.2 % padec v prodaji [6]. Padec prodaje namiznih računalnikov v prvem četrtletju leta 2013 znaša že 13.9 %, kar predstavlja prodajo zgolj še 76 milijonov tovrstnih računalnikov [37]. Po drugi strani pa je opaziti ogromen porast tabličnih računalnikov, ki so zaradi svoje majhnosti in uporabnosti vse bolj priljubljeni med ljudmi. Tablični računalniki so danes cenovno dostopni, zato njihova prodaja strmo narašča. Do konca letošnjega leta naj bi bilo prodanih preko 227 milijonov tabličnih računalnikov, medtem ko naj bi prodaja pametnih telefonov narastla za 7.3 % na skupno število preko 1 bilijon [29, 30]. Tovrstne mobilne naprave so zmogljive in uporabne, zato je rast pričakovana tudi v prihodnosti.

Spletne strani predstavljajo marsikateremu podjetju velik delež posla in osnovo za predstavitev svojih storitev. Seveda so za mobilne naprave na voljo tudi domorodne (angl. native) aplikacije, vendar je njihova izdelava za majhno podjetje predraga in nesmiselna. Tu imamo v mislih predvsem tovrstne aplika-

cije spletnih mest in ne namenskih aplikacij, ki so gotovo smiselne in uporabne. Izdelava lastnih domorodnih aplikacij mora pokriti vse priljubljene operacijske sisteme, kar na koncu privede do obširnejšega vzdrževanja, težje analitike uporabnikov ter dražjih stroškov. Menimo, da je izdelava lastnih aplikacij primerna, ko imamo veliko množico uporabnikov ter zelo dobro prepoznavnost med ljudmi, kot jo imajo spletne strani *facebook.com*, *twitter.com*, *24ur.com*, *siol.tv* in druge. Za predstavitvene in prodajne spletne strani majhnih podjetij je težko pričakovati, da bodo uporabniki pripravljene uporabljati domorodne aplikacije, saj ima konkurenca svoje storitve in kontakte gotovo lažje dosegljive na spletnih straneh.

Globalne statistike kažejo, da vse več ljudi uporablja mobilne naprave tudi za brskanje po svetovnem spletu. Lastniki spletnih strani in trgovci vedo, da se na njihovih straneh vedno več zadržujejo prav mobilni uporabniki ter da se tem potencialnim kupcem njihova spletna stran ne prikazuje najbolje. Potreben je velik miselni preskok za rešitev težave oblikovanja spletnih strani, da le te nudijo uporabnikom pozitivno izkušnjo med brskanjem ne glede s katero napravo do nje dostopajo. Oblikovalci in razvijalci se sedaj srečujemo s pristopom odzivnega spletnega oblikovanja, s katerim skušamo doseči optimalno delovanje spletne strani na množici različnih naprav.

## 1.1 Pregled uporabljenih naprav na slovenskih spletnih straneh

Stopili smo v stik z lastniki nekaj slovenskih prepoznavnih spletnih strani, ki so nam omogočili vpogled v analitične podatke. S pridobitvijo analitičnih podatkov spletnih strani, ki zaradi svoje vsebine privabljajo široko množico odjemalcev, smo želeli preveriti trenutne deleže uporabnikov, ki za brskanje po spletu uporabljajo mobilne naprave in ugotoviti, ali je tudi pri nas opaziti globalni trend porasta tovrstnih obiskovalcev. Osredotočili smo se na delež mobilnih naprav v letu 2011 ter v tekočem letu 2013, zato smo iz obeh let primerjali enako petmesečno obdobje in ugotovitve vpisali v spodnjo tabelo.

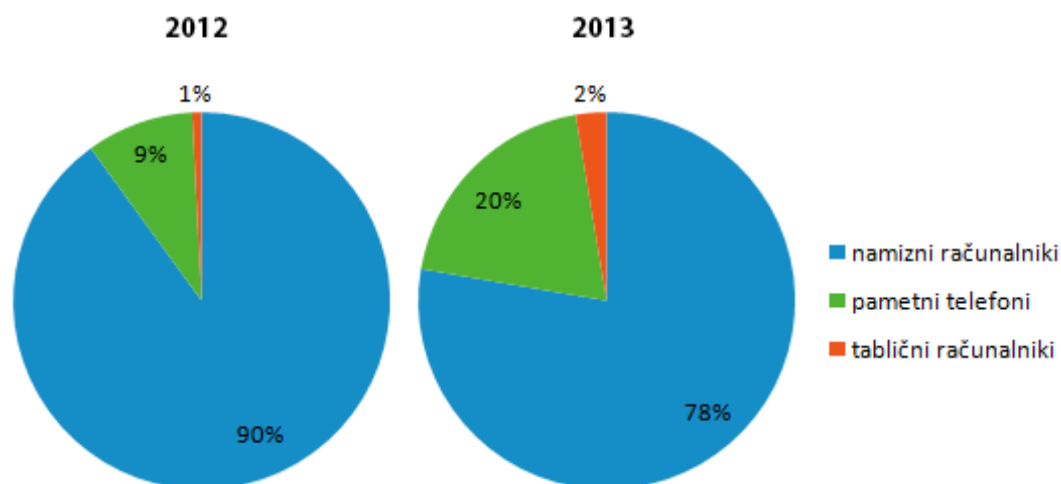
Naslov spletne strani	Delež mobilnih naprav v letu 2011	Delež mobilnih naprav v letu 2013
<i>www.delo.si</i>	1,58 %	12,21 %
<i>www.slovenskenovice.si</i>	0,89 %	9,54 %
<i>www.dolenjskilist.si</i>	0,48 %	7,10 %
<i>www.ena.com</i>	0,78 %	11,04 %

Tabela 1.1: Primerjava deležev uporabnikov, ki so v letu 2011 in 2013 za brskanje po nekaterih priljubljenih slovenskih spletnih straneh uporabljali mobilne naprave.

Glede na globalne spremembe prodaje mobilnih naprav, je porast uporabe tovrstnih naprav za brskanje po spletu opaziti tudi med Slovenci, saj se celotni delež pri zelo priljubljenih spletnih straneh v letošnjem letu giblje nekje okoli 10 % (tabela 1.1). Ta delež mobilnih uporabnikov je za tovrstne spletne strani velik dosežek, saj so na njih vsebine namenjene predvsem daljšemu brskanju oz. branju, obenem pa le te še niso prilagojene za velikosti mobilnih zaslonov. V analitičnem poročilu spletne strani *www.delo.si* smo pri mobilnih napravah opazili kar 90,6% delež podpore zaslonov na dotik, preostanek mobilnih naprav pa uporablja krmilno ročico ali pa zgolj tipkovnico na napravi.

Največ dostopov pametnih telefonov in tabličnih računalnikov je gotovo opaziti predvsem pri spletnih straneh, ki so pretežno mobilne narave, kot so denimo vozni redi, socialna omrežja, vremenske napovedi in podobne vsebine.

Spletne strani, kot je denimo *www.prevoz.org*, ki je namenjena ponujanju in iskanju prevozov, dodatno privabljajo tisto skupino uporabnikov, ki navadno ne sedijo za namiznim računalnikom. Omenjeno spletno stran je obiskovalo 10% mobilnih uporabnikov že v lanskem letu 2012, letos pa jo uporablja kar 22% mobilnih uporabnikov (slika 1.1).



Slika 1.1: Primerjava petmesečnega obdobja deležev uporabljenih naprav spletne strani *www.prevoz.org* v letih 2012 in 2013.

Ob dejstvu, da prodaja mobilnih naprav narašča lahko pričakujemo veliko rast mobilnih uporabnikov na spletu tudi v prihodnjih letih. Odzivno spletno oblikovanje se sooča s težavo prikazovanja spletnih strani na množici različno velikih zaslonih. Deleži mobilnih uporabnikov v spletu so narasli do te mere, da bo smiselno prilagoditi vse spletne strani in ponuditi optimalno izkušnjo vsem uporabnikom, ne glede na vrsto naprave.

# Poglavje 2

## Značilnosti in omejitve spleta

Oblikovanje spletnih strani je zahteven proces, saj se oblikovalci prilagajamo mnogo dejavnikom, s katerimi se srečujemo na spletu. Zavedati se moramo, da danes uporabniki do spleta dostopajo z različnimi napravami, katerih velikost zaslona se razlikuje. Na vseh teh napravah so lahko nameščeni najrazličnejši operacijski sistemi s svojimi posebnostmi. Uporabniki imajo proste roke tudi pri izbiri spletnih brskalnikov, ki so prava nadloga mnogim spletnim oblikovalcem, saj vsak od njih včasih obliko spletne strani prikazuje malce drugače. Delo nam otežujejo še slike ter videoposnetki z visoko ločljivostjo, ki so danes modni hit in še dodatno obremenjujejo predvsem uporabnike s slabšo internetno povezavo.

Skratka seznam omejitev bi lahko še dopolnjevali. V nadaljevanju bomo obrazložili vse pomembne značilnosti in omejitve, ki se jih moramo oblikovalci zavedati pri izdelavi spletnih strani.

### 2.1 Naprave za dostop do spleta

John Paul Titlow [16] je začetke dostopa do spleta opisal z naslednjimi besedami: “Se še spomnite, ko je dostop do interneta pomenilo sedeti za mizo pri okornem računalniku s katodnim monitorjem, ob katerem smo čakali konec hrupa povezave na klic? Ti dnevi so na srečo že dolgo za nami in danes vse bolj dostopamo do spleta z množico različnih majhnih in velikih naprav.”

Dejansko stanje danes je, da imamo možnost dostopa do spleta skoraj povsod, brez povezovalnih kablov ali z njimi, z visokimi hitrostmi tudi med letom

v letalu ali pa med vožnjo na mestnem avtobusu, kar nam omogoča množica različnih naprav (slika 2.1). Za industrijo je možnost prodaje pametnih telefonov ter prenosnih in tabličnih računalnikov strmo narastla, izboljšale pa so tudi priložnosti slehernega uporabnika, ki lahko s pomočjo tovrstnih mobilnih naprav kjerkoli in kadarkoli dostopa do informacij dostopnih na spletu. Do spleta lahko dostopamo tudi preko televizijskih sprejemnikov, kjer lahko poslušamo glasbo, dostopamo do socialnih omrežij, brskamo po spletu, pregledujemo elektronsko pošto ali zgolj beremo novice.



Slika 2.1: Množica različnih naprav, s katerimi lahko danes dostopamo do svetovnega spleta [4].

Uporabniki od oblikovalcev vedno bolj zahtevajo prilagajanje spletne strani posameznim napravam, zato je naše delo zaradi specifičnih lastnosti naprav še toliko bolj oteženo.

## 2.2 Ločljivost zaslona

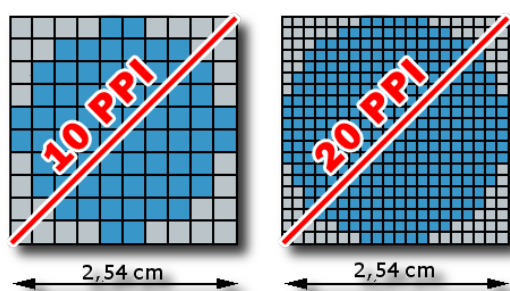
Ko govorimo o ločljivosti, mislimo na širino in višino zaslona naprave, ki je opisana s slikovnimi točkami (angl. pixels). Večja ločljivost na zaslonu iste velikost pomeni boljšo ostrino prikazane slike. Določene kombinacije širine in višine so standardizirane, zato so jim dodeljene kratice [27]. Zaradi množice različnih ločljivosti se zaslone razvršča tudi po razmerju stranic. Poznamo naslednja razmerja in njihove največkrat uporabljene ločljivosti zaslonov, s katerimi uporabniki dostopajo do spleta:

- **3:2** (HVGA 480 x 320, DVGA 960 x 640),
- **4:3** (SVGA 800 x 600, XGA 1024 x 786, UXGA 1600 x 1200),
- **5:4** (SXGA 1280 x 1024),
- **16:10** (WXGA 1280 x 800, WXGA+ 1440 x 900, WSXGA+ 1680 x 1050, WUXGA 1920 x 1200) ter
- **16:9** (HD 1366 x 768, HD+ 1600 x 900, FHD 1920 x 1080).

Obenem moramo pri oblikovanju spletnih strani biti pozorni tudi na orientacijo uporabniškega zaslona. Poznamo dva tipa orientacije in sicer pokončno (angl. portrait) ter ležečo (angl. landscape). Uporabniki nam z množično uporabo mobilnih naprav narekujejo željo po prilagodljivosti spletnih strani, obenem pa tudi oblikovalci vedno težje zagotavljamo ustrezen prikaz na vseh različnih napravah.

## 2.3 Gostota slikovnih točk

Poleg ločljivosti zaslonske slike je zelo pomembna tudi gostota slikovnih točk, ki je tesno povezana z velikostjo zaslona naprave. Slika ločljivosti 1280 x 1024 je na mobilni napravi z velikostjo zaslona 5 palcev dokaj ostra, medtem ko bo videti na velikem filmskem platnu zelo slaba. Gostoto slikovnih točk označujemo s številom posameznih slikovnih točk po diagonali na površini velikosti enega palca (PPI) ali enega centimetra zaslona (PPCM).



Slika 2.2: Primer dveh različnih gostot slikovnih točk na velikosti enega palca (2,54 cm).

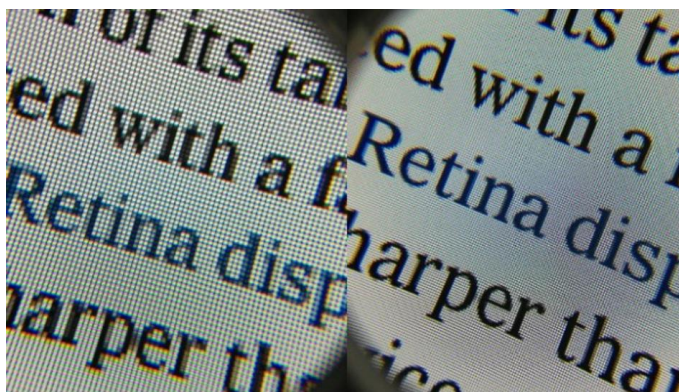
Kot vidimo na sliki 2.2, je z več slikovnimi točkami na zaslonu lažje prikazati gladke krivulje rastrskih slik ter z njimi na določeni razdalji prepričati človeško oko, da ne opazi posameznih slikovnih točk.

Naprava	Diagonala zaslona	Ločljivost zaslona	PPI
Samsung X-Cover	3.6" (9.14 cm)	320 x 480	158
iPhone 4 (Retina)	3.5" (8.89 cm)	640 x 960	326
iPhone 5 (Retina)	4.0" (10.16 cm)	1136 x 640	326
Sony Xperia S	4.3" (10.92 cm)	1280 x 720	342
iPad 1 in 2	9.7" (24.64 cm)	1024 x 768	132
iPad 3 in 4 (Retina)	9.7" (24.64 cm)	2048 x 1536	264
Sony Xperia Tablet Z	10.1" (25.65 cm)	1920 x 1200	224
Samsung Galaxy Tab 2 in 3	10.1" (25.65 cm)	1280 x 800	149

Tabela 2.1: Primerjava gostote slikovnih točk na nekaterih priljubljenih mobilnih napravah [31].

Za primerjavo naj omenimo, da ima 24 palčni zaslon namiznega računalnika pri ločljivosti 1920 x 1080 zgolj 91 PPI. Večja gostota slikovnih točk naj bi pomenila ostrejšo sliko, vendar pa se moramo zavedati, da za prikaz slike z visoko gostoto točk potrebujemo več grafičnega pomnilnika in procesorske moči. Razumljivo je, da nas proizvajalci zaradi lastnih koristi na nek način silijo v uporabo večje gostote slikovnih točk na zaslonu.

Primer blagovne znamke Apple so zasloni Retina, s katerimi dosegajo dovolj visoko gostoto, da človeško oko ne more opaziti posameznih slikovnih točk pri tipični razdalji (slika 2.3). Trdijo, da je za pametne telefone ta mejna vrednost okoli 300 PPI, če opazujemo zaslon iz razdalje od 25 do 30 centimetrov. Pri Retina zaslonih so slikovne točke zgoščene do te mere, da jih človeško oko povezuje v neprekinjene krivulje, s tem pa naj bi dosegli ostrino kot v zelo dobro natisnjeni knjigi, česar prej ni bilo mogoče doseči z nobeno elektronsko napravo [7].



Slika 2.3: Primerjava ostrine besedil na klasičnih in Retina zaslonih.

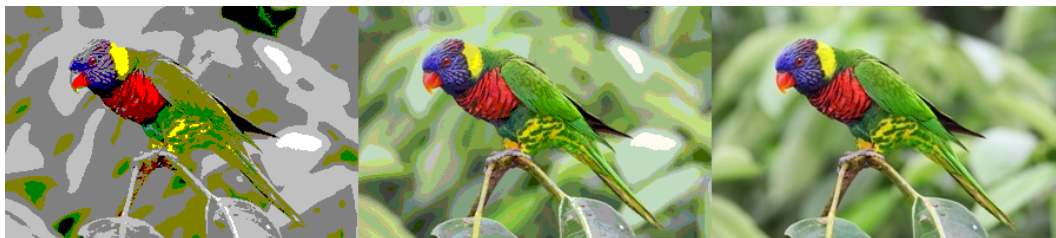
Ta učinek se doseže tako, da sliki štirikrat povečamo število slikovnih točk prikazanih na enako velikem zaslonu. Kot lahko opazimo v tabeli 2.1 pri napravah z zasloni Retina, pomeni to dvakratno povečanje širine in višine zaslonske ločljivosti, s čimer se doseže dvakratnik gostote slikovnih točk v PPI.

Pri oblikovanju spletnih strani za naprave z različno gostoto slikovnih točk se zopet pojavijo določena vprašanja glede prilagoditve. Zaradi vedno večjega števila naprav z zasloni Retina, so oblikovalci primorani preko omrežja dostavljati nekajkrat večje slike kot poprej, kar obremenjuje uporabnikovo internetno povezavo.

## 2.4 Barvna globina

Barve se lahko na različnih zaslonih prikazujejo različno. Ključno vlogo ima barvna globina, ki določa število barv ene slikovne točke. Za nadzor vsake točke na zaslonu naprave je dodeljen majhen del pomnilnika. Navadno označujemo barvno globino s številom bitov za nadzor ene slikovne točke (angl. BPP – bit per pixel), lahko pa je označena tudi s številom bitov na kanal (angl. BPC – bit per channel). Pri slednjem moramo število bitov zaradi treh osnovnih barv sistema RGB pomnožiti s tri, da dobimo število bitov na točko.

Slika z barvno globino 8 BPP ima za vsako slikovno točko na voljo 256 ( $2^8$ ) barv. Poznamo tudi večje barvne globine: 16, 24, 32, 48 ter 64 bitov na slikovno točko. Barvam, ki za svoj zapis uporabljajo 24 bitov pravimo tudi prave barve (angl. True colors), saj lahko z njimi zapišemo kar 16 777 216 ( $2^{24}$ ) različnih barv, kar je dovolj za prikaz vseh odtenkov, ki jih je zmožno zaznati človeško oko [8]. Po statistiki zmore kar 98 % današnjih računalnikov prikazovati 24 ali 32 bitno sliko [22], kjer vsaki osnovni barvi sistema RGB (rdeča, zelena, modra) pripada za zapis v pomnilniku po 8 bitov. Razlika med 24 in 32 bitno sliko je zgolj 8 dodatnih bitov za prosojnost.

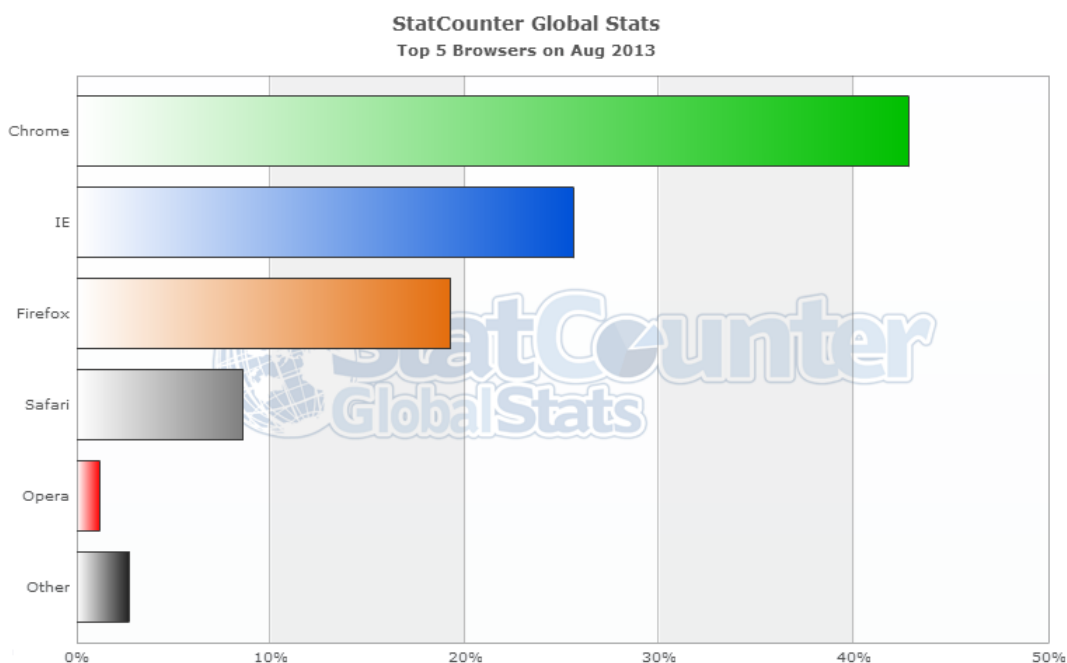


Slika 2.4: Primer fotografije s 4, 8 ter 24 bitno barvno globino [38].

Barvna globina ima v tesni povezavi z ločljivostjo slike velik vpliv na velikost slikovne datoteke, zato morajo biti le te prilagojene spletu. Oblikovalci se pri optimizaciji slik največkrat poslužujemo tehnike prilagajanja ločljivosti posameznim zaslonom ter izbire ustreznih metod stiskanja slik (JPEG, GIF, PNG).

## 2.5 Spletni brskalniki

Vsakodnevno brskanje po spletu ne bi bilo mogoče brez uporabe spletnih brskalnikov. To so računalniški programi, ki omogočajo prikazovanje dokumentov HTML ter drugih večpredstavnih vsebin na spletu. So grafični vmesnik za vse na spletu dostopne informacije, ki so s pomočjo brskalnika prebrane in prikazane spletnemu uporabniku. Danes so najbolj razširjeni spletni brskalniki Firefox, Chrome, Opera, Safari ter Internet Explorer, vendar pa so načini razumevanja kode spletnih strani pri nekaterih brskalniki različni. Velika težava starejših spletnih brskalnikov je njihova varnostna pomanjkljivost ter neupoštevanje standardov za prikaz spletnih strani.



Slika 2.5: Primerjava uporabe spletnih brskalnikov v avgustu 2013 [40].

Kot opazimo na sliki 2.5 ima spletni brskalnik Google Chrome največji delež uporabnikov in sicer 42.8%. Sledi mu brskalnik Internet Explorer z deležem 25.5%. Posamezne različice tega brskalnika si delijo deleže 11.3% (ver. 10), 5.2% (ver. 9), 8.3% (ver. 8), 0.5% (ver. 7) ter 0.2% (ver. 6). Oblikovalcem so velik trn v peti predvsem brskalniki starejši od različice Internet Explorer 9. Na tretjem mestu po uporabi je spletni brskalnik Mozilla Firefox s 19.3% deležem, takoj za njim pa brskalnik Safari z 8.6% deležem. Brskalnik

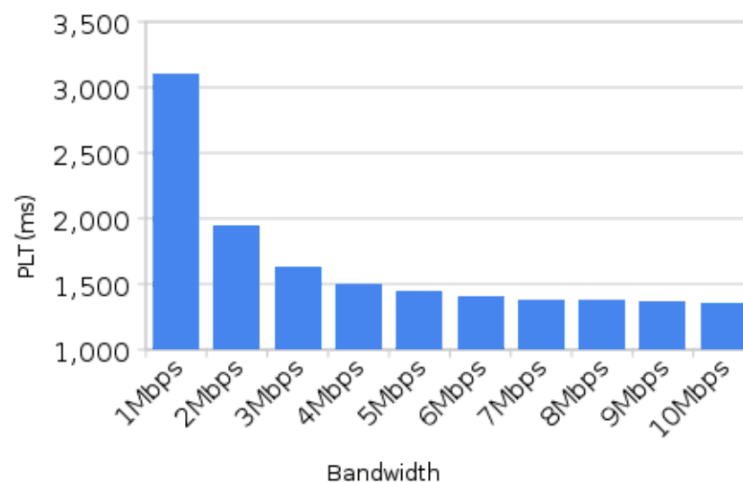
Opera je prav tako zanesljiv, a vseeno zelo podcenjen, saj ga uporablja le 1.2% uporabnikov po svetu. Zaradi navad spletnih uporabnikov se srečujemo z nezanimljivim deležem uporabe starejših in problematičnih brskalnikov, zato smo spletni oblikovalci za posamezne brskalnike primorani upoštevati določene prilagoditve.

Izmed vseh spletnih brskalnikov pozitivno izstopa brskalnik Opera Mini, ki je bil primarno zasnovan za mobilne uporabnike leta 2006 in učinkovito rešuje problem prikaza spletnih strani, ki še niso prilagojene tovrstnim napravam. Brskalnik Opera Mini zahteva prenos spletnih strani skozi njihove strežnike, kjer jih le ti obdelajo in stisnejo, preden so poslani do mobilnih naprav, kar zmanjša količino prenesenih podatkov in pospeši čas prenosa tudi do trikrat [36]. Brskalnik je podprt na večini priljubljenih operacijskih sistemih (Android, iOS, Symbian, Windows Mobile). Predhodna obdelava spletne strani na njihovem strežniku bistveno poveča združljivost s spletnimi stranmi, ki niso prilagojene mobilnim napravam.

## 2.6 Internetna povezava

Za dostop do spleta potrebujemo hitro in zanesljivo internetno povezavo. Pasovna širina izraža hitrost internetne povezave, ki jo označujemo v prenesenih bitih na sekundo (bit/s, kbit/s, Mbit/s, Gbit/s). Večja kot je pasovna širina, hitreje naj bi se spletna stran prenesla po komunikacijskem kanalu do uporabnika. Predvsem hitrost nalaganja spletne strani vpliva na pozitivno uporabniško izkušnjo, katera nam daje povečano možnost, da se bo uporabnik še kdaj vrnil na spletno stran. Pasovna širina je zelo pomemben dejavnik zmogljivosti prenosa, vsekakor pa ni edino merilo za izboljšanje hitrosti nalaganja spletnih strani.

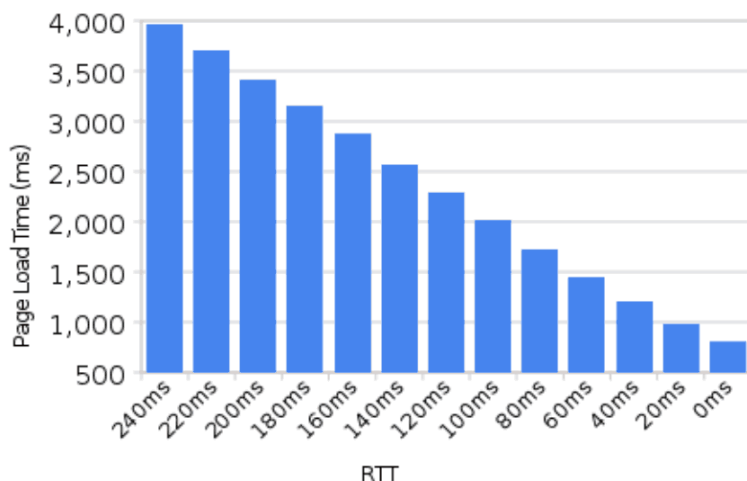
Zelo pomemben je tudi čas vrnitve RTT (angl. Round-Trip Time) oz. latenca, ki izraža čas od trenutka pošiljanja paketa do sprejema potrditve. Čas prenosa celotne spletne strani ni neposredno odvisen od pasovne širine. Rezultati preizkusov prikazujejo povprečni čas naložitve 25 priljubljenih spletnih strani z različnimi pasovnimi širinami ter enakim odzivnim časom 60 ms.



Slika 2.6: Primerjava časov naložitve 25 priljubljenih spletnih strani z uporabo različnih pasovnih širin ter omejenim časom RTT [1].

Kot lahko opazimo na sliki 2.6, se z večanjem pasovne širine zmanjša čas nalaganja spletnih strani, vendar se ta učinek ustali že nekje pri hitrosti 5 Mbit/s. Glavni razlog za to predstavlja zastarel protokol HTTP/1.1, ki za nadzor zastojev prenosa paketov TCP uporablja strategijo počasnega začetka (angl.

Slow start). Ta metoda ob vzpostavitvi povezave TCP za prenos postopoma zvišuje količino prenesenih paketov in s tem preprečuje nastanek prezasedenosti omrežja. Ob prenosu velikega števila majhnih datotek se celotna pasovna širina zaradi počasnega začetka le redko izkoristi.



Slika 2.7: Čas potreben za prenos 25 priljubljenih spletnih strani z omejeno pasovno širino ter različnimi časi RTT [1].

Rezultati drugega preizkusa na sliki 2.7 prikazujejo čase prenosov spletnih strani, kjer je pasovna širina omejena na 5 Mbit/s v odvisnosti od različnih časov RTT in sicer od 240ms do 0ms. Ti rezultati jasno nakazujejo, da so časi naložitve spletnih strani linearno povezani s časi vrnitve RTT. Odzivni čas v katerem se spletna stran naloži je za uporabnike zelo pomemben in je ključ do dobre uporabniške izkušnje. Za izboljšavo slednjih moramo imeti zmogljiv spletni strežnik na povezavah internetnih ponudnikov z veliko pasovno širino, do katerih imajo ciljni uporabniki kratke dostopne čase. Za hiter prenos spletne strani do uporabnika se moramo oblikovalci potruditi, da za ogled strani uporabimo čim manj zahtev HTTP ter da optimiziramo vsebino, ki mora biti prenesena s spletnega strežnika.

# Poglavje 3

## Uporabljene tehnologije

### 3.1 Označevalni jezik HTML5

HTML5 je označevalni jezik za oblikovanje in predstavitev vsebin na svetovnem spletu, ki je osrednja tehnologija današnjega interneta. To je že peta sprememba standarda HTML od leta 1990, katera ima osrednji cilj izboljšati jezik z naprednimi multimedijskimi tehnologijami. HTML5 je trenutno še v fazi razvoja, predstavlja pa pomembno poglavje področja spletnega oblikovanja.

Označevalni jezik HTML5 dodaja nove značke za vstavljanje multimedijskih vsebin, kot so denimo `<video>`, `<audio>`, `<canvas>` in druge. Zasnovane so za lažje vključevanje in upravljanje multimedije, ne da bi se bilo potrebno zatekati k lastnim vtičnikom.

Številne nove značke kot so `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>` in druge nam pomagajo poleg same izdelave strukture spletne strani tudi pri semantični obogatitvi vsebine. Z izbiro teh elementov lahko bolje opišemo pomembne dele spletne strani, kar pripomore k splošni preglednosti dokumenta in bo nekoč vplivalo tudi na optimizacijo za spletne iskalnike SEO. Svetovno znan spletni iskalnik Google naj nebi še prilagajal svojih tehnik razumevanja posodobljenemu jeziku HTML5, predvsem zaradi splošne nerazširjenosti. Trdijo, da nameravajo svoje algoritme prilagoditi jeziku HTML5 šele, ko bo le ta s svojo razširjenostjo in s pravilno implementacijo omogočal lažje indeksiranje spletnih strani, obenem pa trenutna uporaba te tehnologije naj nebi koristila ali škodila pri samem uvrščanju med zadetki [26].

## 3.2 Slogovni jezik CSS3

Kaskadne stilske podloge, ki jih zapišemo tudi s kratico CSS (angl. Cascading Style Sheets) se uporabljajo v obliki preprostega slogovnega jezika, s katerim nadzorujemo prezentacijo spletnih strani. Glavni namen slogovnega jezika CSS je, da ločimo strukturo vsebine od dizajna spletne strani. S slogovnim jezikom CSS imamo možnost lažjega dodajanja in urejanja stilov, obenem pa poskrbimo za večjo preglednost dokumentov HTML. Slogovne lastnosti lahko zapišemo v posebne datoteke s končnico `.css`, katere vključimo v glavo dokumenta in se s tem znebimo ponavljanja kode znotraj vsakega dokumenta.

Sintaksa jezika je preprosta in sestoji iz angleških ključnih besed, ki poimenujejo stilske lastnosti. Slogovno pravilo je sestavljeno iz enega ali več selektorjev in deklaracijskega bloka, znotraj katerega so zapisane stilske lastnosti.

```
1 .selektor {
2   background-color: #888;
3   text-align: center;
4   font-family: Arial;
5   width: 100%;
6   color: #F00;
7 }
```

Selektor določa ime, na katerega se slogovna pravila nanašajo. Znotraj dokumenta HTML lahko elementom dodajamo selektorje in s tem določimo katera slogovna pravila veljajo zanj.

```
1 <div class="selektor">
2   Vsebina bloka div.
3 </div>
```

Novosti, ki jih prinaša posodobitev jezika CSS3 so animacije, stolpci, gradienti, odsevi, prehodi in pa tudi medijske poizvedbe, ki so veliko pripomogle ravno k razvoju odzivnega spletnega oblikovanja. Vse te zmožnosti jezika CSS3 sedaj nadomeščajo in izpodrivajo predhodne rešitve animacij Flash ter slik GIF. Ker je koda jezika navadno majhna, učinkovita in med sodobnimi brskalniki dobro podprta, je uporaba slogovnega jezika CSS3 v področju spletnega oblikovanja zelo pomembna [2].

## 3.3 Skriptni jezik JavaScript

JavaScript je objektni skriptni jezik, ki spletnim razvijalcem omogoča izdelavo interaktivnih spletnih strani. Z njim lahko popestrimo statične spletne strani, jim dodamo dinamične učinke ali pa razvijemo zahtevnejše aplikacije. Jezik podpirajo velika programska podjetja, zato je kot odprtokodni jezik na voljo vsem. Kodo JavaScript lahko vstavljamo znotraj dokumenta HTML in z njo implementiramo dinamično izvajanje v spletnem brskalniku. Sodobni spletni brskalniki podpirajo jezik JavaScript, zato ga mnogo spletnih oblikovalcev uporablja kot orodje za preverjanje vnesenih podatkov v spletne obrazce, izdelavo interaktivnih navigacij, upravljanje z okni ter za delo s piškotki.

## 3.4 Programski jezik PHP

PHP je široko uporabljen in odprtokodni programski jezik, ki se izvaja na spletnem strežniku in je zelo primeren za razvoj dinamičnih spletnih strani. Seznam funkcij, ki jih jezik PHP omogoča so dokumentirane na uradni spletni strani *www.php.net*. Kodo PHP lahko vključimo tudi v dokument HTML znotraj značk `<?php` ter `?>`, kot prikazuje spodnji primer.

```
1 <html>
2   <head>
3     <title>Detekcija locljivosti zaslona</title>
4     <script>document.cookie='piskotek='+screen.width;
5       </script>
6   </head>
7   <body>
8     <?php
9       $piskotek = $_COOKIE["piskotek"];
10      if ($piskotek <= 480)
11        echo "majhna locljivost";
12      else
13        echo "vecja locljivost";
14    ?>
15  </body>
16 </html>
```

Koda prikazuje primer združitve dokumenta HTML z jezikoma JavaScript, ki se izvede na odjemalčevi strani in PHP, ki se izvede na strani spletnega strežnika. V nasprotju z običajnim dokumentom HTML, koda PHP ni poslana neposredno spletnemu brskalniku, temveč jo pred tem izvede spletni strežnik, ki nato brskalniku pošlje rezultat v obliki HTML.

Pri oblikovanju spletnih strani gre največkrat za souporabo večih tehnologij, ki nam omogočajo razvoj kompleksnih aplikacij.

# Poglavje 4

## Odzivna spletna stran

### 4.1 Predhodne rešitve

Že s prvimi mobilnimi napravami, ki so imele možnost brskanja po spletu se je pojavila težava prikaza spletnih strani. Te naprave so se razlikovale od namiznih naprav predvsem v tem, da so imele zaslon veliko manjši, posledično tudi manjšo zaslonsko ločljivost, slabšo internetno povezavo ter manjšo procesorsko in pomnilniško zmogljivost. Brskanje po takratnih spletnih straneh s takšno napravo je bilo težavno ali pa sploh ni bilo mogoče.

Za rešitev tega problema so takrat oblikovalci iznašli WAP (angl. Wireless Application Protocol), ki za osnovo uporablja jezik XML. Gre za skupek tehnologij in protokolov s katerim je bil omogočen dostop mobilnim telefonom do osnovnih spletnih vsebin.

Ta rešitev ni bila idealna, saj je bila izdelava in oblikovanje takšnih spletnih strani precej omejeno (slika 4.1). Besedil je bilo malo, grafika pa osnovna z uporabo nekaj majhnih sličic. Zaradi vseh omejitev mobilnega telefona, so morale biti spletne strani WAP prilagojene in narejene ločeno od prvotne spletne strani. Večinoma uporabniki z brskanjem po spletnih straneh WAP niso bili zadovoljni.



Slika 4.1: Mobilni telefon, ki za spletno brskanje uporablja tehnologijo WAP.

S prihodom naprednejših telefonov so bile priljubljene posebej prilagojene spletne strani za mobilne uporabnike. Te strani so se največkrat nahajale na poddomeni ali pa v posebni mapi prvotne spletne strani (npr.: *m.example.com*, *wap.example.com*, *www.example.com/mobile/*), na katere so preusmerjali vse mobilne obiskovalce. To je bil že drugi poskus ločenega oblikovanja spletnih strani za mobilne uporabnike. Z njo so lastniki spletne strani želeli prilagoditi svojo stran za mobilne naprave na drugem spletnem naslovu, vendar navadno tudi za točno določeno širino zaslona (slika 4.2). Težava nepravilnega prikaza spletne strani nastopi že, če mobilni uporabnik pošlje povezavo URL nekemu po elektronski pošti ali pa jo deli preko socialnih omrežjih ter jo kdorkoli odpre na namiznem računalniku in obratno.



Slika 4.2: Prikaz mobilne spletne strani *m.delo.si* na starejših telefonih.

Zaradi različnih povezav med prvotno in mobilno spletno stranjo prihaja tudi do dvojnega prometa, kar povzroča nevšečnosti pri optimizaciji za spletne iskalnike (SEO) ter težave pri analizi obiskovalcev spletne strani.

Kmalu so se pojavile prve naprave, kot so pametni telefoni in tablični računalniki, opremljeni z naprednimi operacijskimi sistemi Android in iOS. Oblikovalcem se je odprlo mnogo možnosti, saj so uporabniki imeli na voljo napredne naprave z dovolj procesorske moči in hitrejšo internetno povezavo. Še vedno pa je ostajala ena težava in sicer velikost zaslona.



Slika 4.3: Drsniki za premikanje po strani ter pomanjšana spletna stran predstavljajo neprijetno izkušnjo uporabnika med brskanjem.

Kot prikazuje slika 4.3, se običajne spletne strani s fiksno širino na pametnih telefonih ne prikazujejo najbolje, kajti velikost pisav, slik ter drugih vsebin je na majhnem zaslonu zelo pomanjšana in težko berljiva. Pri povečevanju takih spletnih strani na mobilni napravi, se je moral uporabnik premikati po strani s horizontalnim in vertikalnim drsnikom, kar ne omogoča najboljše uporabniške izkušnje za brskanje po spletu.

Napačno je že samo razmišljanje, da moramo izdelovati spletne strani s fiksno širino. Kot primer lahko na sliki 4.4 vidimo prvo spletno stran, ki jo je objavil CERN v aprilu leta 1993 in se še danes brezhibno prikazuje na mobilnih napravah [41].



Slika 4.4: Prva spletna stran objavljena leta 1993.

Ta svetovno znana spletna stran je že odzivna na vseh napravah, saj vsebuje zgolj besedilo, ki ima lastnost prilagajanja širini zaslona. To kaže, da je splet že od samega začetka prilagodljiv, vendar smo zaradi prevladujočega osredotočanja na namizne računalnike oblikovali spletne strani s fiksnimi širinami.

Današnje spletne strani so veliko bolj obsežne, zato moramo oblikovalci prilagoditi prikaz vedno večji množici različnih naprav, kjer je vsaka posebna s svojo ločljivostjo zaslona. Ob poplavi pametnih telefonov ter tabličnih računalnikov se je pojavilo mnogo vprašanj o načinih prilagoditve spletnih strani in vsebin za posamezne naprave in brskalnike, ki se pojavijo na trgu, vendar ne s prilagojenimi spletnimi stranmi na drugem naslovu.

## 4.2 Odzivno spletno oblikovanje

Odzivno spletno oblikovanje je pristop izdelave spletne strani, ki cilja na optimalno izkušnjo uporabnikov, ne glede na to, s katere naprave dostopajo do spletne strani. Dandanes namizni računalnik še zdaleč ni edini način s katerim si ljudje ogledujemo spletne strani, namreč do spleta lahko dostopamo že preko pametnih telefonov, tabličnih računalnikov, prenosnih računalnikov, televizijskih sprejemnikov in mnogo drugih naprav, za katere današnje spletne strani še niso prilagojene. Koncept, ki ga je v maju leta 2010 predstavil Ethan Marcotte je problem reševal s tekočo postavitvijo elementov nadzorovanih s pomočjo slogovne kode CSS3 ter z uporabo prilagodljivih slik [12].



Slika 4.5: Primer odzivne spletne strani na različnih napravah ob različnih orientacijah.

Gre torej za spletno stran, ki zazna uporabnikovo napravo in se odzove s prilagoditvijo vsebine strani tako, da je ta čim bolj vidna na njegovi napravi (slika 4.5). Bistvena prednost tovrstnega pristopa je, da imamo eno spletno stran za vse naprave, kar nam omogoča lažje posodabljanje vsebine, lažjo analizo obiskovalcev ter boljšo optimizacijo za spletne iskavnike (SEO).

Zaradi množice različnih omejitev internetnega omrežja in zmožnosti mobilnih naprav, so številni razvijalci spletnih strani prišli do spoznanja, da odzivno spletno oblikovanje ne zajema zgolj klasično oblikovanje spletne strani, temveč

tudi strežniške prilagoditve posameznim napravam. Za optimalno delovanje odzivnih spletnih strani je potrebno glede na zmožnosti naprave pošiljati le najnujnejše podatke, pri čemer nam pomagajo optimizacijske tehnike opisane v nadaljevanju.

### 4.2.1 Tekoča postavitve elementov

Pred uporabo odzivnih spletnih strani so se le te oblikovale na sistemu elementov s fiksnimi merami. Veljalo je nenapisano pravilo, da širina strani ni smela presegati 960 slikovnih pik, katerega so se morali držati vsi oblikovalci, če so želeli, da se njihova spletna stran pravilno prikazuje na najbolj razširjenih zaslonskih resolucijah.

Mobilne naprave postajajo vse manjše, po drugi strani pa namizni monitorji postajajo vse večji. Za oblikovalce je velik dosežek, da ne rabijo več prilagajati spletne strani za posamezno napravo. Koncept tekoče postavitve, ki ga uporabljamo pri odzivnem spletnem oblikovanju zahteva, da je velikost elementov določena z relativnimi enotami, kot so odstotki, namesto absolutnih enot, kot so slikovne pike. Pomembno je, da poznamo ta pristop, preden se lotimo oblikovanja odzivnih spletnih strani. Z uporabo pristopa tekoče postavitve elementov se lahko prilagodi vse elemente tudi na večjih zaslonih, zaradi izračunov, ki temeljijo na odstotkih.

100 %			
75 %			25 %
25 %	25 %	25 %	25 %

Slika 4.6: Primer tekoče postavitve elementov v mreži.

Tekoče elemente definiramo v mreži, kjer jim velikost znotraj strukture določimo v odstotkih. Mreža je razdeljena na določeno število stolpcev, da

ohranjamo urejanje postavitev čim bolj enostavno. Vsak element znotraj mreže ima definirano širino, koliko stolpcev v mreži zavzame, zapisano z odstotki (slika 4.6). V takem sistemu, kjer so vsi elementi znotraj mreže definirani z relativnimi enotami, si tako ob spremembi naprave ali zaslona vsi elementi prilagodijo širino in višino sorazmerno glede na predhodni element.

Tudi pri definiranju pisav spletnih strani je pomembno, da njihove velikosti izražamo v relativnih enotah. Najbolj priljubljene tovrste enote so `em` in odstotki `%`, kjer velja pravilo, da je `1em` enako `100%`, kar je enako v ostalih enotah `16px` oz. `12pt` [15]. Prednost uporabe relativnih enot pri pisavah je, da se velikost pisave prilagaja na lastnost predhodnega elementa (slika 4.7), kar omogoča oblikovalcem lažjo prilagodljivost pri oblikovanju spletnih strani za različno velike zaslone.

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	Naključno besedilo	<b>Naključno besedilo</b>
<code>font-size: 12pt</code>	Naključno besedilo	Naključno besedilo
<code>font-size: 16px</code>	Naključno besedilo	Naključno besedilo
<code>font-size: 100%</code>	Naključno besedilo	<b>Naključno besedilo</b>

Slika 4.7: Primer besedil, kjer so pisave izražene v relativnih in absolutnih enotah.

Pri izdelavi tekoče postavitev elementov so nam danes v pomoč že ustvarjene mreže, ki omogočajo hiter razvoj odzivnih spletnih strani. Med priljubljenimi tovrstnimi orodji poznamo Bootstrap, Foundation, Skeleton in druge [24]. Hiter razvoj z naštetimi orodji ima tudi svojo slabost, saj je koda brez poznavanja in prilagoditve njene velikosti hitro preobsežna za večino potreb oblikovalcev pri izdelovanju spletnih strani.

## 4.2.2 Prilagodljive slike in druge medijske datoteke

S tekočo postavitvijo elementov smo dosegli, enostavno prilagodljivost konstrukcije spletne strani. Ta rešitev bi bila dovolj, če bi na spletnih straneh uporabljali le tekstovne vsebine, ki se brez velikega truda lepo lomijo v vrsticah znotraj prilagodljivih elementov. Spletne strani pa vsebujejo vedno več slikovnih in video vsebin, ki imajo fiksno širino.

Težava nastopi že, ko imamo sliko večjo, kot pa je element znotraj katerega je vstavljena. V takem primeru ostanejo mreža in njeni elementi na svojem mestu, saj so definirani z absolutnimi enotami, vendar presežek prevelike slike uide iz svojega elementa. Vsem slikam moramo določiti omejitve, da ustrezajo v tem prilagodljivem sistemu elementov.

```
1 | img {  
2 |   max-width: 100%;  
3 | }
```

S tem pravilom znotraj slogovne datoteke CSS smo omejili vse slike v svoje predhodne elemente. Slike se bodo prilagajale na poljubno širino predhodnega elementa, dokler bo le ta manjši od velikosti slike. Današnji spletni brskalniki podpirajo sorazmerno pomanjševanje slik, kar pomeni, da se ob spreminjanju velikosti elementa, slika znotraj njega ustrezno prilagaja, obenem pa ohranja razmerje med širino in višino.

S to tehniko lahko rešujemo tudi vstavljanje videoposnetkov in drugih obogatitvenih vsebin. Pravzaprav lahko selektorje CSS združimo in naredimo univerzalno pravilo za medijske datoteke.

```
1 | img, embed, object, video {  
2 |   max-width: 100%;  
3 | }
```

Z značko `<video>` dosežemo prikaz videoposnetkov s privzetim predvajalnikom brskalnika, kar nam omogoča sodobna tehnologija HTML5. Na spletu se pogosto srečujemo, da nekatere video vsebine vstavljamo iz drugih spletnih strani, kjer se pojavlja težava odzivnosti, predvsem pri predvajalnikih Flash. Za rešitev te težave je na spletu na voljo knjižnica JavaScript, dostopna na [www.fitvidsjs.com](http://www.fitvidsjs.com), ki rešuje težavo odzivnosti tovrstnih vsebin.

### 4.2.3 Medijske poizvedbe znotraj slogovne kode CSS3

Zgolj z uporabo tekoče postavitve elementov ter prilagodljivih slik lahko izdelamo spletno stran, ki se bo dokaj lepo prilagajala majhnim spremembam zaslonov. S spreminjanjem širine svojega brskalnika lahko opazujemo kako se mreža, elementi ter vsebina prilagajajo širini svojih predhodnih elementov. Že ob takem opazovanju bomo opazili, da nam bo spletna stran vseč le do neke točke, kjer bi bilo potrebno narediti strukturno spremembo elementov (slika 4.8). Takrat pridejo na vrsto medijske poizvedbe, da naredijo tiste male spremembe, s katerimi je spletna stran spet dobro vidna, tudi na zelo majhnih oz. velikih zaslonih.

Medijske poizvedbe nam omogočajo definiranje različnih skupin slogovnih pravil glede na zapisan pogoj. Funkcije, ki jih lahko znotraj CSS3 uporabljamo v pogojih so širina spletnega brskalnika, orientacija naprave, razmerje med širino in višino zaslona ter mnogo drugih. Znotraj posamezne medijske poizvedbe imamo možnost določanja posameznih slogovnih pravil, s pomočjo katerih lahko poljubno oblikujemo elemente ter jih umestimo na pravo mesto znotraj tekoče postavitve elementov.

```
1 @media screen and (min-width: 1024px)
2   and (orientation: landscape) {
3     /* slogovna prilagoditev elementov strani za naprave
4     z ločljivostjo najmanj 1024 slikovnih točk v ležeči
5     orientaciji */
6   }
```



Slika 4.8: Elementi na različno velikih zaslonih zahtevajo strukturne spremembe v mreži tekoče postavitve.

### 4.3 Optimizacijske tehnike

Na spletu je opaziti mnogo kritik, da so odzivne spletne strani zaradi dodatne kode še večje in počasnejše, vendar za to ni kriv tovrstni pristop, ampak sama implementacija s katerimi se na spletu trenutno srečujemo [9]. Vedeti moramo, da je pristop odzivnega spletnega oblikovanja dokaj nov in priljubljen, vendar vseeno premalo raziskan, zato zaradi pomanjkanja znanj oblikovalcev nekatere spletne strani niso najboljše optimizirane.

Med naprednejšimi oblikovalci in razvijalci je na spletu mnogo razprav glede izdelave spletnih strani, ki dobro delujejo na številnih različnih napravah. Nekateri so naklonjeni za optimizacijske pristope pri odjemalcu, drugi se bolj nagibajo k učinkoviti strežniški rešitvi. Luke Wroblewski je s svojim člankom predstavil predlog združitve obeh strani optimizacij v pristop RESS, ki naj bi ga sestavljala tako odzivno spletno oblikovanje, kot optimizacijske komponente na strežniški strani [20].

Z implementacijo odzivnega spletnega oblikovanja dosežemo, da imamo eno spletno stran, ki se prilagaja uporabniškemu zaslonu na odjemalčevi strani. Do odzivnih spletnih strani dostopamo na enotni URL povezavi, kar pomeni prenos identične vsebine za množico vseh različnih naprav. Pomanjkljivost tega pristopa je, da določene naprave prenašajo nekatere datoteke in podatke, ki jih ne potrebujejo ali pa so za njih preobsežni.

Čas, ki je potreben za prikaz spletne strani predstavlja uporabniku zelo velik faktor njegove pozitivne izkušnje med brskanjem. Pri časovnih zakasnitvah nalaganja spletnih strani poznamo tipične uporabniške reakcije, ki so določene s človeškimi zaznavnimi sposobnostmi [14]:

- Zakasnitev od 0 ms do 100 ms - spletna stran je hitra, saj se naloži takoj.
- Zakasnitev od 100 ms do 300 ms - spletna stran se zdi počasna.
- Zakasnitev od 300 ms do 1000 ms - največja meja za prijetno brskanje.
- Zakasnitev več kot 1 s - zdi se, da naprava izvaja nekaj zahtevnega.
- Zakasnitev več kot 10 s - uporabnik se bo odločil oditi iz spletne strani.

Ob spoznanju vseh značilnosti in omejitev spleta, ki smo jih opisali v poglavju 2 smo prepričani, da lahko nekatere težave izboljšamo z uporabo opti-

mizacijskih pristopov na spletnem strežniku. Z rešitvami na strežniški strani dosežemo, da preko omrežja pošiljamo le najnujnejše podatke, ki jih odjemalec potrebuje za prikaz odzivne spletne strani prilagojene zmogljivostim njegove naprave.

V nadaljevanju smo opisali nekaj priljubljenih ter učinkovitih optimizacijskih metod na strežniški in odjemalčevi strani, ki smo jih nato tudi implementirali v praktičnem primeru.

### 4.3.1 Prilagodljive slike

Kadar govorimo o velikosti spletne strani, so navadno slike tisti viri, ki predstavljajo največji delež prenosa s spletnega strežnika. Programska rešitev prilagodljivih slik (angl. Adaptive Images), ki jo je predstavil Matt Wilcox, učinkovito rešuje problem dostavljanja velikih slik do mobilnih naprav [18]. S tehniko prilagodljivih slik skušamo prenašati po omrežju le slike, ki ustrezajo zasloni ločljivosti uporabnika. Uporabniku se ob dostopu do spletne strani najprej naloži dokument HTML. V glavi dokumenta `<head>`, se izvede kratka koda, ki s programskim jezikom JavaScript zapiše sejni piškotek, v katerem je shranjena ločljivost uporabnikovega zaslona (največja vrednost izmed višine in širine) ter gostota slikovnih točk.

```
1 <script type="text/javascript">
2 document.cookie='resolution='+Math.max(screen.width,
   screen.height)+("devicePixelRatio" in window ? ","+
   devicePixelRatio : ",1")+'; path
3 </script>
```

Skupaj z brskalnikovo zahtevo po sliki, se strežniku pošlje tudi sejni piškotek z informacijo o ločljivosti uporabnikovega zaslona. Spletni strežnik sprejme zahtevo za sliko in pogleda v strežniško datoteko `.htaccess`, da bi se prepričal, če so v njej zapisane kakšne posebnosti dostavljanja. V tej datoteki zapišemo pogoj, ki vsako zahtevo slike preusmeri na datoteko `adaptive-images.php`, katera bo poskrbela za dostavljanje optimalno velikih slik. S spodnjim pravilom smo v pogoju preusmeritve določili formate slik JPEG, GIF in PNG.

```
1 RewriteRule \.(?:jpe?g|gif|png)$ adaptive-images.php
```

V preusmerjeni datoteki `adaptive-images.php` nam je ponujena določitev posameznih razredov velikosti slik, ki jih želimo dostavljati odjemalcem. Odločili smo se za naslednje razrede ločljivosti slik.

```
1 | $resolutions = array(1600, 1200, 1024, 940, 768, 480);
```

Programski jezik PHP ima to zmožnost, da pogleda sejni piškotek ter prebere največjo ločljivost uporabniškega zaslona. Sledi preverjanje ali pomanjšana slika izbrane ločljivosti že obstaja na strežniku. V primeru, da ustrezno pomanjšane slike še ni v predpomnilniški mapi `ai-cache`, jo spletni strežnik naprej ustvari ter shrani za naslednji dostop in nato dostavi brskalniku. Ta metoda prilagajanja slik za svoje delovanje potrebuje podporo spletnih piškotkov. V primeru, da le teh naprava ne podpira ali pa so iz poljubnih razlogov onemogočeni, se napravam, ki imajo v uporabniškem agentu besedo `mobile` dostavi najmanjša slika, ostalim pa največja.

Za uporabo pristopa dostavljanja prilagojenih slik ne potrebujemo nobenih sprememb v kodi HTML pri označevanju slik. Implementacija v odzivno oblikovano spletno stran je zelo preprosta, saj se rešitev odlično dopolnjuje s tehniko prilagodljivih slik opisano v poglavju 4.2.2. Z uporabo te programske rešitve dostavljamo primerno velike slike tudi trenutno priljubljenim napravam z zaslona Retina, ki imajo visoko gostoto slikovnih točk. Uporabniška izkušnja na spletnih straneh, ki uporabljajo prilagodljive slike je zelo pozitivna, predvsem za mobilne naprave z majhnimi zaslona, manj zmogljivimi procesorji ter za naprave s slabšo povezavo do svetovnega spleta. Na takih napravah bi se slike brez optimizacije v polni velikosti nalagale zelo počasi, povzročale bi dolge zakasnitve pri nalaganju celotne spletne strani ter nepotreben promet, ki še najbolj prizadene uporabnike s plačljivo podatkovno povezavo do spleta. Obenem je proces optimizacije slik za urednika spletne strani popolnoma transparenten in mu za izboljšanje uporabniške izkušnje ne povzroča dodatnega truda.

### 4.3.2 Zakasnjeno nalaganje slik

Če želimo uporabnikom zagotoviti hitre čase nalaganja spletnih strani, so slike ter njihov prenos prvi kandidat za optimizacijo. Zelo priljubljena tehnika za pospešitev nalaganja spletnih strani se imenuje `LazyLoad`. Gre za programsko

rešitev napisano v jeziku JavaScript, ki jo je predstavil Matt Mlinac in za svoje delovanje potrebuje knjižnico jQuery. Z uporabo zakasnjene naložnje se naložijo le slike znotraj vidnega polja brskalnika, ostale pa šele, ko se z drsnikom premaknemo do njih [17].

Deluje na principu, da pri vseh slikah zamenjamo vrednost v atributu izvorne slike `src` z neko drugo sliko, ki je praviloma zelo majhna. To nam omogoča, da se vsa ostala struktura spletne strani naloži pred velikimi slikami. Pot do izvorne slike je shranjena v atributu `data-original`, tako da se lahko naloži kasneje. Taka rešitev nam pride zelo v pomoč, saj drugače nimamo popolnega nadzora, kdaj se posamezne slike pričnejo prenašati s spletnega strežnika.



Slika 4.9: Primer prikaza galerije slik na pametnem telefonu s tehniko zakasnjene naložnje slik, kjer se sprva naložijo le prvi dve sliki.

Največja prednost uporabe zakasnjene naložnje slik je, da ne obremenjujemo internetne povezave s slikami, ki jih še ni potrebno prenesti s spletnega strežnika (slika 4.9). Ob obisku spletne strani z veliko slikami izven vidnega polja imamo posledično manj nepotrebnih zahtev HTTP, s čimer dosežemo, da se spletne strani naložijo veliko hitreje. Te tehnike so najbolj veseli uporabniki s slabšo internetno povezavo, v nekaterih primerih pa je opaziti tudi zmanjšanje obremenitve spletnega strežnika.

Za namestitev je potrebna sprememba kode HTML povesod, kjer vstavljamo slike. V slikovni atribut `src` vpišemo pot do začasne slike, ki se prenese ob naložitvi dokumenta HTML. Za optimalni učinek delovanja te tehnike je priporočljivo, da je velikost začasne slike čim manjša, pot do nje pa povesod enaka. Tako bo brskalnik za prenos začasne slike potreboval zgolj eno zahtevo, majhna slika pa se bo prenesla hitro. Pot do prave slike vpišemo v atribut `data-original`. Programska koda poskrbi za prepis vrednosti v atribut `src`, ko pride slika v vidno polje. V nadaljevanju zahteva brskalnik od spletnega strežnika željeno sliko, saj se je vrednost v atributu `src` spremenila. Tako se prenos slike začne šele, ko smo prepričani, da je v vidnem polju uporabnika. Možna je tudi nastavitev občutljivosti, kjer določimo razdaljo izven vidnega polja, ob kateri želimo, da se nalaganje slik začne, s čimer je proces skrivanja slik za uporabnika transparenten. Pri vstavljanju slik v HTML dokument moramo biti pozorni, da imajo vse slike definirano širino in višino, bodisi kot atribut ali v slogovni kodi CSS, sicer vtičnik znotraj oblikovanja spletne strani ne deluje pravilno. Zaradi različnih dimenzij med začasno in izvirno sliko lahko pride do motečih preskokov vsebine.

Zavedati se moramo, da se z uporabo zakasnjene nalaganja slik zanašamo na podporo programskega jezika JavaScript v uporabnikovem brskalniku. Danes sicer večina brskalnikov omogoča ta jezik, vendar ga lahko uporabniki izključijo. Da je prikaz slik vedno mogoč, smo vnesli klasično kodo za vstavljanje slike v notranjost bloka `<noscript>`, ki ga podpirajo vsi brskalniki. Za lažje upravljanje sloga vseh slik, ki uporabljajo programsko rešitev Lazy-Load, je priporočljivo da ustvarimo poseben razred in ga določimo slikam, kot prikazuje spodnji primer.

```
1 <script type="text/javascript" src="lazyload.js">
  </script>
2 
3 <noscript>
  </noscript>
```

Koda za uporabo te tehnike je pregledna in učinkovita, njen učinek pa je zelo dobro sprejet pri mobilnih uporabnikih s slabšo internetno povezavo, kajti za delovanje potrebuje dodatno knjižnico JavaScript z velikostjo zgolj 3.4KB.

### 4.3.3 Predpomnilnik brskalnika

Ob prenosu obsežnih spletnih strani se srečujemo z vedno več dostopi do spletnega strežnika. Spletne strani vsebujejo veliko slik, datotek JavaScript ter slogovnih datotek CSS. Vsak posamezen dostop upočasnjuje prikaz spletne strani in obremenjuje tako uporabnikovo internetno povezavo, kot tudi spletni strežnik.

Ob prvem obisku spletne strani in kasneje pa imamo možnost, da uporabnika obvestimo o nekaterih statičnih virih, ki jih ni potrebno vedno znova prenašati, saj se ne spreminjajo tako pogosto. Komunikacijski protokol HTTP/1.1 omogoča spletnim brskalnikom lokalno predpomnjenje statičnih virov, ki se ne spreminjajo med brskanjem po spletni strani. Če nastavimo datum poteka ali največjo starost v glavah HTTP za statične elemente spletne strani, brskalniku povemo, da naj sredstva, ki so že bila predhodno prenesena, naloži iz lokalnega diska in ne preko omrežja. S to tehniko spletnim obiskovalcem zmanjšamo dostopni čas do statičnih elementov, ker so le ti preneseni iz brskalnikovega predpomnilnika. Obenem uporabniki tudi manj obremenjujejo svojo internetno povezavo, kar posledično privede do hitrega nalaganja strani ter do manjše obremenitve spletnega strežnika.

Starejši spletni brskalniki so za uporabo svojega predpomnilnika zahtevali v odgovoru spletnega strežnika željo po predpomnjenju, ki mora biti zapisana v glavi HTTP. Novejši spletni brskalniki uporabljajo heurističen pristop za odločanje pri predpomnjenju sredstev, ki v glavah HTTP nimajo izrecno določenega datuma poteka.

Če želimo optimizirati spletne strani in uporabiti vse koristi predpomnjenja pri vseh spletnih brskalnikih, se priporoča, da v konfiguracijo strežnika za statične vire izrecno nastavimo datume poteka, ki jih nato spletni strežnik pošlje brskalniku ob odgovoru v glavi HTTP. V kategorijo statičnih virov spadajo slike, datoteke JavaScript, slogovne datoteke CSS in druge binarne objektno datoteke (medijske datoteke, dokumenti PDF, datoteke Flash ...). Nekatere velike spletne strani, zaradi zmanjšanja obremenitve spletnega strežnika in množice uporabnikov, uporabljajo zahtevo po predpomnjenju tudi za nestatične datoteke HTML, vendar le tem poteče čas predpomnjenja že v nekaj minutah.

Protokol HTTP ponuja dva načina, s katerima lahko spletni strežnik obvesti brskalnik o času predpomnjenja.

```
1 Cache-Control: max-age=604800
2 Expires: Sun, 04 Aug 2013 20:00:00 GMT
```

Pri prvem načinu definiramo čas v sekundah, pri drugem pa natančen datum, ko mora spletni brskalnik znova zahtevati vir iz spletnega strežnika. Ker oba načina omogočata isti rezultat, lahko v konfiguraciji spletnega strežnika omogočimo le enega.

Konfiguracija strežniških nastavitev je možna, četudi nimamo dostopa do spletnega strežnika in sicer znotraj datoteke `.htaccess`, ki jo ustvarimo ali dopolnimo v mapi za katero želimo uveljaviti spremembe.

```
1 ExpiresActive On
2
3 ExpiresByType text/css "access plus 1 week"
4 ExpiresByType image/png "access plus 1 week"
5 ExpiresByType image/jpeg "access plus 1 week"
6 ExpiresByType image/gif "access plus 1 week"
7 ExpiresByType image/svg+xml "access plus 1 week"
8 ExpiresByType application/x-javascript "access plus 1
   week"
```

V zgornji konfiguraciji smo omogočili ter nastavili čase poteka za nekaj statičnih virov, ki se ne spreminjajo tako pogosto. Vrste virov moramo navesti v obliki zapisa MIME [33].

#### 4.3.4 Stiskanje vsebin z metodo Gzip

Čas, ki je potreben za prenos podatkov po omrežju od zahtevka HTTP ter do njegovega odgovora, se lahko bistveno zmanjša z odločitvami spletnih oblikovalcev. Ker ne vemo kakšno pasovno širino imajo končni uporabniki internetnih strani, stremimo k temu, da jih s prenosom podatkov čim manj obremenjujemo. Stiskanje vsebine zmanjšuje čas prenosa, zaradi zmanjšanja velikosti podatkov, ki jih je potrebno prenesti skozi omrežje.



Komunikacijski protokol HTTP/1.1 spletnim brskalnikom omogoča, da v glavi svoje zahtevke sporočijo ali želijo odgovor prejeti v stisnjeni obliki ter katere metode stiskanja podpirajo.

```
1 | Accept-Encoding: gzip, deflate, sdch
```

Spletni strežnik iz glave zahtevke HTTP razbere željo odjemalca po stiskanju vsebine. Če spletni strežnik omogoča katero iz med ponujenih metod stiskanja, bo v svojem odgovoru HTTP o tem obvestil spletnega uporabnika.

```
1 | Content-Encoding: gzip
```

Gzip je najbolj priljubljen in učinkovit način stiskanja vsebine, ki je trenutno na voljo. Razvit je bil leta 1992 v projektu GNU in standardiziran z dokumentom RFC 1952 [28]. Metoda Gzip za stiskanje vsebine brez izgube uporablja algoritem Deflate, ki nam omogoča zmanjšanje prenosa s spletnega strežnika tudi do 70% in več (slika 4.10). Njegova učinkovitost se razlikuje predvsem glede na podatke, ki jo želimo stisniti. Več kot imamo enakih znakov in pa ponavljajočih se vzorcev, boljša je učinkovitost stiskanja. Kljub temu, da vsi priljubljeni spletni brskalniki že dolgo podpirajo tehniko stiskanja Gzip, je še vedno mnogo spletnih strani, ki te možnosti ne izkoriščajo.

 index.php /gzip	GET	200 OK	text/html	8.5 KB 598 KB
 jquery.js /gzip	GET	200 OK	application/x-javascript	16.0 KB 52.8 KB
 modernizr.js /gzip	GET	200 OK	application/x-javascript	8.2 KB 27.8 KB
 style.css /gzip	GET	200 OK	text/css	1.9 KB 5.3 KB
 testni-svg.svg /gzip	GET	200 OK	image/svg+xml	12.8 KB 30.7 KB

Slika 4.10: Izkoristek uporabe tehnike stiskanja Gzip je opazen v zadnjem stolpcu.

Metodo stiskanja Gzip smo omogočili na spletnem strežniku znotraj datoteke `.htaccess`, ki jo ustvarimo ali dopolnimo v mapi za katero želimo uveljaviti spremembe.

Nepoznane vrste virov moramo prehodno definirati v obliki zapisa MIME [33], kot prikazuje spodnji primer konfiguracije.

```
1 <IfModule mod_mime.c>
2   AddType application/x-javascript .js
3 </IfModule>
4 <IfModule mod_deflate.c>
5   AddOutputFilterByType DEFLATE text/html text/css
6     application/x-javascript image/svg+xml
7 </IfModule>
```

### 4.3.5 Orodje Modernizr

Večina današnjih spletnih brskalnikov že podpira prednosti novih tehnologij jezika HTML5 in CSS3. Ob vseh prednostih, ki jih ponujajo sodobne spletne tehnologije in naprave naletimo razvijalci spletnih strani na pogosto težavo in sicer na slabo podporo starejših spletnih brskalnikov ter nezanesljivo informacijo o zmožnostih naprav.

Programska rešitev Modernizr je majhna knjižnica funkcij napisana v programskem jeziku JavaScript, s pomočjo katerih lahko izvemo ali brskalnik, ki ga uporabljamo že omogoča določeno napredno tehnologijo. Funkcije za detekcijo tehnologij predstavljajo bolj zanesljivo informacijo o zmožnosti brskalnika, kot pa je prepoznavanje zmožnosti iz uporabniškega agenta, ki si ga lahko uporabniki zaradi raznih razlogov spremenijo. Modernizr ponuja teste za preko 40 različnih funkcij, znotraj katerih je opravljen test zmožnosti brskalnika [35].

Na spletni strani orodja Modernizr je na voljo generiranje lastne knjižnice po meri, kjer lahko določimo, katere od funkcij detekcije tehnologij bomo potrebovali. S premišljeno izbiro uporabljenih funkcij in s stiskanjem generirane knjižnice lahko ogromno privarčujemo na velikosti datoteke. Vključevanje knjižnice v dokument HTML se izvede z sledečo kodo.

```
1 <script type="text/javascript" src="modernizr.js">
2   </script>
```

Orodje Modernizr ob nalaganju strani izvede vse teste. Rezultate zapiše v obliki imen razredov znotraj značke `<html>` na začetku dokumenta HTML.

```
1 <html class="js canvas webgl no-touch geolocation
   draganddrop websockets boxshadow textshadow opacity
   cssanimations csscolumns cssgradients no-
   cssreflections csstransforms csstransforms3d
   csstransitions video audio localstorage
   sessionstorage svg">
```

Iz seznama rezultatov testiranja razberemo imena razredov, ki nam povedo, katere od funkcij so podprte v brskalniku in katere ne. Kot vidimo je v seznamu rezultatov razred `no-touch`, kar pomeni, da naprava ne podpira zaslona na dotik. Znotraj slogovne kode CSS3 lahko sedaj določimo, kaj se zgodi v primeru, da uporabnik obiše spletno stran z zaslonom na dotik. V spodnjem primeru smo izkoristili informacijo o uporabniškem zaslonu za spremembo višine vnošnega polja ter gumba iskalnika, kar bo omogočilo lažjo uporabo pri tovrstnih obiskovalcih.

```
1 <html class="js touch">
2   <style>
3     .touch input { height: 45px; }
4   </style>
5   <input type="text" value="Iskalnik" />
6   <input type="submit" value="Išči" />
7 </html>
```

Orodje prav tako ustvari objekt z imenom `Modernizr`, ki vsebuje rezultate vseh funkcij detekcije, do katerega lahko dostopamo s skriptnim jezikom JavaScript. V spodnjem primeru smo izkoristili informacijo o zmožnosti prikazovanja vektorskih slik SVG, ki nam omogoča prilagoditev prikaza nadomestnih slik PNG pri vseh brskalnikih, kjer vektorske slike niso podprte [13].

```
1 <script src="jquery.min.js"></script>
2 <script type="text/javascript">
3 if(!Modernizr.svg) {
4   $('img[src*="svg"]').attr('src', function() {
5     return $(this).attr('src').replace('.svg', '.png');
6   });
7 }
8 </script>
```

Orodje Modernizr omogoča detekcijo podpore tudi ostalih naprednih funkcij slogovnega jezika CSS3, metode vleci in spusti, HTML5 video in avdio, možnost geološke lokacije, podporo spletnih vtičnikov ter mnogo drugih sodobnih funkcij.

Razvijalcem spletnih strani je s tem orodjem omogočena ključna informacija pri ugotavljanju podpore nekaterih novih tehnologij v uporabnikovem brskalniku, kar je zelo zaželeno pri metodi progresivnega oblikovanja. Pri spletnih brskalnikih, kjer napredne tehnologije niso podprte, se moramo odpraviti določenim funkcionalnostim. Z uporabo orodja Modernizr lahko lažje prilagodimo odzivno spletno stran tako, da bo ustrezala zmožnostim spletnih brskalnikov in naprav.

### 4.3.6 Vektorske slike SVG

Slike SVG so močno orodje za prikaz vektorskih slik v brskalniku. Značilnosti vektorskih slik je, da jih lahko povečujemo na katerokoli velikost, brez izgube kvalitete. Za shranjevanje osnovnih lastnosti kot so točke, krivulje, oblike, pisave, gradienti, filtri, skripte in animacije uporablja format XML. Velikost datotek SVG je manjša v primerjavi s formati shranjevanja rastrskih slik.

Slike SVG ponujajo odlično lastnost resolucijske neodvisnosti za prikazovanje grafike na spletu, kar je želja vseh spletnih oblikovalcev pri izdelavi odzivnih spletnih strani. Tehnologija vektorskih slik SVG na spletu ni nova, vendar vključevanje tovrstnih slik v spletne strani še ni zelo razširjeno. Največji vzrok neuporabe je drugačen pristop dela z vektorskimi slikami, na katerega oblikovalci še niso vajeni.

Vsi današnji spletni brskalniki podpirajo prikaz vektorskih slik SVG. Težave s prikazom imajo le spletni brskalniki, starejši od različice Internet Explorer 9 ter brskalniki na operacijskih sistemih starejših od različice Android 3 [23]. Za ustrezen prikaz v spletnih brskalnikih, ki ne podpirajo vektorskih slik SVG, smo opisali alternativno rešitev z uporabo orodja Modernizr v poglavju 4.3.5.

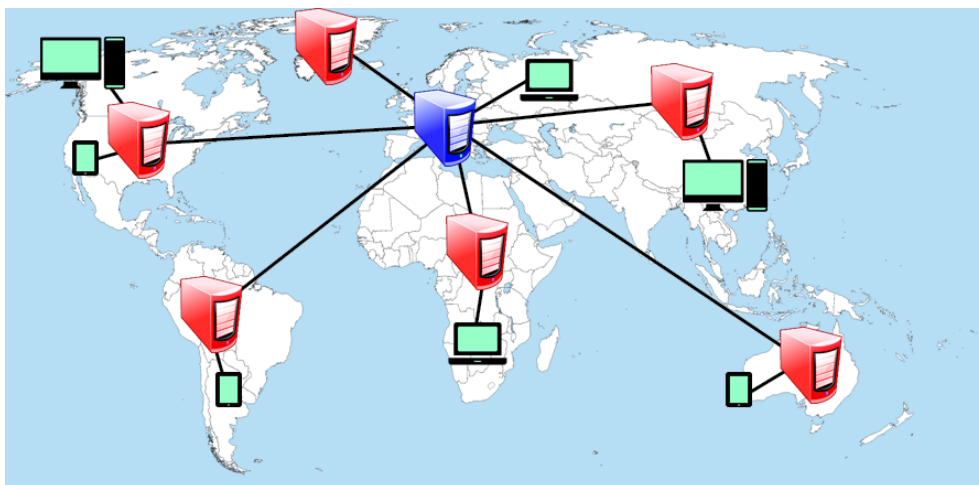
Vektorske slike SVG so idealne za izdelavo kompleksnih uporabniških vmesnikov, navigacijskih ikon, logotipov, grafov, zemljevidov, vektorskih ilustracij, vzorcev in ozadij spletnih strani. Na prvi pogled slike SVG ne delujejo dosti manjše od slik v rasterskem formatu PNG, GIF ali JPG. Ker so slike SVG v formatu XML in na splošno zajemajo zelo omejen nabor znakov, jih lahko zlahka skompresiramo oz. stisnemo, preden pridejo v uporabo na splet [39]. V praksi pomeni, da bo stisnjena datoteka nekajkrat manjša (tudi do 70 odstotkov) od izvirne datoteke. Formati rastrskih slik, kot so PNG, GIF in JPG pa že uporabljajo določeno metodo kompresije, zato jih ne moremo več izrazito stisniti, brez slabšanja kvalitete slike.

Z uporabo vektorskih slik SVG izkoriščamo njihovo resolucijsko neodvisnost, zaradi katere ne potrebujemo več shranjevati posameznih različic enakih slik na spletnem strežniku, kot je to potrebno pri uporabi rasterstskih slik. Vektorske slike SVG se ustrezno prilagodijo tudi na zaslonih Retina, obenem pa za ta učinek ni potrebno preko omrežja prenesti nič več podatkov. Vstavljanje vektorskih slik v dokument HTML je identično, kot vstavljanje ostalih slik.

### 4.3.7 Sistem dostave s strežniki CDN

Za izboljšanje dostopnih časov moramo zagotoviti, da se množice virov, ki jih vključujemo v spletno stran, ne prenašajo iz spletnih strežnikov s slabimi dostopnimi časi. Geografske razdalje med spletnim strežnikom in odjemalci predstavljajo težavo velikih dostopnih časov, saj morajo podatki potovati daleč in skozi ogromno omrežne opreme.

Tehnika CDN (angl. Content Delivery Network) je povezan sistem različnih spletnih strežnikov po svetu z isto vsebino, kar omogoča njeno dostavljanje številnim uporabnikom z izjemno hitrostjo, saj se vsebina prenaša iz strežnika, ki je posameznemu uporabniku najbližji (slika 4.11). Vsi strežniki CDN po svetu se sinhronizirajo s centralnim strežnikom, kar je pogoj za zagotavljanje dostavljanja identične vsebine vsem odjemalcem.

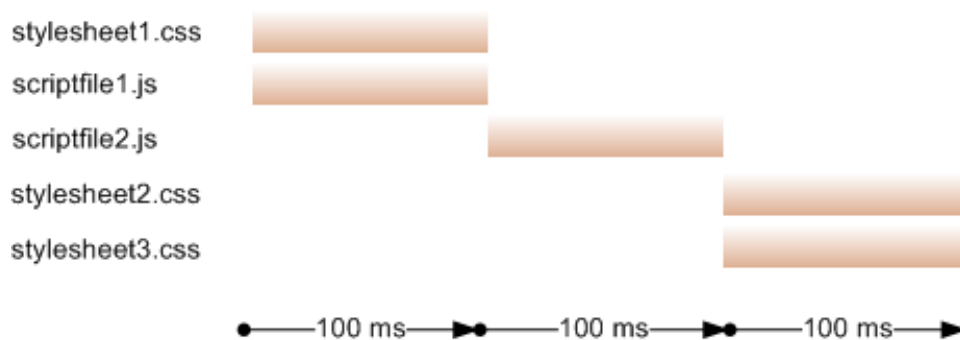


Slika 4.11: Prikaz strukture omrežja spletnih strežnikov CDN razporejenih po celem svetu.

Te tehnike zagotavljanja hitrih dostopnih časov se poslužujejo predvsem velike spletne strani, z ogromno slikami in video vsebinami, do katerih množično dostopajo uporabniki iz celega sveta. Za majhna podjetja so stroški vzpostavitve lastnih strežnikov CDN precej visoki, zato se lahko poslužujemo tudi posameznih ponudnikov teh storitev, kot je denimo *www.cdn77.com*, kjer imajo že izgrajeno omrežje spletnih strežnikov po svetu in ponujajo tovrstne storitve po ugodnih cenah.

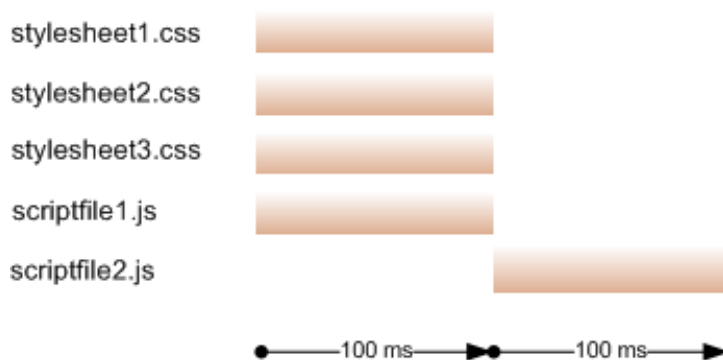
### 4.3.8 Vrstni red vključevanja virov

Za hiter prikaz spletne strani je zelo pomemben vrstni red vključevanja programske kode JavaScript in slogovne kode CSS. Številni spletni brskalniki onemogočajo prenos ostalih virov, ki so v dokument HTML vključeni kasneje kot datoteke JavaScript. Kot prikazuje slika 4.12, se prenos ostalih virov začne šele, ko je programska koda JavaScript v celoti prenesena do odjemalca, kar povzroča daljše čase nalaganja spletnih strani.



Slika 4.12: Prikaz prenosa virov pri mešanem vrstnem redu [34].

Po drugi strani pa opažamo vzporeden prenos, če so ostali viri že v fazi prenosa, pred zahtevo datoteke JavaScript (slika 4.13). Zaradi teh lastnosti spletnih brskalnikov vključujemo datoteke JavaScript na konec dokumenta HTML, tik pred značko `</body>`, slogovno kodo CSS pa vključimo v glavi `<head>`, kar nam bistveno izboljša čas nalaganja spletne strani.



Slika 4.13: Prikaz prenosa virov v optimiziranem vrstnem redu [34].

Pri vključevanju virov moramo biti tudi pazljivi, da ne uporabljamo preusmeritev, ki so za uporabnike s slabšo internetno povezavo časovno zelo potratne in da za prenos virov porabimo čim manj zahtev HTTP. Zaradi tega razloga smo se odločili za združevanje programskih kod iste vrste v eno datoteko.

### 4.3.9 Hitro odzivanje povezav pri mobilnih napravah

Z množico opisanih optimizacijskih tehnik dosežemo, da se spletna stran prenese hitro do odjemalca, še vedno pa njeno delovanje zavirajo določene lastnosti naprav z zasloni na dotik. Težava tovrstnih naprav je, da njihovi spletni brskalniki čakajo približno 300 ms od trenutka, ko se obiskovalec dotakne gumba do izvršitve dogodka [3]. Razlog za ta zamik je čakanje brskalnika na morebiten dvojni dotik uporabnika. Oblikovalci in razvijalci odzivnih spletnih strani vemo, da v večini med brskanjem po spletnih straneh ne potrebujemo hitrega dvojnega dotika in da čas zakasnitve ne prinese najboljše uporabniške izkušnje, zato se trudimo tovrstni čas odstraniti.

Z uporabo knjižnice `fastbutton.js` [25] dosežemo, da odjemalcem z zasloni na dotik ni potrebno več čakati na zakasnitev 300 ms.

Omenjeno knjižnico vključimo na konec dokumenta poleg knjižnice jQuery, ki jo potrebujemo za delovanje. V telo dokumenta `<body>` vstavimo ime funkcije `initFastButtons`, ki se izvede ob naložitvi spletne strani ter identifikator `fastclick`, kot prikazuje spodnji primer.

```
1 <body onload="initFastButtons();" id="fastclick">
2   <!-- množica povezav brez zakasnitve 300 ms -->
3   <script type="text/javascript" src="fastclick.js">
4     </script>
5   <script type="text/javascript" src="jquery.js">
6     </script>
7 </body>
```

Ta koda nam omogoča hitro akcijo gumbov in povezav znotraj celotne spletne strani.

V primeru, da se želimo izogniti zakasnitvam le na določenem delu spletne strani, ovijemo zelene povezave znotraj značke `<span>`, kateri določimo identifikator `fastclick`, kot prikazuje spodnji primer.

```
1 <body onload="initFastButtons();">
2   <!-- množica povezav z zakasnitvijo 300 ms -->
3   <span id="fastclick">
4     <!-- množica povezav brez zakasnitve -->
5   </span>
6   <script type="text/javascript" src="fastclick.js">
7     </script>
8   <script type="text/javascript" src="jquery.js">
9     </script>
10 </body>
```

Sodobne tehnologije nam omogočajo zagotoviti izvajanje vseh povezav in gumbov brez zakasnitev, s čimer zabrišemo razliko med spletnimi ter domodrodnimi aplikacijami na mobilnih napravah. Uporabniška izkušnja med brskanjem po spletni strani je tako izboljšana, saj ima odjemalec občutek hitrega odzivanja spletne strani.

#### 4.3.10 Zmanjšanje velikosti datotek CSS in JS

Pri vključevanju datotek slogovne kode CSS in programske kode JavaScript moramo stremeti k temu, da je njihova velikost čim manjša. To lahko dosežemo tudi z odpravo nepotrebnih znakov iz kode (presledki, nove vrstice, komentarji). Za ta korak se odločimo tik pred objavo spletne strani, saj je preglednost stisnjene kode za razvoj spletne strani bistveno slabša. Za sam proces odprave nepotrebnih znakov smo uporabili priljubljene orodji na spletnih mestih *jscompress.com* ter *cssminifier.com*, ki pomanjšajo velikost kode zelo hitro in učinkovito. S to metodo bistveno pripomoremo k času prenosa tovrstnih datotek, kar posledično izboljša čas, ki je potreben za prikaz celotne spletne strani.



## Poglavje 5

# Implementacija odzivne spletne trgovine

Odločili smo prenoviti spletno stran podjetja Vini d.o.o. iz Mokronoga, ki se ukvarja s prodajo, tiskom in vezenjem promocijskega tekstila. Podjetje se s tovrstno dejavnostjo uspešno ukvarja že od leta 1998. V podjetju se zavedajo porasta različnih mobilnih naprav, zato si odločno želijo prilagoditi njihovo spletno stran in ponuditi svoje storitve uporabnikom na najrazličnejših napravah v obliki spletne trgovine.

### 5.1 Delovni tok oblikovanja

V predhodnih poglavjih smo podrobno raziskali področje odzivnega spletnega oblikovanja ter predstavili ključne značilnosti in omejitve spleta, na katere moramo biti pozorni pri izdelavi spletnih strani. Opisali smo tudi tehnologije, ki nam omogočajo nov pristop k izdelovanju spletnih strani za množico uporabnikov, ki brskajo po spletu z najrazličnejšimi napravami. Oblikovalci svoje delo opravljamo za množico različnih naprav s svojimi posebnostmi, zato je nujna tudi prilagoditev delovnega toka implementacije [5, 10]. V tem poglavju se bomo osredotočili na drugačen pristop oblikovanja odzivnih spletnih strani in na nekaterih mestih poudarili razlike od klasičnega oblikovanja.

### **5.1.1 Popis elementov spletne strani**

Zelo pomemben korak na samem začetku projekta je popis elementov, kjer moramo od naročnika izvedeti čim več informacij o njegovih željah. Pogost pojav je tudi, da vsebino črpamo iz predhodne spletne strani naročnika. Naša naloga je presoditi in se odločiti, katere vsebine bomo uporabili in katere ne ter ustvariti seznam elementov za celotno spletno stran. Ta seznam lahko vsebuje poleg besedila tudi slike, obrazce, gumbe, zemljevide in ostale dodatke spletne strani.

Na podlagi obstoječe spletne strani ter želj naročnika smo se odločili za seznam naslednjih pomembnih elementov spletne trgovine:

1. logotip podjetja,
2. navigacija strani,
3. iskalni obrazec,
4. kontaktni podatki,
5. galerija premikajočih slik,
6. vrstica lokacije uporabnika na strani,
7. slikovni seznam kategorij artiklov,
8. navigacija po kategorijah artiklov,
9. predstavitev artikla,
10. predstavitev objave,
11. obrazec za naročilo,
12. obrazec za povpraševanje,
13. obrazec za prijavo na e-novice,
14. prostor za obvestila ter
15. prostor za socialna omrežja.

S seznamom pomembnih delov spletne strani se lahko oblikovalci lažje organiziramo pri opredelitvi strukture strani, kjer vsi elementi iz seznama dobijo svoje mesto v tekoči mreži postavitve.

### 5.1.2 Oblikovanje besedil

Vsebina je najpomembnejša, saj uporabniki spletno stran obiskujejo ravno zaradi nje, zato potrebujemo za nadaljevanje procesa pregledati vsa besedila, ki jih imamo na voljo. Pri naročniku, kjer prenavljamo staro spletno stran si lahko pomagamo že z njeno obstoječo vsebino.

Osredotočimo se zgolj na oblikovanje besedil, za katere moramo biti prepričani, v kakšni obliki jih naročnik želi predstaviti. Oblikovano besedilo mora biti berljivo in smiselno strukturirano, kot bo tudi na končni spletni strani. Vemo, da se uporabniki pogosto izogibajo dolgih besedil, zato jih poskušamo na tem mestu čim bolj strniti. Oblikujemo ga v obliki HTML, ki nam omogoča lažje vstavljanje v izdelano strukturo tekoče postavitve elementov.

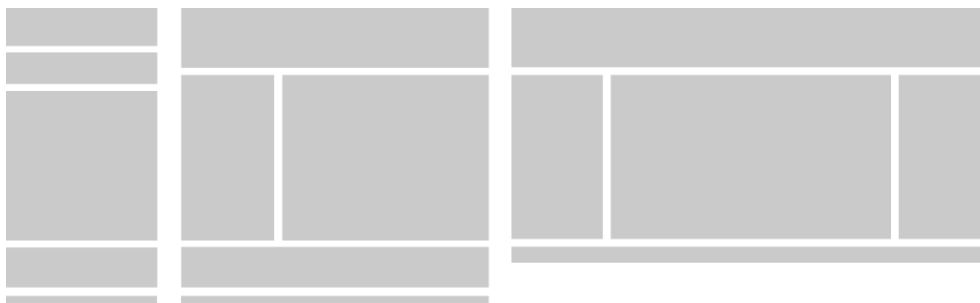
### 5.1.3 Struktura tekoče postavitve elementov

Vse elemente iz seznama v poglavju 5.1.1 bomo na čim lažji način predstavili znotraj mreže tekoče postavitve elementov. V fazi načrtovanja tekoče postavitve elementov moramo pozorno preučiti cilje, ki jih mora spletna stran zadostiti.

Pomembno je zavedanje kje, kdaj in zakaj bodo uporabniki uporabljali to spletno trgovino na mobilnih napravah ter katere so ključne informacije, ki jih potrebujejo uporabniki v trenutku, ko pridejo na spletno mesto preko mobilnih naprav. Pri tovrstnem načrtovanju moramo za določitev mesta elementov v mreži upoštevati želje naročnika, zmožnosti naprav, pripravljeno vsebino ter dobre prakse izdelovanja spletnih strani.

Preprost in časovno hiter način je risanje skic mreže elementov, kjer lažje dobimo vtis o končni strukturi spletne strani. Pri risanju skic se skušamo čim bolj osredotočiti na strukturo spletne strani za zaslone treh najbolj priljubljenih tipov naprav (pametni telefoni, tablični računalniki in namizni računalniki).

Že osnovno znanje jezikov HTML in CSS3 nam omogoča, da kreiramo mrežo tekoče postavitve elementov (slika 5.1) s preprostimi bloki `<div>`, ki jih hitro in brez težav nadzorujemo z medijskimi poizvedbami znotraj slogovne kode CSS3.



Slika 5.1: Struktura tekoče postavitve elementov za zaslone treh priljubljenih tipov naprav.

Prvi osnutki strukture elementov spletne strani nam že omogočajo preprosto demonstracijo, ki jo lahko pokažemo naročniku, saj bo na primeru lažje razumel delovanje odzivnih spletnih strani. Odločili smo se, da naročniku predstavimo osnutke strukture elementov za štiri različne tipe spletne strani:

- strukturo prve strani,
- strukturo objave,
- strukturo artikla ter
- strukturo naročila.

Ob demonstraciji se lahko nam ali naročniku vedno pojavi sveža ideja, kako so besedila in slike umeščene znotraj elementov mreže, zato tudi v tem koraku ne smemo pozabiti na risanje skic.

Od naročnika smo skušali izvedeti dodatne želje glede oblikovanja in funkcionalnosti obrazca za naročilo artikla. Z risanjem skic smo pri naročniku prišli do ugotovitve, kaj vse mora obrazec za naročilo vsebovati. Želja naročnika je bila, da si uporabniki lahko sami izračunajo ceno posameznega artikla, katera je odvisna od barve, velikosti ter števila naročenih artiklov (slika 5.2).

VPC cena brez DDV:	3,97 €
<b>VPC cena z DDV:</b>	<b>4,84 €</b>
MPC cena z DDV:	6,78 €

Barva: rdeča

Velikost:

S  M  L  XL  XXL  3XL

kos

Slika 5.2: Prikaz obrazca za naročilo artiklov.

Proces skiciranja in pripravljanja demonstracij lahko ponavljamo, dokler ni naročnik zadovoljen s strukturo spletne strani. Pogosto imajo naročniki vedno nekaj popravkov, zato je pomembno, da za naše osnutke ne porabimo preveč časa.

Prav iz tega razloga je uporaba programskih orodij za grafično oblikovanje (primer: Adobe Photoshop) pri oblikovanju odzivnih spletnih strani pogosto nesmiselna in zamudna, saj moramo že najmanjše popravke upoštevati v vseh različnih dokumentih, ki navadno predstavljajo prikaz za posamezne tipe naprav. S sodelovanjem naročnika v procesu načrtovanja strukture postavitve elementov lahko zagotovo pričakujemo manj naknadnih popravkov.

#### 5.1.4 Oblikovanje v spletnem brskalniku

Pomembno je, da preizkušamo našo mrežo tekoče postavitve elementov z vstavljenim besedilom ter morebitnimi slikami v brskalniku. Tekstovna besedila imajo lastnost prilagodljivosti širini brskalnika, kar nam omogoča hiter preizkus, kakšen je prikaz vsebine na različno velikih zaslonskih ločljivostih.

Tekočo mrežo postavitve elementov smo med oblikovanjem preizkušali s spreminjanjem širine brskalnika Chrome, ki odlično služi tudi kot razvijalno orodje. Preizkušanje odzivnosti spletne strani je priljubljeno tudi z uporabo spletnih mest kot sta recimo *www.responsivetest.net* ali *www.deviceponsive.com*, kjer naročniku še lažje predstavimo funkcionalnosti odzivnih spletnih strani.

### 5.1.5 Oblikovanje pri različnih prelomnih točkah

Ko so vse skice potrjene s strani naročnika in imamo informacije o njegovih željah, se lahko lotimo obširnejšega oblikovanja spletne strani. Pomembno je, da začnemo oblikovati najprej za najmanjše zaslone, kajti le to nas prisili k osredotočenosti prikaza za mobilne naprave [19]. V dizajn spletne strani vključujemo čim manjše datoteke za skrajšanje časa prenosa, porabe baterije in izboljšanje hitrosti ter odzivnosti spletnega mesta. Še boljše je, če v dizajn namesto grafičnih datotek vključujemo uporabo slogovnih tehnik CSS3, kot so preliv, obrobe, sence ter določene tipografije za naslove. V nasprotnem primeru se lahko hitro zgodi, da je spletna stran prevelika in prepočasna.

S prioriteto osredotočenostjo na najmanjše vidno polje pri mobilnih napravah dosežemo, da uporabniku sprva na primeren način prikažemo najpomembnejše elemente spletne strani.

#### Logotip podjetja

Logotip ima prav gotovo visoko prioriteto biti na vrhu spletne strani, ne glede, če do nje dostopamo z najmanjšim zaslonom. Namen logotipa je lažja prepoznavnost podjetja, ima pa tudi komunikativno in reklamno vrednost, saj so pogosto domiselno oblikovani, da bi si ga uporabniki čim hitreje zapomnili.

#### Kontaktne podatke

Drugi element, ki ga želimo umestiti na vrhu spletne strani je prostor za kontaktne podatke. Z visoko postavitvijo le teh smo lahko prepričani, da bo vsak obiskovalec vedel kako stopiti v stik s trgovcem, če bo potreboval dodatne informacije ali pa zgolj pomoč pri naročanju artiklov. Pomembno je, da se podjetje hitro odziva in odgovarja na vprašanja obiskovalcev, saj lahko le tako zadržimo dobro uporabniško izkušnjo uporabnikov spletne strani.

#### Iskalnik vsebine

Za enostavnejše iskanje po katalogu artiklov in ostali vsebini smo v dizajn spletne strani dodelili visoko mesto tudi obrazcu za iskanje. Nekaterim uporabnikom predstavlja to najlažji način za dostop do želenih vsebin. Odločili smo se, da bo obrazec za iskanje privzeto skrit. S tem smo zagotovili, da bo ob

obisku najmanjših mobilnih naprav v vidnem polju še vedno dovolj prostora za prikaz ostale vsebine. Za prikaz iskalnega obrazca je dodeljena dovolj velika in prepoznavna lupa, ki predstavlja simbol za iskanje (slika 5.3).

### Navigacija spletne strani

Navigacija strani mora biti pri odzivnih spletnih straneh prilagodljiva mediju. Obširne in zahtevne navigacije, ki so dobro vidne na velikih zaslonih je nemogoče prikazovati na enak način napravam z majhnimi zasloni. Zavedati se moramo, da naprave z zasloni na dotik nimajo miške, s katerimi bi imeli možnost lebdenja nad določenim elementom, zato večnivojskih menijev, ki zahtevajo miško, za tovrstne naprave ne uporabljamo. Odločili smo se, da so logotip podjetja, kontaktni podatki, ikona za prikaz iskalnega obrazca ter ikona za navigacijo po strani v zgornji vrstici združeni, obenem pa še vedno zelo pregledni.



Slika 5.3: Prikaz logotipa, kontaktnih podatkov, ikone za iskanje ter navigacijo po strani pri napravah z majhnim zaslonom.

Za navigacijo strani smo uporabili priljubljeno ikono, ki je na spletu prepoznana kot navigacijski simbol (slika 5.3). Zaradi prostorske stiske smo se pri manjših zaslonih odločili, da bo navigacija strani privzeto zaprta. Navigacijska ikona je pravzaprav gumb, ki ob izbiri odpre meni, s katerim zavzamemo večji del zaslona (slika 5.4).



Slika 5.4: Prikaz odprtega menija navigacije po spletni strani pri napravah z majhnim zaslonom.

Pri napravah z zaslonom na dotik se za interakcijo uporablja človeški palec, s katerim težko dosežemo natančnost izbire. Izkušnja uporabnika, ki je povzročil nenamerno dejanje na spletni strani zaradi premajhne velikosti izbirnega elementa je zelo neprijetna. Uporabniški vmesnik mora imeti elemente z dovolj veliko površino, ki omogočajo preprosto možnost izbire, brez povzročanja nenamernih dejanj. Za prijetno uporabniško izkušnjo se priporoča velikost izbirnih elementov od 7 mm do 9 mm, za pogosto uporabljene elemente pa tudi več [21].

### Navigacija po katalogu izdelkov

Zaradi preglednosti in velikega števila artiklov smo se odločili za ločeno navigacijo po katalogu artiklov. Naš cilj je, da uporabniku čim bolj poenostavimo iskanje zelenih artiklov, zato smo se odločili za večnivojski padajoči meni, ki za svoje delovanje ne potrebuje motečega osveževanja strani.



Slika 5.5: Prikaz hitrega iskanja zelenih artiklov na majhni mobilni napravi z večnivojskem padajočim menijem.

Kot prikazuje slika 5.5, je v začetku navigacija po katalogu izdelkov na mobilni napravi z majhnim zaslonom zaprta in prostorsko nepotratna. Majhnost omogoča, da so lahko v istem vidnem polju še logotip, kontaktni podatki, iskalnik ter ostala vsebina. Odločitev, do česa želi dostopati uporabnik prepuščamo njemu samemu. Ob izbiri kataloga izdelkov, bo le ta zasedel večino uporabnikovega zaslona, obenem pa mu bo omogočal hiter pregled storitev in enostavno izbiro zelenih artiklov.

### Prikaz artiklov

V seznamu artiklov izpostavimo le najpomembnejše podatke, na katere se obiskovalci ob ogledu množice artiklov največkrat osredotočajo (naziv artikla, teža in vrsta materiala ter cena). Slike morajo biti dovolj velike, da si jih obiskovalci ogledajo, a vseeno ne smejo zavzemati celotnega zaslona, če želimo poleg njih prikazati tudi omenjene pomembne podatke. Pri vstavljanju slik smo pozorni tudi na strežniško prilagoditev, opisano v poglavju 4.2.2, kjer dostavljamo napravam prilagojene velike slike.

Item Name	Weight (Teža)	Material (Material)	Price (že od)
GILDAN 64400L	150g/m <sup>2</sup>	100% bombaž	4,75 €
GILDAN 64400L	150g/m <sup>2</sup>	100% bombaž	4,75 €
B&C Angel LSL	145g/m <sup>2</sup>	100% bombaž	5,45 €
B&C Angel LSL	145g/m <sup>2</sup>	100% bombaž	5,45 €
B&C Pink Slub	180g/m <sup>2</sup>	93% bombaž, 7% elasthan	6,83 €
B&C Classic	120g/m <sup>2</sup>	100% bombaž	5,99 €
KARIBAN K321 Cindy	180g/m <sup>2</sup>	100% bombaž	5,23 €
KARIBAN K328 Carla	180g/m <sup>2</sup>	100% bombaž	8,80 €

Slika 5.6: Prikaz seznama artiklov spletne strani na pametnih telefonih in tabličnih računalnikih.

Ko smo zadovoljni s prikazom spletne strani pri najmanjšem zaslonu, postopoma širimo brskalnik do te mere, da nam dizajn vizualno ni več všeč in tam je potrebna točka preloma [5]. Na tem mestu lahko s pomočjo medijskih poizvedb v slogovni kodi CSS3 določimo drobne spremembe, s katerimi bo dizajn znova vizualno ustrezen. V točkah preloma se lahko poleg drobnih sprememb pisav in velikosti elementov odločimo tudi za drugačno razporeditev tekoče mreže elementov (slika 5.6). Pri uporabi prelomnih točk moramo nujno izhajati iz vsebine in ne iz naprave. Šele z vstavljenjo vsebino lahko presodimo, kdaj je spletna stran za oko prijazna in kdaj ne.

Na trg vsak dan prihajajo nove naprave z različnimi zasloni, zato se na trenutno stanje ni priporočljivo zanašati. Seveda to ne pomeni, da bolj razširjenim zaslonskim ločljivostim ne posvetimo dodatne pozornosti pri definiranju prelomnih točk.

```
1 @media screen and (min-width: 768px) {  
2 }
```

S točkami preloma, ki nam jih omogoča slogovni jezik CSS3, skušamo vedno prikazati spletno stran čim bolje, ne glede na širino zaslona ter njegovo orientacijo. Pomagamo si tudi z medijskimi poizvedbami, ki v svojem pogoju vključujejo gostoto slikovnih točk. V spodnjem primeru smo vključili tri različne pogoje za detekcijo gostote slikovnih točk naprave, ki imajo isto funkcijo, vendar jih spletni brskalniki različno podpirajo. Ta pristop nam omogoča različne prilagoditve dizajna za naprave z zasloni Retina. Obenem smo s to medijsko poizvedbo ciljali na tovrstne naprave, ki so v pokončnem položaju.

```
1 @media screen and (-webkit-min-device-pixel-ratio:  
2 2.0),  
3 screen and (-o-min-device-pixel-ratio: 2/1),  
4 screen and (min-resolution: 192dpi),  
5 screen and (orientation: portrait) {
```

Kljub temu, da prioritarno prilagajamo dizajn spletne strani vsebini in ne napravam, nam medijske poizvedbe omogočajo, da lahko posvetimo posebno pozornost tudi najpogostejšim ločljivostim zaslonov.

The screenshot shows the Vini website interface. At the top left is the Vini logo (a stylized bird) and the text 'vini'. To the right is a search bar labeled 'Iskalnik...'. Below the logo and search bar are navigation links: 'NAŠA PONUDBA', 'AKCIJA', 'E-KATALOG', 'CENIK', 'POVPRAŠEVANJE', and 'PODJETJE'. Contact information 'T: 07 / 34 99 226' and 'E: info@vini.si' is also present.

On the left side, there is a vertical menu titled 'Katalog izdelkov' with the following categories: 'Majice', 'Polo majice', 'Puloverji', 'Jakne', 'Flis jope', 'Brezrokavniki', 'Srajice', 'Hlače', and 'Drug promocijski material'. Each category has a downward arrow.

The main content area displays six clothing items in a 2x3 grid:

Item Name	Weight	Material	Price
GILDAN 64400L	150g/m <sup>2</sup>	100% bombaž	že od 4,75 €
B&C Angel LSL	145g/m <sup>2</sup>	100% bombaž	že od 5,45 €
B&C Pink Slub	180g/m <sup>2</sup>	93% bombaž, 7% elastan	že od 6,83 €
B&C Classic	120g/m <sup>2</sup>	100% bombaž	že od 5,99 €
KARIBAN K321 Cindy	180g/m <sup>2</sup>	100% bombaž	že od 5,23 €
KARIBAN K328 Carla	180g/m <sup>2</sup>	100% bombaž	že od 8,80 €

Slika 5.7: Prikaz seznama artiklov spletne strani na napravah z večjimi zasloni.

Pri večjih zasloneh smo se odločili, da je navigacija po strani in katalogu izdelkov že odprta in s tem zelo pregledna na namiznih računalnikih (slika 5.7). Zaradi večjega vidnega polja pri velikih zasloneh, smo se odločili na tovrstnih napravah prikazati tudi večji logotip ter dobro viden iskalnik v glavi spletne strani.

V določenih točkah prelomov, smo za prikaz seznama artiklov pri večjih zasloneh določili strukturno spremembo tekoče mreže elementov in s tem omogočili večje število artiklov prikazanih v eni vrstici. Pri zaslonski ločljivosti s širino 1600 slikovnih točk je v vrstici seznama artiklov lahko prikazanih kar šest artiklov.

Ključno je, da ob prilagoditvah, ki nam jih omogoča slogovni jezik CSS3 z medijskimi poizvedbami, poleg funkcionalnih sprememb, vedno poskrbimo tudi za lep izgled dizajna. Le tako lahko zagotovimo popoln prikaz spletne strani ne glede na zaslonsko ločljivost naprave obiskovalca.

### 5.1.6 Preizkus na različnih napravah in brskalnikih

Zaradi nekaterih posebnosti posameznih naprav je priporočljivo spletno stran v razvoju preizkušati tudi na dejanskih napravah. Obenem opazujemo tudi hitrost nalaganja spletne strani ter ali omogoča predstavitev vsebine na optimalen način. Le tako smo lahko prepričani, da se bo spletna stran pri odjemalcu pravilno odzivala in ustrezno prikazala. Spletno trgovino smo preizkusili z različnimi brskalniki na napravah:

- Samsung X-cover,
- Apple iPhone 4,
- Nokia Lumia 920,
- Apple iPad 4,
- Sony Xperia Z Tablet,
- prenosnem računalniku s 17-palčnim zaslonom ter na
- namiznem računalniku s 24-palčnim zaslonom.

#### Vidno polje naprave

Pri preizkusu spletne strani na mobilnih napravah sprva opazimo, da medijske poizvedbe ne delujejo kot bi želeli. Celotna spletna stran je pomanjšana in elementi so videti premajhni. Težava večine mobilnih brskalnikov je, da jim moramo predhodno znotraj glave dokumenta `<head>` določiti vidno polje naprave, sicer bodo našo odzivno spletno stran prikazali kot pomanjšano različico spletne strani namenjeno za velike zaslone.

```
1 <meta name="viewport" content="width=device-width,  
   initial-scale=1, maximum-scale=1">
```

Z zgornjo kodo smo določili, da je vidno polje brskalnika dejanska širina naprave in omejili možnost pomanjševanja ter povečevanja vidnega polja, saj ne želimo imeti horizontalnih drsnikov, kot so jih imele predhodne neprilagojene spletne strani. Z uporabo te vrstice kode, se tekoča postavitve elementov lepo prilagaja tudi na mobilnih napravah, ne glede na orientacijo.

### Podpora medijskih poizvedb CSS3 v brskalnikih Internet Explorer

Med preizkušanjem smo opazili tudi, da brskalnik Internet Explorer različice 8 ter starejši ne omogočajo izvajanje medijskih poizvedb slogovnega jezika CSS3. Zaradi precej visokega deleža uporabe tega brskalnika, smo oblikovalci primorani k izvajanju dodatnih prilagoditev.

```

1 <!--[if lt IE 9]>
2 <script type="text/javascript" src="css3-
   mediaqueries.js "></script>
3 <![endif]-->

```

Za rešitev te težave smo uporabili posebno knjižnico [32], ki nam omogoča nemoteno delovanje pri teh zastarelih brskalnikih. Omenjena knjižnica se prenese do odjemalca le pri ciljnih brskalnikih, zapisanih v pogojnem stavku.

### Poročanje gostote slikovnih točk brskalnika IE 10 Mobile

Med preizkusom spletne trgovine na množici naprav smo opazili težavo pri pametnih telefonih z operacijskim sistemom Windows Phone 8 in privzetim brskalnikom Internet Explorer 10 Mobile, ki vedno poroča o enaki gostoti slikovnih točk, ne glede na to, če je to res ali ne. Omenjeno težavo so pri podjetju Microsoft že priznali in naj bi bila rešena v naslednji različici spletnega brskalnika. Za rešitev te težave smo v spletno stran vključili posebno prilagoditev [11].

```

1 <script type="text/javascript">
2 if (navigator.userAgent.match(/IEMobile\/10\.0/)) {
3     var msViewportStyle = document.createElement("
         style");
4     msViewportStyle.appendChild(
5         document.createTextNode(
6             "@-ms-viewport{width:auto!important}"
7         )
8     );
9     document.getElementsByTagName("head")[0].
10         appendChild(msViewportStyle);
11 }
12 </script>

```

Koda ob ujemanju mobilnega spletnega brskalnika Internet Explorer 10 ustrezno ukrepa z vstavitvijo novega elementa `<style>` v katerem je slogovno pravilo `@-ms-viewport{width:auto!important}`, ki prepíše nastavitev CSS3 o širini zaslona naprave. Ta prilagoditev nam omogoča, da spletna stran deluje brezhibno tudi pri napravah s tovrstnim spletnim brskalnikom.

Na ostalih napravah in spletnih brskalnikih nismo opazili potrebe po dodatnih prilagoditvah, saj se vsebina prikazuje brezhibno. S preizkušanjem spletne strani na dejanskih napravah smo odkrili nekatere pomanjkljivosti spletnih brskalnikov in jih uspešno odpravili.

## 5.2 Rezultati testiranja

Z odzivnim spletnim oblikovanjem nam je uspelo prikazati spletno stran na različnih zaslonih, obenem pa z vrsto optimizacijskih pristopov prilagoditi količino prenesenih podatkov.

Želeli smo ugotoviti, koliko posamezna optimizacijska metoda pripomore naši odzivni spletni trgovini. Odločili smo se, da bomo učinke optimizacij izmerili s pomočjo spletnega brskalnika Chrome in njegovih vključenih orodij za opazovanje omrežnih prenosov. Znotraj teh orodij imamo tudi možnost spreminjanja lastnosti naprave, vključno z ločljivostjo zaslona. Za preizkus smo uporabljali internetno povezavo s pasovno širino 2Mbit/s, kjer je bil povprečni dostopni čas do spletnega strežnika 40 ms. V spodnji tabeli so rezultati testiranja količine in časa prenosa za omenjeno spletno stran pri dveh različnih zaslonskih ločljivostih.

Optimizacijska metoda	Ločljivost 320 x 480	Ločljivost 1920 x 1080
Brez optimizacije	2,7 MB ~ 12,6 s	2,7 MB ~ 15,8 s
Prilagodljive slike	283 KB ~ 1,73 s	346 KB ~ 2,81 s
Zakasnjeno nalaganje slik	198 KB ~ 1,54 s	367 KB ~ 2,41 s
Stiskanje vsebine z Gzip	99 KB ~ 1,18 s	261 KB ~ 1,89 s
Vrstni red CSS in JS	99 KB ~ 907 ms	261 KB ~ 1,63 s
Zmanjšanje CSS in JS	95 KB ~ 704 ms	257 KB ~ 1,49 s
Predpomnilnik brskalnika	3,6 KB ~ 244 ms	3,6 KB ~ 251 ms

Tabela 5.1: Rezultati testiranja količine in časa prenosa celotne spletne strani pri dveh različnih zaslonskih ločljivostih z uporabo opisanih optimizacijskih tehnik.

Kot lahko opazimo iz podatkov v tabeli 5.1, je spletna stran brez optimizacij navadno velika in počasna. K velikosti spletne strani najbolj pripomorejo slike, ki so na spletni strežnik naložene tipično neoptimizirane in zelo velike.

S prvo strežniško optimizacijo smo prilagodili dostavljanje slik primernih za posamezne velikosti zaslonov. Učinek pohitritve spletne strani je v tem koraku najbolj opazen, saj je prenos slik prilagojen zaslonski ločljivosti naprave, kar je za delo urednika vsebine spletne strani popolnoma transparentno.

Zakasnjeno nalaganje slik nam onemogoča prenos tistih slik, ki še niso v vidnem polju. To metodo smo preizkusili na seznamu artiklov, kjer pri najmanjšem zaslonu ne moremo opazovati več kot dva artikla naenkrat. Posledično smo s to metodo zmanjšali število zahtev HTTP za prenos slik, kar je vidno v tabeli rezultatov. Učinek te optimizacije pride še bolj do izraza pri velikem številu slik, ki jih ne moremo prikazati v vidnem polju.

Z optimizacijsko tehniko stiskanja vsebine smo bistveno zmanjšali velikost dokumentov HTML, JavaScript ter CSS, ki poleg slik predstavljajo ogromen delež prenosa virov spletne strani. Znotraj dokumenta HTML smo optimizirali še vrstni red skript JS ter slogovnih datotek CSS. Vse slogovne datoteke smo združili v eno, ki smo jo še dodatno pomanjšali z odstranjevanjem nepotrebnih znakov. Enak postopek smo naredili tudi za skriptne datoteke JavaScript.

Na koncu smo preverili še količino in hitrost prenosa spletne strani z uporabo brskalnikovega predpomnilnika in dosegli zelo dobre rezultate pri ponovnem obisku spletne strani.

Množici odjemalcev z različnimi napravami smo z odzivnim spletnim oblikovanjem ter z uporabo opisanih optimizacijskih tehnik zagotovili optimalen prikaz spletne strani s hitrimi dostopnimi časi, ki so poleg vsebine spletne strani ključ do pozitivne uporabniške izkušnje.



# Poglavje 6

## Sklepne ugotovitve

V diplomskem delu smo ugotovili, da sodobni pristopi izdelave odzivnih spletnih strani zahtevajo izjemno širok spekter znanj. Pri oblikovanju spletnih strani za množico različnih naprav moramo posvečati posebno pozornost zmogljivosti mobilnih naprav, saj imajo le te v primerjavi z namiznimi računalniki bistveno manjše vidne površine ter slabše zmogljivosti procesiranja. Na hitrost prikaza spletne strani bistveno vplivajo tudi same omejitve internetnih povezav, ki so pri mobilnih napravah navadno počasnejše ter dražje.

Naš cilj diplomskega dela je bil podrobno raziskati področje odzivnega spletnega oblikovanja ter izdelati primer spletne trgovine, ki ponuja optimalno izkušnjo uporabnikom na najrazličnejših napravah, kar nam je uspelo doseči z uporabo različnih optimizacijskih tehnik. Z metodo progresivnega izboljševanja spletne strani smo zagotovili dostop do osnovnih funkcionalnosti spletnih vsebin tudi uporabnikom z manj naprednimi napravami oz. brskalniki. S tem smo poskrbeli za ustrezen prikaz spletne strani denimo tudi uporabnikom, ki ne omogočajo spletnih piškotkov ali skriptega jezika JavaScript. Danes se spletni brskalniki z novjšimi različicami počasi izpopolnjujejo in podpirajo posodobljene spletne standarde, kar nam daje upanje, da bo delo spletnih oblikovalcev v prihodnosti vsaj malo bolj poenostavljeno. Implementacije različnih in učinkovitih optimizacijskih metod nam zagotavljajo, da odjemalci preko omrežja prejmejo le najnujnejše podatke, ki so ključni dejavnik za hitre čase nalaganja spletnih strani. Večine opisanih optimizacijskih tehnik bi se morali oblikovalci posluževati že pri dosedanjem oblikovanju spletnih strani, vendar se je potreba za tovrstne pristope izrazito pojavila šele

z oblikovanjem za množico različnih mobilnih naprav, ki so trenutno v porastu. Celotna združitev odzivnega spletnega oblikovanja z optimizacijskimi tehnikami je ključ do dobre uporabniške izkušnje odjemalca.

Vsem spletnim oblikovalcem in razvijalcem, ki se odločajo za pristop k odzivnemu spletnemu oblikovanju svetujemo, da prav tako sprva preučijo njegove prednosti in slabosti ter nato izboljšajo uporabniško izkušnjo s primernimi optimizacijskimi tehnikami, ki najbolj ustrezajo njihovi spletni strani.

# Literatura

- [1] M. Belshe (2010) More Bandwidth Doesn't Matter (Much). Dostopno na: <https://www.belshe.com/2010/05/24/more-bandwidth-doesnt-matter-much/>
- [2] D. Cederholm, CSS3 For Web Designers, New York: A Book Apart, 2010
- [3] R. Fioravanti (2011) Creating Fast Buttons for Mobile Web Applications. Dostopno na: [https://developers.google.com/mobile/articles/fast\\_buttons](https://developers.google.com/mobile/articles/fast_buttons)
- [4] B. Frost (2012) This is the web. Dostopno na: <http://bradfrostweb.com/blog/post/this-is-the-web/>
- [5] S. Hay, Responsive Design Workflow, US: Pearson Education, 2013
- [6] B. Hillen (2012) PC sales to decline in 2012 for the first time in 11 years. Dostopno na: <http://www.slashgear.com/pc-sales-to-decline-in-2012-for-the-first-time-in-11-years-10251339/>
- [7] S. Jobs (2010) Steve Jobs said about Retina Display. Dostopno na: [http://www.firstpost.com/topic/person/steve-jobs-steve-jobs-said-about-retina-display-video-KdX\\_NVYpw7Q-2600-7.html](http://www.firstpost.com/topic/person/steve-jobs-steve-jobs-said-about-retina-display-video-KdX_NVYpw7Q-2600-7.html)
- [8] D. B. Judd , G. Wyszecki, Color in Business, Science and Industry . New York: Wiley-Interscience, 1975
- [9] T. Kadlec (2012) Blame the Implementation, Not the Technique. Dostopno na: <http://timkadlec.com/2012/10/blame-the-implementation-not-the-technique/>

- 
- [10] T. Kadlec, *Implementing Responsive Design*, US: Pearson Education, 2013
- [11] T. Kadlec (2013) *Windows Phone 8 and Device-Width*. Dostopno na: <http://timkadlec.com/2013/01/windows-phone-8-and-device-width/>
- [12] E. Marcotte, *Responsive Web Design*, New York: A Book Apart, 2011
- [13] T. Motto (2013) *Mastering SVG use for a retina web, fallbacks with PNG script*. Dostopno na: <http://toddmotto.com/mastering-svg-use-for-a-retina-web-fallbacks-with-png-script/>
- [14] J. Nielsen (1993) *Response Times: The 3 Important Limits*. Dostopno na: <http://www.nngroup.com/articles/response-times-3-important-limits/>
- [15] K. Schaeffer (2008). *CSS Font-size*. Dostopno na: <http://kyleschaeffer.com/development/css-font-size-em-vs-px-vs-pt-vs/>
- [16] J. P. Titlow (2011) *Top 5 Internet Devices of 2011*. Dostopno na: [http://readwrite.com/2011/12/12/top\\_5\\_internet-connected\\_devices\\_of\\_2011](http://readwrite.com/2011/12/12/top_5_internet-connected_devices_of_2011)
- [17] M. Tuupola (2013) *Lazy Load Plugin for jQuery*. Dostopno na: <http://www.appelsiini.net/projects/lazyload>
- [18] M. Wilcox (2013) *Adaptive Images, Deliver small images to small devices*. Dostopno na: <http://adaptive-images.com/details.htm>
- [19] L. Wroblewski, *Mobile First*, New York: A Book Apart, 2011
- [20] L. Wroblewski (2011) *RESS: Responsive Design + Server Side Components*. Dostopno na: <http://www.lukew.com/ff/entry.asp?1392>
- [21] L. Wroblewski (2010) *Touch Target Sizes*. Dostopno na: <http://www.lukew.com/ff/entry.asp?1085>
- [22] (2013) *Browser Display Statistics*. Dostopno na: [http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp)

- 
- [23] (2013) Can I use SVG. Dostopno na:  
<http://caniuse.com/svg>
- [24] (2013) CSS frameworks. Dostopno na:  
[http://en.wikipedia.org/wiki/CSS\\_frameworks](http://en.wikipedia.org/wiki/CSS_frameworks)
- [25] (2013) Fastbutton.js. Dostopno na:  
<https://gist.github.com/fwebdev/2168307>
- [26] (2010) Google's View of HTML5 In Terms Of SEO and Crawling. Dostopno na:  
<http://www.seroundtable.com/archives/023106.html>
- [27] (2013) Graphic display resolution. Dostopno na:  
[http://en.wikipedia.org/wiki/Graphics\\_display\\_resolution](http://en.wikipedia.org/wiki/Graphics_display_resolution)
- [28] (2013) Gzip. Dostopno na:  
<http://en.wikipedia.org/wiki/Gzip>
- [29] (2013) IDC Tempers Long-Term Tablet Forecast. Dostopno na:  
<http://www.businesswire.com/news/home/20130829005302/en/IDC-Tempers-Long-Term-Tablet-Forecast-Competing-Technologies>
- [30] (2013) IDC Worldwide Mobile Phone Market Forecast. Dostopno na:  
<http://www.idc.com/getdoc.jsp?containerId=prUS24302813>
- [31] (2013) List of displays by pixel density. Dostopno na:  
[http://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density)
- [32] (2013) Make CSS3 Media Queries work in all browsers. Dostopno na:  
<https://css3-mediaqueries-js.googlecode.com/>
- [33] (2013) MIME Types Complete List. Dostopno na:  
<http://reference.sitepoint.com/html/mime-types-full>
- [34] (2012) Minimize round-trip times. Dostopno na:  
<https://developers.google.com/speed/docs/best-practices/rtt>
- [35] (2013) Modernizr Documentation. Dostopno na:  
<http://modernizr.com/docs/>

- [36] (2013) Opera Mini. Dostopno na:  
[http://en.wikipedia.org/wiki/Opera\\_Mini](http://en.wikipedia.org/wiki/Opera_Mini)
- [37] (2013) PC Shipments Post the Steepest Decline Ever in a Single Quarter.  
Dostopno na:  
<http://www.idc.com/getdoc.jsp?containerId=prUS24065413>
- [38] (2013) Rainbow Lorikeet. Dostopno na:  
[http://en.wikipedia.org/wiki/Rainbow\\_Lorikeet](http://en.wikipedia.org/wiki/Rainbow_Lorikeet)
- [39] (2013) Scalable Vector Graphics. Dostopno na:  
[http://en.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics)
- [40] (2013) StatCounter Global Stats. Dostopno na:  
<http://gs.statcounter.com/#browser-ww-monthly-201308-201308-bar>
- [41] (2013) The World Wide Web project. Dostopno na:  
<http://info.cern.ch/hypertext/WWW/TheProject.html>

# Slike

1.1	Primerjava petmesečnega obdobja deležev uporabljenih naprav spletne strani <i>www.prevoz.org</i> v letih 2012 in 2013. . . . .	4
2.1	Množica različnih naprav, s katerimi lahko danes dostopamo do svetovnega spleta [4]. . . . .	6
2.2	Primer dveh različnih gostot slikovnih točk na velikosti enega palca (2,54 cm). . . . .	8
2.3	Primerjava ostrine besedil na klasičnih in Retina zaslonih. . . . .	9
2.4	Primer fotografije s 4, 8 ter 24 bitno barvno globino [38]. . . . .	10
2.5	Primerjava uporabe spletnih brskalnikov v avgustu 2013 [40]. . . . .	11
2.6	Primerjava časov naložitve 25 priljubljenih spletnih strani z uporabo različnih pasovnih širin ter omejenim časom RTT [1]. . . . .	13
2.7	Čas potreben za prenos 25 priljubljenih spletnih strani z omejeno pasovno širino ter različnimi časi RTT [1]. . . . .	14
4.1	Mobilni telefon, ki za spletno brskanje uporablja tehnologijo WAP. . . . .	20
4.2	Prikaz mobilne spletne strani <i>m.delo.si</i> na starejših telefonih. . . . .	20
4.3	Drsniki za premikanje po strani ter pomanjšana spletna stran predstavljajo neprijetno izkušnjo uporabnika med brskanjem. . . . .	21
4.4	Prva spletna stran objavljena leta 1993. . . . .	22
4.5	Primer odzivne spletne strani na različnih napravah ob različnih orientacijah. . . . .	23
4.6	Primer tekoče postavitve elementov v mreži. . . . .	24
4.7	Primer besedil, kjer so pisave izražene v relativnih in absolutnih enotah. . . . .	25
4.8	Elementi na različno velikih zaslonih zahtevajo strukturne spremembe v mreži tekoče postavitve. . . . .	27

---

4.9	Primer prikaza galerije slik na pametnem telefonu s tehniko zakasnjene nalaganja slik, kjer se sprva naložijo le prvi dve sliki.	31
4.10	Izkoristek uporabe tehnike stiskanja Gzip je opazen v zadnjem stolpcu.	35
4.11	Prikaz strukture omrežja spletnih strežnikov CDN razporejenih po celem svetu.	40
4.12	Prikaz prenosa virov pri mešanem vrstnem redu [34].	41
4.13	Prikaz prenosa virov v optimiziranem vrstnem redu [34].	41
5.1	Struktura tekoče postavitve elementov za zaslone treh priljubljenih tipov naprav.	48
5.2	Prikaz obrazca za naročilo artiklov.	49
5.3	Prikaz logotipa, kontaktnih podatkov, ikone za iskanje ter navigacijo po strani pri napravah z majhnim zaslonom.	51
5.4	Prikaz odprtega menija navigacije po spletni strani pri napravah z majhnim zaslonom.	52
5.5	Prikaz hitrega iskanja zelenih artiklov na majhni mobilni napravi z večnivojskem padajočim menijem.	53
5.6	Prikaz seznama artiklov spletne strani na pametnih telefonih in tabličnih računalnikih.	54
5.7	Prikaz seznama artiklov spletne strani na napravah z večjimi zasloni.	56

# Tabele

1.1	Primerjava deležev uporabnikov, ki so v letu 2011 in 2013 za brskanje po nekaterih priljubljenih slovenskih spletnih straneh uporabljali mobilne naprave. . . . .	3
2.1	Primerjava gostote slikovnih točk na nekaterih priljubljenih mobilnih napravah [31]. . . . .	8
5.1	Rezultati testiranja količine in časa prenosa celotne spletne strani pri dveh različnih zaslonkih ločljivostih z uporabo opisanih optimizacijskih tehnik. . . . .	60